On Latency in the Internet of Things

Naveed Nawaz

Submitted in accordance with the requirements for the degree of

Doctor of Philosophy

School of Electronic and Electrical Engineering

University of Leeds, UK

(December, 2019)

Supervisors

Dr. Des McLernon, Dr. Syed Ali Raza Zaidi, Dr. Maryam Hafeez, Dr. Li Zhang

Date of submission

23.12.2019

Dedicated to my parents, my wife and my son

Acknowledgments

In the first place, I would like to thank ALLAH, the Almighty, for blessing me with the strength and endurance to make all my achievements possible.

I would like to express my sincere gratitude to my supervisor, Dr. Des. McLernon, for his advice, guidance, and encouragement throughout my PhD research work. Thank-you for all the knowledge you transferred to me and the confidence you put in me. Thank-you for your endless kindness, care, and support. I could not have imagined having a better mentor for my PhD study.

My gratitude, appreciation and admiration go to Dr. Syed Ali Zaidi for his support, availability and and guidance. Thank you for all the knowledge and constructive conversations that we went through.

Finally, thanks to Dr. Maryam Hafeez for her support since my arrival to Leeds. Thank-you for your time, generous guidance, patience, and encouragement. Thanks for your patience and welcoming behaviour every day which encouraged me to widen my research from various perspectives.

Thanks also to the University of Engineering and Technology, Lahore for their financial support and funding without which, none of this work would have been possible.

My sincere appreciation goes: to my family who always encouraged and motivated me to reach this point in my life; to my parents who taught me the alphabet; to my wife for her companionship, support, care, endless patience, constant reassurance and steadfast belief; to my brothers and sisters for their belief and support. Finally, I would like to thank all my colleagues and friends who made my time at Leeds university such a pleasure. Thank-you so much!

Copyright

Declaration

The candidate confirms that the work submitted is his own, except where work which has formed part of jointly authored publications has been included. The contribution of the candidate and the other authors to this work has been explicitly indicated below. The candidate confirms that the work submitted is his own and that appropriate credit has been given within the thesis where reference has been made to the work of others.

The work in chapter 4 is submitted in the Internet of Things Journal for publication as follows

- Nawaz, N., Hafeez, M., Zaidi, S. A. R., McLernon, D. C., and Ghogho, M. "Fog as a Service: Towards Minimizing Latency of IoT using an Efficient Selection Strategy."

The work in Chapter 5 has appeared in the following publication

- Nawaz, N., Hafeez, M., Zaidi, S. A. R., McLernon, D. C., and Ghogho, M. (2017, May). "Throughput enhancement of restricted access window for uniform grouping scheme in IEEE 802.11 ah," IEEE International Conference on Communications (ICC) (pp. 1-7).

The work in chapter 6 is submitted to Transactions on Wireless Communications for publication as follows

- Nawaz, N., Hafeez, M., Zaidi, S. A. R., McLernon, D. C., and Ghogho, M. "WiFi-Halow for Internet of Things: Estimation of Restricted Access Window's Duration and Performance Evaluation."

Contents

List of Figures

List of Tables

Nomenclature

AID	Association Identifier
AP	Access Point
BEB	Binary Exponential Back-off
BI	Beacon Interval
CL	Cloud Layer
CSMA/CA	Carrier Sense Multiple Access/Collision Avoidance
DCF	Distributed Coordination Function
DIFS	Distributed Inter Frame Space
DSSS	Direct Sequence Spread Spectrum
DTMC	Discrete Time Markov Chain
EPSM	Enhanced Power Saving Mode
FaaS	Fog as a Service
FHSS	Frequency Hopping Spread Spectrum
FIFO	First In First Out
FL	Fog Layer
FLGW	Fog Layer Gateway
HT	High Throughput

ICT	Information and Communication Technologies
IEEE-SA	IEEE standard Association
IFC	IoT-Fog-Cloud
IL	IoT Layer
IoT	Internet of Things
ISM	Industrial, Scientific, and Medical
ITS	Intelligent Transport System
LIFO	Last In First Out
LRL	Long Retry Limit
M2M	Machine to Machine
MAC	Medium Access Control
MCCA	Multi-user polling Controlled Channel Access
MIMO	Multiple Input Multiple Output
NLOS	Non Line of Sight
OFDM	orthogonal frequency division multiplex
PHY	Physical Layer
QoS	Quality of Service
RAW	Restricted Access Window
RPS	RAW Parameter Set

RS	Random Service
RTS/CTS	Request to send/Clear to Send
RTTs	Round Trip Times
SDP	Service Discovery Protocol
SFE	Speed Frame Exchange
SIG	Signal field
SRL	Short Retry Limit
TDMA	Time Division Multiple Access
TWT	Target Wake Time
V2V	Vehicle to Vehicle
WLAN	Wireless Local Area Network

Abstract

In this thesis, we propose the use of IoT-Fog-Cloud (IFC) and Restricted Access Window (RAW) mechanism of IEEE 802.11ah standard to reduce latency in large-scale deployment of IoT devices having limited storage, computation and transmission capabilities. To achieve this end, we use a service discovery protocol to assess the capabilities of the devices and propose a strategy to offload tasks from one layer to the other in IFC paradigm using a queuing-theoretic framework which considers the storage, computation and transmission capabilities of the devices in each layer before actually offloading tasks to it. Moreover, we develop an efficient selection strategy in this thesis for the acquisition of nodes in the fog layer when many candidates are offering their services in the vicinity. The results show that our proposed strategy enhances the overall capability of fog layer by inclusion of efficient fog nodes in the network and thus reduces the latency in processing of tasks being offloaded to the fog layer.

Finally, we develop a comprehensive framework based on Discrete Time Markov Chain (DTMC) to characterize RAW which allows nodes to be divided into groups, and permits only one group to access the shared medium within a certain duration of time (RAW slot). By the use of the proposed framework, we determine the size of group of nodes for each RAW slot by quantifying the duration of the RAW slot required to transmit a given number of data packets when nodes in the group have only one, or more (finite or infinite) number of data packets. In this way, the number of collisions due to contention among large number of nodes in IoT is reduced significantly which results in reduction of latency of data packets in the network.

Chapter 1

Introduction

In this chapter, the motivation behind the study of latency issues in Internet of Things (IoT), along with some latency-sensitive applications, is presented. For densely-deployed IoT applications, we highlight how latency becomes a critical challenge, and describe the current techniques to reduce different components of latency and their limitations. At the end of this chapter, we highlight the limitations of the related work, present our contributions to reduce latency in this thesis, and summarize how the rest of the thesis is organized.

1.1 Motivation

The advent of technology has brought about revolutionary changes to the way we communicate, socialize and interact nowadays. The use of personal gadgets such as mobile phones, tablets, laptops, etc., in our daily routine nowadays is quite common. Already, the connected devices that are in use worldwide have outnumbered the total world population [2]. All this is due to a remarkable progress in wireless connectivity and miniaturization of electronic components. The trend to manufacture and deploy a multitude of smart sensors, actuators, and controllers, etc., in an urban and sub-urban environment, and then connecting them to the internet is on the rise. It is anticipated that in the coming years, almost everything imaginable will exchange information both intelligently and automatically. At the moment, only 1% of the physical objects in the world are connected to the internet [3]. With the increasing trend of connecting almost everything of daily usage, an exponential growth in the IoT is imminent. According to the predictions by EE, a British mobile network operator and internet service provider company, each UK household will have 50 connected devices on average by the year 2023 [4]. The total number of IoT devices worldwide, as projected by Statistica, is expected to reach 75.44 billion by 2025 – a five-fold increase in ten years from year 2015 to 2025 as shown in Fig. 1.1 [1]. CISCO, a worldwide leader providing state-of-the-art Information Technology (IT)-related products and services, puts forward an estimate that there will be around 500 billion connected things in the world by the end of year 2030 [5]. Such a massive number of IoT devices has a potential to generate enormous revenues and is expected to become a multi-trillion dollar industry [6]. However, the scale of future growth is highly dependent on a host of factors such as a resolution of technical barriers as well as the proclivity of industry to come up with state-of-the art solutions aiming to attract the customer's demands.

With technology encroaching more and more in our lifestyle, consumers in today's market demand that applications and services yield results instantly. According to a recent study by Forrester Consulting, 47% of users expect a web page to load in less than two seconds, and 40% of them will abandon a web page if it is not loaded within the first three seconds of the request [7]. Consumers demand a real time

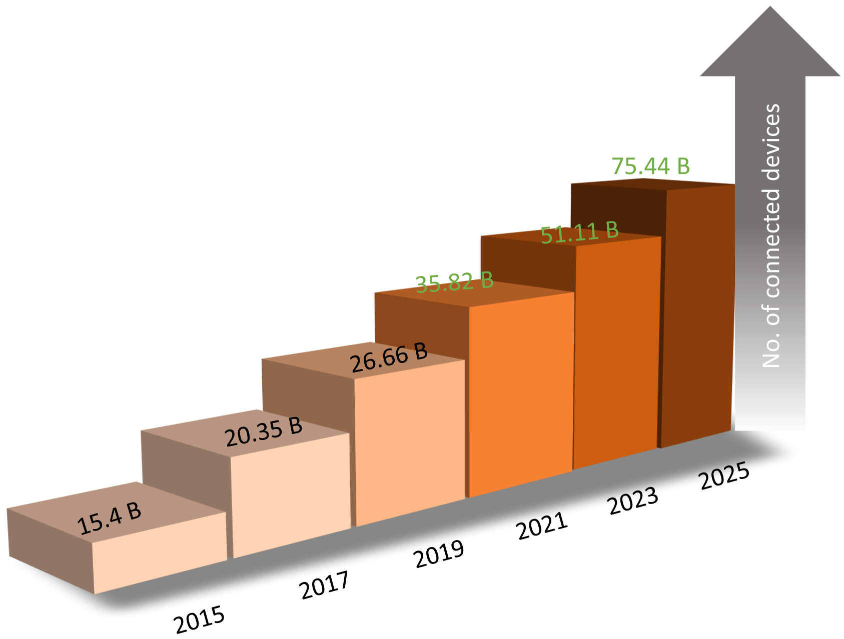

Figure 1.1: Total number of connected devices worldwide projected by Statistica [1]

response for almost all their actions, e.g., would a person, after flipping the switch, be willing to wait for a light to come on? A large scale adoption of IoT-solutions in our real world is possible only when it can produce results in real time, even for the most common applications of daily usage. Hence, provision of low latency solutions becomes a big challenge in almost all future applications.

IoT, being a network of connected things, is expected to be mainly composed of miniaturized devices having limited resources such as storage capacity, processing, and transmission power. These devices cannot keep data in their buffers for long due to limited storage capacity and may result in loss of either the newly arriving packets, or the ones previously stored in the memory. The limited processing power of such devices may result in further delays as they take quite long time to process an information before its transmission. Moreover, the constraints of devices to transmit limited power aggravates the situation even further and leads to even more delays. The net result of all these delays is that the device has to remain active for a longer duration. Therefore, a low latency solution becomes a critical challenge in the perspective of resource constrained IoT devices and motivates future research to provide state-of-the art solutions in this regard.

Spurred by a wide range of applications, the explosive growth in the number of connected things in the future will result in a dense deployment of IoT devices, especially in urban and sub-urban environments. All these devices will be required to transmit information over a shared medium. In order to access the medium, the use of protocols based on a Time Division Multiple Access (TDMA), a collision-free access scheme, may be helpful in a reduction of energy consumption (as a device may become active during its allocated time slot and remain inactive for the entire remaining duration). However, in practical, when devices are generating data sporadically, it may not be possible to schedule such a large number of devices appropriately. This may result in a dramatic degradation in the utilization of channel. The use of contention-based random access protocols, such as Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) in which the nodes contend to access the shared medium, may be practical for IoT traffic, but the network becomes severely overburdened especially when a massive number of devices attempt to access the shared medium at a time. This results in severe collisions of packets, leads to wastage of channel utilization, and causes huge data losses along with unbounded network delays. Even though IoT devices, in most of the use cases, transmit a small amount of data intermittently, the resource strapped devices have to remain involved in the transmission process for longer periods and may not be able to survive extended power losses and long network delays. This motivates us to focus on advanced techniques for the densely deployed IoT network with an aim to minimize latency by improving channel access schemes.

The enormous amount of data generated by IoT devices will require huge processing capabilities. By employing other devices such as cloud servers, tasks can be processed on behalf of IoT devices. Although these servers usually have vast processing and storage capabilities and can perform tasks almost instantaneously, the remote location of these servers may result in extraordinary delays and this motivates us to find alternative ways of offloading tasks with a focus to develop low latency solutions.

1.2 Internet of Things and Latency

The IoT is a network of smart, web-enabled physical objects/things that can communicate, sense and interact with the external environment over the internet by the use of embedded processors, sensors and communication hardware. In this section, we present challenges arising from the development of various IoT solutions from the perspective of latency and introduce some of the latency-sensitive use cases of IoT and enunciate some of the ways to minimize various components of latency.

1.2.1 Latency-sensitive Applications

Some of the possible applications of IoT where low-latency solutions are of significant importance, as shown in figure 1.2, are as follows:

1. **Remote Healthcare and medical intervention:** Integration of communication networks with the healthcare industry has already brought substantial transformation in the way high-resolution medical records are accessed and a high-definition video is transmitted in real time. With the support of a massive number of connected devices such as e-health wearables, and by keeping medical records electronically, we stride towards predictive healthcare and perform individual pharmaceutical analysis. There is an increasing trend of

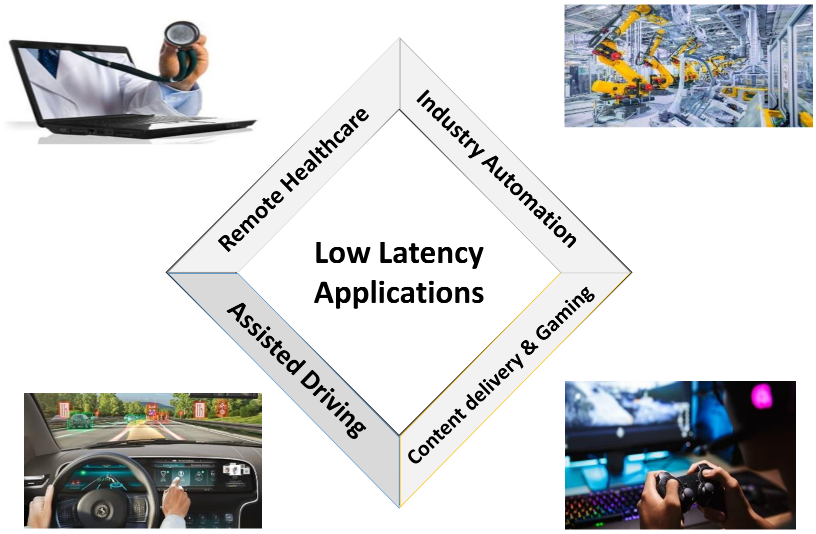

Figure 1.2: Some of the latency-sensitive applications

decentralization of hospitals, tele-medicine, remote care and mobile care where medical records can be made available in an emergency while a patient is in the ambulance. With low latency solutions, we are now making headway to remote healthcare by breaking the obstacle of geographical boundaries in most complex medical interventions and surgeries. In tele-surgery, the entire procedure can be controlled by a surgeon at a remote site by the use of a robotic arm. The use of tactile sensors on surgical equipment can give a real feel to healthcare professionals. In this way, they can exploit the palpation skills remotely while watching a real-time video stream. It is determined from the real experiments that the maximum tolerable one-way delay in case of tele-surgery is 150 ms [8]. Although the use of haptic feedback in tele-surgery can improve the accuracy in real feel, it puts further stringent constraints on latency. Tele-surgery, in the presence of haptic feedback, requires end-to-end round trip times (RTTs) of less than 10 ms [9]. Hence, low latency solutions in remote healthcare and medical interventions are of significant importance.

2. **Assisted Driving and Transport Services:** In order to reduce traffic congestion on roads and improve the efficiency, safety, and sustainability of a traffic management system, the use of leading edge information and communication technologies (ICT) along with the deployment of smart sensors in smart cities is on the rise, transitioning the traffic to an intelligent transport system (ITS). Today, state-of-the-art mobile communications have paved a way for many user-friendly applications in the automotive sector which share real time information among commuters and drivers. The revolution in the automotive market is under way where the manufacturers such as Ford, Audi, Jaguar Land Rover or Volkswagen, etc., are focusing to enhance user experiences in driving by developing autonomous or assisted driving. The massive number of sensors and actuators being used in both vehicles and transport infrastructures would need ultra high reliability which is not possible without achieving low latency communication solutions, e.g., the maximum tolerable end to end latency during exchange of a message for a reliable operation of an automated overtaking system for vehicles is 10 ms [10]. Thus, low latency solutions are essential for the reliability of automated, and/or assisted driving in vehicle-to-vehicle (V2V) communication.

3. **Entertainment: Content Delivery and Gaming:** In the gaming industry, the consumer experience can be enhanced by adding the capabilities of virtual and augmented reality which enable the gamer to play with the feel of the real world. Bio-sensing, a useful way to detect people in the game and capture motion while interacting with other objects, is another useful way to improve the gamer experience. Moreover, development of new interfaces (e.g., gloves)

to introduce haptic capabilities, use of real time videos and audios along with a multitude of other sensors can improve the consumer experience in both content delivery and gaming. Low latency is one of the main technical challenges to deliver these kinds of entertaining services in the gaming industry, e.g., for virtual reality and augmented reality, the delay threshold to provide smooth consumer experience is 15 ms [11].

4. **Industry Automation:** Industry 4.0, a European Commission vision to improve efficiency in manufacturing industry, emphasizes the need to adopt intelligent networking of logistics and product development with a focus to collect and exchange information along the whole life-cycle of the product. The use of a multitude of intelligent, interconnected sensors to carry out industrial monitoring and tracking has been in vogue in industry nowadays. The evolution of a production system that can operate beyond the factory premises and can be optimized in real-time according to the information being received from multiple vendors is a competitive trend in manufacturing business. Low latency IoT solutions, integrated with the industrial automation, are key enablers to improve the efficiency of manufacturing and production processes [12].

In the next subsection, we present how latency is a critical challenge for various IoT solutions.

1.2.2 Low Latency: A Critical Challenge

As mentioned previously, the number of IoT devices and sensors in the world is projected to increase to the tune of billions in future. This raises huge concerns about the deployment, maintenance and handling of such a huge number of devices which contend for the access of a shared medium during information transmission, and may face longer delays, especially under heavy load conditions. Moreover, the centralized location of servers may lead to long propagation delays in information exchange. The prolonged delays may cause IoT devices to remain active for a longer period and result in an increased energy consumption. Hence, low latency is a big challenge for an efficient energy management of devices – an important factor to preserve the lifespan of IoT devices.

With the emergence of low-cost embedded computing devices in various IoT solutions, most of the devices and sensors are expected to be resource-constrained in terms of size, memory storage capacity, processing and transmission power as well. The packets generated by these devices may have to queue up in their buffer and wait for their turn when they are faced with delays. This may result in the loss of important information as newly arriving data may not be able to find enough memory space to remain in the buffer of these devices. Therefore, low latency is required to ensure smooth information exchange without any loss of data in a multitude of such resource-constrained devices.

Above all, low latency is critical for many delay-sensitive and real time applications. For example, a delay of a few hundred milliseconds may degrade the human perception of interactive multimedia quite dramatically. Similarly, there are many real time applications such as live streaming, video conferencing, networked games, video surveillance, online data analytic, and tactile internet etc., where response within a certain delay threshold is a must for smooth operation. Moreover, many mission-critical applications such as public safety, military and disaster recovery etc., can only be deployed if the response time is instantaneous or within a tolerable delay limit. In a world fraught with technological advancements, it is quite obvious that people demand instantaneous results even in commonly used IT-related applications such as uploading a picture on Instagram, loading a web page, or playing a video on Youtube, etc. Therefore, the expectation that the IoT solutions will be part of our real world will necessitate that the results are also achieved in real-time. This makes latency a critical challenge in the future.

1.2.3 Towards Minimizing Latency

Latency, an important performance metric of a network and critical for many delay-sensitive applications, has four fundamental components for communication of information along a route as shown in Fig. 1.3 and are discussed briefly as follows:

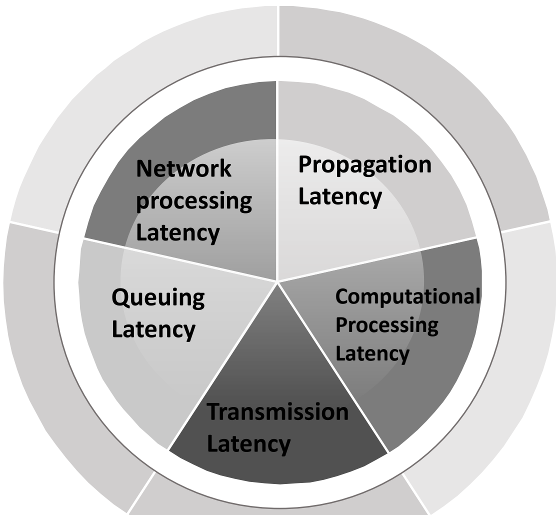

Figure 1.3: Components of latency

1. **Propagation Latency:** The delay incurred by the data packet to actually travel from a source to the destination is called the propagation latency. Let d be the distance a data packet has to cover during its transmission from source to the destination (expressed in meters), and v be the propagation velocity of the wave through the medium (in m/sec), then the propagation latency is given as $\zeta_p = \frac{d}{v}$. The propagation delay is one of the major factors that

affect the system performance in high-speed networks. By the use of advanced techniques such as optical fiber, the delay for propagation of waves through the medium can be achieved as low as 5 μsec per kilometer [13]. In order to reduce propagation latency, the use of a network of distributed computational devices near to the end users called "fog computing" is preferred over the conventional centralized computing cloud servers. The fog computing plays a vital role to facilitate delay-sensitive real time IoT applications. Due to a close proximity of computational devices with the end consumers, the packets generated by the IoT devices have to travel a much shorter distance in order to perform computation of their tasks, and hence the propagation delay can be reduced significantly.

2. **Processing Latency:** The delay incurred by the network node to process the data packet is called processing latency. Processing delay can be divided into two types, i.e., Network processing latency, and computation latency.
   The delay incurred by the data packet while accessing the shared medium is called network processing latency. The network processing latency depends upon a host of factors such as the intensity of the number of packets arriving to access the medium, the number of nodes contending for the access of the shared medium, and the efficiency of the protocols controlling the access of the shared medium. With a dense deployment of IoT nodes in large scale wireless networks, the efficiency of medium access control protocols reduces drastically, and may lead to inordinate network latency. This may be improved by dividing the nodes into several groups and assigning each group to a specific time duration, called Restricted Access Window (RAW) in IEEE 802.11ah standard where the nodes belonging to the same group can contend with each other for the medium access within the RAW duration.

The delay incurred by the network node to compute the task in the data packet is called the computational latency. It is dependent on the performance of the hardware, e.g., the processing latency in high speed routers is typically of the order of microseconds. The computational processing latency can be reduced to a significant degree by the use of the devices having an improved hardware performance. In IoT applications, a repercussion of having resource-constrained end-user's devices in terms of processing power is long computational delays. This can be reduced by offloading task data from IoT to the fog devices in the vicinity provided the delay incurred in transmission, propagation and processing of packets in the fog layer are lesser than nodal processing delay of the devices. This is possible if there is a sufficient link bandwidth between the devices in the IoT and the fog layer. The fog devices are quite close to the end users, and have better computational capability to process the task data. However, the deployment of efficient fog devices over the entire geographical footprint or selection of efficient devices is quite critical for latency reduction.

3. **Transmission Latency:** The delay incurred in transmitting a data packet through the medium due to its limited bandwidth is called the transmission latency, or the serialization delay. Let R be the bandwidth of the link through which the data packet is to be transmitted (expressed in bits per second), and L be the length of the data packet (in bits), then the transmission latency is found as $\zeta_T = \frac{L}{R}$. In IoT, where the length of the data packet is usually quite small as compared to the link speed, the transmission latency is of negligible value of the order of nanoseconds, e.g., the transmission latency for a 64 byte data packet, a typical size of a data packet, over a link speed of 10 Gbps is approximately 51.2 nanosecond.

4. **Queuing Latency:** Queuing latency is the delay incurred by a data packet when it has to wait in the queue of the device before being served. A queue is formed if the rate at which the packets arrive is greater than the rate at which they are departing from the device, and in such a case, the data packet has to wait for its turn in the buffer of the device. There are several reasons that contribute towards queuing latency. For example, the device may have to re-transmit a data packet when it is faced with a collision with other data packets, and cause the bottleneck for other packets to wait longer in the queue. The slow computation capability of the device is another factor that may result in an increased queuing latency as it takes longer to process the data packet that is being served in the queue. Moreover, the limited bandwidth of the link between the transmitter and receiver and larger-sized data packets also result in a bottleneck in transmission of data packets and contribute towards increased delays for other data packets in the queue of the device.
   Due to limited storage capacity of IoT devices, only a finite amount of packets can be held in the queue. Therefore, a device may have to discard some of its received packets when its buffer is full. The approaches such as fog computing, deployment of efficient hardware, use of an efficient task offloading strategy, transmission of smaller-sized packets, availability of enough link-bandwidth, and use of an efficient medium access control protocol to reduce network latency can help reduce the bottlenecks that may occur during the transmission of packets successfully, and are helpful in reducing the queuing latency.

Table 1.1 shows the various components, along with their typical values, which contribute to the overall latency when a data packet is transmitted over a link after it is processed by a typical device [14].

Table 1.1: Components of network latency along with their typical values for the transmission of a 1250 byte data packet over a 1 Gbps link at a distance of 200 km by a device having processing speed of 100 MIPS

Latency Components	**Typical range of value**
Transmission Latency	$10\mu s$
Propagation Latency	$1,000\mu s$
Computational processing Latency	$10\mu s - 100\mu s$
Network processing Latency	$0 ... \infty$
Queuing Latency	$0 ... \infty$

1.3 Current Work & its Limitations

Despite the obvious promising avenues to deal and minimize the latency for ultra-dense IoT deployment, researchers in the field are still trying to overcome the embedded challenges from various perspectives. Next, we highlight some of these critical issues and present limitations of the current work.

1. **IoT-Fog-Cloud paradigm:** In order to meet the demands of the generation of large volumes of task data by a wide variety of IoT solutions, the integration of IoT devices with the cloud infrastructure is proposed in [15], [16] and the enabling technologies for such an integration are presented in [17]. An insight into the architecture, implementation and performance of the cloud-based IoT solutions is provided in [18]. To address the challenges posed by the integration of IoT with the cloud servers, the articles in [19]-[20][21][22] present an IoT-Fog-Cloud (IFC) model by introducing an additional layer of computing devices between the cloud and end users. The characteristics of fog computing are described in [23]. In [24], [25], the ability of fog computing to handle issues such as big data, congestion and latency for real-time applications is highlighted. Several reference architectures for fog computing are presented in [26]-[27][28][29][30][31][33][34]. In [26], Dastjerdi et. al presented a fog computing architecture in which the tasks are processed by local fog devices instead

of involving the cloud. A model using fog devices for delay sensitive applications is proposed in [27]. In [28], a framework to model service delays in the IFC paradigm is proposed along with a delay minimizing policy. A platform to meet demands of the industrial processes using fog computing is proposed in [29]. The connectivity of sensors with the fog nodes using device-to-device (D2D) based communication is proposed in [30], and a network architecture consisting of fog nodes is presented in [31]. In [34], a general framework for IFC applications is presented.

2. **Efficient selection strategy for new fog nodes:** All the existing network architectures for the IFC paradigm are based on a general assumption that the fog nodes are chosen a priori and the information about the formation of the fog network is completely known. However, in a practical scenario, especially when Fog as a Service (FaaS) is utilized, the fog nodes may leave the network at a random time. Moreover, the traffic load may also increase unpredictably in reality. Therefore, the network may have to select a new fog node dynamically so that the tasks may be offloaded to it to achieve an improved QoS. When the selection process for a new fog node is initiated, there may be many candidate devices in the locality offering to join as fog nodes. In [35], Lee et al. adopted a model that used exploration and exploitation structures from the online k-secretary framework [36] for the selection of the best fog node to be included in the fog network. The model is then extended in [37] by updating the network size dynamically and providing an online algorithm for selection of fog nodes with an aim to minimize latency. However, in these works, it is assumed that a fog node has an updated information about the processing capabilities of all the existing nodes in the network during the entire selection process which may not be possible in a practical scenario as some of the nodes may leave the

network at any time. The classical decision making approaches are presented in [38]-[39][40] in which the choice of a new candidate depends only on the capabilities of those competing for the selection during the ongoing process and not on the already acquired fog nodes in the network. But they are inefficient when it comes to the selection of a large number of fog nodes.

3. **Task offloading:** As described earlier, one major drawback of resource-constrained devices in IoT solutions is that they may not be able to process all of the task data on their own. This not only leads to longer processing delays but also may cause the nodes to discard some of the packets arriving in their buffer due to a limited storage capability. In [41], the authors propose a general framework for an IFC paradigm and offloading policy to minimize latency. In [42], Alameddine et al. formulated a dynamic task offloading framework mathematically. Other strategies for the task offloading to nearby devices are developed in [43], and [44]. The authors formulated a computational offloading game to model the competition between IoT users and efficiently allocate the processing power of fog nodes in [45]. A cooperative task offloading policy between two fog nodes is formulated in [32] and an analysis to offload tasks among multiple fog nodes is presented in [33]. From the perspective of latency, the task offloading depends on a host of factors such as storage, processing, and the transmission capabilities of devices, and delays in transmission of packets from one device to the other. So it becomes quite a challenging task to devise a strategy for distribute tasks among different network nodes.

4. **Grouping of nodes and modeling RAW:** For ultra-dense IoT infrastructure deployment, the number of nodes contending for the medium access simultaneously can be reduced by dividing the nodes into groups and letting only the nodes within a group contend with each other in RAW, which

helps reduce network latency to a great extent. In [46] and [47], the authors adopted an approach presented in [48] and developed a mathematical model for the Group-Synchronized Distributed Coordination Function (GS-DCF) mechanism – a scenario similar to RAW. Khorov et. al. developed a model for finding throughput in [49] and presented an analysis of the influence of network and traffic conditions on the optimal station grouping parameters. Other related works, in which the performance of the RAW mechanism is evaluated analytically, are presented by the authors in [50]-[51][52][53][54][55][56][57][58][59][53][55][60]. In all these works, the nodes contending for the medium access are either considered to have either an infinite number of packets, or a single data packet in their buffer. The analytical models developed in [61]-[62][63] assume that the nodes are in saturated mode, and mathematical frameworks developed in [64]-[65][66][67] assume that the nodes are in unsaturated mode. However, in reality, a node may have a finite number of packets and may receive another data packet from the upper layer while it is in the process of transmitting the previous one i.e., the generic mode. In [68], an algorithm is presented via simulations to adapt the RAW parameters in real time for saturated, unsaturated and generic modes. However, none of these works provide an analytical framework which can deal with the generic mode along with both saturated and unsaturated modes that could be applied to find the duration of time required to transmit a given number of packets, and find parameters of RAW.

A summary of the techniques dealing with the latency issues along with a breakdown of relevant literature in each category is presented in Table 1.2.

Table 1.2: Categories of current literature dealing with the latency issues in IoT

Latency Components	Typical range of value
IoT-Fog-Cloud paradigm	[16]-[35]
Efficient selection of new fog nodes	[36]-[41]
Task offloading	[42]-[45]
Grouping of nodes and modeling RAW	[47]-[67]

1.4 Thesis Outline and Contribution

In this thesis, we aim to study latency issues that come across as a result of the dense deployment of devices in future IoT solutions, especially in an urban and sub-urban environment. Generally, we focus on implementation and modeling of the IFC paradigm, where tasks are offloaded from one layer to the other layer with the aim of reducing latency, and we develop a selection strategy for deployment of efficient fog nodes so that the task data can be processed near to the end users. To avoid congestion and contention in ultra-dense IoT deployment, we present a general framework of RAW in IEEE 802.11ah and estimate the duration of the beacon interval that can be adopted to evaluate the performance of the standard while nodes are grouped together in WiFi HaLow. The thesis is organized into 7 chapters, as shown in figure 1.4, and the remaining parts of it are outlined as follows:

- **Chapter 2:** In this chapter, we present a literature review and theoretical background that we will use in the rest of the thesis. In particular, we will focus on the IEEE 802.11ah standard and fog computing which provide useful techniques to deal with the latency issues in IoT. Moreover, we present a background related to Markov chains and queuing-theoretic models which are utilized in developing a framework for RAW and task offloading in the IFC paradigm that are used throughout this thesis.

- **Chapter 3:** In this chapter, we present a queuing-theoretic model for task offloading with an aim of reducing latency in a three-layered IFC paradigm dependent on the storage and processing capacity of devices. We discuss an algorithm to find out when to trigger the selection process of a new fog node so that tasks could be processed in nearby computing devices instead of sending data to the cloud servers.

- **Chapter 4:** In this chapter, an algorithm to acquire new fog nodes in the IFC paradigm is presented which uses of an efficient selection strategy and results in enhancing the overall capability of the network to process the task data within the fog layer. Then, we develop an analytical framework to find the efficiency of our algorithm in terms of time spent in the acquisition of a new fog device by the use of our selection strategy.

- **Chapter 5:** In this chapter, we develop an analytical framework for RAW and present a model where a frame is divided into two sub-frames and the duration of RAW slots in each sub-frame is chosen according to the size of the group in a uniform grouping scheme. We demonstrate that the throughput under our proposed scheme can be significantly enhanced when compared to a conventional implementation.

- **Chapter 6:** In this chapter, we present a Markov-chain based analytical framework to find the mean duration of time until any of the contending nodes transmits its data packet in a RAW slot, and analyze the time required to transmit a given number of packets in a RAW slot for saturated, unsaturated and generic modes. We organize the nodes according to a uniform grouping scheme and analyze the throughput for all the modes. We then estimate the duration of time required to transmit a given number of packets by a group of nodes assigned to a RAW slot efficiently. In this way, we find the duration of a beacon interval in IEEE 802.11ah standard to improve the overall throughput.

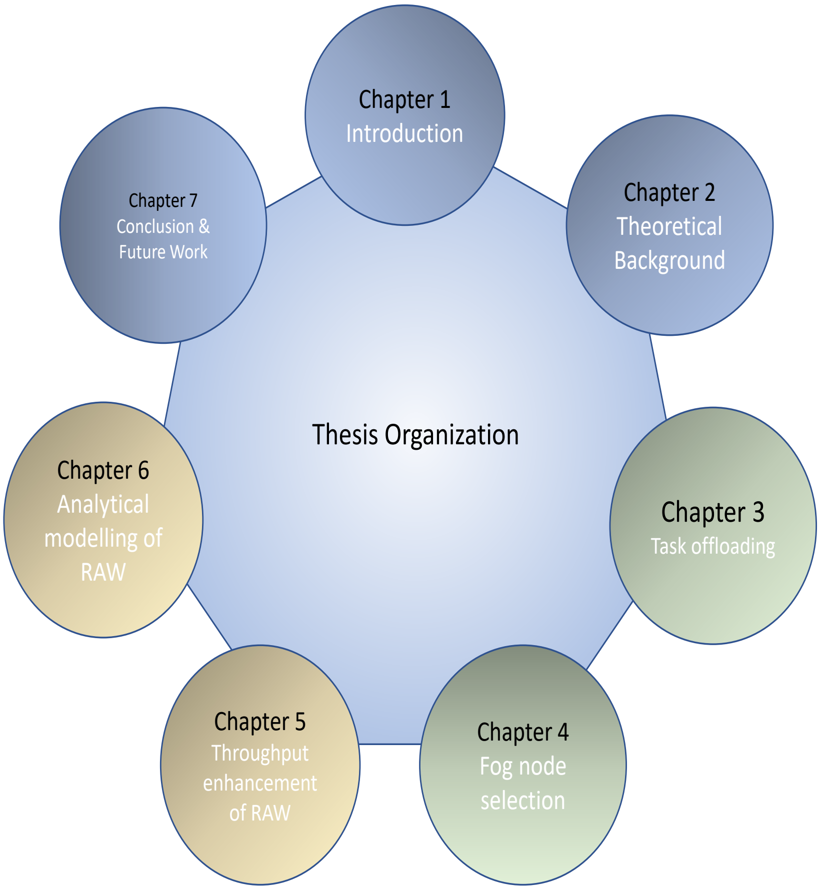

Figure 1.4: Thesis organization

- **Chapter 7:** This chapter concludes the thesis and discusses possible future directions for research.

1.5 List of Publications

- Nawaz, N., Hafeez, M., Zaidi, S. A. R., McLernon, D. C., & Ghogho, M. (2017, May). "Throughput enhancement of restricted access window for uniform grouping scheme in IEEE 802.11 ah," IEEE International Conference on Communications (ICC) (pp. 1-7).

- Nawaz, N., Hafeez, M., Zaidi, S. A. R., McLernon, D. C., & Ghogho, M. "Fog as a Service: Towards Minimizing Latency of IoT using an Efficient Selection Strategy", (submitted to the Internet of Things Journal).
- Nawaz, N., Hafeez, M., Zaidi, S. A. R., McLernon, D. C., & Ghogho, M. "WiFi-Halow for Internet of Things: Estimation of Restricted Access Window's Duration and Performance Evaluation", (submitted to Transactions on Wireless Communications).

Chapter 2

Background Theory

In this chapter, we present a general overview of the thesis and a brief literature review of the standard approaches used to deal with the issue of latency. We provide a comparison of IEEE 802.11ah with other competing standards for IoT, describe some of its salient features, and present an IoT-Fog-Cloud (IFC) network architecture. We then discuss some of the mathematical tools such as queuing-theoretic models and Markov chains which are useful for the development of analytical frameworks to reduce latency in IoT.

2.1 General Overview

The grand vision of connectivity of anything, anyone, anywhere and at anytime is realizable with the deployment of smart, miniature IoT devices in a cost effective way. Slow responsiveness is one of the major challenges not only for latency-sensitive applications but for almost all use cases in IoT. This is mainly due to the deployment of devices having limited resources, most of which do not have a sufficient capability to process tasks on their own and so may cause long processing and queuing delays. Due to limited storage capacity, it may not be possible for them to hold data for long durations. Moreover, the contention among a large number of devices for the medium access may result in collision of data packets, and cause even more delays when nodes attempt to re-transmit the same data again and again. Such inordinate delays cause devices to stay active for a longer duration which may not be possible for most of the IoT nodes. Therefore, low latency is a key challenge for a wide range of IoT solutions. While researchers from both academia and industry have been focusing on latency-related issues for many years in the past, there is still a room for significant improvement in the design space of IoT in this regard.

In this thesis, the strategies to reduce latency in the design space of deployment of resource constrained IoT devices are presented. In the context of reducing propagation and processing delays for such IoT deployments, the design space is explored to include fog computing to augment the existing IoT-Cloud continuum and a comprehensive queuing theoretic model is presented which includes a service discovery protocol in between the IoT and fog nodes such that the capabilities of the devices related to fog computing could be assessed before actually offloading tasks to them. Unlike a fixed infrastructure of cloud server over an entire geographical footprint, new avenues for formation of fog computing are explored by use of Fog as a Service where already existing devices in the vicinity of end users are deployed and an efficient strategy for selection of fog nodes in the fog infrastructure dynamically is presented in this thesis. With the deployment of billions of IoT devices in future, the contention among nodes are expected to increase significantly when all the nodes access the shared medium simultaneously by the use of conventional Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA). The IEEE 802.11ah standard introduces a technique to reduce collision among the nodes contending for the medium access according to which only a group of nodes can access the shared medium in a RAW slot. However, the choice of a large duration of RAW slot may result in wastage of some of its portion in a scenario when all the nodes in the group assigned to the RAW slot have transmitted their data packets and result in idle time at the end of the RAW slot. The choice of a small duration of RAW slot may cause a significant performance degradation as more nodes contend for the medium access at the beginning of RAW slot, resulting in a relatively higher probability of collision at the beginning of RAW slot as compared to the end of RAW slot when there may be lesser number of nodes contending for the medium access. Therefore, the choice of an optimal duration of RAW slot is necessary to reduce mean number of collisions of data packets during the RAW slot. In this context, a comprehensive framework is developed for RAW in this thesis which is useful to quantify the duration of RAW slot required to transmit a given number of data packets.

In this chapter, we present different IEEE standards and their extensions for the deployment of IoT use cases and discuss some background theory necessary for the development of analytical frameworks aiming to reduce latency in IoT.

2.2 IEEE 802.11 standards

In order to facilitate the wireless connectivity to portable and moving devices within a local area, IEEE 802.11 is a standard for a wireless local area network (WLAN) developed by the IEEE standards Association (IEEE-SA) – an organization responsible for the design, development and standardization of technologies emerging on an international arena. The standard defines specifications related to both medium access control (MAC) and physical layer (PHY), and specifies an over-the-air interface between two wireless devices or a wireless device and a base station, and enables transmission at the rate of either 1 Mbps or 2 Mbps using the 2.4 GHz band. It is based on either Frequency Hopping Spread Spectrum (FHSS) or Direct Sequence Spread Spectrum (DSSS) and achieves a transmission range of 20 feet. To cope with the new challenges posed by technological advancements, the standard is updated through some amendments. The main characteristics of these amendments are delineated as follows [69], [70]:

- IEEE 802.11a: Enables high speed up to 54 Mbps using the 5 GHz band, achieves a transmission range up to 75 feet, and is based on using orthogonal frequency division multiplex (OFDM) instead of FHSS or DSSS.
- IEEE 802.11b: Although has a speed up to 11 Mbps using 2.4 GHz, but it extends the transmission range up to 150 feet.
- IEEE 802.11e: Adds Quality of Service (QoS) and multimedia support to the previously available standards.
- IEEE 802.11g: Utilized for transmission over shorter distances at a speed up to 54 Mbps using 2.4 GHz frequency band with a range of 150 feet.
- IEEE 802.11n: Aims to increase throughput with a baseline goal to achieve speeds up to 600 Mbps, uses the 2.4 GHz and 5 GHz frequency bands, and adopts Multiple Input Multiple Output (MIMO) antenna technology, and may achieve transmission ranges of 175+ feet.
- IEEE 802.11ac: High throughput (up to a speed of 1 Gbps for multi-station and 500 Mbps for a single-link) by using MIMO spatial streams, multi-user MIMO on the 5 GHz band [71].
- IEEE 802.11ad: Aims to achieve throughput as high as 7 Gbps using a frequency band of 60 GHz [72].

Table 2.1: Typical ranges of candidate standards for IoT

Standard	Frequency band	Data rate	Range
SigFox	< 1 GHZ	300 bps	50 km outdoors
LoRa	< 1 GHZ	<50 Kbps	10-20km
Thread	2.40-2.48GHz	250Kbps	30 m indoors
Zigbee	<1GHZ, 2.4-2.480GHz	250Kbps	30m/1.5km (indoors/outdoors)
EnOcean	< 1 GHZ	125Kbps	30m/1.5km (indoors/outdoors)
6LoWPAN	<1GHZ, 2.4-2.480GHz	250-4Mbps	5-10 km
Z-Wave	908.42MHz	40 Kbps	30 m indoors
NB-IoT	cellular	250Kbps	1km (urban), 10km (rural)
IEEE 802.11ah	< 1 GHZ	150Kbps +	100-1000 m

- IEEE 802.11ah: Enables formation of groups of stations and transmission of a large number of nodes, and uses license-exempt Industrial, Scientific, and Medical (ISM) frequency bands below 1 GHz. Therefore, it can provide a coverage range up to a distance of 1 km, and allows the use of battery-operated equipment because of low power consumption [73].

Most of these standards have an operating frequency of 2.4 GHz, or 5 GHz and require higher powers to extend the range which may result in an increased power consumption. In IEEE 802.11ah, the use of a sub-1GHz operating frequency not only increases the coverage range with a relatively less requirement of transmission power, but also enables signals to penetrate through the buildings and other obstacles with ease, and enables non-line of sight (NLOS) operation. Most of the devices used in IoT applications such as smart metering, traffic monitoring, soil monitoring for crops, supply chain monitoring, forest fire detection in remote regions, instant notifications of leakages in pipelines, availability of parking spaces, etc., transmit sporadic, minimal data at a low data-rate and need to keep power consumption to a bare minimum (to ensure long battery life). However, these devices are densely deployed in an urban and sub-urban environment and there will be hundreds, or thousands of nodes connected to a single AP [74]. While many of these applications

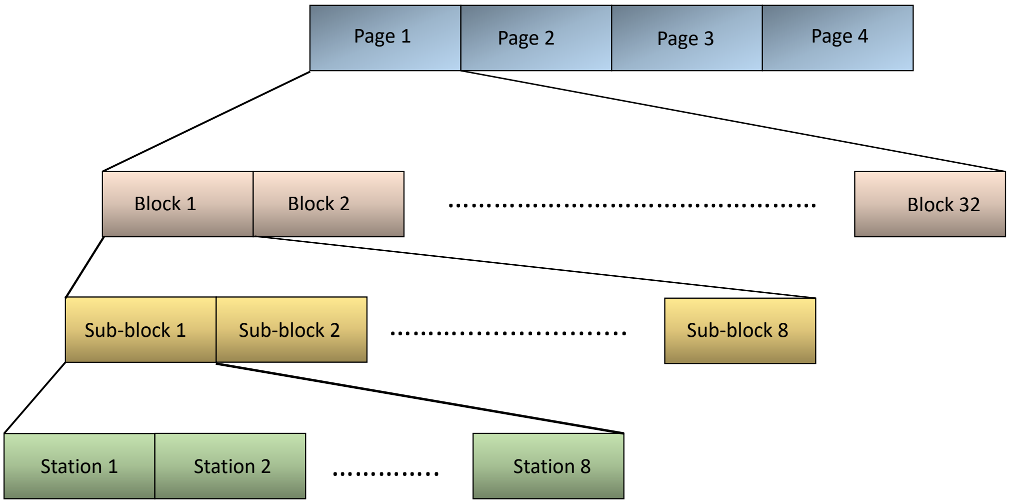

Figure 2.1: Hierarchical structure of AID

require them to transmit within the range of few hundred meters, some of them require covering a distance of many kilometers. The table 2.1 shows different candidate standards for future implementation of IoT [75] along with their frequency bands, data rate, coverage range, and typical transmission power. Some of the salient features of IEEE 802.11ah which make it a strong candidate for the deployment of large scale sensor networks and IoT devices [76] are discussed in the next subsection.

2.2.1 Salient Features of IEEE 802.11ah

- **Association Identifier:** In IEEE 802.11 ah, a hierarchical structure is defined in such a way that that Access Point (AP) can assign each node a 13-bit unique identity called Association Identifier (AID). In this way, quite a large number of nodes can be addressed as compared to the legacy IEEE 802.11 standard [77]. The hierarchical association structure consists of 4 levels such

Table 2.2: Energy consumption of a typical device in different modes

Mode	**Energy consumption (mW)**
Active (Transmission)	255
Active (Receiving/Carrier sensing)	135
Sleep	1.5

as page, block, sub-block and node-index as shown in Fig. 2.1. Such a structure facilitates in the organization of nodes into various groups where nodes having similar constraints or traffic patterns can be grouped together. By organizing nodes into the groups (and adopting the techniques described in IEEE 802.11ah), a large number of devices may access the channel efficiently, and can save sufficient energy.

- **Enhanced Power Saving Mode (EPSM):** In IEEE 802.11ah, EPSM is specifically useful for low-powered sensor devices in IoT [77]. In this mode, the AP exchanges an information element called Target Wake Time (TWT) when it gets associated with devices. Therefore, the nodes wake up at TWT for exchange of data and do not have to remain active for the entire beacon interval in IEEE 802.11ah [78]. In contrast, the legacy IEEE 802.11 has a power saving mode in which the stations alternate between active and sleep mode. Therefore, the nodes using EPSM in IEEE 802.11ah can remain in sleep mode for relatively longer durations as compared to when they use the power saving mode in legacy IEEE 802.11. Table 2.2 lists the energy consumption of a typical device during active and sleep modes [79] where it can be seen that energy consumed by the device is significantly reduced when it is in sleep mode as compared to the other modes. Hence, the energy consumption of devices can be reduced significantly by the use of EPSM in IEEE 802.11ah.

- **Restricted Access Window (RAW):** As described earlier, 802.11ah introduces a channel access mechanism where the nodes are assigned to groups by

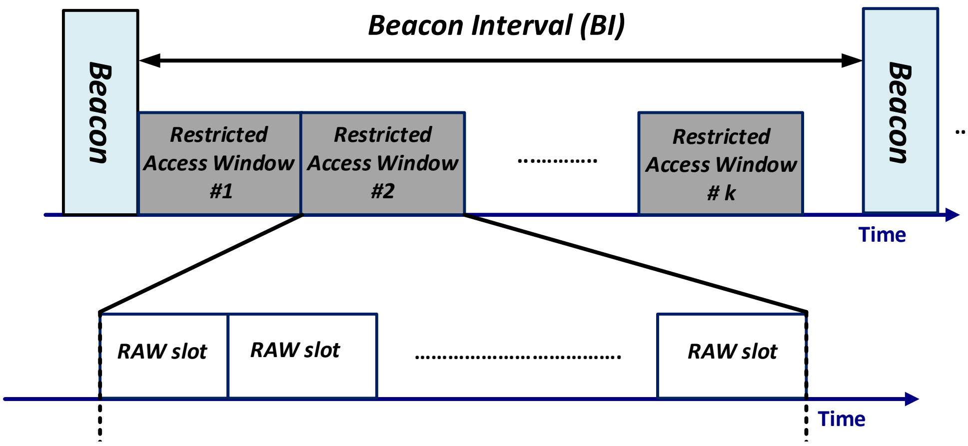

Figure 2.2: Structure of the restricted access window (RAW) and a beacon interval in the IEEE 802.11ah standard.

the use of AID. In such a standard, the nodes in a group are allowed to transmit data (after contending with each other for the medium access) within a restricted interval of time called RAW. A beacon interval (BI) consists of one or more RAWs, and each RAW can be further divided into time slots called "RAW slots" as shown in Fig. 2.2. The AP assigns one specific RAW number to each group by the use of an information element called a RAW Parameter Set (RPS) included in the beacon [58], and all the nodes have to wake up at the beginning of each beacon interval to receive information related to the RAW number assigned to them. In each RAW-slot, the nodes among the same group contend for channel access using Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA). The AP manages RAW allocation dynamically and assigns a RAW-number only to those stations which have data ready for transmission or receiving. This paves the way to reduce contention and collisions among the nodes to a significant extent, and thus leads to a reduction in network latency and an improvement in overall throughput.

- **Short MAC Header:** IEEE 802.11ah aims at improving throughput by the

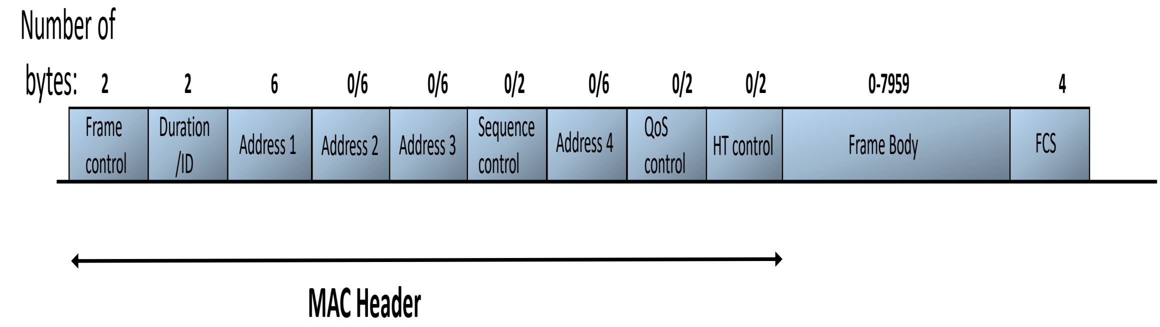

Figure 2.3: Frame structure of 802.11

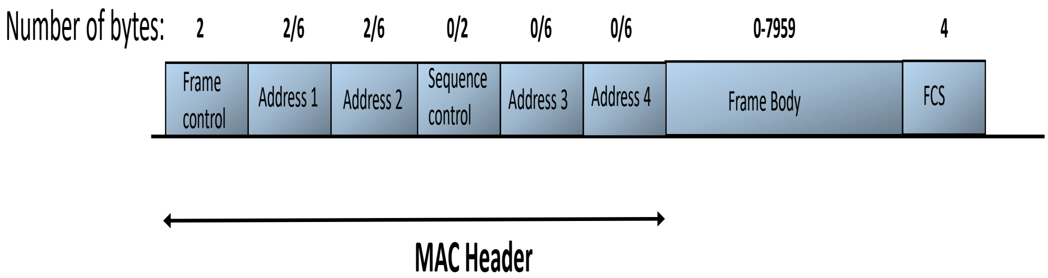

Figure 2.4: Frame structure of 802.11 ah with short MAC header

use of a short MAC header. Figure 2.3 and 2.4 show the frame structure of the legacy IEEE 802.11 and 802.11ah respectively. It is evident that MAC header in 802.11ah is relatively shorter as compared to 802.11. This is achieved in IEEE 802.11ah by replacing a 6-byte MAC address of IEEE 802.11 MAC with 2-byte AID, eliminating duration/ID (so that virtual carrier sensing is not supported), and moving QoS and High Throughput (HT) fields to Signal field (SIG) in PHY header, which results in a saving of 12 bytes in the MAC header. Such a compact header results in an improvement of throughput to a significant degree in IoT especially when short packets are to be transmitted as is the case with most of the sensors in IoT.

- **Speed Frame Exchange:** In order to speed up exchange of packets between

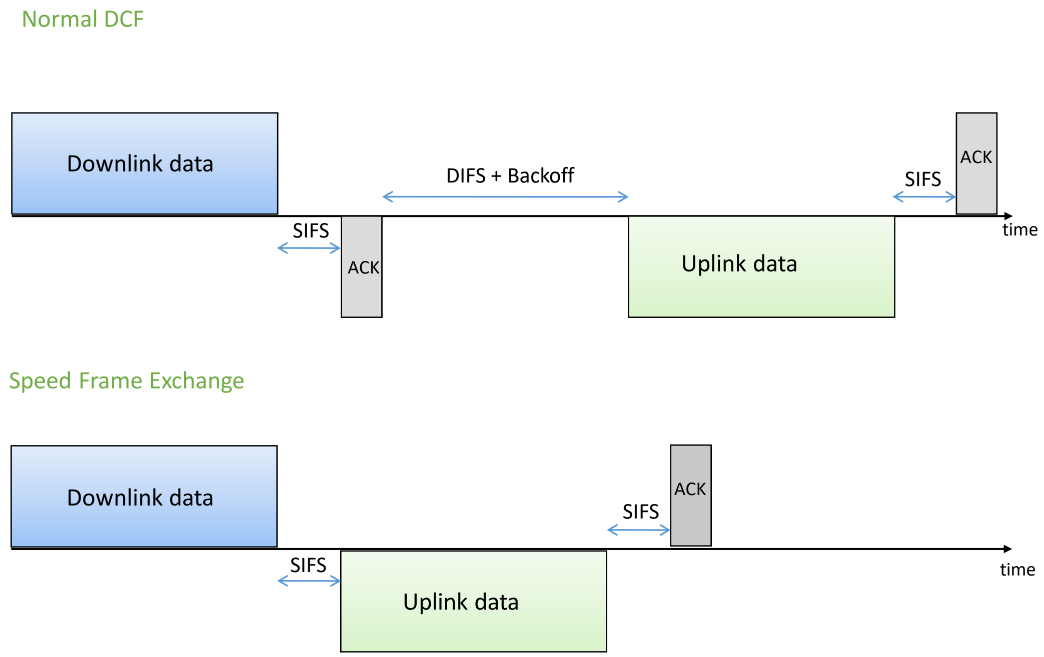

Figure 2.5: Speed frame exchange in IEEE 802.11ah compared with the conventional IEEE 802.11.

AP and receiver, an ACK overhead may be completely eliminated in IEEE 802.11ah by a method called Speed Frame Exchange (SFE). In this method, the receiver node, in response to the received data, sends a data packet instead of ACK. This reduces the gap between uplink and downlink transmissions to SIFS only. Fig 2.5 presents the comparison of a normal Distributed Coordination Function (DCF) scheme with speed frame exchange which reduces the net active time of the node [80]. However, SFE is possible only if both AP and station have data to transmit in reply and is most effective when devices have an equal number of packets in uplink and downlink. This method reduces the overall overhead, improves the throughput gain, saves energy and reduces latency because of a reduction in the time that the nodes have to remain active [78].

As described previously, the nodes in each group contend with each other for the medium access in a RAW slot in IEEE 802.11ah according to DCF and adopt CSMA/CA as a channel access mechanism in the same way as is done in the legacy IEEE 802.11 standard and this is described briefly in the next subsection.

2.2.2 DCF and CSMA/CA

In DCF mode, the nodes communicate directly with each other, without a central coordinator. Based on different ways to transmit data and handle possible collisions, two channel access mechanisms are described as follows [61]:

1. **Basic Access Scheme:** It is based on a two-way handshake procedure in which a node sends its data and waits for the acknowledgment (ACK) from the receiver.
2. **Request to Send/Clear to Send (RTS/CTS) Access scheme:** It is a four way handshake procedure in which the node sends a request to access the channel through a message called RTS. AP replies back with a message CTS which allows it to send its data. The node then sends its data and waits for the ACK.

To access the shared channel, all nodes in DCF contend on the basis of CSMA/CA protocol [81] where a node having a packet of data for transmission senses the shared channel and if the medium remains idle for some duration called Distributed Inter Frame Space (DIFS), the node waits a random amount of time and then starts to transmit its data. However, if the medium becomes busy during the DIFS interval, the node keeps on sensing the medium until it finds the medium idle for DIFS duration. There are four possible stages as follows: [82],[61]

1. **DIFS Stage:** A node having a packet of data first enters into the DIFS stage during which it keeps on sensing the channel. If the channel remains free

for DIFS interval, then the node moves to the next stage, called the Back-off stage. However, in case the channel is sensed busy at any time during the DIFS interval, the node restarts the DIFS stage immediately.

2. **Back-off Stage:** Each node in a back-off stage has a back-off counter whose value is randomly selected from a set of values called the contention window (CW). The size of such a window is $[0 \ \ CW_{max}]$ where CW_{max} is the maximum value a node can take during the backoff stage. In this stage, the size of CW increases becomes double with the number of successive re-transmission attempts, and is called Binary Exponential Back-off (BEB) Window. The backoff counter decrements by 1 with each time slot if the channel is found idle and freezes if the channel is sensed busy at any time during the backoff stage. As the back-off counter reaches zero, the node moves to the transmit stage.

3. **Transmit Stage:** The stage during which the node transmits its data is called the transmit stage. After the payload information is transmitted, the node enters the ACK stage.

4. **ACK Stage:** In this stage, the node waits for the ACK from the receiver. After ACK for the transmitted data is received or the node has waited for the ACK duration, it goes to the DIFS stage so that it can follow the above procedure for transmission of another packet of data.

One important feature of DCF is that all stations try to re-transmit if data is received in error. Each node has two counters called the Short Retry Counter (SRC) and the Long Retry Counter (LRC) whose maximum values are defined by the standard and are the Short Retry Limit (SRL) and the Long Retry Limit (LRL). In an RTS/CTS scheme, a station sends an RTS and waits for a CTS for a duration called the CTS Timeout. If the node detects no CTS in this duration, SRC increments by 1.

Similarly, LRC increases by 1 if a data packet is sent and its acknowledgment is not received during a certain time interval (ACK Timeout). The counters reset to 0 if either the packet is received successfully or they reach their specified maximum limit [83].

2.3 IoT, Fog, Cloud (IFC) paradigm

Propagation and nodal processing delays are the major components of latency which can be reduced by the introduction of fog computing devices as an intermediary between the IoT-Cloud continuum. In this section, we briefly describe the IoT-cloud network architecture, compare fog computing with cloud computing and introduce the IFC paradigm.

Cloud computing refers to a network of multiple computers and servers which are connected to each other over the internet and are usually placed at a centralized location. The conventional IoT-Cloud system is based on two layers as follows:

1. The IoT layer consists of devices such as mobile phones, computers, tablets, sensors, actuators, controllers, etc. Most of the IoT devices are resource constrained in terms of storage and processing capabilities and can not perform theirs task computation on their own.
2. The Cloud layer consists of a group of servers, possessing extensive storage and processing capabilities and are placed at a centralized location, usually far away from the devices in the IoT layer.

The integration of cloud servers with IoT offers many advantages as remotely located cloud servers provide unlimited virtual processing capabilities on demand, facilitate integrating, aggregating and sharing an enormous amount of data generated by the IoT devices and provide highly scalable, low-cost solutions with an improved overall performance. However, there are some downsides when cloud technology is integrated with IoT services, e.g., large down times and outages due to a centralized solution connected over the internet, vulnerability of the system to the cyber-attacks and data losses during the transfer of thousands of gigabytes of information by means of globally connected channels, and most significantly, the high latency due to a remote location of the cloud servers [84].

Fog refers to a network of devices which have relatively moderate computing and storage capabilities as compared to the cloud servers, and are located closer to the end users. The fog computing solutions offer a distributed decentralized infrastructure as compared to the centralized system in cloud computing. However, this warrants the deployment of a large number of fog devices distributed over the entire geographical footprint. Since the fog devices are located closer to the end users, they are able to provide instant connections and perform computation of a significant amount of data on their own, without the need of sending it to the remotely located servers. In this way, fog computing offers low latency solutions to a wide variety of delay-sensitive IoT applications [85]. Table 2.3 summarizes the major differences between the fog and cloud computing, highlighting their salient features [86].

Table 2.3: Comparison of salient features of Fog computing with the cloud computing.

	Cloud	Fog
Architecture	Centralized	Distributed
Location	Remote	Closer to the end users
Computing capabilities	Higher	Moderate
Number of nodes	Few	Very large
Analysis	Long-term	Short-term
Latency	High	Low

Fog computing acts as a mediator between the physical devices and the remote servers, and is basically an extension of cloud computing. Both the fog and cloud computing devices provide storage and computing facilities and complement each other [87]. Such an architecture consists of three layers as follows:

1. IoT Layer.
2. The fog layer consisting of devices with medium processing and storage capabilities and are located near to the end users.
3. The cloud layer consisting of centralized servers.

The fog devices are located in a distributed manner over the geographical footprint and are in close proximity to the end users. Therefore, it may be more efficient to offload data collected by the resource-constrained end user's devices to the fog for processing and temporary storage. This obviates the need for sending all the data to the remotely located cloud servers and only data related to long term analysis and permanent storage is forwarded to the cloud servers. Thus, the introduction of fog devices in the IoT-cloud continuum reduces the overall network traffic and latency. Moreover, a flexible, a scaleable IoT solution having a relatively lower operational cost can be developed by the use of distributed fog devices [88]. Therefore, by regulating which information is to be sent to the cloud server and which is to be processed locally in fog devices, the cloud servers are offloaded and more efficient processing, analysis and data storage in the overall network can be performed.

2.4 Queuing Theoretic model

In order to determine the most feasible way of providing services using the available constrained resources, mathematical models involving queuing systems are exploited. The queuing system is characterized by the probabilistic properties of both arrival and service processes, service structure and service disciplines described as follows.

2.4.1 Service Disciplines in Queue

The service structure determines how many servers are being utilized in the system and how large is the capacity of the system, i.e., the maximum number of service requests that can wait for the services in the system. The service discipline in a queue describes the way in which the incoming requests are dealt. Some of the most frequently utilized service disciplines are [89]:

- First In First Out (FIFO) - the request which arrives earlier is served earlier and leaves the system first.
- Last In First Out (LIFO) - the request which arrives later is served earlier and leaves the system.
- Random Service (RS) - the request is chosen randomly for service provision among those present in the queue.
- Priority - the request is served according to the given order of priority.

In a queuing theoretic model, performance measures such as the number of requests in the system, the number of requests waiting for being served, the response time of a service request, and utilization of servers in the system are investigated.

2.4.2 Traffic Intensity in Queuing System

Considering that the queuing system is based on infinite population with the mean inter-arrival time $\frac{1}{\lambda}$ and mean service time $\frac{1}{\mu}$, then the the traffic intensity of the queuing system is given as

$$\rho = \frac{\text{mean service time}}{\text{mean inter-arrival time}} \tag{2.1}$$
$$= \frac{\lambda}{\mu}. \tag{2.2}$$

Due to the random nature of arrival and service times in the system, the queues are likely even when the mean service time is less than mean inter-arrival time of requests i.e., $\rho < 1$. The system is considered as overloaded when $\rho > 1$, i.e., the rate of arrival of requests to the system is greater than the rate at which they are being served.

2.4.3 Parameters of Queuing System

In order to gauge the performance of a queuing system, it is characterized by the following four important parameters:

- **The length of the queue**, L_q: expected number of the requests waiting in the queue for the service provision.
- **The length of the queuing system**, L_s: expected number of requests in the system including the ones being served.
- **Waiting time in the queuing system**, W_s: expected waiting time for the request to be served.
- **Waiting time in the queue**, W_q: expected waiting time for a request before it is served.

2.4.4 Little's Law

Assuming that the parameters characterizing the system remain the same over time, then the relationship between the length of the queue and the time associated with the it is governed by the well known **Little's law** as [90]

$$L_s = \lambda W_s, \tag{2.3}$$

$$L_q = \lambda W_q. \tag{2.4}$$

2.4.5 Kendall's Notation

In queuing-theoretic models, we assume that the inter-arrival times of the service requests and their service times are are independent and identically distributed (i.i.d) random variables. Let $A(t)$ be the arrival process representing the distribution of inter-arrival times of the service requests, i.e.,

$$A(t) = \Pr\{\text{inter-arrival times} < t\}, \tag{2.5}$$

and the time it takes the system to process the service request, denoted by $B(t)$, is

$$B(t) = \Pr\{\text{service times} < t\}. \tag{2.6}$$

The parameters of the queuing system model are described by a notation introduced by **Kendall** as $A/S/m/K/N/D$, where A represents the distribution function of the inter-arrival times, S represents the distribution function of the service times, m is the number of servers, K is the capacity i.e., the number of requests in the system, N is the size of the population from which the requests are generated and D is the service discipline to serve the requests. In this notation, D, N and K are omitted if the service discipline is FIFO, the population size is infinite and the system can accommodate an infinite number of requests. Some of the famous queuing theoretic models are discussed in the following subsections.

2.5 Markov Chain Model

A stochastic process, in general, is a family of random variables denoted by X_θ, where θ belong to some index set Θ. The stochastic processes are often used to model the processes that develop in time or in space according to the rules of probability theory. In time domain, the stochastic process is called the discrete time when Θ represents specific time points and is a set of integers and the subscript θ of random variables is replaced by n. Therefore, the family of random variables belonging to discrete time stochastic process is denoted by X_n. Similarly, the stochastic process in time domain is called the continuous time when Θ is some interval of the real line and θ is replaced by t, and family of continuous time stochastic process's random variables is denoted by $X(t)$.

A random variable may depend on the previous values of the process, e.g., a random variable, X_n associated with the discrete time stochastic process may depend on the previous values $X_{n-1,}X_{n-2,}......$ or a random variable, $X(t)$ associated with the continuous time stochastic process may depend on the values $X(u)$ for $u < t$. The conditional distribution in which the process at time t_k depends on all the previous values is of the form

$$\Pr\{X_{t_k}|X_{t_{k-1}}, X_{t_{k-2}}, ..., X_{t_1}\}, \tag{2.7}$$

where $t_k > t_{k-1} > ....t_1$.

In order to simplify the forecast about the future state of a stochastic processes, an assumption, called the Markov Property, that most of the processes observe is: given the present state $X_{t_{k-1}}$ in a stochastic process, the future state X_{t_k} is independent of all the previous states $X_{t_{k-2}}X_{t_{k-3}}, ..., X_{t_1}$. In mathematical terms, the Markov Property is given as

$$\Pr\{X_{t_k}|X_{t_{k-1}}, X_{t_{k-2}}, ..., X_{t_1}\} = \Pr\{X_{t_k}|X_{t_{k-1}}\}. \tag{2.8}$$

2.5.1 Important Definitions

In order to describe a process that changes over time, Markov chains are used. The change from one state to another in a process is described by the probability distributions in the Markov Chain where the conditional distribution of future states depends on the most recent state when some information of the past states is already given. Therefore, Markov chains are developed for the stochastic processes where it is appropriate to assume that the process satisfies the Markov Property and are an important mathematical tool in evaluation and analysis of stochastic processes.

State space S

In a Markov chain, the state space S is a set of values that the random variable X_{t_k} may take on. Let the state space of the Markov chain is represented as $S = \{1, 2, ..., M\}$ where M is a finite value, then the random variable can take on the values from S and the Markov chain is said to be finite. Note that the Markov chain is said to be infinite for the case when $M = \infty$.

Initial distribution

The probability distribution of the Markov chain at time $t = 0$ is called the initial distribution. Let the probability that the Markov chain begins with the state i , where $i \in S$ i.e., $Pr\{X_{t_0} = i\}$ be denoted by $\pi_0(i)$. Then, $\pi_0(i) \geq 0, \quad \forall\, i \in S$, and $\sum_{i \in S} \pi_0(i) = 1$.

State transition probabilities

Let $P_{i,j}$ represents the conditional probability that the process is in state i of Markov chain at any time t_k and then jumps to state j at time t_{k+1}. This is called the state transition probability and is given as

$$P_{i,j} = \Pr\{X_{t_{k+1}} = j \mid X_{t_k} = i\}. \tag{2.9}$$

For a Markov chain consisting of M states, the state transition matrix P is a matrix of size $M \times M$ where $P_{i,j}$ is an element at the i^{th}row and the j^{th} column of the matrix P and represents the probability of transition from state i to state j in the Markov chain.

Probability of following a specific path

The probability that a Markov chain follows a specific path in the sequence $i_0 \rightarrow i_1 \rightarrow i_2 \rightarrow ..... \rightarrow i_n$ is given as

$$\begin{aligned} &\Pr\{X_{t_0} = i_0, X_{t_1} = i_1, ..., X_{t_n} = i_n\} \\ &\quad = \Pr\{X_{t_0} = i_0\} * \Pr\{X_{t_1} = i_1 | X_{t_0} = i_0\} * ....\Pr\{X_{t_n} = i_n | X_{t_{n-1}} = i_{n-1}\}. \end{aligned}$$

By using the Markov Property, we get

$$\Pr\{X_{t_0} = i_0, X_{t_1} = i_1, ..., X_{t_n} = i_n\} = \pi_0(i_0) * P_{i_1,i_0} * P_{i_2,i_1} * ....P_{i_n,i_{n-1}}. \quad (2.10)$$

Probability that the chain is at state j after n steps

Let $\pi_n(i)$ denote the probability distribution that the process is at state i at time $t = t_n$, i.e., $\pi_n(i) = \Pr\{X_{t_n} = i\}$. Then, the probability that the process is at state j at time t_{n+1} is

$$\pi_{n+1}(j) = \Pr\{X_{t_{n+1}} = j\}.$$

By use of total probability law, we get

$$\begin{aligned}\pi_{n+1}(j) &= \sum_{i=1}^{M} \Pr\{X_{t_n} = i\}\Pr\{X_{t_{n+1}} = j|X_{t_n} = i\} \\ &= \sum_{i=1}^{M} \pi_n(i)P_{i,j}.\end{aligned}$$

In matrix notation, let π_n be a row vector such that $\pi_n = (\pi_n(1), \pi_n(2), ..., \pi_n(M))$. Then, the probability that the chain is in state j at time t_{n+1} in matrix notation is

$$\pi_n = \pi_0 P^n, \quad (2.11)$$

where P is the state transition matrix.

Chapter 3

Task Offloading and Service Discovery in the IFC Paradigm

In this chapter, we develop a queuing-theoretic model in the IFC paradigm where the devices in both IoT and Fog layer are considered to have two queues such as the computation queue for processing of of tasks locally and the transmission queue for offloading the task to the devices in upper layer, and the servers in the cloud have a computation queue only. We present an algorithm for task offloading by the devices both in the IoT and Fog layers with an aim to minimize latency after discovering the capabilities of all the fog and cloud servers present in the network using a service discovery protocol (SDP).

3.1 Introduction

In order to cope with the low responsiveness of cloud-based networks, fog computing has surfaced as an attractive solution especially when a large number of IoT devices are deployed in the urban or sub-urban environment. Most of these IoT devices have limited resources in terms of processing power, transmission power and storage capacity, e.g., sensors deployed to monitor environment and traffic, the ones used in smart fire alarm, smart door lock, smartwatches, smart home appliances, fitness trackers, etc. may not have enough resources to process and analyze data on their own or may cause extra-ordinary processing delays if they attempt to process data locally. In such a scenario, the data packets arriving at the IoT devices may have to face long processing delays, and the data packets may have to be dropped when their buffers become full, which may cause a loss of information. One possible solution is to offload task data to the cloud servers which usually have immense processing capability and can process the task data. However, the cloud servers are generally present at a centralized location and it may take quite a long propagation time to transmit data from the IoT to cloud servers. Moreover, it causes an extraordinary burden on the network to transmit the data of a huge numbers of devices deployed in various IoT solutions. Therefore, cloud-based IoT solutions may suffer due to their low responsiveness.

The latency in IoT solution may be reduced by the use of an additional layers of devices in between the IoT and cloud servers, called the Fog layer as shown in Fig. 3.1. The devices in the fog layer are usually located closer to the IoT devices and have low to moderate processing and storage capacity. This may help in the reduction of both the processing and propagation latency. However, it may lead to even longer delays in processing the tasks if data packets are offloaded to a fog device which is either incapable of handling the task data or already has quite a long computation

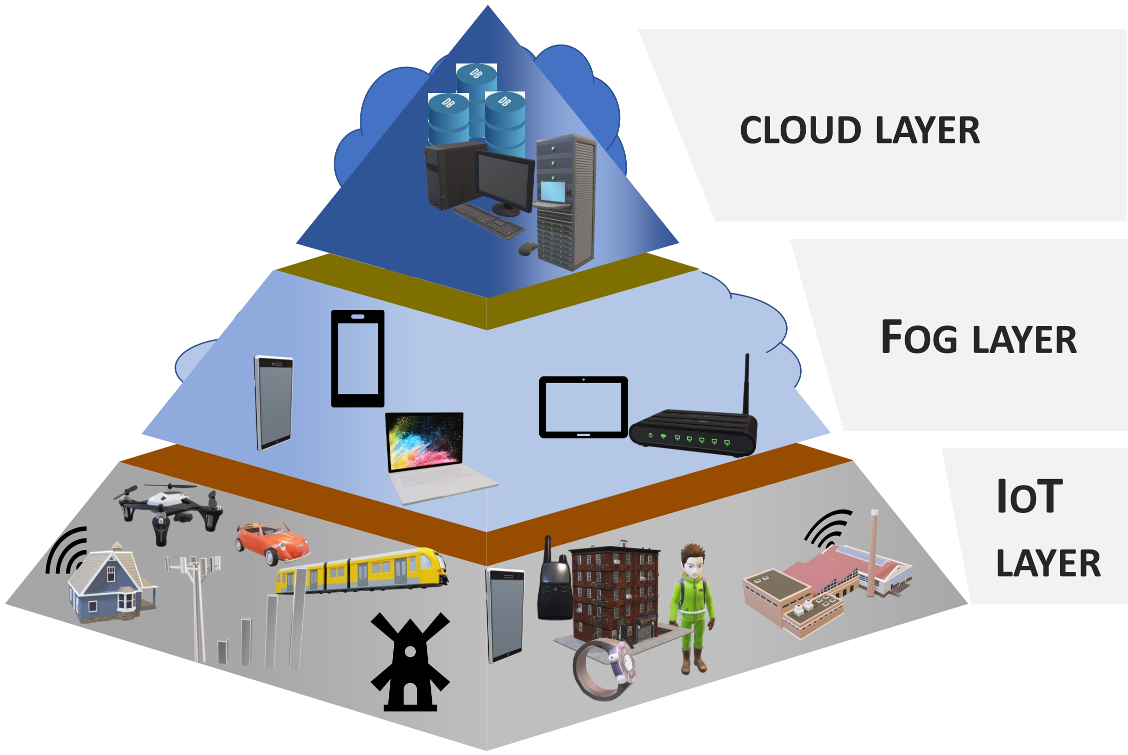

Figure 3.1: IoT-Fog-Cloud paradigm.

queue, as the data may then have to be offloaded again to some other entity in the network such as a neighboring fog node or the cloud server. Therefore, task offloading poses a new challenge in the IFC paradigm and necessitates the need for an algorithm for task offloading which could address the issue of low responsiveness of various IoT solutions.

3.1.1 Related Work

In [41], the authors propose a general framework for the IFC paradigm and offloading policy between fog and cloud servers with an aim to minimize latency. In [42], Alameddine et al. formulated the dynamic task offloading problem mathematically, and a strategy for task offloading to nearby devices is devised by Chen et al. in

[43]. Another strategy based on the density of IoT devices to offload tasks from IoT devices to the servers is analyzed in [44]. The authors formulated a computational offloading game to model the competition between IoT users and efficiently allocate the processing power of fog nodes in [45]. A cooperative task offloading policy between two fog nodes is formulated in [32] and an analysis to offload tasks among multiple fog nodes is presented in [33]. In all these works, the tasks are offloaded from one layer to the other without any discovering the capabilities of the devices in the other layer, without the use of any service discovery protocol and may prove inefficient for an ultra-dense deployment of IoT devices, especially when heavy tasks are offloaded to fog devices in FL.

3.1.2 Contribution

In this chapter, we

- develop a queuing theoretic model in the IFC paradigm and determine queuing delays in devices IoT, Fog and cloud layer,
- find the capabilities of fog nodes by the use of a service discovery protocol
- devise an algorithm to offload task data among these layers which aims to reduce latency in the network.

3.2 System Model

In this chapter, we consider an IFC network architecture $\Psi(\mathcal{I}, \mathcal{F}, \mathcal{C})$ comprising $\mathcal{I}$ devices in IoT Layer (IL), $\mathcal{F}$ devices in Fog Layer (FL) and $\mathcal{C}$ servers in the Cloud Layer (CL) where it is assumed that the devices in each both IL and FL have a limited storage and processing capability, whereas the cloud servers have vast

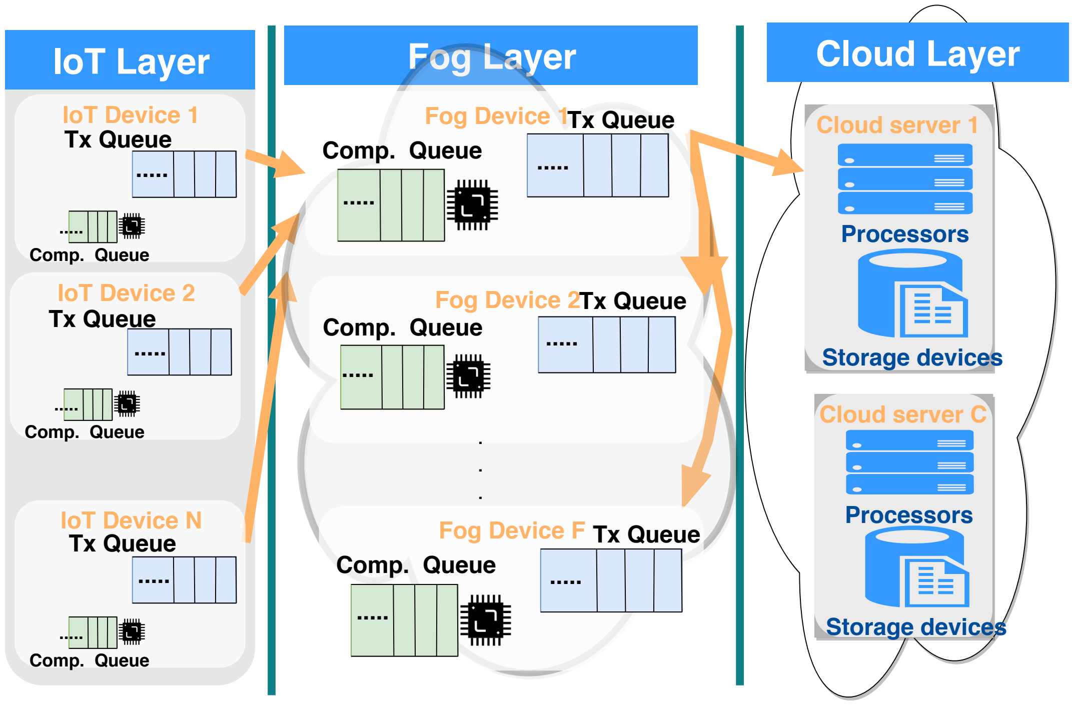

Figure 3.2: Queuing-theoretic IFC model.

resources for processing, storage and data analysis. Therefore, the devices in both IL and FL may have two types of queues, i.e., the computation queue formed by the tasks that arrive while some of the previous ones are yet to be processed, and the transmission queue which is formed when more than one tasks are required to be offloaded to the other entities in the network.

Such a queuing-theoretic based IFC network architecture used in our system is shown in Fig. 3.2. We assume that the arrival of tasks at the i^{th} device in IL, $\forall\ i \in \mathcal{I}$, f^{th} device in FL, $\forall\ f \in \mathcal{F}$, and at cloud servers in CL, is according to Poisson processes at rates λ_i, λ_f, and λ_c respectively. Moreover, we assume that all these devices process the tasks on the basis of a First Come First Served (FCFS) queuing discipline. Considering the limited processing capability of devices in both IL and

FL, we assume that a proportion of tasks arriving at each of these devices is offloaded to the other entities, e.g., IL's devices can offload tasks to fog nodes or cloud servers, and fog nodes can offload tasks to the cloud server. We define the offloading ratio as the ratio of the number of tasks to be offloaded to the total number of tasks arriving at a given time. Let $\chi_i,\ \chi_f$ be the offloading ratio for the i^{th} device in IL and the f^{th} device in FL respectively, then $\chi_i\lambda_i$ tasks arrive in the transmission queue of the i^{th} device in IL, and $\chi_f\lambda_f$ arrive in the transmission queue of the f^{th} device in FL for offloading, whereas $(1-\chi_i)\lambda_i$ tasks arrive in the computation queue of the i^{th} device in IL, and $(1-\chi_f)\lambda_f$ tasks arrive in the computation queue of the f^{th} fog device so that they can be processed locally.

Once tasks arrive in the transmission queue of device i in IL for offloading, we assume that they are offloaded either to a device in FL or to a cloud server at a rate $\hat{\mu}_i$. Also, we let the tasks arriving in the transmission queue of device f in FL be offloaded to the cloud server at a rate $\hat{\mu}_f$. Similarly, let the tasks arriving in the computation queue of device i in IL be processed locally at a rate μ_i, and the tasks arriving in the computation queue of fog device f processed locally at a rate μ_f. We define the service duration for tasks in the computation queue of a device in IL as the duration of time to process the task locally in the computation queue of an IoT device and is equal to $\frac{1}{\mu_i}$, and the service duration for tasks in the transmission queue of the device in IL as the duration of time to offload a task either to FL or CL, and is equal to $\frac{1}{\hat{\mu}_i}$. Similarly, the service duration for tasks in the computation queue of a device in FL as the duration of time to process the task locally in the computation queue of a fog device in FL and is equal to $\frac{1}{\mu_f}$, and the service duration of tasks in the transmission queue of the device in FL as the duration of time to offload a task either to a neighbouring fog device in FL or to CL, and is equal to $\frac{1}{\hat{\mu}_f}$. The service duration for tasks in the cloud server is the duration of time required to process the the tasks in the cloud server and is equal to $\frac{1}{\mu_c}$. In our system model, it is assumed that the service duration is independent and exponentially distributed for devices in computation and transmission queues of both IL and FL as well as in computation queue of cloud server. Before offloading the task data from the IL to a particular fog node in FL, we propose a service discovery architecture that isolates the IL from the FL by a Fog Layer Gateway (FLGW), and data is offloaded only to a fog node which is capable to process the task in our system model. FLGW is a central node that discovers all devices and the services that they can render in the FL in a domain $\mathcal{F}$ by the use of a service discovery protocol (SDP).

Considering the limited storage capacity of both the IoT and fog devices, we assume that a node i in IL can accommodate C_i tasks in its computation queue, and $\hat{C}_i$ tasks in its transmission queue, whereas node f in FL can accommodate C_f tasks in its computation queue, and $\hat{C}_f$ tasks in its transmission queue. However, as the cloud servers at a centralized location have a vast storage capacity to accommodate the task data in practice, we assume that cloud servers can accommodate an infinite number of tasks in its computation queue, i.e., $C_c = \infty$. With a vast processing power of the cloud servers in the network, we consider that the tasks are processed by the cloud server locally once they are arrive to it either from IL or from FL in our system model.

In the following subsections, we first present two different modes of SDP to discover resources in FL, and then develop a queuing-theoretic model to estimate waiting time in both IL and FL in both processing and offloading task data in the IFC paradigm.

3.2.1 Service Discovery Protocol

There are three possible stages in SDP, named as Discovery, Registration and Update stages. During the discovery stage, the FLGW and fog nodes communicate with each other for the first time and discover the hosting services of each other. In the registration stage, the fog nodes report their resources and estimated time to process the requests in the queue to the FLGW which then saves this information so that the look up on the available resources in FL can be provided. The update stage is used to refresh or remove records about a fog node. FLGW of domain $\mathcal{F}$ then announces the estimated waiting time of the fog nodes and the services that each node can render to the nodes in IL. The devices in IL record the estimated waiting times and the service capability of the fog nodes in a Resource table that they maintain, and utilize it when making a decision as to which fog node to offload a task data. The data corresponding to the estimated waiting time, round trip time, node ID, and service capability of each fog nodes in domain $\mathcal{F}$ is recorded in the Resource table and is updated by both FLGW and the IoT device itself; the FLGW announces the estimated waiting times and services offered by the fog nodes in the domain, and IoT devices measure the round-trip delay in between IL and FL. SDP can have two different modes of operation, called poll mode and the push mode described as follows:

1. **Poll mode:** In the poll mode, the sequence of information flow during discovery, registration and update stages is shown in fig. 3.3 . In this mode, FLGW polls the fog nodes in FL and discovers their presence and their hosting services by sending a FLGW Discovery (FLGWD) message periodically. The fog nodes receive the FLGWD message and respond with a Fog Node Discovery (FND) message which contains information about it such as the status of fog nodes and the expected time until its expiration. In this way, FLGW keeps

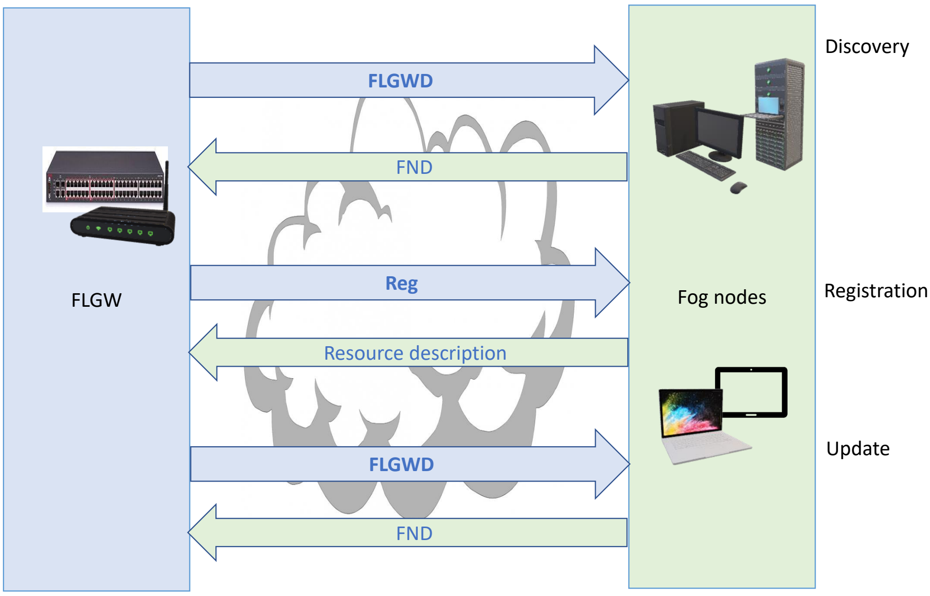

Figure 3.3: Discovery, registration and update stage of the poll mode in service discovery protocol.

a record of FND messages in a database and moves either to the registration stage when the FND message it has received is the first one from the fog node or to update stage if it has already received a FND message from the same fog node previously. During the registration stage, the FLGW sends a requests to fog node, receives a response from the fog node containing information about the resources associated with it, and updates them in its database to enable look-ups at a later stage. In the update stage, the fog node sends a FND message (either independently or as a response to the FLGWD message) containing the information about the expected time until expiration, and a flag to indicate FLGW to update information about its resources, or its attributes.

2. **Push Mode:** In push mode, the sequence of information flow during discovery, registration and update stages is shown in Fig. 3.4. In this mode, the fog nodes announce their presence periodically by sending a FND message to the FLGW. The FLGW does not respond to the FND message during the

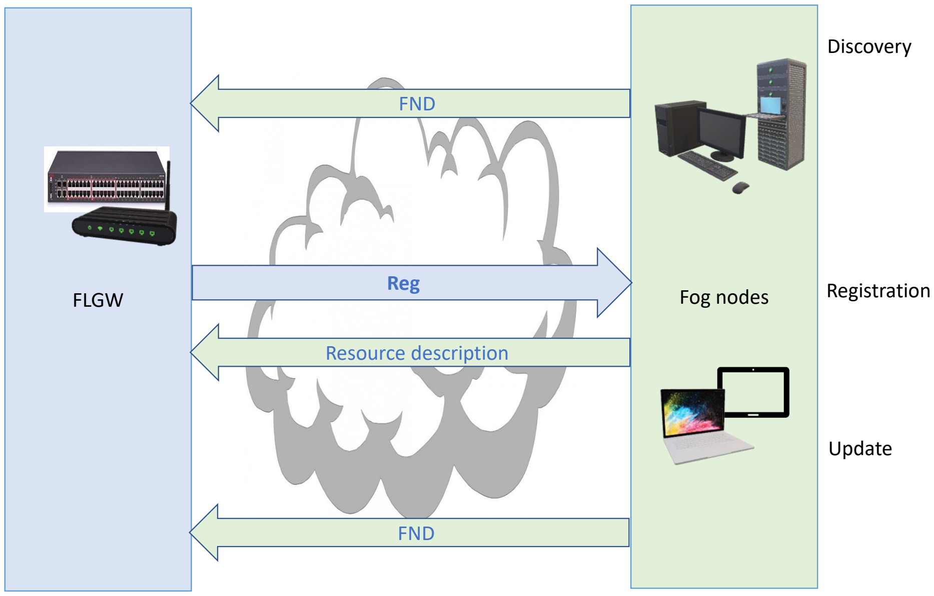

Figure 3.4: Discovery, registration and update stage of push mode in service discovery protocol.

discovery stage; it rather records the contents of FND and moves either to registration stage if it has received FND from a particular fog node for the first time, or to the update stage if it has received a FND message from this fog node previously. During the registration stage, the FLGW records the resources of the fog nodes discovered during the discovery stage in a manner similar to the poll mode. Similarly, during the update stage in the push, the fog node refreshes its information in FLGW if some of its parameters have changed by sending FND message.

Finally, a list of the most frequently used symbols, along with their descriptions is presented in table 3.1.

Table 3.1: A list of most frequently used symbols and their description.

Description	**Symbol**
The rate at which tasks arrive at IoT device i	λ_i
The rate at which tasks arrive at fog node f	λ_f
The rate at which tasks arrive at clod server c	λ_c
The rate at which tasks are processed in computation queue of IoT device i	μ_i
The rate at which tasks are processed in computation queue of fog node f	μ_f
The rate at which tasks are processed in computation queue of cloud server c	μ_c
The rate at which tasks are offloaded from transmission queue of IoT device i	$\hat{\mu}_i$
The rate at which tasks are offloaded from transmission queue from fog node f	$\hat{\mu}_f$
The ratio of tasks offloaded to the total tasks arriving at IoT device i	χ_i
The ratio of tasks offloaded the total tasks arriving at IoT device f	χ_f
Size of the data packet arriving at IoT device i	ℓ_i
Storage capacity in computation queue of IoT device i	$C_i\ell_i$
Storage capacity in transmission queue of IoT device i	$\hat{C}_i\ell_i$

3.2.2 Queuing Delays

In order to predict the waiting time to process a task, or offload it by the devices in all three layers, i.e., IL, FL and CL, according to the assumptions described in our proposed system model, we first evaluate the queuing delays for both the computation queues and the transmission queues of the devices in this subsection. Two important theorems useful in finding the waiting time for processing tasks in devices are as follows.

Theorem 3.1

For an M/M/1/K queuing theoretic model, where the tasks arrive according to the Poisson process at a rate λ, i.e., the inter-arrival times are independent and exponentially distributed random variables with parameter λ, the service times are assumed to be independent and exponentially distributed with parameter μ, and there is a single server in the system. Let $N(t)$ denote the number of tasks in the system at any time t, then the mean time for processing a task in a system that has a capacity to accommodate K tasks in the system including the one under service, is

$$E[W] = \frac{E[N]}{\lambda(1 - P_K)},$$

where $E[N]$ is the mean number of tasks in the queuing system and is given as $E[N] = \frac{\rho(1-(K+1)\rho^k + K\rho^{K+1})}{(1-\rho)(1-\rho^{K+1})}$, and ρ is the utilization ratio, i.e., $\rho = \frac{\lambda}{\mu}$, and $P_k = \frac{\rho}{\sum_{j=0}^{k} \rho^j}, \quad \forall\ k = 0, 1,\ \ldots, K.$

Proof: See Appendix 3A.

Theorem 3.2

For an M/M/1 queuing theoretic model, where the tasks arrive according to the Poisson process at a rate λ, the service times are assumed to be independent and exponentially distributed with parameter μ, and there is a single server in the system, the mean time for processing a task in a system is

$$E[W] = \frac{1}{\mu - \lambda}.$$

Proof: See Appendix 3B.

Let S_i be the storage capacity of IoT device i in the FL which is divided into computation and transmission queue in such a way that the storage capacity of computation queue in IoT device i is $C_i = \chi_i S_i$, and the storage capacity of the transmission queue of the fog node f is $\hat{C}_i = (1-\chi_i)S_i$,then the queuing model can be developed as $M/M/1/C_i$ for the device i in the IL where the tasks arrive according to the Poisson process at a rate $(1-\chi_i)\lambda_i$, and are processed according to exponential distribution at the rate of μ_i on FCFS basis, . Therefore, the mean response time to process a task by the IoT device i is found by using theorem 3.1 as

$$E[W_i] = \frac{E[N_i]}{(1-\chi_i)\lambda_i(1-P_{C_i})}, \tag{3.1}$$

where $E[N_i] = \frac{\rho_i\left(1-(C_i+1)(\rho_i)^{C_i}+C_i(\rho_i)^{C_i+1}\right)}{(1-\rho_i)\left(1-(\rho_i)^{C_i+1}\right)}$, $\rho_i = \frac{(1-\chi_i)\lambda_i}{\mu_i}$,and $P_{C_i} = \frac{(\rho_i)^{C_i}}{\sum_{j=0}^{C_i}(\rho_i)^j}$. In our system model, the tasks arriving in the buffer of IoT device are sent to the transmission queue at the rate of $\chi_i\lambda_i$. Then, once a task is sent to the transmission queue for offloading, it faces a waiting delay before it is actually transmitted either to the fog device or to the cloud server over a wireless channel. As described in the system model, the transmission queue of a device i in IL having the capacity $\hat{C}_i$ can be modeled as $M/M/1/\hat{C}_i$ where the tasks arrive at the rate of $\chi_i\lambda_i$ according to the Poisson process and are transmitted to the FL with probability p_i^f at a rate of $\hat{\mu}_i^f$ following an exponential distribution. Therefore, the mean time to transmit the task data from IoT device i to the fog node f is found by using theorem 3.1 as

$$E[\hat{W}_i^f] = \frac{E[\hat{N}_i]}{p_i^f \chi_i \lambda_i (1 - P_{\hat{C}_i})}, \tag{3.2}$$

where $E[\hat{N}_i] = \frac{\hat{\rho}_i \left(1 - \left(\hat{C}_i + 1\right)(\hat{\rho}_i)^{\hat{C}_i} + \hat{C}_i (\hat{\rho}_i)^{\hat{C}_i + 1}\right)}{(1-\hat{\rho}_i)\left(1 - (\hat{\rho}_i)^{\hat{C}_i + 1}\right)}$, $\hat{\rho}_i = \frac{p_i^f \chi_i \lambda_i}{\hat{\mu}_i}$. For a data packet of size ℓ_i, the service rate for offloading task from device i in IL to fog node f in FL is

$$\hat{\mu}_i^f = \frac{p_i^f \chi_i \lambda_i \ell_i}{R_i^f}, \tag{3.3}$$

where R_i^f is the data rate for transmission of data packet over a link in between device i to node f. For a device i offloading data in between IL and FL through a channel of bandwidth B_i^f, and having a transmission power of $P_{tx,i}$, the data rate of transmission is given as

$$R_i^f = B_i^f \log_2 \left(1 + \frac{P_{tx,i} h_i^f}{P_N + \sum_{j=1, j \neq i}^{\mathcal{I}} P_{tx,j} h_j^f}\right), \tag{3.4}$$

where P_N is the noise power, and h_i^f is the channel gain between device i and the fog node f. Similarly, the tasks are transmitted to the CL with probability p_i^cat a rate of of $\hat{\mu}_i^f$ following an exponential distribution. Therefore, the mean time to transmit the task data from IoT device i to the cloud server c is found by using theorem 3.1

as

$$E[\hat{W}_i^c] = \frac{E[\hat{N}_i]}{p_i^c \chi_i \lambda_i (1 - P_{\hat{C}_i})}, \tag{3.5}$$

where $E[\hat{N}_i] = \frac{\hat{\rho}_i \left(1 - \left(\hat{C}_i + 1\right)(\hat{\rho}_i)^{\hat{C}_i} + \hat{C}_i (\hat{\rho}_i)^{\hat{C}_i + 1}\right)}{(1-\hat{\rho}_i)\left(1 - (\hat{\rho}_i)^{\hat{C}_i + 1}\right)}$, $\hat{\rho}_i = \frac{p_i^c \chi_i \lambda_i}{\hat{\mu}_i^c}$, and the service rate while offloading task from device i in IL to cloud server c in CL for a data packet of size ℓ_i is

$$\hat{\mu}_i^c = \frac{p_i^c \chi_i \lambda_i \ell_i}{R_i^c}, \tag{3.6}$$

where R_i^c is the data rate for transmission of data packet over a link in between device i to server c. For a device i offloading data through a channel of bandwidth B_i^c in between IL and CL, and having a transmission power of $P_{tx,i}$, the data rate of transmission is given as

$$R_i^c = B_i^c \log_2 \left(1 + \frac{P_{tx,i} h_i^c}{P_N + \sum_{j=1, j \neq i}^{C} P_{tx,j} h_j^c}\right), \tag{3.7}$$

where P_N is the noise power, and h_i^c is the channel gain between device i and the cloud server c.

Let S_f be the storage capacity of a fog node f in the FL which is divided into computation and transmission queue in such a way that the storage capacity of computation queue in fog node f is $C_f = \chi_f S_f$, and the storage capacity of the transmission queue of the fog node f is $\hat{C}_f = (1 - \chi_f) S_f$, then the node f having a capacity to accommodate C_f tasks in its computation queue, where the tasks arrive according to the Poisson process at a rate $(1-\chi_f)\lambda_f$, and are processed according to exponential distribution at the rate of μ_f, then the queuing model can be developed as $M/M/1/C_f$ for each device f in FL. Therefore, the mean response time to process a task by the fog node f is found by using theorem 3.1 as

$$E[W_f] = \frac{E[N_f]}{(1-\chi_f)\lambda_f(1-P_{C_f})}, \tag{3.8}$$

where $E[N_f] = \frac{\rho_f\left(1-(C_f+1)(\rho_f)^{C_f}+C_f(\rho_f)^{C_f+1}\right)}{(1-\rho_f)\left(1-(\rho_f)^{C_f+1}\right)}$, $\rho_f = \frac{(1-\chi_f)\lambda_f}{\mu_f}$,and $P_{C_f} = \frac{(\rho_f)^{C_f}}{\sum_{j=0}^{C_f}(\rho_f)^j}$. The fog node f having a capacity to accommodate $\hat{C}_f$ tasks in its transmission queue may offload tasks to the cloud server, which can then be modeled as $M/M/1/\hat{C}_f$ where the tasks arrive at the rate of $\chi_f\lambda_f$ following the Poisson process and are transmitted at a rate of $\hat{\mu}_f$ according to an exponential distribution. Therefore, the mean time to transmit the task data from fog node f is found by using theorem 3.2 as

$$E[\hat{W}_f^c] = \frac{E[\hat{N}_f]}{\chi_f\lambda_f(1-P_{\hat{C}_f})}, \tag{3.9}$$

where $E[\hat{N}_f] = \frac{\hat{\rho}_f\left(1-(\hat{C}_f+1)(\hat{\rho}_f)^{\hat{C}_f}+\hat{C}_f(\hat{\rho}_f)^{\hat{C}_f+1}\right)}{(1-\hat{\rho}_i)\left(1-(\hat{\rho}_i)^{\hat{C}_i+1}\right)}$, $\hat{\rho}_f = \frac{\chi_f\lambda_f}{\hat{\mu}_f}$. For a data packet of size ℓ_i, the service rate for offloading task from fog node f in FL to the cloud server c in CL is

$$\hat{\mu}_f = \frac{\chi_f\lambda_f\ell_i}{R_f^c}, \tag{3.10}$$

where R_f^c is the data rate for transmission of data packet over a link in between node f to server c. For a fog node f offloading data through a channel of bandwidth B_f^c and having a transmission power of $P_{tx,f}$, the data rate of transmission is given as

$$R_f^c = B_f^c \log_2\left(1 + \frac{P_{tx,f}h_f^c}{P_N + \sum_{j=1,j\neq i}^{\mathcal{F}} P_{tx,j}h_j^c}\right), \tag{3.11}$$

where P_N is the noise power, and h_f^c is the channel gain between fog node f and the server c.

Note that when SDP is used, the tasks are only offloaded to a fog device if it has the capability and resource-availability to process the data. In such a scenario, the fog nodes do not have to offload any data to the cloud server and hence, $\chi_f \to 0$ by the use of SDP in FL. In practice, the computational capability of the cloud server is quite large[91], and therefore, we assume that the cloud servers have an infinite storage capacity. For the tasks arriving at the cloud server according to the Poisson process at a rate λ_c, and are processed by the cloud servers at the rate of μ_c following an exponential distribution, then the computation queue in the cloud server can be modeled as $M/M/1$, and hence the the delay in the computation queue of the cloud server is found by using theorem 3.2 as

$$E[W_c] = \frac{1}{\mu_c - \lambda_c}. \tag{3.12}$$

For a finite arrival rate at the cloud server, λ_c, as $\mu_c \to \infty$, $E[W_c] \to 0$, and for small data packets in IoT, the time required to process a task data is negligibly small i.e., $\mu_c \to \infty$. Hence, the delay in processing a data packet in the cloud server approaches zero.

3.3 Task Offloading Strategy

In this subsection, we propose a strategy for offloading tasks between devices in the IFC paradigm. After discovering the services offered by the nodes in FL. In IL, a task arriving at IoT device i is either processed locally, or is offloaded to a node f in the FL if it has the capability to process the task. Given that node i in the IL finds a device in FL that can process the task, latency in processing the task when it is offloaded from device i in IL to the fog node f in FL is equal to the sum of the delay in transmission of task from device i to the fog node f and the delay in the computation queue of the fog node f, and is found as

$$\mathcal{L}^{i \to f} = E[\hat{W}_i^f] + E[W_f] + E[T_i^f]. \tag{3.13}$$

where $E[T_i^f]$ is the mean propagation delay between IL and FL. However, when node i in IL is unable to find a device capable of processing the task in fog layer, it offloads its data to the cloud layer, and the latency in processing the task after offloading from device i to the cloud server c is found as

$$\mathcal{L}^{i \to c} = E[\hat{W}_i^c] + E[W_c] + E[T_i^c]. \tag{3.14}$$

where $E[T_i^c]$ is the mean propagation delay between IL and CL. Similarly, the latency in processing the task when it is offloaded from fog node f to the cloud server c is found as

$$\mathcal{L}^{f \to c} = E[\hat{W}_f^c] + E[W_c] + E[T_f^c]. \tag{3.15}$$

where $E[T_f^c]$ is the mean propagation delay between FL and CL. The strategy for either processing or offloading the tasks from one layer to the other in the IFC continuum along with execution of SDP is given in Algorithm 3.1. According to the proposed algorithm, the task is processed locally by the IoT device if the response time in the computation queue of the device is less than the time it takes the packet in offloading to either the FL or CL and then getting it processed by the device to which it is offloaded. The task data is offloaded to a fog node in FL when the least latency in offloading and processing the task from IoT device to the fog device is less than (or equal to) that in offloading and processing the task from IoT device to the cloud server, i.e., $\mathcal{L}^{i\to f} \leq \mathcal{L}^{i\to c}$. The available resources and estimated time in processing of tasks is obtained by running the SDP in between FLGW and the devices in FL in our proposed algorithm, and a task is offloaded to the fog node only when the fog node has sufficient resources to process the task. However, the task data is offloaded to the cloud server from IoT device in case when the least latency in offloading the task from IoT layer to the fog layer and then getting it processed once the data packet is propagated to the fog node is greater than that in offloading the task from IoT layer to the cloud server and then getting it processed once the data packet has propagated to the cloud server, i.e., the task is offloaded from IL to cloud server when $\mathcal{L}^{i\to f} > \mathcal{L}^{i\to c}$. A flow chart illustrating the algorithm for task offloading in IFC is shown in Fig. 3.5.

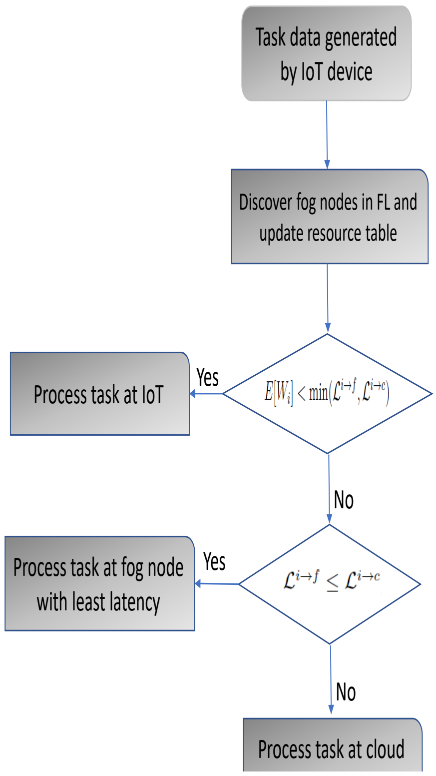

Figure 3.5: Flow chart illustrating the algorithm for task offloading.

Algorithm 3.1 Algorithm for task offloading in IFC paradigm.

Input: Task data generated by IoT devices

for (each task data)

  discover fog nodes in FL and update Resource table

  if($E[W_i] < \min(\mathcal{L}^{i\to f}, \mathcal{L}^{i\to c})$)

    Process task at IoT

  else

    if($\mathcal{L}^{i\to f} \leq \mathcal{L}^{i\to c}$)

      find the fog node f with least latency and Process task at fog node f

    else

      Process task at cloud server c

    end

  end

end

3.4 Performance Evaluation

In this section, we evaluate the performance of devices in the IFC paradigm by performing extensive simulations and present the effectiveness of our proposed offloading strategy. We consider an IFC network architecture consisting of 100 IoT devices, 5 fog nodes and 2 cloud servers. The simulation parameters for performance evaluation are presented in Table 3.2

Table 3.2: Simulation parameters of queuing theoretic model of IFC network archiecture with SDP.

Parameter	**Value**
Link bandwidth between IL and FL, B_i^f	54 Mbps
Link bandwidth between IL and CL, B_i^c	100 Mbps
Link bandwidth between FL and CL, B_f^c	500 Mbps
processing delay in computation queue of IoT device $1/\mu_i$	2 msec
processing delay in in computation queue of fog node $1/\mu_f$	1 msec
service rate in computation queue of cloud server $1/\mu_c$	50 μsec
Transmission power of IoT device $P_{tx,i}$	23 dBm
Transmission power of fog device $P_{tx,f}$	115 dBm
Storage capacity of IoT device S_i	32 KB
Storage capacity of fog device S_f	256 MB
Storage capacity of cloud server S_c	100 TB
length of a data packet, ℓ_i	1 KB
Mean propagation delay in offloading tasks between IoT and fog node $E[T_i^f]$	1.5 msec
Mean propagation delay in offloading tasks from IL to CL $E[T_i^c]$	3 msec
Mean propagation delay in offloading takss from FL to CL $E[T_f^c]$	6 msec

First, we investigate the impact of task arrival and service rates in the computation queue of devices in IL, FL and CL. We can see from Fig. 3.6 that the delay in processing tasks is significantly small for all devices in the IFC paradigm when the rate at which the tasks are served is significantly less as compared to the arrival of tasks. However, when the rate of arrival of tasks becomes comparable with the rate at which the tasks are processed, the delay in cloud servers is less as compared to the other devices. This is mainly because of the fast computing power of cloud server as well as the deterministic service duration. Moreover, it can be seen that delays incurred by the devices in IL while processing tasks in the computation queue are relatively small as compared to the devices in FL when both of them are faced with equal traffic intensity. The reason behind this phenomenon is that devices in IL have a limited storage capacity, and some of tasks are offloaded to the devices in FL or CL, which reduces the number of tasks waiting for their turn for getting processed in the computation queue of the IL's devices.

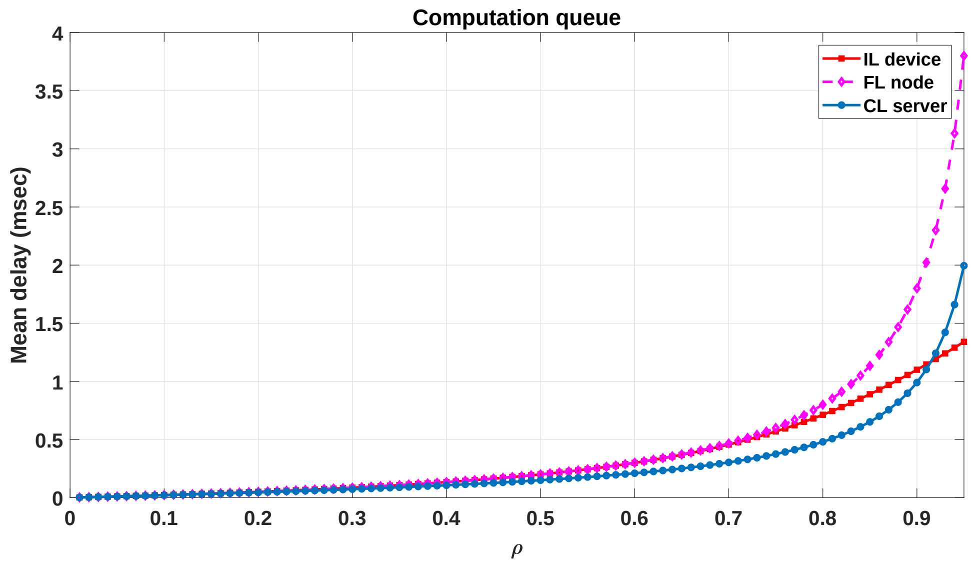

Figure 3.6: Mean delay in computation queue of devices in IL, FL, and CL for different utilization ratio.

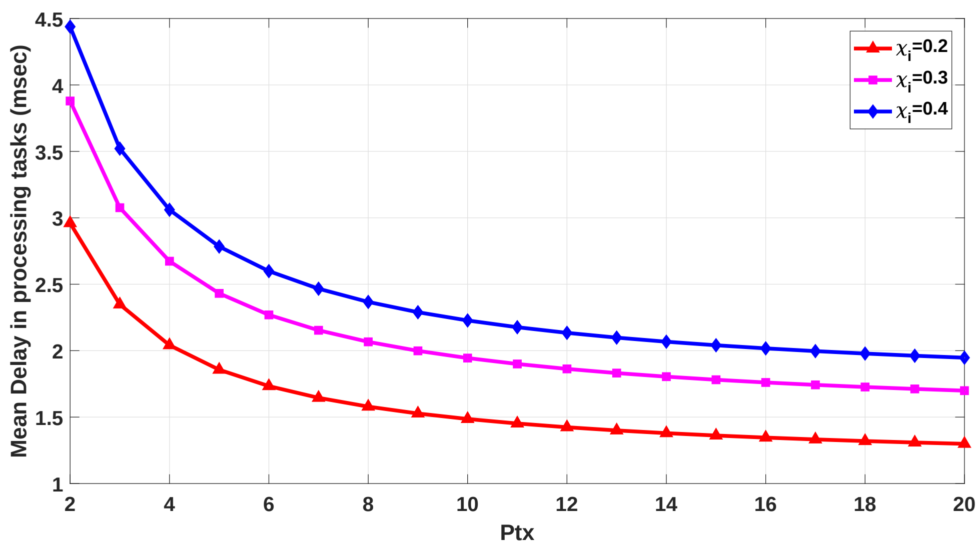

Figure 3.7: Mean delay in processing tasks in the IFC paradigm for different transmission power.

The impact of variation of the offloading ratio in the IFC paradigm for different transmission powers of the IoT devices on mean delay in processing tasks is shown in Fig. 3.7. It can be observed from the figure that the mean delay in processing tasks reduces with a reduction of offloading ratio. This is because more and more data packets are processed in the computation queue of the IoT device when the offloading ratio is small. Also, it can be seen that the mean delay in processing reduces with an increase in the transmission power of the IoT device, as it takes relatively less time to transmit a data packet when an IoT device has the capability to transmit data with more power.

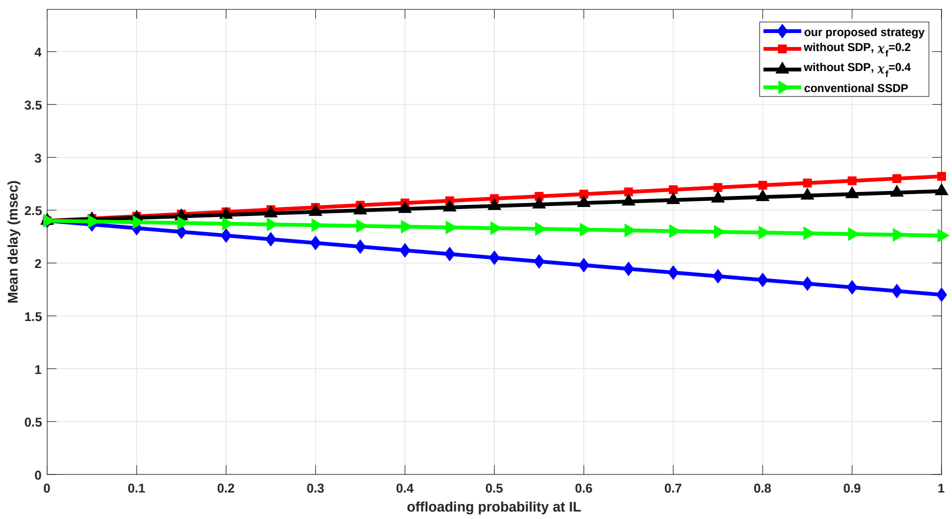

Figure 3.8: The comparison of mean latency in our proposed offloading strategy and the conventional offloading without SDP.

The comparison of mean latency in the network after using SDP in our proposed queuing theoretic based strategy, the conventional offloading using Simplified Service Discovery Protocol (SSDP) and the offloading strategy without the use of SDP for different values of offloading probabilites at IL is presented in Fig. 3.8. Fig. 3.8 shows that the mean latency in processing the tasks reduces significantly by the use of our proposed offloading strategy which uses the service discovery protocol along with the queuing based model. This is because tasks are sent to a fog node only when it has enough resources for processing them on its own in our proposed strategy after evaluation of the estimated latency while offloading tasks from one layer to the other. In this way, the fog nodes do not have to further offload their task data to any other neighbouring fog nodes or to the cloud servers, and data once offloaded to fog nodes is processed by them locally. It can be seen that when SDP is not used, the mean delay increases with the likelihood of fog nodes to offload their data further to other entities of the network.

3.5 Conclusions

In this chapter, we investigated the problem of latency reduction by task offloading in the IFC paradigm. We developed a queuing-theoretic model for each layer in the IFC paradigm and developed a strategy of offloading only to the devices which have the capability to process them. We used a service discovery protocol to find the availability of resources before offloading tasks to the fog layer. In this way, the latency in processing the tasks is shown to have reduced significantly as data arriving at the fog node is processed in it locally and the likelihood of offloading it further to some other device in the vicinity approaches zero in our strategy. We derived analytical results on delay performance of IoT tasks processing by the use of our proposed offloading strategy, and performance evaluations are presented to illustrate the performance of our proposed strategy and demonstrate the superior performance of our strategy over the existing schemes which do not use the service discovery protocol and queuing theoretic model.

3.6 Appendix

3.6.1 Appendix 3A

Let K be the capacity of an $M/M/1$ system and the probability that there are k tasks in the queue is

$$P_k = \frac{\rho^k}{\sum_{i=0}^{K} \rho^i}, \qquad \forall\ k = 0, 1, 2, ..., K.$$

Then, the mean number of tasks in the queue is found as

$$E[N] = \sum_{k=1}^{K} k P_k,$$

which is simplified to

$$E[N] = \frac{\rho(1-(K+1)\rho^K + K\rho^{K+1})}{(1-\rho)(1-\rho^{K+1})},$$

and the mean response time to process the tasks becomes

$$E[W] = \sum_{k=0}^{K-1} \frac{k+1}{\mu} \frac{\rho^k P_0}{1-P_k},$$

which is also simplified as

$$E[W] = \frac{E[N]}{\lambda(1-P_K)}.$$

3.6.2 Appendix 3B

For an $M/M/1$ queuing system, let $N(t)$ be the number of tasks in the system at time t,and considering that the system is in state k when $N(t) = k$, the probability of transition from state k to state $k+1$ during time h is

$$P_{k,k+1}(h) = (\lambda h+o(h))(1-(\mu h+o(h))+\sum_{k=2}^{\infty}(\lambda h+o(h))^{k}(\mu h+o(h))^{k-1}, \qquad \forall\, k = 0, 1, 2....$$

which is simplified as

$$P_{k,k+1}(h) = (\lambda h + o(h)), \qquad \forall\ k = 0, 1, 2....$$

Similarly, the probability of transition from state k to state $k-1$ during h is

$$P_{k,k-1}(h) = (\lambda h + o(h)), \qquad \forall\ k = 0, 1, 2....$$

Then the probability that k tasks are in the queue is

$$P_k = (1-\rho)\rho^k,$$

where $\rho = \frac{\lambda}{\mu}$. Then, the mean number of tasks in the queue is

$$E[N] = \sum_{k=0}^{\infty} kP_k = (1-\rho)\rho \sum_{k=1}^{\infty} k\rho^{k-1},$$

which is simplified as

$$E[N] = \frac{\rho}{1-\rho}.$$

By Little's Law, the mean time to process a task is $E[W] = \frac{E[N]}{\lambda}$,

which is simplified as

$$E[W] = \frac{1}{\mu - \lambda}.$$

Chapter 4

Selection Strategy for Fog Node

In this chapter, we develop an algorithm for the dynamic selection of a new fog node in the network and present a mathematical analysis for the proposed selection strategy which aims at minimizing the latency of network by acquiring efficient candidates in the fog layer of the IFC paradigm. We then evaluate the performance of our selection strategy by comparing it with the conventional ones.

4.1 Introduction

As described in chapter 2, the introduction of a fog layer in the IoT-cloud continuum reduces the overall network latency provided that a network of efficient fog devices could be established closer to the end-users which has the capability to store and process tasks in the network. In this context, FaaS opens up new avenues of opportunity for the devices present in the vicinity of end users to offer their services as fog nodes, and this may help in improving the overall capability of the network without any additional cost of deployment of a fixed infrastructure. In such a setup, the fog devices may withdraw their services from the fog network intermittently. Moreover, the IoT traffic load may also increase unpredictably. Therefore, the fog network may have to acquire additional nodes dynamically for an efficient functioning of the network. As there may be many candidate devices offering their services as fog nodes, the acquisition of efficient devices in the network becomes a challenging task. In this regard, an efficient strategy to select a fog node among the candidate devices with an aim to enhance the overall processing and storage capability of the network is of significant importance.

4.1.1 Related Work

Several classical general decision making approaches are presented in [38]-[39][40]. However, the use of such techniques proves inefficient for selection of a large number of fog nodes in the network. In [38], Fergusen et. al., proposed a selection strategy in which a best-ranked candidate as compared to all the preceding ones is selected with an aim to maximize the probability of selecting the best-ranked candidate. Although the probability of selecting the best ranked candidate in such a strategy is maximized, it does not guarantee that the best candidate will always be selected. Therefore, the mean rank of the candidates being selected during the process is not maximized in this strategy. Blanc et. al. presented an improved algorithm for selecting a new candidate with an aim to maximize the mean of the rank of selected candidates in [39]. In both these strategies, there occurs a significant degradation in a worst case scenario when no candidate is found to be better than the previously encountered ones until the last candidate is encountered (a case quite likely when the best-ranked candidate happens to have been encountered among the initially rejected ones). Gilbert et. al., in [40], presented a selection strategy where a candidate having either the best or second best candidate is selected and maximizes the probability of selecting the best-ranked candidate but does not maximize the mean of the rank of the selected candidates.

4.1.2 Contribution

In this chapter, we define a layered architecture for a network consisting of IoT, fog and the cloud layers. We develop:

- an algorithm for the dynamic selection of a new fog node in the network without the need for updated information about the existing fog nodes in the network when many candidate nodes are offering their services with FaaS.
- an analytical model for the dynamic selection of a new fog node with the aim to maximize the mean of the capabilities of the acquired fog nodes, thereby adding new nodes to the fog network that result in the reduction of the processing time of the tasks generated by the IoT devices.
- evaluate the performance of our selection strategy for fog nodes and compare it with the conventional ones.

4.2 System Model

Consider a network architecture $\Psi(\mathcal{I}, \mathcal{F}, \mathcal{C})$ comprising $\mathcal{I}$ devices in the IoT Layer (IL), $\mathcal{F}$ devices in the Fog Layer (FL) and $\mathcal{C}$ servers in the Cloud Layer (CL) where the tasks are offloaded from one layer to the other (depending on the availability of resources such as computational power and storage capacity of the devices in each

Table 4.1: A list of most frequently used symbols and their description

Description	**Symbol**
Number of candidate fog nodes offering their services	N
Number of tasks arriving at fog node f	N_F^f
Number of tasks offloaded to the cloud server c	$N^{f \to c}$
Probability that a candidate fog node at sequence k is selected	γ_h^k
Probability that a candidate fog node is examined at sequence k	γ_e^k
Mean rank of fog node acquired as being the best among all r candidates	$\mu_{(r)}$
Mean rank of fog node acquired as being second-best among all r candidates	$\mu_{(r-1)}$
Mean of the rank of the acquired fog nodes	$V_N(\Gamma_1, \Gamma_2)$
Mean number of candidate fog nodes during the selection process	$T_N(\Gamma_1, \Gamma_2)$

layer) as follows: a device in the IL either processes a newly arrived task locally or offloads it to the FL; a device in the FL receives the task data from the IL (or from the neighboring fog devices) and either processes it on their own or offloads it to one of the other devices such as neighboring fog devices and the servers in CL; and the server in the CL processes all the data received from the fog layer on its own. After the processing of the task data, the response is then sent back to the appropriate relevant IoT devices such as actuators, controllers, etc. Let N_F^f be the total number of tasks arriving at the fog node f already present in the network, and $N^{f \to c}$ denote the number of tasks offloaded by the fog node f to the cloud server c. In our system model, it is assumed that the existing infrastructure of nodes in the fog layer continues to process tasks data while the centralized fog node initiates the process of acquisition of a new fog node in the network when a large number of tasks are being offloaded to the cloud servers, i.e., when the ratio $N^{f \to c}/N_F^f$ reaches a limit e_M, where the value of e_M could be adjusted in implementation based on the given traffic demands and latency requirements. Moreover, it is considered that the lifetime of the network is significantly long as compared to the latency required in the node selection process. We consider that the processing and storage capabilities of candidate nodes are independent of each other. Once the selection process is initiated, we assume that the centralized fog node f encounters candidates in random order and is able to retrieve information regarding the processing capabilities of the candidates one by one as soon as they are encountered. The centralized fog node then decides whether to select or reject a candidate encountered during the inspection according to the information retrieved in our system model. A list of most frequently used symbols along with their description, used throughout this chapter, is presented in table 4.1.

4.2.1 The Selection Strategy

We characterize an efficient strategy for the dynamic selection of a new fog node in the network as N candidate fog nodes offer their services by utilizing FaaS as shown diagrammatically in Fig. 4.1.

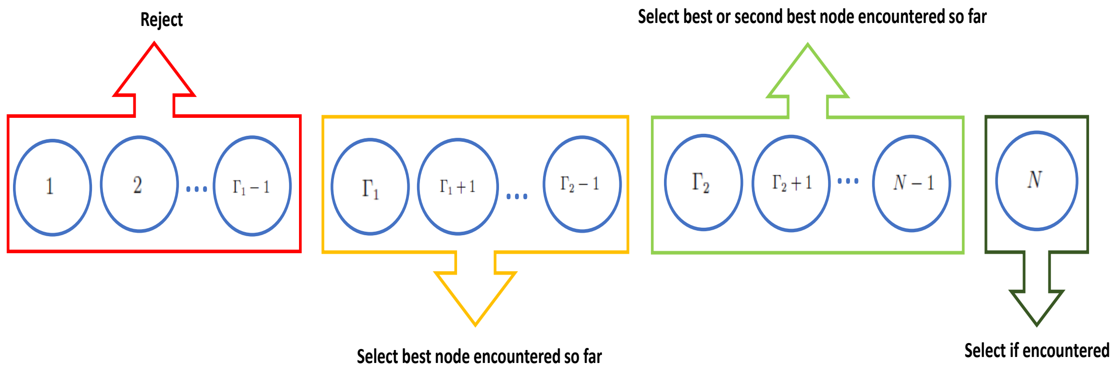

Figure 4.1: The diagrammatic representation of the proposed selection strategy

The algorithm for the fog node selection policy is given as Algorithm 4.1 in which the candidate fog nodes are examined one by one, and the processing and storage capability of each candidate under examination is determined as its "rank". In the proposed algorithm, although the candidates encountered during the first Γ_1 examinations are not selected during the selection process where $1 \leq \Gamma_1 < N$, their ranks are recorded for comparison with the candidate nodes which are yet to be examined.

Therefore, after Γ_1 rejections, a candidate having a better rank is selected and

---

Algorithm 4.1 Algorithm to select a new fog node in the fog network.

Input: Number of candidate fog nodes N

for (each candidate examined at sequence number $k, \forall\ k \in \{1, 2, ..., N\}$)

  find the processing power of the candidate k as rank(k)

  if($k \ \leq \Gamma_1$) **then**

    reject the candidate while keeping record of its rank

  else-if ($\Gamma_1 < k \ < \Gamma_2$) **then**

    if(rank(k) $>$ max. rank(1 to $k-1$) **then**

      select the fog node k & stop further examinations

    end

  else-if($\Gamma_2 \leq k < N$) **then**

    if(rank(k) $>$ max./second max. rank(1 to $k-1$)) **then**

      select the fog node k & stop further examinations

    end

  else

    select the fog node k

  end

end

---

then the process of examination of any further nodes is terminated. However, if no candidate fog node is found to have a better relative rank as compared to all the previously encountered fog nodes till the examination of $(\Gamma_2 - 1)$, $\forall\ \Gamma_1 < \Gamma_2 \leq N$, candidate fog nodes, the selection strategy is changed in such a way that the fog node chooses a first encountered candidate fog node which has either best or second-best relative rank as compared to all the candidates encountered previously. Also, if neither the best nor the second best ranked fog node is encountered after inspection of $N-1$ fog nodes (a case likely when both the best and second best fog nodes happen to be encountered during the first Γ_1 inspections), the N^{th} fog node is chosen without any comparisons with the previously encountered ones. This is because there are no more fog nodes left for inspection in the selection process.

4.2.2 Analytical Modeling of Selection Strategy

In this subsection, we present an analytical model of the of the proposed algorithm 4.1 for the selection of a new fog node and find an efficient selection strategy among the N candidate fog nodes with an objective of finding the best ranked fog node on average, thus minimizing latency in processing of tasks.

4.2.2.1 Probability of examination and selection of a candidate

We first find the probabilities of examination and selection of the candidate fog nodes encountered at a given sequence number when the selection process is initiated. As described in the proposed algorithm, the candidate encountered at sequence number k is selected if it is examined and is found best among all the preceding ones encountered at sequence number $\{1, 2, ...k-1\}$, $\forall\ k \in \{\Gamma_1+1,\ \Gamma_1+2, ..., \ \Gamma_2-1\}$. Since the occurrence of the best candidate at any sequence number is equally likely, the probability that the candidate encountered at sequence number k is best among all those encountered so far is $1/k$. Then, the probability that a candidate encountered at sequence number k is selected is

$$\gamma_h^k = \left(\frac{1}{k}\right)\gamma_e^k, \qquad \Gamma_1 < k < \Gamma_2 \tag{4.1}$$

where γ_e^k is the probability that the candidate is examined at a sequence number k. According to the proposed algorithm, the selection process terminates as soon as a candidate is selected. Therefore, a candidate at a sequence number k is examined only if none of the candidates encountered from sequence number $\Gamma_1 + 1$ to $k - 1$ is selected. As described previously, the candidate at sequence number k is selected if it is the best among all those encountered so far where $\Gamma_1 < k < \Gamma_2$. Thus, the probability that the candidate is examined at a sequence number k is

$$\gamma_e^k = \prod_{j=\Gamma_1+1}^{k-1} \left(1 - \frac{1}{j}\right), \quad \Gamma_1 < k < \Gamma_2$$

which simplifies to

$$\gamma_e^k = \frac{\Gamma_1}{k-1}, \quad \Gamma_1 < k < \Gamma_2. \tag{4.2}$$

Therefore, the probability that a candidate encountered at sequence number k is selected is

$$\gamma_h^k = \frac{\Gamma_1}{k(k-1)}, \quad \Gamma_1 < k < \Gamma_2. \tag{4.3}$$

If no node could be selected even after examination of the candidate fog nodes at sequence number $\Gamma_2 - 1$, the criteria for selection is changed according to the proposed strategy and any node examined thereafter is selected if its rank is found to be either the best or second-best when compared to the preceding ones. However, a candidate is examined at a sequence number Γ_2 if none of the candidates encountered at sequence number $\{\Gamma_1 + 1,\ \Gamma_1 + 2,\ ...,\ \Gamma_2 - 1\}$ is found best among the preceding ones. Therefore, the probability that the candidate is examined at a sequence number Γ_2 is

$$\gamma_e^{\Gamma_2} = \frac{\Gamma_1}{(\Gamma_2 - 1)}. \tag{4.4}$$

The node examined at sequence number Γ_2 is selected when it is either the best or the second-best among all the preceding candidates. As the occurrence of the best or the second best candidate at any sequence number is equally likely, the probability that the candidate encountered at sequence number Γ_2 is either the best or second-best among all the Γ_2 nodes encountered is $2/\Gamma_2$. The candidate node encountered at sequence number Γ_2 is selected when it is examined at sequence Γ_2 and is found either the best or second-best among all Γ_2 candidates. Therefore, the probability that the candidate encountered at sequence number Γ_2 is selected is

$$\begin{aligned} \gamma_h^{\Gamma_2} &= \left(\frac{2}{\Gamma_2}\right) \gamma_e^{\Gamma_2}, \\ &= \frac{2\Gamma_1}{\Gamma_2(\Gamma_2 - 1)}. \end{aligned} \tag{4.5}$$

A candidate fog node encountered at a sequence number q, $\forall\ \Gamma_2 < q \leq N$, is examined only when the candidate fog nodes at sequence numbers $\{\Gamma_1 + 1, \Gamma_1 + 2, ..., \ \Gamma_2 - 1\}$ are not found as better-ranked than the preceding ones, and the candidates examined at sequence number $\{\Gamma_{2,}\ \Gamma_2{+}1,\ ...,\ q{-}1\}$ are found neither the best nor the second best-ranked as compared to all the preceding ones. Therefore, as the occurrence of best or the second best at a given sequence number is equally likely, the probability that the candidate fog node encountered at sequence number q is examined is

$$\gamma_e^q = \left[\prod_{j=\Gamma_1+1}^{\Gamma_2-1} \left(1 - \frac{1}{j}\right)\right] \left[\prod_{j=\Gamma_2}^{q-1} \left(1 - \frac{2}{j}\right)\right],$$

$\forall\ \Gamma_2 < q \leq N$, and this simplifies to

$$\gamma_e^q = \frac{\Gamma_1(\Gamma_2 - 2)}{(q-1)(q-2)}. \tag{4.6}$$

The candidate examined at a sequence number q, $\forall\ \Gamma_2 < q < N$ is selected when it is found to be either the best or the second best among the preceding nodes. Since the occurrence of either the best or the second at a given sequence number is equally likely, the probability that the candidate is either the best or the second best among all q nodes encountered so far is $2/q$. The candidate node encountered at sequence number q, $\forall\ \Gamma_2 < q < N$ is selected when it is examined at sequence number q and is found either the best or second-best among all the q candidates encountered so far. Therefore, the probability that the candidate encountered at sequence number q is selected is

$$\begin{aligned} \gamma_h^q &= \left(\frac{2}{q}\right)\gamma_e^q \\ &= \frac{2\Gamma_1(\Gamma_2 - 2)}{q(q-1)(q-2)},\ \forall\ \Gamma_2 < q < N. \end{aligned} \tag{4.7}$$

Moreover, if none of the fog nodes is selected after the examination of $N-1$ candidate fog nodes, the fog node encountered at sequence number N is selected without its comparison with the preceding candidate nodes and the selection process is terminated. Then, the probability that the N^{th} fog node is selected is equal to the probability that N^{th} fog node is examined and is given as

$$\gamma_h^N = \gamma_e^N = \frac{\Gamma_1(\Gamma_2 - 2)}{(N-1)(N-2)}. \tag{4.8}$$

4.2.2.2 Mean of the rank of the acquired fog nodes

Let $X_1, X_2, X_3, ...X_N$ be the ranks of the candidate fog nodes examined during the selection process at sequence number $1, 2, 3, ..., N$ respectively. A candidate fog node examined at a sequence number r is selected if it is encountered at a sequence number in the range of $[\Gamma_1+1, \quad \Gamma_2-1]$, and is the best among the ones encountered so far, or if it is encountered at a sequence number in the range of $[\Gamma_2, \quad N-1]$ and is either the best or the second best among the ones encountered so far. Let $\mu_{(r)}$ be the mean value of the rank of the candidate fog node encountered at the r^{th} sequence number if it is the best-ranked encountered so far among all the r candidates, and this is defined as

$$\mu_{(r)} := E[X_r \mid X_r = X_{(r)}], \tag{4.9}$$

where $X_{(r)}$ represents the maximum of the set $\{X_1, X_2, ..., X_r\}$, and let $\mu_{(r-1)}$ be the mean value of rank of r^{th} fog node if it is the second-best ranked encountered so far among all the r candidates, and is defined as

$$\mu_{(r-1)} := E[X_r \mid X_r = X_{(r-1)}], \tag{4.10}$$

where $X_{(r-1)}$ represents the second-maximum of the set $\{X_1, X_2, ..., X_r\}$. In the case when both the best and the second best-ranked candidate fog nodes have already been encountered at a sequence number $\{1, 2, ..., \Gamma_1\}$, then as described in the proposed selection strategy, none of the subsequent candidate fog nodes could be chosen until the inspection of $(N-1)$ candidate fog nodes. In such a scenario, the N^{th} fog node is chosen automatically without any comparison with any of the preceding ones. Let the mean value of the rank of the candidate fog node encountered at the N^{th} sequence number which is to be acquired without any comparison with the preceding candidates be $\mu_{_N}$, and it is defined as

$$\mu_{_N} := E[X_N]. \tag{4.11}$$

The equations above provide the mean value of the rank of a candidate fog node when it is chosen during the selection process at any sequence number according to the proposed fog network formation policy. Two important theorems to find the mean rank of the acquired fog node as being either the best or second best encountered during the selection process are as follows:

Theorem 4.1

Let $X_1, X_2, ..., X_n$ be continuous independent and identically distributed (i.i.d) random variables, each of which follows a standard uniform distribution (i.e., from 0 to 1), then the mean of the largest among them, denoted by $\mu^U_{(n)}$, is

$$\mu^U_{(n)} = \frac{n}{n+1}, \tag{4.12}$$

and the mean of the second largest among them, denoted by $\mu^U_{(n-1)}$, is

$$\mu^U_{(n-1)} = \frac{n-1}{n+1}. \tag{4.13}$$

Proof: See Appendix 4A.

Theorem 4.2

Let $X_1, X_2, ..., X_n$ be i.i.d, normally distributed with mean μ and variance σ^2, i.e., $X^{\mathcal{N}}_k \sim \mathcal{N}(\mu, \sigma^2)$, then the mean of the largest among them, denoted by

$\mu^{\mathcal{N}}_{(n)}$, is

$$\mu^{\mathcal{N}}_{(n)} = \mu - \sigma \left[\Phi^{-1} \left(\frac{1-\alpha}{n-2\alpha+1} \right) \right], \tag{4.14}$$

where $\alpha = 0.375$, and $\Phi^{-1}(p)$ is the inverse normal function with parameter p. Also, the mean of the second-largest among n normally distributed i.i.d's, denoted by $\mu^{\mathcal{N}}_{(n-1)}$, is

$$\mu^{\mathcal{N}}_{(n-1)} = \mu - \sigma \left[\Phi^{-1} \left(\frac{2-\alpha}{n-2\alpha+1} \right) \right]. \tag{4.15}$$

Proof: See Appendix 4B.

In order to acquire the most efficient fog node on average when N candidates are willing to offer their service, we devise a fog node selection strategy with an aim to maximize the mean of the rank of the acquired fog node. Let $V_N(\Gamma_1, \Gamma_2)$ be the mean of the rank of the acquired fog nodes following the proposed selection strategy. Our objective is to find the values of Γ_1 and Γ_2 to maximize $V_N(\Gamma_1, \Gamma_2)$, i.e., the values Γ_1^* and Γ_2^* such that the strategy for the fog node selection consists of examining the first Γ_1^* candidate fog nodes while maintaining the record of their ranks, and then from that point on, select a fog node if it is the best among the ones encountered so far. However, if no fog node is selected after the examination of $(\Gamma_2^* - 1)$ fog nodes, then select a fog node if it is either best or second best so far. In order to determine the threshold values Γ_1^* and Γ_2^*, we compute the mean of the ranks of the selected fog node $V_N(\Gamma_1, \Gamma_2)$ for each value of Γ_1 and Γ_2. It is obtained by summing the expected value of the fog node selected during the inspection at the k^{th} sequence number times the probability of being selected at that inspection, including the case in which the fog node selects the last fog node without any inspection. Given

$1 \leq \Gamma_1 \leq N-1$, and $\Gamma_2 > \Gamma_1$, we get the mean rank of the selected nodes as

$$V_N(\Gamma_1, \Gamma_2) = \sum_{k=\Gamma_1+1}^{\Gamma_2-1} \mu_{(k)} \gamma_h^k + \sum_{k=\Gamma_2}^{N-1} \left(\frac{\mu_{(k)} + \mu_{(k-1)}}{2} \right) \gamma_h^k + \mu_N \gamma_h^N. \quad (4.16)$$

Substituting the probability of selection of candidate nodes from (4.3), (4.5), (4.7), & (4.8), we get

$$V_N(\Gamma_1, \Gamma_2) = \sum_{k=\Gamma_1+1}^{\Gamma_2-1} \left(\frac{\mu_{(k)} \Gamma_1}{k(k-1)} \right) + \left(\frac{(\mu_{(\Gamma_2)} + \mu_{(\Gamma_2-1)}) \Gamma_1}{\Gamma_2(\Gamma_2 - 1)} \right) + \sum_{k=\Gamma_2+1}^{N-1} \left(\frac{(\mu_{(k)} + \mu_{(k-1)}) \Gamma_1 (\Gamma_2 - 2)}{k(k-1)(k-2)} \right) + \left(\frac{\mu_N \Gamma_1 (\Gamma_2 - 2)}{(N-1)(N-2)} \right). \quad (4.17)$$

For the case when the rank of the fog nodes is distributed uniformly between 0 to 1, we use Theorem 4.1 to find the mean of the rank of the selected fog nodes as

$$V_N^U(\Gamma_1, \Gamma_2) = \sum_{k=\Gamma_1+1}^{\Gamma_2-1} \left(\frac{\Gamma_1}{k^2 - 1} \right) + \left(\frac{2\Gamma_1(\Gamma_2 - 0.5)}{\Gamma_2(\Gamma_2^2 - 1)} \right) + \sum_{k=\Gamma_2+1}^{N-1} \left(\frac{2\Gamma_1(\Gamma_2 - 2)(k - 0.5)}{k(k^2-1)(k-2)} \right) + \left(\frac{\Gamma_1(\Gamma_2 - 2)}{2(N-1)(N-2)} \right). \quad (4.18)$$

When the rank of the fog nodes is distributed normally, each with mean μ and variance σ^2, the mean of the rank of the selected fog nodes is evaluated (by using Theorem 4.2) as

$$V_N^{\mathcal{N}}(\Gamma_1, \Gamma_2) = \sum_{k=\Gamma_1+1}^{\Gamma_2-1} \frac{\Gamma_1}{k(k-1)} \left[\mu - \Phi^{-1} \left(\frac{1-\alpha}{k - 2\alpha + 1} \right) \right] + \sum_{k=\Gamma_2+1}^{N-1} \left(\frac{\Gamma_1(\Gamma_2 - 2)}{2k(k-1)(k-2)} \right) \Theta_k + \frac{\Gamma_1}{\Gamma_2(\Gamma_2 - 1)} \Theta_{\Gamma_2} + \frac{\mu \Gamma_1 (\Gamma_2 - 2)}{(N-1)(N-2)}, \quad (4.19)$$

where $\Theta_k = 2\mu - \Phi^{-1}\left(\frac{1-\alpha}{k-2\alpha+1}\right) - \Phi^{-1}\left(\frac{2-\alpha}{k-2\alpha+1}\right)$, and $\Phi^{-1}(p)$ is the inverse normal function with parameter p.

4.2.2.3 An efficient selection strategy

The discrete first-order derivative of $V_N(\Gamma_1, \Gamma_2)$ with respect to both Γ_1 and Γ_2 respectively is

$$\frac{\triangle V_N(\Gamma_1, \Gamma_2)}{\triangle \Gamma_1} = V_N(\Gamma_1 + 1, \Gamma_2) - V_N(\Gamma_1, \Gamma_2), \tag{4.20}$$

$$\frac{\triangle V_N(\Gamma_1, \Gamma_2)}{\triangle \Gamma_2} = V_N(\Gamma_1, \Gamma_2 + 1) - V_N(\Gamma_1, \Gamma_2), \tag{4.21}$$

and the desired values of Γ_1 and Γ_2 are the ones that satisfy the conditions

$$\left(\frac{\triangle V_N(\Gamma_1, \Gamma_2)}{\triangle \Gamma_1}\right)\left(\frac{\triangle V_N(\Gamma_1 - 1, \Gamma_2)}{\triangle \Gamma_1}\right) \leq 0, \tag{4.22}$$

$$\left(\frac{\triangle V_N(\Gamma_1, \Gamma_2)}{\triangle \Gamma_2}\right)\left(\frac{\triangle V_N(\Gamma_1, \Gamma_2 - 1)}{\triangle \Gamma_2}\right) \leq 0. \tag{4.23}$$

Therefore, the values which provide the maximum mean rank of the selected fog node can be obtained for different values of the total number of candidate fog nodes that are offering their services in the vicinity, i.e., N.

In order to obtain a closed form expression for values of Γ_1, Γ_2, we use the method of least-square estimation on the values of Γ_1, Γ_2 as a function of the total number of candidates being examined (N), and obtain

$$\Gamma_1^* = \lfloor \alpha_1 N^{\beta_1} + 0.5 \rfloor, \tag{4.24}$$

$$\Gamma_2^* = \lfloor \alpha_2 N^{\beta_2} + 0.5 \rfloor, \tag{4.25}$$

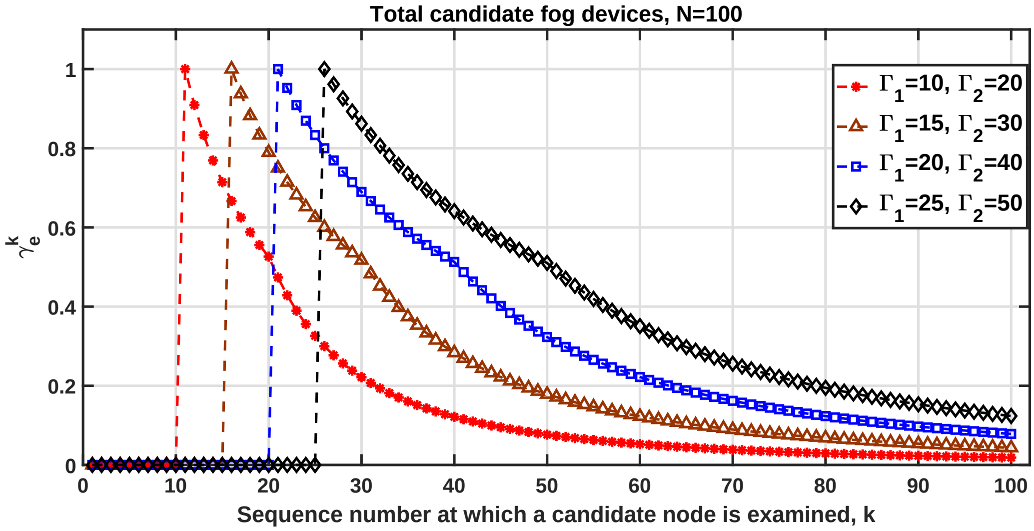

Figure 4.2: Probability of examining a fog node at a sequence number k during the selection process when the total number of fog devices to be examined $N = 100$.

where the symbol $\lfloor x + 0.5 \rfloor$ represents the nearest integer value to x, and the values of coefficients α_1, α_2 and exponents β_1, β_2 depend on the type of distribution and their relevant parameters. For the case when the processing power of the candidate fog nodes is assumed to be uniformly distributed from 0 to 1, the coefficients are evaluated as $\alpha_1 = 0.6566$, $\alpha_2 = 1.309$, and the exponents as $\beta_1 = 0.6798$, $t\beta_2 = 0.6618$. Similarly, when the processing power of fog nodes is distributed normally with mean $\mu = 0.5$ and standard deviation of $\sigma = 0.125$, the coefficients are evaluated as $\alpha_1 = 0.3131$, $\alpha_2 = 0.6718$, and the exponents as $\beta_1 = 0.9015$, $\beta_2 = 0.889$.

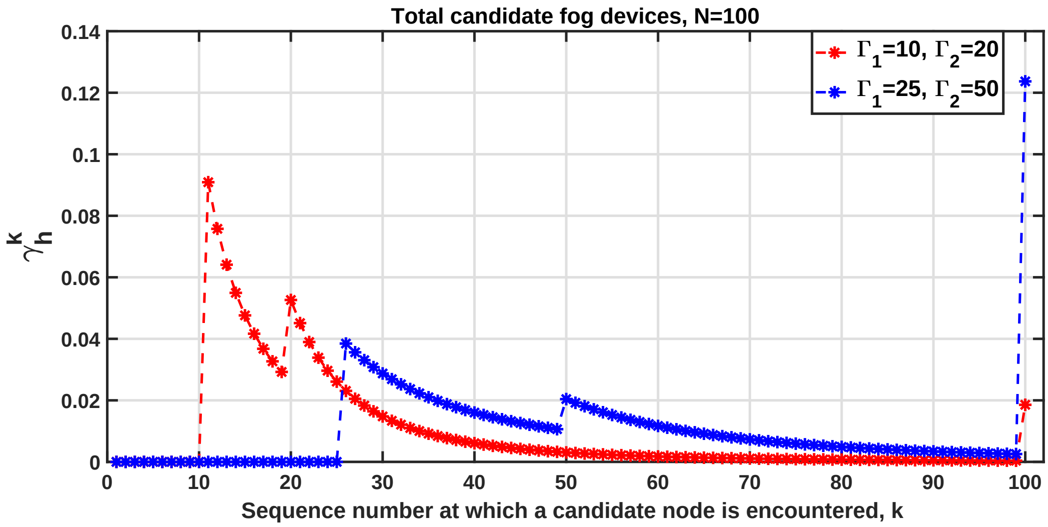

Figure 4.3: Probability of selecting a fog node at a sequence number k according to the fog node selection strategy when the total number of fog devices to be examined $N = 100$.

4.2.2.4 The proportion of candidates examined

The average number of fog nodes examined using the proposed strategy for the given values of Γ_1^* and Γ_2^* is

$$T_N(\Gamma_1^*, \Gamma_2^*) = \sum_{k=\Gamma_1^*+1}^{N} k\gamma_h^k. \tag{4.26}$$

Using the values from (4.3), (4.5), (4.7), and (4.8), we get

$$T_N(\Gamma_1^*, \Gamma_2^*) = \frac{N\Gamma_1^*(\Gamma_2^* - 2)}{(N-1)(N-2)} + \frac{2\Gamma_1^*}{\Gamma_2^* - 1} + \sum_{k=\Gamma_1^*+1}^{\Gamma_2^*-1} \frac{\Gamma_1^*}{k-1} + \sum_{k=\Gamma_2^*+1}^{N-1} \frac{2\Gamma_1^*(\Gamma_2^* - 2)}{(k-1)(k-2)}, \tag{4.27}$$

where the values Γ_1^*, Γ_2^* are obtained according to the proposed selection strategy depending on whether the ranks of the candidates are distributed according to the standard uniform distribution or the normal distribution. Therefore, the proportion

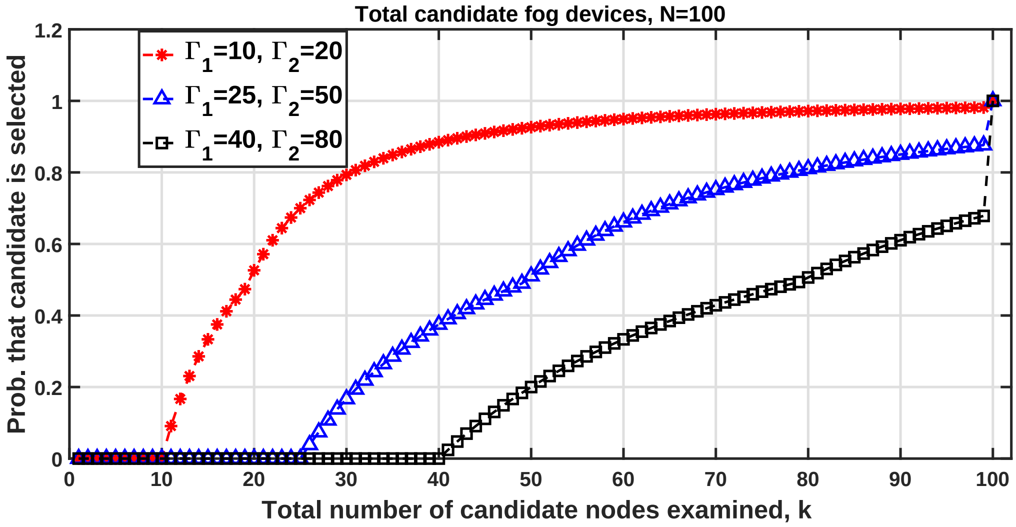

Figure 4.4: Probability that candidate fog device is selected after k examinations according to the fog node selection strategy when total number of fog devices to be examined $N = 100$.

of candidates required to be examined can be found by dividing the average number of fog nodes examined by the total number of candidate fog nodes willing to offer their services in the selection process. This leads us to quantify how fast the process of selection of a new fog node is terminated according to the given strategy.

4.3 Numerical Results and Performance Evaluation

In this section, we evaluate the proposed fog node selection strategy, verify the analytical results through simulation, and then compare them with the already existing models of Ferguson, Blanc, and Gilbert. The selection strategy proposed by Ferguson [38] compares the candidate encountered at sequence number k with all the preceding $k-1$ candidates (after rejection of first Γ_1 candidates encountered), and

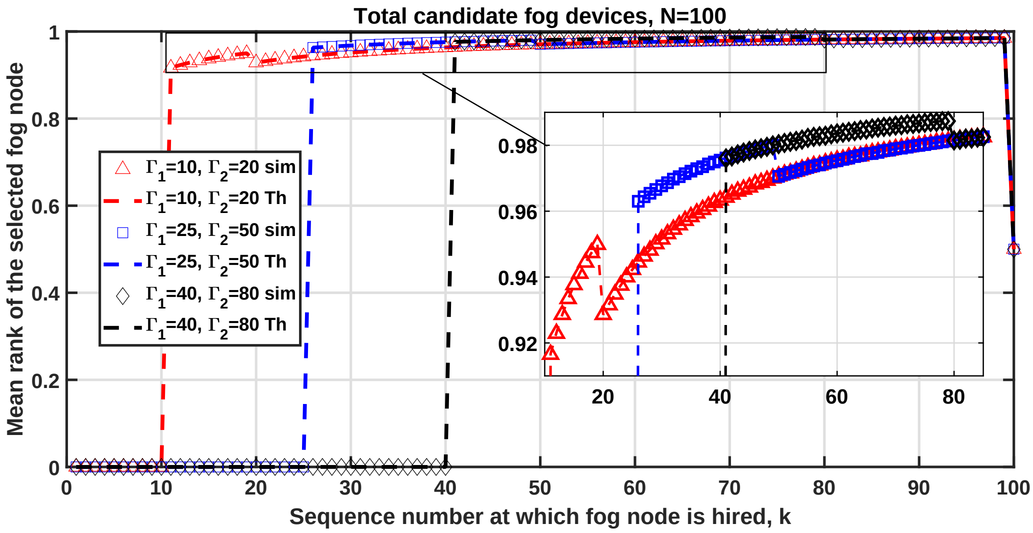

Figure 4.5: Mean rank of the fog device selected at sequence k when the rank of candidate devices is distributed uniformly from 0 to 1.

terminates the selection process whenever a best among the preceding candidates is encountered. The Ferguson's strategy is based on a threshold value of Γ_1 which is evaluated to maximize the probability of selection of best-ranked candidate. In contrast, Gilbert [40] proposed the selection strategy which is based on two threshold values, i.e., Γ_1 and Γ_2, where first Γ_1 candidates are rejected, and the selection process is terminated either when a best among the all the previously examined candidates is encountered until the inspection of Γ_2 candidates or when a best or second-best candidate among all the previously examined candidates is encountered at a sequence number k, $\forall\ k \geq \Gamma_2$. The values of Γ_1 and Γ_2 in Gilbert selection strategy are evaluated to maximize the probability of selection of best or second best-ranked candidate. The strategy proposed by Blanc [39] is based on termination of selection process when a best among the preceding candidates is encountered and has only one threshold value of Γ_1 in the same way as presented by Ferguson. However, the value of Γ_1 is evaluated to maximize the mean rank of the selected candidates instead of maximizing the probability of selection of best-ranked candidate. Our proposed selection strategy is a blend of Gilbert and Blanc approaches in a way that

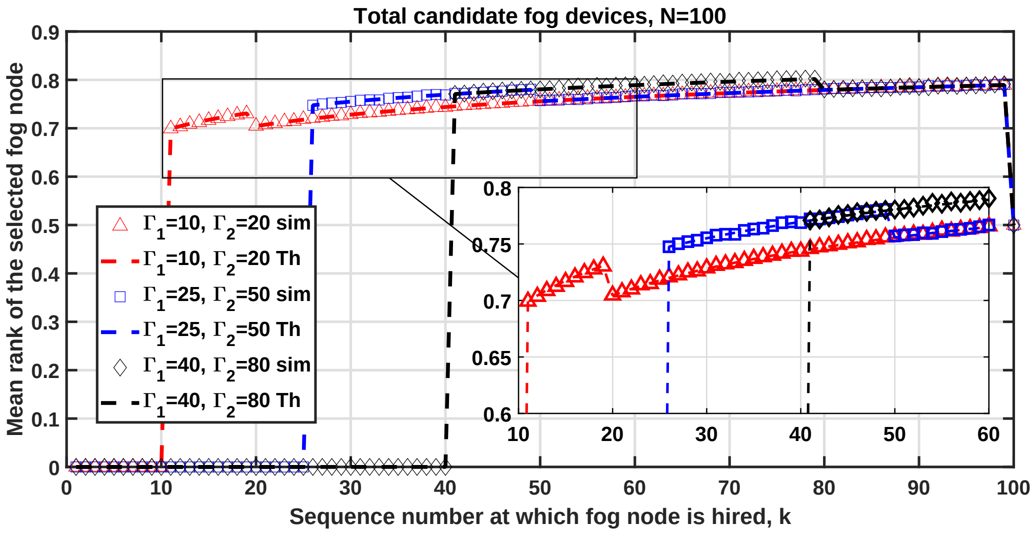

Figure 4.6: Mean rank of the fog device selected at sequence k when the rank of candidate devices is distributed normally with mean 0.5 and standard deviation 0.125.

it is based on two threshold values Γ_1 and Γ_2 as suggested by Gilbert, and the values of Γ_1 and Γ_2 are evaluated to maximize the mean rank of the selected candidates instead of maximizing the probability of selection of best-ranked candidates as is done in Blanc strategy.

We consider that the fog nodes, deployed with an aim to process the task data generated by the IoT devices, can communicate with both the existing fog nodes as well as the candidates offering their services within an area of 100×100 m^2. IoT devices are connected with the fog node and when the selecting process of a new fog node is triggered, we assume that the fog node already in the network may examine as many as N candidates, depending on the rate at which the new candidates offer their services. The fog node then decides about the acquisition of one of the candidates examined during the selection process.

Fig. 4.2 shows the probability of examining the candidate fog node at a sequence number k when the total candidates offering their service, $N = 100$ with a random set of threshold values $(\Gamma_1, \Gamma_2) = \{(10, 20),\ (15, 30),\ (20, 40),\ (25, 50)\}$. However,

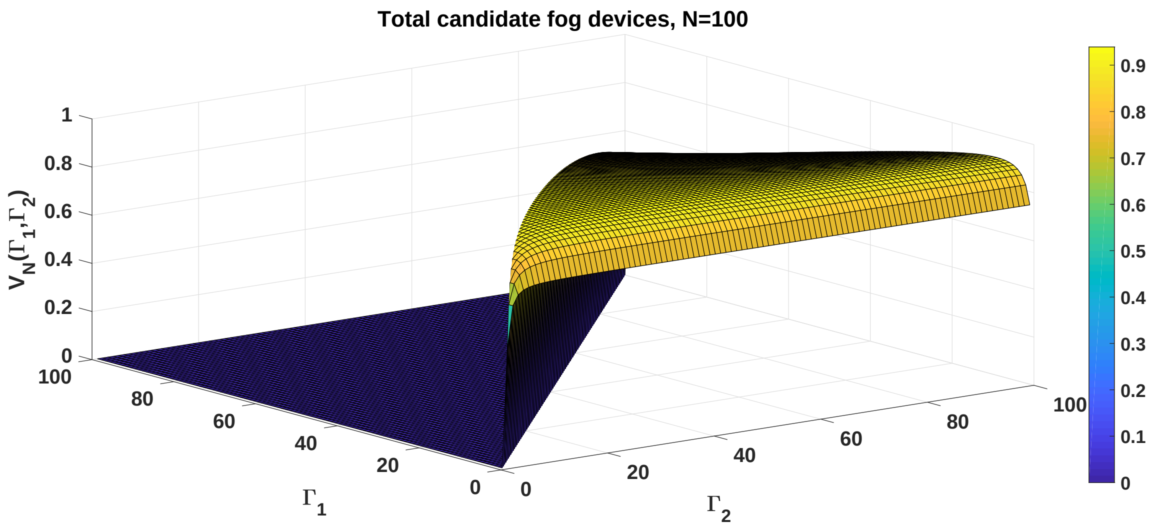

Figure 4.7: Mean of the ranks of the selected node when ranks are distributed according to the standard uniform distribution for given values of Γ_1, Γ_2 according to the node selection strategy when $N = 100$.

the unique values of set (Γ_1, Γ_2) that maximize the mean of the rank of selected fog node are obtained from eq. 4.24 & 4.25 in section 4.2.2.3 under the assumption that the rank of the candidate fog nodes is distributed according to the standard uniform or normal distribution with mean 0.5 and standard deviation 0.125, and are dependent on the total number of candidate fog nodes that are offering their services in the vicinity, i.e., N. Since the selection process starts after the ignoring first Γ_1 candidates, it is certain that the first candidate after Γ_1 will be examined and hence the probability of a candidate appearing at a sequence number $\Gamma_1 + 1$ being examined is 1. During the selecting process, as more and more candidates are examined, the chance of occurrence of the best candidate increases and hence the probability of terminating the selection process increases with each candidate's occurrence. So the probability that the candidate k is examined decreases with k. As the criteria of selection changes after $\Gamma_2 - 1$ candidates, Fig. 4.2 shows a gradual decrease in probability of examination of candidates after $\Gamma_2 - 1$. This is because the occurrence of either the best or the second best candidate becomes more likely after

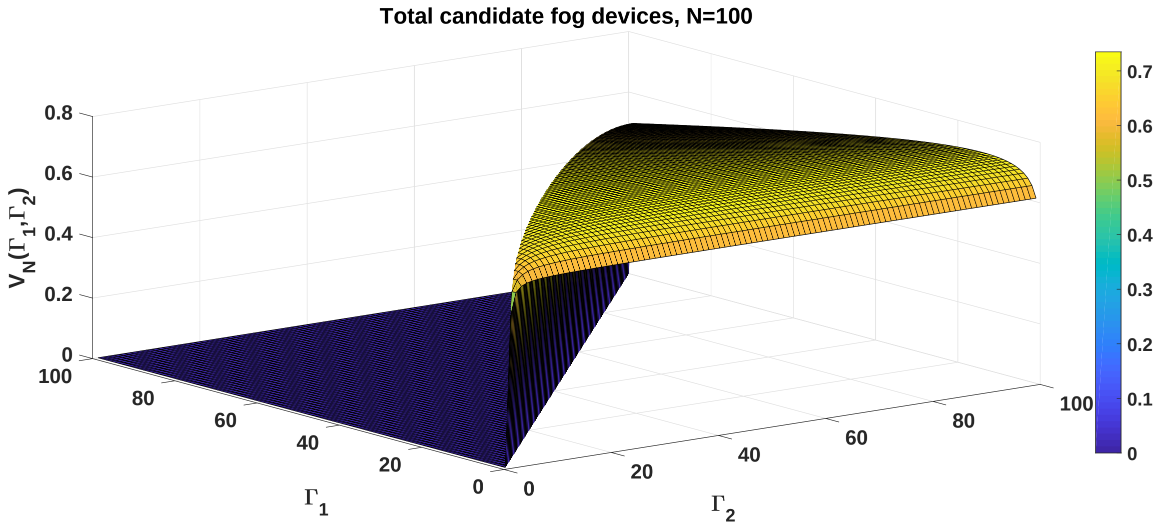

Figure 4.8: Mean of the ranks of the selected node when ranks are distributed normally with mean 0.5 and standard deviation of 0.125 for given values of Γ_1, Γ_2 according to the node selection strategy when $N = 100$.

examination of $\Gamma_2 - 1$ candidates and leads to the selection of candidates in a lesser number of examinations afterwards. The probability of selecting the fog node during the examination of the k^{th} candidate is shown in Fig. 4.3. The selecting process is assumed to have candidates $N = 100$ in aggregate and simulations are performed for two random set of values, i.e., $(\Gamma_1, \Gamma_2) = \{(10, 20), (25, 50\}$, where the values of Γ_1 and Γ_2 are selected randomly in Fig. 4.3. However, the unique values of (Γ_1, Γ_2) that maximize the mean of the rank of selected fog node are obtained from eq. 4.24 & 4.25 in section 4.2.2.3 under the assumption that the rank of the candidate fog nodes is distributed according to the standard uniform or normal distribution with mean 0.5 and standard deviation 0.125, and are dependent on the total number of candidate fog nodes that are offering their services in the vicinity, i.e., N. It is evident that as the selection probability decreases with the number of inspections of the fog nodes as the process terminates after finding of the best fog node in the process. However, after examination of $\Gamma_2 - 1$ candidates, the probability of selection of a fog node rises to a higher value because of leniency in the selection criteria of candidates

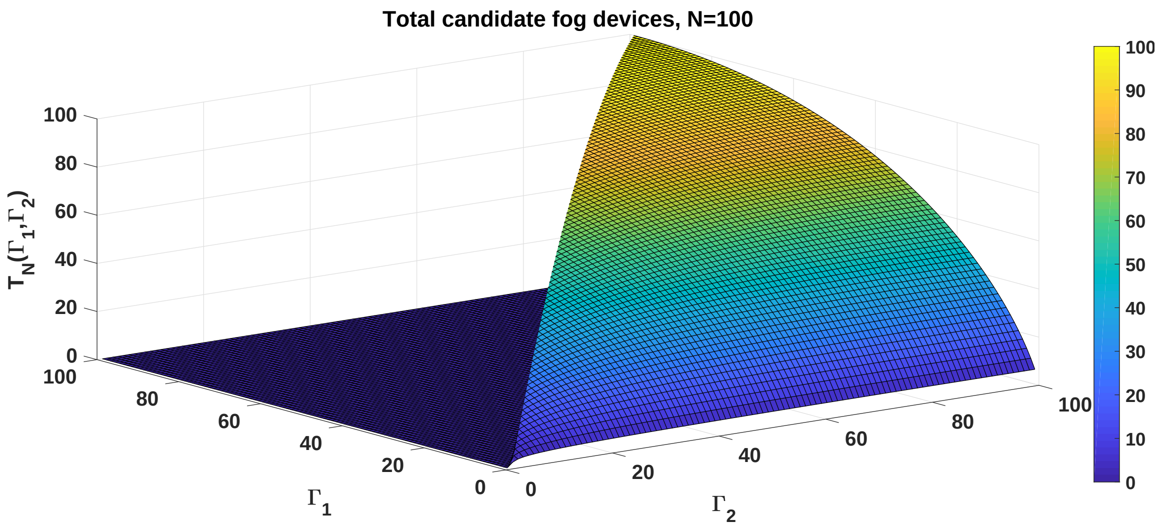

Figure 4.9: Mean number of candidates examined according to the selection strategy for the given values of Γ_1, Γ_2 when $N = 100$.

which is changed in a way that a node encountered with either the best or second best rank is selected instead of selecting only the best-ranked. The selection process continues until a fog node is selected according to the proposed strategy. Fig. 4.4 shows the probability that the selection process is terminated after the encounter of candidate fog nodes at a given sequence number k. It is found that as the selection of candidates is initiated after a certain number of fog nodes, Γ_1, the probability of termination of the selection process increases with an increase in the sequence number at which the fog nodes are examined and compared with the preceding the nodes.

When it is assumed that the rank of the fog nodes is distributed uniformly between 0 to 1, and N is the total number of candidates available to offer their services in the fog network, the mean rank of the one selected at a sequence number k is shown in Fig. 4.5. The reason behind this phenomenon is that as the number of rejected candidates increases, the chance to come across with a better fog node improves and hence the the mean processing power of the selected one increases as the selection process continues. This trend of increasing mean processing power continues until the inspection of $\Gamma_2 - 1$ fog nodes after which a sudden dip in the expected processing power of the fog node can be seen in Fig. 4.5. This is mainly because the quality of fog nodes decreases at an encounter number Γ_2 due to acceptance of either the best or second best fog node candidate so far instead of selecting only the best one. Fig. 4.6 shows the mean rank of the selected fog node at a sequence number k when the rank of the fog nodes is distributed according to the normal distribution with mean 0.5 and standard deviation 0.125 and exhibits a similar trend as in the case when the candidate fog nodes had ranks distributed uniformly.

When the rank of the candidate fog nodes is distributed according to the standard uniform distribution, the mean of the rank of the acquired fog nodes for the given values of Γ_1 and Γ_2 according to the proposed selection strategy is shown in Fig. 4.8 where the number of candidate fog nodes is $N = 100$. The point of maximum $(\Gamma_1^*,\ \Gamma_2^*)$ is found as the criteria of selection of fog node in our proposed selection strategy. Similarly, the mean of the rank of the acquired fog nodes when the candidates are distributed according to a normal distribution with mean 0.5 and standard deviation 0.125 is shown in Fig. 4.8 for a total of $N = 100$ candidate fog nodes.

The mean number of nodes examined before the selection of one of the candidate fog node in the network is shown in Fig. 4.9 which shows that as the value of threshold Γ_1, Γ_2 is increased, the examination of a larger number of nodes is required.

The bar chart shown in Fig. 4.10 illustrates the performance of our proposed strategy as compared to the other three conventional selection strategies of Ferguson, Blanc and Gilbert in terms of the mean rank acquired from the available candidate fog nodes for different values of N, i.e., $N \in \{10,\ 50, 100,\ 500\}$. During the evaluation of each strategy, it is assumed that the rank of the candidate fog nodes is assumed to be uniformly distributed in the range of 0 to 1 and the occurrence of best-ranked candidate at any sequence number is considered to be equally likely.

It can be seen in Fig. 4.10 that the strategy proposed by Fergusen et. al. [38], is the worst performing among all the proposed strategies. The strategies proposed by Gilbert [40] and Blanc [39] are found to be somewhat closer to our proposed strategy. However, our proposed strategy is still found to be best performing as compared to all the other three conventional strategies. This is owing to the fact that the Blanc strategy [39] also aims to achieve the maximum of the mean of the rank of the acquired candidates, and the Gilbert [40] strategy changes its selection criteria if it is not able to find a best candidate after examination of a certain number of candidates. Similarly, Fig. 4.11 shows the comparison of performance of our proposed strategies with the conventional strategies when the rank of the fog nodes is distributed according to the normal distribution with mean 0.5 and standard deviation 0.125 and shows that our proposed strategy performs better when compared to the conventional ones.

The major drawback of all these selection strategies is that they reject a certain number of candidates while building a perception about the availability of the rank of the candidates and help making an informed decision about the selection of a new candidate. When a candidate having a best-rank among all N candidates is rejected, the performance of most of the selection strategies degrades significantly.

In Fig. 4.12, we compare the performance of our proposed strategy with the other conventional strategies when a best-ranked candidate is encountered at sequence number 1 and the rank of the fog nodes is distributed according to the standard uniform distribution. It is evident from the bar chart that strategies proposed by Fergusen [38] and Blanc [39] could achieve the mean rank of the acquired fog nodes as 0.5 under such a scenario. The strategy proposed by Gilbert [40] could achieve the mean rank as 0.65 when $N = 500$, whereas our proposed strategy outperforms all the conventional schemes and achieves the mean rank of the acquired fog node as high as 0.91 when $N = 500$. Similar trends can be seen in Fig. 4.13 when the rank fog nodes are distributed normally with mean rank of 0.5 and standard deviation of 0.125 and our proposed strategy performs much better when compared to the conventional schemes.

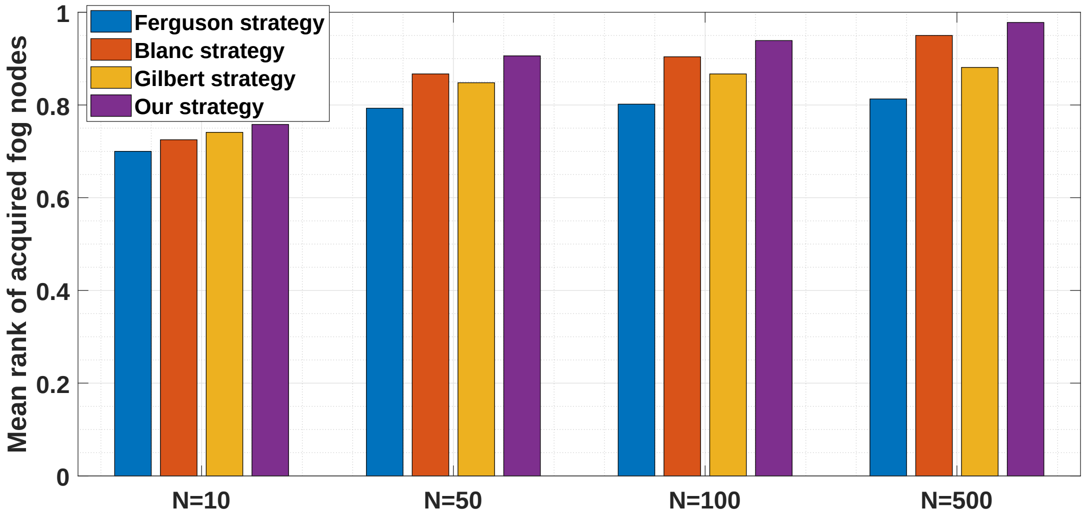

Figure 4.10: Comparison of performance of our proposed strategy with the conventional selection strategies in terms of the mean rank of the acquired fog nodes when $N \in \{10, 50, 100, 500\}$ and the rank of candidate is distributed according to the standard uniform distribution.

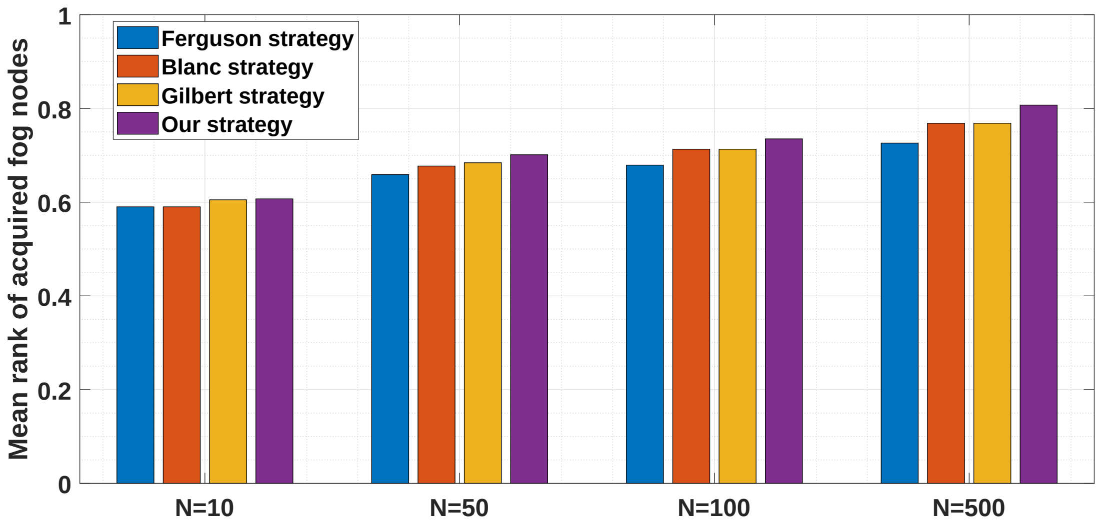

Figure 4.11: Comparison of performance of our proposed strategy with the conventional selection strategies in terms of the mean rank of the acquired fog nodes when $N \in \{10, 50, 100, 500\}$ when the rank of candidate is distributed according to the normal distribution with mean 0.5 and standard deviation 0.125.

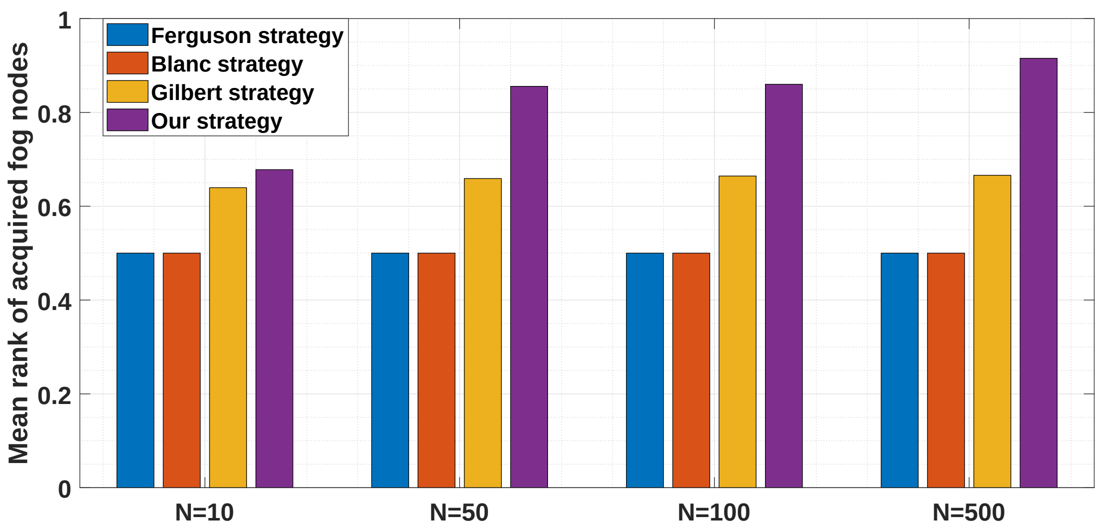

Figure 4.12: Comparison of performance of our proposed strategy with the conventional selection strategies in terms of the mean rank of the acquired fog nodes when $N \in \{10, 50, 100, 500\}$ when the best candidate appears at sequence number 1 and ranks are distributed according to the standard uniform distribution.

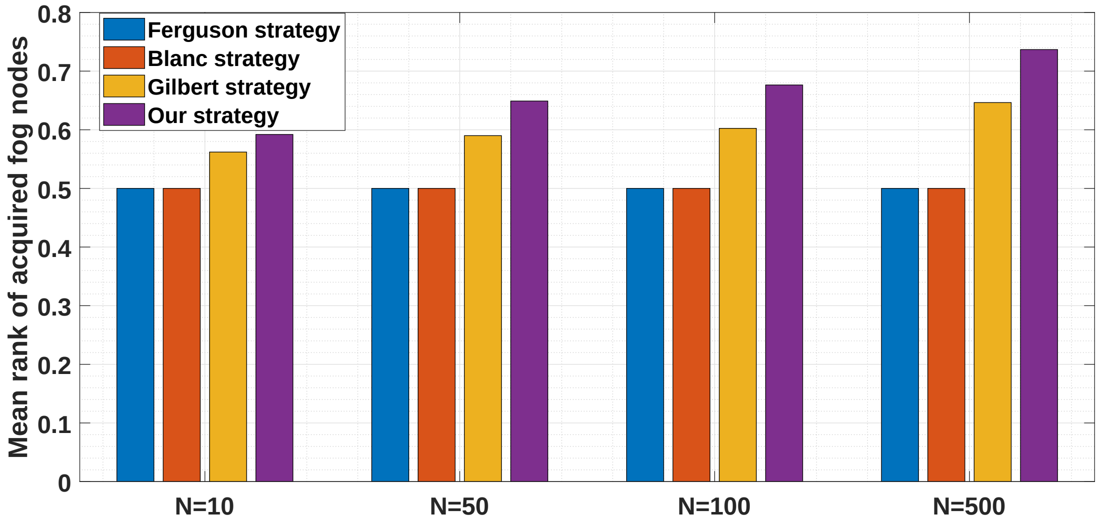

Figure 4.13: Comparison of performance of our proposed strategy with the conventional selection strategies in terms of the mean rank of the acquired fog nodes when $N \in \{10, 50, 100, 500\}$ when the best candidate appears at sequence number 1 and ranks are distributed according to the normal distribution with mean rank 0.5 and standard deviation 0.125.

4.4 Conclusions

The IoT network can be managed efficiently by introducing an intermediary layer between IoT and cloud which is called the Fog layer and consists of devices with small to medium range of processing power. These devices are located closer to the end users and the tasks processed in the fog layer help in reducing the overall network traffic towards the cloud data centers and help reduce the requirements of communication bandwidth and minimize the overall latency. FaaS enables ordinary computing devices in the vicinity to offer their services to be used in the fog network, and hence improves the overall capability without any additional cost of deployment of a fixed infrastructure. As there may be many candidate computing devices offering their services for the network, the selection of a most effective among them opens up new avenues of research. In this chapter, we developed an algorithm to formulate the strategy to select the best or second best fog node which maximizes the overall computing capability of the fog network. The acquisition of fog nodes with better processing capabilities results in reduction of the overall latency in processing the tasks generated by the IoT.

4.5 Appendix

4.5.1 Appendix 4A

Let $X_1, X_2, ..., X_n$ be continuous independent and identically distributed (i.i.d) random variables, each with pdf $f(x)$ and cdf $F(x)$, then the density of the k^{th} smallest random variable is the probability that one of them is in an interval $[x,\ x+\epsilon]$, exactly $k-1$ out of the remaining $n-1$ are less than x, and is given as

$$f_{(n,k)}(x) = \sum_{i=1}^{n} Pr\{X_i \in [x,\ x+\epsilon]\} \times \frac{(n-1)!}{(k-1)!(n-k)!} \times [F(x)]^{k-1}[1-F(x)]^{n-k}.$$

Since all X_i are identically distributed, we get

$$f_{(n,k)}(x) = \frac{n!}{(k-1)!(n-k)!}\ f(x)\ [F(x)]^{k-1}[1-F(x)]^{n-k}. \tag{4.28}$$

For $X_1, X_2, ..., X_n$ uniformly distributed from 0 and 1, each with pdf $f(x) = 1$ and cdf $F(x) = x$, the density of the k^{th} smallest random variable among all n uniformly distributed random variables is

$$f_{(n,k)}(x) = \frac{n!}{(k-1)!(n-k)!} x^{k-1}(1-x)^{n-k}.$$

Since $f_{(n,k)}(x)$ is a density function, so

$$\int_0^1 f_{(n,k)}(x) dx = 1.$$

Therefore, for a n uniformly distributed random variables, we get

$$\frac{n!}{(k-1)!(n-k)!} \int_0^1 x^{k-1}(1-x)^{n-k} dx = 1. \tag{4.29}$$

The mean of the k^{th} smallest of n random variables is then found as

$$\mu^U_{(n,k)} = \int x f_{(k,n)}(x)$$
$$= \frac{n!}{(k-1)!(n-k)!} \int_0^1 x^k (1-x)^{n-k} dx.$$

Using (4.29) to evaluate the integral, we get

$$\mu^U_{(n,k)} = \frac{n!}{(k-1)!(n-k)!} \times \frac{k!(n-k)!}{(n+1)!} = \frac{k}{n+1}.$$

Therefore, the mean of the largest (i.e., the n^{th} smallest among n random variables distributed according to standard uniform distribution) is

$$\mu^U_{(n)} = \frac{n}{n+1},$$

and the mean of the second largest (i.e., the $(n-1)^{th}$ smallest among n random variables distributed according to the standard uniform distribution) is

$$\mu^U_{(n-1)} = \frac{n-1}{n+1}.$$

4.5.2 Appendix 4B

Let $X_1, X_2, ..., X_n$ be independent and identically distributed random variables, then density of the k^{th} smallest value among them all k random variables is

$$f_{(n,k)}(x) = \frac{n!}{(k-1)!(n-k)!} f(x) \, [F(x)]^{k-1} [1-F(x)]^{n-k}. \tag{4.30}$$

and when X_i's are normally distributed with mean μ and variance σ^2, then density of the k^{th} smallest value among them all k random variables is

$$f_{(n,k)}(x) = \frac{n!}{(k-1)!(n-k)!} \times \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \times \left[\frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{x} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} dx\right]^{k-1} \times \left[1 - \frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{x} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} dx\right]^{n-k}. \tag{4.31}$$

Then, the mean of the k^{th} smallest among all n random variables is

$$\mu_{(n,k)}^{\mathcal{N}} = \int_{-\infty}^{\infty} x f_{(n,k)}(x) dx.$$

By using the values from (4.31) and using the approximation [92], the mean of the largest among all n i.i.d random variables distributed according to the standard normal becomes

$$\mu_{(n)}^{\mathcal{N}} \cong \mu - \sigma \left[\Phi^{-1}\left(\frac{1-\alpha}{n-2\alpha+1}\right)\right], \tag{4.32}$$

where $\alpha = 0.375$, and $\Phi^{-1}(p)$ is the inverse normal function with parameter p. Also, the mean of the second largest among all n i.i.d random variables distributed according to the standard normal is found by using approximation [92] as

$$\mu_{(n-1)}^{\mathcal{N}} \cong \mu - \sigma \left[\Phi^{-1}\left(\frac{2-\alpha}{n-2\alpha+1}\right)\right]. \tag{4.33}$$

Chapter 5

Modeling of RAW

In this chapter, we present a statistical model to organize the nodes into groups, quantify the throughput for saturated, unsaturated and generic modes, and develop an analytical model for finding the mean of the time required to transmit a given number of data packets using an absorbing Markov Chain. With the help of our analytical framework, we find that the duration of time required to transmit a given number of data packets by a group of nodes assigned to a RAW slot.

5.1 Introduction

In the previous chapter, we assumed that AP allocates the duration of the RAW slot which may vary dependent on the number of data packets which are required to be transmitted in the RAW slot and the number of contending nodes in a group assigned to the RAW slot. In this chapter, we estimate the duration of RAW slot in IEEE 802.11ah by use of a Markov chain model which could be allocated by a group of nodes. Dependent on the number of data packets in the buffer of each node, the groups in IEEE 802.11ah can be classified to be in saturation mode, unsaturated mode, or generic mode. In saturation mode, each node of the group is assumed to have an infinite number of data packets in its buffer. Therefore, after successful transmission of a data packet, another one is made readily available by the node. Hence, each node in saturation mode contends for the channel access during the entire duration of their allocated RAW slot and the effective number of nodes contending for the medium remains constant in the RAW slot. In unsaturated mode, each node in a group is assumed to have only one data packet in its buffer. Therefore, the number of nodes contending for the medium access reduces with each successful transmission in unsaturated mode.

In generic mode, each node in the group has a finite number of data packets in its buffer. Therefore, after successful transmission of its data packet, the node may or may not have a data packet for transmission. In IoT and machine to machine (M2M) communications, only some devices such as the ones used in industrial applications may have enough memory to store large amounts of data in their buffer and therefore, can act in a saturated mode.

Most of the real networks, e.g., smart grid, Intelligent Transportation Systems (ITS), indoor/outdoor surveillance systems, etc., [93], [94] have resource-constrained devices in terms of processing capabilities, transmission power and memory, and therefore either operate in unsaturated mode or in generic mode [95]. Therefore, it is quite useful to develop an analytical model for different modes such as saturated, unsaturated and generic modes and evaluate the performance of IEEE 802.11ah both of which are presented in this chapter.

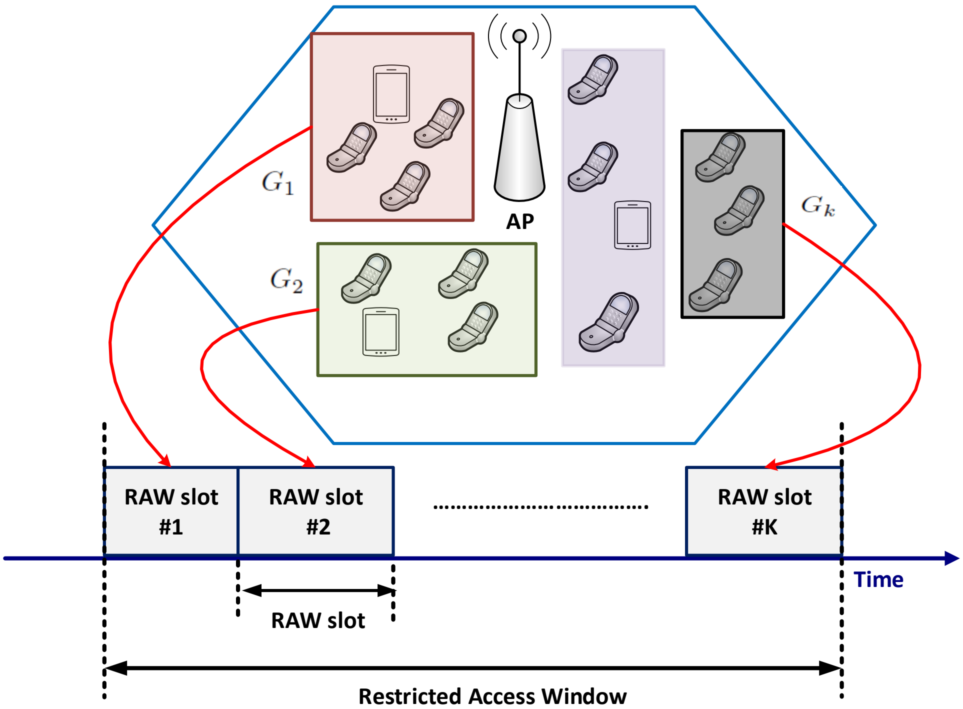

Figure 5.1: Node assignment to various RAW slots in IEEE 802.11ah according to a grouping scheme.

5.1.1 Related Work

Several past studies have focused on the performance analysis of contention based channel access protocols. Bianchi in [61] presented a Discrete Time Markov Chain (DTMC) model to find the transmission probability of a node and found the saturation throughput of IEEE 802.11 DCF. Hu et. al and Robinson et. al, extended his model in [62], [63] respectively. The mean value analysis [48], and fixed point analysis [96] are other approaches to find the probability of transmission and collision without using DTMC for the legacy IEEE 802.11 network. As described in chapter 1, the authors in in [46] and [47] develop a mathematical model for the grouping of nodes which can be used in RAW. In [49], an analysis to find the influence of network and traffic conditions for different station grouping parameters is presented. Some of the other related works which evaluate the performance of the RAW mechanism are presented in [50]-[60], where the nodes are assumed to be either in saturation mode or unsaturated mode. However, in reality, a node may have a finite number of data packets and may receive another data packet from the upper layer while it is in the process of transmitting the previous one, i.e., the generic mode.

5.1.2 Contribution

In this chapter, we present an analytical framework to:

- find the mean duration of time until any of the contending nodes transmits its data packet in a RAW slot, and analyze the time required to transmit a given number of data packets in a RAW slot for saturated, unsaturated and generic modes by developing Markov Chains in section 5.3.2.
- present two practical grouping schemes for nodes, i.e., uniform and random grouping, and then analyze the throughput for saturated, unsaturated and generic modes for these grouping schemes in section 5.3.3.
- estimate the duration of time required to transmit a given number of data packets by a group of nodes assigned to a RAW slot efficiently, and thus find the duration of the beacon interval in the IEEE 802.11ah standard to improve the overall throughput.

5.2 System Model

In this chapter, we consider a network of N nodes which are divided among K groups and each group is assigned to a RAW slot as shown in Fig. 5.1. For the sake of analytical tractability, we investigate the problem at MAC layer assuming that the channel impairments can be tackled by the physical layer processing. It is also assumed that there are no hidden terminals in the network and all the data packets are sent from source to the AP in a single hop. Moreover, each node is assumed to have a fixed-length data packet. The duration of a RAW slot is divided into mini-slots (each with the time duration σ). We develop stochastic models for saturated, unsaturated and generic modes by describing a sequence of states where each state depends only on the state attained in the previous event, and develop a Markov Chain for each mode by assuming that a group of n nodes is allocated to a RAW slot at its beginning.

Let $\gamma_c^{[n]}$ and $\gamma_s^{[n]}$ be the probability of collision and success respectively, where the superscript $[n]$ represents the number of nodes contending for the medium access in a RAW slot. A list of the most frequently used symbols throughout this paper along with their description is shown in table 5.1.

In our model, the state of the system at a given time describes the number of nodes contending for the medium access and the number of data packets transmitted successfully in a RAW slot, i.e., the system is in state (k, i) when k nodes contend for the medium access to transmit their data packets and i data packets have already been transmitted successfully in a RAW slot. Let there be n nodes in a group assigned to the RAW slot which contend with each other for the medium access, then at the beginning of a RAW slot when the transmission of data packets is yet to be started, the system is in state $(n, 0)$. Based on the assumption about whether the nodes in the group assigned to the RAW have a single, multiple or infinite number of data packets in their buffer, the system models for the three scenarios (saturated, unsaturated and general) using Markov Chain are as follows.

Table 5.1: A LIST OF MOST FREQUENTLY USED SYMBOLS AND THEIR DESCRIPTION

Description	**Symbol**
Probability of success when n nodes are contending for medium	$\gamma_s^{[n]}$
Probability of collision when n nodes are contending for medium	$\gamma_c^{[n]}$
Number of transmission attempts when n nodes are contending for medium	$R^{[n]}$
Number of slots a node backsoff during r^{th} transmission attempt	β_r
Average attempt rate when n nodes are contending for medium	$A(\gamma_c^{[n]})$
Time until transmission attempt when n nodes are contending for medium	$T_A^{[n]}$
Mean time to reach absorbing state (m,n) starting from state $(0,n)$ in saturated mode	$E[\zeta_{(n,0)}^{(m,n)}]$
Mean time to reach absorbing state $(n,0)$ starting from state $(0,n)$ in unsaturated mode	$E\left[\zeta_{(n,0)}^{(0,n)}\right]$
Mean time to reach one of the absorbing states $\mathcal{AS}^m$ starting from state $(0,n)$ in generic mode	$E\left[\zeta_{(n,0)}^{\mathcal{AS}^m}\right]$
Set of larger sized groups	Ω_l
Set of smaller sized groups	Ω_s
Group index	G_k
Beacon Interval when nodes are in mode mod, where $mod \in \{sat,unsat,gen\}$	BI_{mod}
Throughput when nodes are in mode mod, where $mod \in \{sat,unsat,gen\}$	Γ_{mod}
Time duration to successfully transmit a packet in basic access mechanism	T_s^{basic}
Time duration when transmission attempt fails in basic access mechanism	T_c^{basic}
Time duration to successfully transmit a packet in RTS/CTS access mechanism	$T_s^{RTS/CTS}$
Time duration when transmission attempt fails in RTS/CTS access mechanism	$T_c^{RTS/CTS}$

Model I: Saturation Mode

The system is in saturation mode when all the nodes in a group assigned to a RAW slot are assumed to have an infinite number of data packets in their buffer. In such a mode, a data packet is made readily available by a node after the successful transmission of its previous packet. In this way, all the nodes contend for the medium access for the entire duration of the RAW slot. Therefore, the probability of collision and success remains constant during the RAW slot in saturation mode. Assuming that n nodes are assigned to the RAW slot and each of these nodes is in saturation mode, the Markov Chain model is shown in Fig. 5.2 where the system remains in the same state when the packet faces collision and moves from state (k,i) to the next state $(k,i+1)$ when the data is transmitted successfully by the node. Hence,

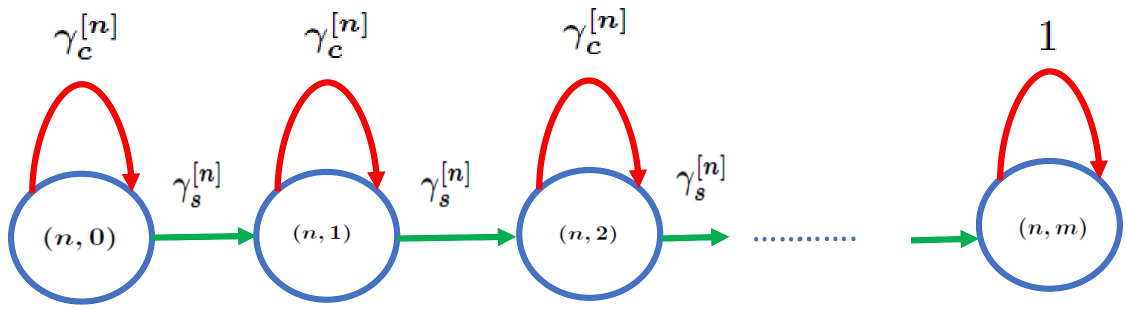

Figure 5.2: Markov Chain model of a system where all n nodes are in saturation mode, i.e., each node has an infinite number of data packets in its buffer.

the probability of transition from state (k, i) to state (k, j) is given as

$$p_{sat,(k,i)}^{(k,j)} = \begin{cases} \gamma_c^{[k]}, & \text{for} \quad j = i \\ \gamma_s^{[k]}, & \text{for} \quad j = i + 1 \\ 0 & \text{otherwise} \end{cases}, \tag{5.1}$$

where $i \in \{0, 1, 2, ...m - 1\}$ and m is the maximum number of data packets that can be transmitted in the system during the RAW slot. When the system has transmitted m data packets successfully, it reaches an absorbing state (n, m) such that $p_{sat(n,m)}^{(n,m)} = 1$ and $p_{sat(n,m)}^{(n,j)} = 0, \ \forall \ j \in \{0, 1, 2, ...m - 1\}$.

Model II: Unsaturated mode

In an unsaturated mode, each node in a group assigned to a RAW slot is assumed to have only one data packet for transmission and does not receive any further data packet from the upper layer before the duration of the RAW slot is over. In such a mode, the contention of the node for the medium access is deemed as complete after successful transmission of its data packet. In this way, the number of nodes contending for the medium access reduces with each successful transmission in a

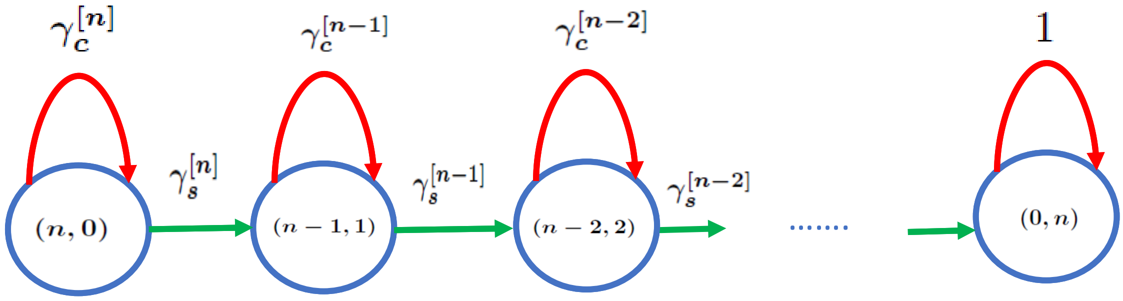

Figure 5.3: Markov Chain model of a system where the nodes are in unsaturated mode, i.e., each node has only one data packet for transmission and the number of contending nodes reduces with each successful transmission.

RAW slot. Assuming that n nodes are assigned to the RAW slot and each of these nodes is in unsaturated mode, the Markov Chain model is shown in Fig. is shown in Fig. 5.3 where the system remains in the same state when the packet faces collision(s), and moves from state (k, i) to state $(k-1, i+1)$ with a successful transmission of a data packet as the node has no more data packets in its buffer to remain a contender for the medium access in such a mode. Hence, the probability of transition from state (k, i) to state (q, j) is given as

$$p_{unsat,(k,i)}^{(q,j)} = \begin{cases} \gamma_c^{[k]}, & \text{for} \quad q = k \ \ \& \ j = i \\ \gamma_s^{[k]}, & \text{for} \quad q = k-1 \ \ \& \ j = i+1 \\ 0 & \quad \text{otherwise} \end{cases}, \tag{5.2}$$

where $k, i \in \{0, 1, 2, ...m-1\}$, and m is the maximum number of data packets that can be transmitted in the system during the RAW slot. After the successful transmission of n data packets in the RAW slot in an unsaturated mode, there remain no more nodes for transmission of data and the system is said to be in the absorbing state $(0, n)$, i.e., $p_{ns,(0,n)}^{(0,n)} = 1$ and $p_{ns,(0,n)}^{(q,j)} = 0$, $\forall\ q, j \in \{0, 1, 2, ...n-1\}$. Note that the system can transmit a maximum of n data packets when n nodes in unsaturated mode are assigned to a RAW slot as each node has only one data packet in such a mode. Hence, $m = n$ in an unsaturated mode.

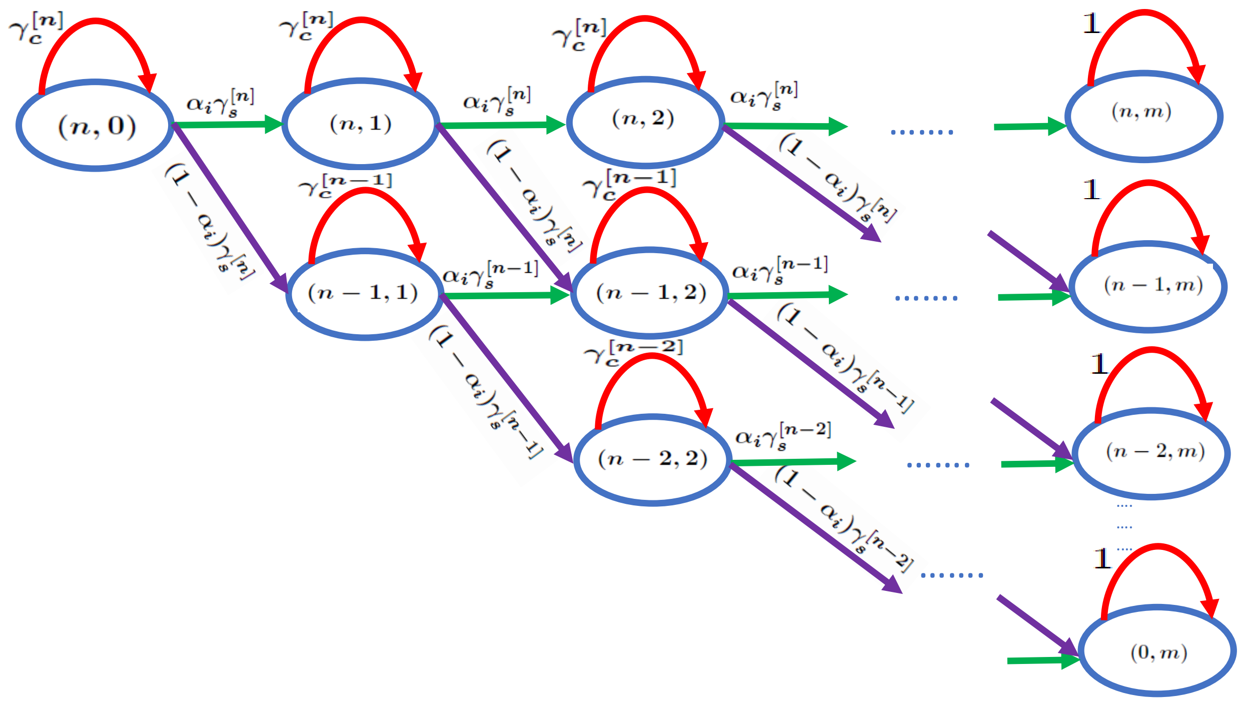

Figure 5.4: Markov Chain model of a system where the nodes are in generic mode i.e., each node may or may not have any further data after the successful transmission of its current packet.

Model III: Generic mode

The system is in a generic mode when each node in a group has a finite number of data packets for transmission in a RAW slot, and may receive another packet from the upper layer before the RAW slot is over. In such a mode, a node after successful transmission of its data packet may or may not have any further data packets. The contention of the node for the medium access is deemed complete if it has no more data packets for transmission. Therefore, the number of nodes contending for the medium access may or may not reduce with each successful transmission when the system is in a generic mode during the RAW slot. We assume that the nodes keep the task data in their buffer when it arrives. Let α_i be the probability that the node i, after successful transmission of its data, still has more packets in its buffer. Assuming that n nodes are assigned to the RAW slot and each of these nodes is in the generic mode, the Markov Chain model is shown in Fig. 5.4 where the system remains in the same state when the data transmitted through the medium faces a collision, and transits to one of the next states when a data packet is transmitted successfully by one of the contending nodes, i.e., the system in state (k, i) transits to state $(k-1, i+1)$ if data is transmitted successfully and the node transmitting the data has no more packets to transmit, or to state $(k, i+1)$ when the data is transmitted successfully and the node transmitting the data still has one or more data packets in its buffer that are yet to be transmitted. Hence, the probability of transition from the state (k, i) to state (q, j) is given as

$$p_{gen,(k,i)}^{(q,j)} = \begin{cases} \gamma_c^{[k]}, & \text{for} \quad q = k \ \ \& \ j = i \\ \alpha_i \gamma_s^{[k]}, & \text{for} \quad q = k \ \ \& j = i+1 \\ (1-\alpha_i)\gamma_s^{[k]}, & \text{for} \quad q = k-1 \ \ \& j = i+1 \\ 0 & \quad \text{otherwise} \end{cases}, \tag{5.3}$$

$\forall\ k \in \{0, 1, 2, ..., n-1\}$ & $i \in \{0, 1, 2, ..., m-1\}$, where m is the maximum number of data packets that can be transmitted in the system during the RAW slot. After the successful transmission of m data packets in the RAW slot in a generic mode, the system is said to be in the absorbing state (k, m), i.e., $p_{gen,(k,m)}^{(k,m)} = 1$ and $p_{gen,(k,i)}^{(k,j)} = 0$, where $0 \leq k \leq n,\ i \neq j$. Note that the system model described by the generic mode reduces to saturated model when $\alpha_i = 1$, and to the unsaturated mode when $\alpha_i = 0$.

The Markov Chain models developed above for saturated, unsaturated and generic modes provide a framework to estimate the duration of the RAW slot and the beacon interval and thus facilitate finding the throughput of the RAW being used in the IoT network in section 5.3.

5.3 Analytical Model for RAW

When any of the n contending nodes attempts to transmit its data packet, the system transits to the next state with probability $\gamma_s^{[n]}$ in time T_s and remains in the same state with probability $\gamma_c^{[n]}$ in time T_c after the transmission attempt is initiated. In section 5.3.1, we find the mean duration of time until an attempt of transmission is initiated by any of the contending nodes and the probability of collision and the success of a data packet for the state transition of the system. Then, the mean time required to transmit a given data packet for saturated, unsaturated and generic mode is found in section 5.3.2 and the throughput of the RAW is calculated according to uniform and random grouping schemes in section 5.3.3.

5.3.1 Aggregate Attempt process

In order to find the mean duration of the waiting time until an attempt of transmission is initiated by any of the contending nodes and the probability of collision and success of data packet for state transition of the system, we define an *aggregate attempt process* as a process in which a given number of nodes assigned to a RAW slot attempt to transmit their packets according to a backoff procedure. In this procedure, each node having a data packet for transmission in the RAW slot enters into a backoff stage by assigning a random value (chosen uniformly from its contention window) to the backoff counter. The node then monitors the channel activity after every min-slot, σ and decrements its backoff counter by 1 if the medium is found idle. When the value of the backoff counter reaches 0, the node starts its transmission. On each successive collision, the size of the contention window of the node gets doubled until it reaches the maximum value $2^{R_{max}-1}W_0$, where W_0 is the size of the initial contention window, and R_{max} is the maximum number of transmission attempts a node can make consecutively before discarding the data packet. The value of R_{max} depends on either the Long Retry Limit (LRL) which is in use for RTS/CTS access mechanism, or the Short Retry Limit (SRL) which is used for basic access mechanism in IEEE 802.11ah. When the packet is transmitted successfully, the size of the contention window is reset to the initial value, i.e., W_0.

We use the mean value analysis as in [96], [46] to find the mean duration of waiting time until any of the contending node initiates an attempt to transmit its data packet. We assume that each node comes across with the same network congestion and faces the same collision probability $p_c^{[n]}$, where n is the total number of nodes contending for the medium access. Moreover, we assume that the collision probabilities associated with each node are independent of each other, and the node discards its data packet when it encounters R_{max} consecutive collisions during its transmission attempts, where the value of R_{max} depends on LRL or SRL according to the access mechanism. Let R be a random variable representing the number of attempts made by the node assigned to the RAW slot for a data packet. Therefore, the average number of transmission attempts made by the node assigned to the RAW slot including the case when the packet is either transmitted successfully or discarded during the R_{max}^{th} attempt is [46]

$$E[R^{[n]}] = \sum_{r=1}^{R_{max}-1} r\left(1-p_c^{[n]}\right)\left(p_c^{[n]}\right)^{r-1} + R_{max}\left(\left(1-p_c^{[n]}\right)\left(p_c^{[n]}\right)^{R_{max}-1} + \left(p_c^{[n]}\right)^{R_{max}}\right), \tag{5.4}$$

which is simplified as

$$E[R^{[n]}] = \frac{1 - \left(p_c^{[n]}\right)^{R_{max}}}{1 - p_c^{[n]}}. \tag{5.5}$$

Let β_r be a random variable representing the number of mini-slots the node waits in the backoff stage before the r^{th} transmission attempt of the data packet and is chosen uniformly from a contention window $[0 \quad \min(2^{r-1}W_0, CW_{max}) - 1]$,where CW_{max} is the maximum value of the contention window. Then, the mean number of mini-slots the node waits during the r^{th} transmission attempt is

$$E[\beta_r] = \frac{2^{r-1}W_0 - 1}{2}. \tag{5.6}$$

Then, the average number of mini-slots the node waits including the case for the R_{max}^{th} attempt when the packet is either transmitted successfully or discarded is then given by

$$E[B^{[n]}] = \sum_{r=1}^{R_{max}-1} \left(p_c^{[n]}\right)^{r-1} (1-p_c^{[n]}) \sum_{j=1}^{r} E[\beta_j] + \left[\left(1 - p_c^{[n]}\right)\left(p_c^{[n]}\right)^{R_{max}-1} + \left(p_c^{[n]}\right)^{R_{max}}\right] \sum_{j=1}^{R_{max}} E[\beta_j], \tag{5.7}$$

which is simplified as

$$E[B^{[n]}] = \sum_{r=1}^{R_{max}} \left(p_c^{[n]}\right)^{r-1} \left(1 - p_c^{[n]}\right)^{I_{\{r<R_{max}\}}} \sum_{j=1}^{r} E[\beta_j], \tag{5.8}$$

and $I_{\{x\}}$ is the indicator function which is equal to 1 when x is true and 0 otherwise. Eq. (5.8) can be further simplified as

$$E[B^{[n]}] = \sum_{r=1}^{R_{max}} \left(p_c^{[n]}\right)^{r-1} E[\beta_r]. \tag{5.9}$$

As each node faces the same collision probability $p_c^{[n]}$ when n nodes are contending for the medium access, the average rate at which a node attempts to transmit its data packet in a mini-slot is

$$A\left(p_c^{[n]}\right) = \frac{E[R^{[n]}]}{E[B^{[n]}] + E[R^{[n]}]}. \tag{5.10}$$

Let p_A be the probability that a node attempts in a mini-slot and is equal to the average attempt rate, i.e.,

$$p_A^{[n]} = A\left(p_c^{[n]}\right). \tag{5.11}$$

When there are n nodes contending for the medium access and a tagged node has started transmitting, it will face collision when any of the remaining $(n-1)$ nodes is also transmitting at the same time. Considering that the probability for each node to transmit is $p_A^{[n]}$, the conditional collision probability is

$$p_c^{[n]} = 1 - (1 - p_A)^{n-1}. \tag{5.12}$$

By numerically solving equations (5.11) and (5.12), the average attempt rate and probability of collision faced by a node can be found. Given that at least one node attempts to transmit in a mini-slot, then the probability of success of a packet in the system where n nodes are attempting to transmit is

$$\gamma_s^{[n]} = \frac{np_A^{[n]}(1 - p_A^{[n]})^{n-1}}{1 - (1 - p_A^{[n]})^n}. \tag{5.13}$$

Then, the probability that a packet is not successful in the system where n nodes are attempting to transmit is $\gamma_c^{[n]} = 1 - \gamma_s^{[n]}$. Since all the n nodes in a group attempt to transmit, each with probability p_A , the mean number of mini-slots until any of the n contending nodes initiates an attempt to transmit is

$$E[T_A^{[n]}] = \frac{1}{1-(1-p_A^{[n]})^n}. \tag{5.14}$$

In this way, the duration of time when the process of state transition is initiated is evaluated and is equal to $\sigma E[T_A^{[n]}]$ where σ is the mini-slot duration.

5.3.2 Mean of the time to transmit a given number of data packets

In this subsection, we find the mean duration time required to transmit a given number of data packets in a RAW slot, i.e., the the mean of the first passage time to move from an initial state $(n, 0)$ to one of the absorbing state. The mean time until any of the k contending nodes initiates an attempt to transmit a data packet is $\sigma E[T_A^{[k]}]$ when k nodes are contending for the medium access, T_c is the duration of time taken by the node when the packet faces collision, and T_s is the time in transmitting a data packet successfully, the total time in transmitting a data packet successfully including the back-off is $T_s + \sigma E[T_A^{[k]}]$, and the total time when a data packet faces collision including the backoff is $T_c + \sigma E[T_A^{[k]}]$ during an attempt.

Case-I: As described in the model for saturated mode, the system in state (k, i) remains in the same state with probability $\gamma_c^{[k]}$, or transits to state $(k, i+1)$ with probability $\gamma_s^{[k]}$, the mean duration of time to transit from state (k, i) to the next state $(k, i+1)$ is

$$E\left[\zeta_{(k,i)}^{(k,i+1)}\right] = \sum_{r=0}^{\infty} \left(T_s + \sigma E[T_A^{[k]}] + r\left(T_c + \sigma E[T_A^{[k]}]\right)\right)\left(\gamma_c^{[k]}\right)^r \gamma_s^{[k]}, \tag{5.15}$$

which is simplified as

$$E\left[\zeta_{(k,i)}^{(k,i+1)}\right] = T_s + \sigma E[T_A^{[k]}] + \frac{\left(T_c + \sigma E[T_A^{[k]}\right)\gamma_c^{[k]}}{1-\gamma_c^{[k]}}. \tag{5.16}$$

Let $\zeta_{(k,i)}^{\mathcal{AS}}$ be the first time the system transits from state (k,i) to one of the absorbing states, $\mathcal{AS} \in \{(n,m)\}$, where m is the maximum number of data packets to be transmitted in the RAW slot. Then, the mean duration of time to reach an absorbing state, starting from state $(n,0)$ is found as

$$E\left[\zeta_{(n,0)}^{(n,m)}\right] = \sum_{i=0}^{m-1} E\left[\zeta_{(n,i)}^{(n,i+1)}\right]. \tag{5.17}$$

Case-II: In unsaturated mode, the system in state (k,i) either remains in the same state with probability $\gamma_c^{[k]}$, or transits to state $(k-1, i+1)$ with probability $\gamma_s^{[k]}$. Then, the mean duration of time to transit from state (k,i) to the next state $(k-1,i+1)$ can be found by using eq. (5.16). As described in the system model, each node in unsaturated mode has only one data packet and hence, the maximum number of data packets that can be transmitted in such a mode is equal to the the number of nodes assigned to the RAW slot at its beginning. Therefore, the absorbing state in this mode is $\mathcal{AS} \in \{(0,n)\}$, and the mean time duration to transmit n data packets, starting from state $(n,0)$ is found as

$$E\left[\zeta_{(n,0)}^{(0,n)}\right] = \sum_{i=0}^{n-1} T_s + \sigma E[T_A^{[[n-i]}] + \frac{\left(T_c + \sigma E[T_A^{[n-i]}]\right)\gamma_c^{[n-i]}}{1-\gamma_c^{[n-i]}}. \tag{5.18}$$

Case III: In generic mode, as described in the system model, the system in state (k,i) remains in the same state with probability $\gamma_c^{[k]}$, or transits either to state $(k,i+1)$ with probability $\alpha_i\gamma_s^{[k]}$ or to state $(k-1,i+1)$ with probability $(1-\alpha_i)\gamma_s^{[k]}$, where α_i is the probability that the node i still has a data packet after successful transmission of its previous data packet. Then, the mean time duration to transit from state (k, i) to one of the next states $\mathcal{N}^k$ where $\mathcal{N}^k \in \{(k, i+1), (k-1, i+1)\}$, is

$$E\left[\zeta_{(k,i)}^{\mathcal{N}^k}\right] = \sum_{r=0}^{\infty} \left(T_s + \sigma E[T_A^{[k]}] + r\left(T_c + \sigma E[T_A^{[k]}]\right)\right) \left(\gamma_c^{[k]}\right)^r \left(\alpha_i \gamma_s^{[k]} + (1-\alpha_i)\gamma_s^{[k]}\right), \tag{5.19}$$

which is simplified as

$$E\left[\zeta_{(k,i)}^{\mathcal{N}^k}\right] = T_s + \sigma E[T_A^{[k]}] + \frac{\left(T_c + \sigma E[T_A^{[k]}\right)\gamma_c^{[k]}}{(1-\gamma_c^{[k]})}. \tag{5.20}$$

Let $\zeta_{(k,i)}^{\mathcal{AS}^m}$ be the first time the system transits from state (k, i) to one of the absorbing states, $\mathcal{AS}^m \in \{(n, m), (n-1, m),\ ....,\ (0, m)\}$, where m is the maximum number of data packets to be transmitted in the RAW slot. Therefore, we get $\zeta_{\mathcal{AS}^m}^{\mathcal{AS}^m} = 0$, i.e., the time it takes the system to transit from an absorbing state to itself, is equal to 0. The mean duration of time to reach one of the absorbing states starting from a transient state (n, i) is then found recursively as

$$E\left[\zeta_{(n,i)}^{\mathcal{AS}^m}\right] = \frac{\left(\left(T_c + \sigma E[T_A^{[n]}\right)\gamma_c^{[n]} + \alpha_i\gamma_s^{[n]} E[\zeta_{(n,i+1)}^{\mathcal{AS}^m}] + (1-\alpha_i)\gamma_s^{[n]} E[\zeta_{(n-1,i+1)}^{\mathcal{AS}^m}]\right)}{\left(1-\gamma_c^{[n]}\right)} + T_s + \sigma E[T_A^{[n]}] \tag{5.21}$$

and the solution of the set of equations obtained in this way provides the mean duration of time to reach to one of the absorbing states from the state $(n, 0)$.

In this way, the mean duration of time required to transmit m data packets is found

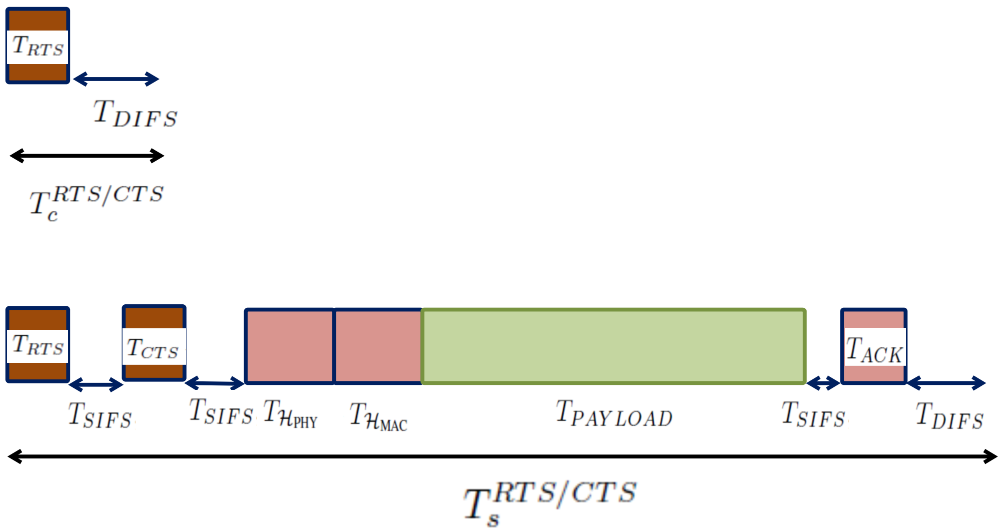

Figure 5.5: The time duration for sucessful and unsuccessful packet transmission in the RTS/CTS access scheme.

for saturated, unsaturated modes and generic modes. To specifically compute the mean time in transmission of a data packet for a given DCF access mechanism, the corresponding values of T_c and T_s are specified as follows: In the basic access mechanism as shown in Fig. 5.6, the values of T_s and T_c, including the ACK timeout which the colliding stations need to wait, are [60]

$$T_c^{basic} = T_{\mathcal{H}_{\mathrm{PHY}}} + T_{\mathcal{H}_{\mathrm{MAC}}} + T_{PAYLOAD} + T_{DIFS} + T_{SIFS} + T_{ACK} + 2T_\delta, \quad (5.22)$$

$$T_s^{basic} = T_{\mathcal{H}_{\mathrm{PHY}}} + T_{\mathcal{H}_{\mathrm{MAC}}} + T_{PAYLOAD} + T_{DIFS} + T_{SIFS} + T_{ACK} + 2T_\delta, \quad (5.23)$$

where $T_{\mathcal{H}_{\mathrm{PHY}}}$, $T_{\mathcal{H}_{\mathrm{MAC}}}$, $T_{PAYLOAD}$, T_{DIFS}, T_{SIFS}, T_{ACK} are the time duration for transmission of physical header, MAC header, payload, DIFS, SIFS, Ack and T_δ is the propagation delay. In the basic access mechanism, it can be found that $T_c^{basic} \cong T_s^{basic}$. In the RTS/CTS access mechanism as shown in Fig. 5.5, the collision can occur only on the RTS frame, so the values of T_s and T_c, including the

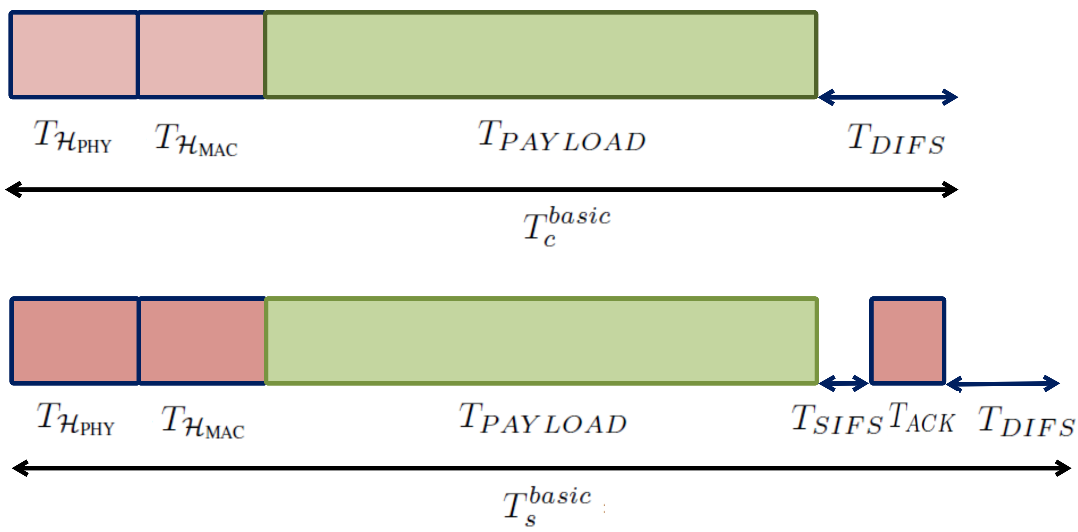

Figure 5.6: The time duration for successful and unsuccessful packet transmission in the basic access scheme.

ACK timeout which the colliding stations need to wait, are

$$T_c^{RTS/CTS} = T_{RTS} + T_{DIFS} + T_{SIFS} + T_{CTS} + 2T_\delta, \tag{5.24}$$

$$T_s^{RTS/CTS} = T_{\mathcal{H}_{\text{PHY}}} + T_{\mathcal{H}_{\text{MAC}}} + T_{RTS} + T_{CTS} + 3T_{SIFS} + T_{PAYLOAD} + T_{DIFS} + T_{ACK} + 4T_\delta, \tag{5.25}$$

where T_{RTS}, T_{CTS} are the time durations for transmission of RTS and CTS respectively, and in the case of collision duration, T_{ACK} is equal to T_{CTS}.

5.3.3 Throughput Analysis

In order to find the overall throughput of RAW, we group nodes and assign them to different RAW slots in this section. We assume that there is a network with a total of N nodes indexed as $n_1, n_2, n_3, ...n_N$. These nodes are divided among K groups indexed as $G_1, G_2, ..., G_K$ and each group G_k is assigned to the k^{th} RAW slot where $k \in 1, 2, 3, ...K$ as shown in Fig. 5.1 [58] according to node assignment the policy given as [97], [58]

$$k = (i + N_{offset}) \mod K, \tag{5.26}$$

where i is the index of the node and $i \in \{1, 2, 3....N\}$, k is the index of RAW slot and K is the total number of groups such that $k \in \{0, 2, 3, ...K - 1\}$. According to this policy, a group of nodes can be of size either $\left\lfloor \frac{N}{K} \right\rfloor$ or$\left(\left\lfloor \frac{N}{K} \right\rfloor + 1\right)$, where $\lfloor x \rfloor$ means the largest integer less than or equal to x. Let Ω_l denote a set of groups indexed as $G_1, G_2, ..., G_{|\Omega_l|}$ where $|\Omega_l|$ is the cardinality of the set and is given as [60]

$$|\Omega_l| = N \mod K. \tag{5.27}$$

Similarly, let Ω_s denote a set of groups indexed as $\{G_{|\Omega_l|+1}, G_{|\Omega_l|+2}, ...G_{|\Omega_l|+|\Omega_s|}\}$, where $|\Omega_s|$ is the cardinality of the set and is given as

$$|\Omega_s| = K \quad - N \mod K. \tag{5.28}$$

Note that $|\Omega_l| + |\Omega_s| = K$. Also, when N is an integer multiple of K, $|\Omega_l| = 0$ and $|\Omega_s| = K$, i.e., all the groups have equal number of nodes when N is an integer multiple of K. Let $|G_l|$ denote the cardinality of each member of set Ω_l, and is given as

$$|G_l| = \left\lfloor \frac{N}{K} \right\rfloor + 1, \tag{5.29}$$

and the cardinality of each member of set Ω_s, denoted by $|G_s|$, is

$$|G_s| = \left\lfloor \frac{N}{K} \right\rfloor. \tag{5.30}$$

Once the nodes are assigned to the groups, and are allocated to the RAW slots, the duration of each RAW slot is determined for different cases in our system model such as saturated, non-saturated, and generic modes and then overall throughput can be found as follows.

Case I: In saturated mode where each node is assumed to have an infinite number of data packets in its buffer, eq. (5.17) provides the mean time duration to transmit d_r data packets in the r^{th} RAW slot. Then, the total beacon interval of RAW in saturated mode when nodes are divided into K RAW slots is given as

$$BI_{sat} = \sum_{r=1}^{|\Omega_s|} E\left[\zeta_{(|G_s|,0)}^{(|G_s|,d_r)}\right] + \sum_{r=1}^{|\Omega_l|} E\left[\zeta_{(|G_l|,0)}^{(|G_l|,d_r)}\right]. \tag{5.31}$$

Case II: In unsaturated mode where each node is assumed to have only one data packet, eq. (5.18) gives the mean duration of time to transmit a number of data packets equal to the number of nodes in the RAW slot. Then, the total beacon interval of RAW in unsaturated mode when nodes are divided into K RAW slots is

$$BI_{unsat} = \sum_{r=1}^{K} E\left[\zeta_{(|G_r|,0)}^{(0,|G_r|)}\right]. \tag{5.32}$$

Case III: In generic mode where each node is assumed to have a finite number of data packet, eq. (5.21) gives the mean time duration to transmit d_r data packets in the r^{th} RAW slot. Then, the total beacon interval of RAW in unsaturated mode when nodes are divided into K RAW slots according to uniform grouping scheme is given as

$$BI_{gen} = \sum_{r=1}^{|\Omega_l|} E\left[\zeta_{(|G_l|,0)}^{\mathcal{AS}^{d_r}}\right] + \sum_{r=|\Omega_l|+1}^{K} E\left[\zeta_{(|G_s|,0)}^{\mathcal{AS}^{d_r}}\right]. \tag{5.33}$$

The throughput is then found in each mode as

$$\Gamma_{mod} = \frac{N}{BI_{mod}}, \tag{5.34}$$

where $mod \in \{sat,\ unsat,\ generic\}$.

Table 5.2: Parameters used in simulations with their symbols

Parameters	Symbol	Value
Payload	$T_{PAYLOAD}$	64 bytes / 1024 bytes
MAC Header	T_{MAC}	272 bits
PHY Header	T_{PHY}	128 bits
ACK	T_{ACK}	240 bits
RTS	T_{RTS}	288 bits
CTS	T_{CTS}	240 bits
SIFS	T_{SIFS}	28 μsec
DIFS	T_{DIFS}	128 μsec
RAW mini-slot	σ	52 μsec
Propagation delay	T_{δ}	6 μsec
Size of Initial Contention Window	W_0	16
Long Retry Limit for R_{max}	LRL	7
Short Retry Limit for R_{max}	SRL	5

5.4 Performance Evaluation and results

In order to evaluate the performance of our model, we perform simulations by using the PHY and MAC layer parameters presented in the draft of the IEEE 802.11ah and IEEE 802.11 standard [98]. These parameters are listed in table 5.2. It is assumed that the channel data rate is 1 Mbps, and we obtain the results for two different payload sizes i.e., 64 bytes and 1024 bytes to cover different traffic patterns.

5.4.1 Mean time until attempt and success probability

We consider that there are n nodes assigned to a group in a RAW slot and find the mean waiting time until any node attempts to transmit in the medium via simulations using Matlab for both SRL ($R_{max} = 5$), and LRL ($R_{max} = 7$). As shown in Fig. 5.7, the theoretical results obtained in section 5.3.1 match well with the simulations. The results show that as the number of contending nodes increases, the mean time until any of the contending nodes attempt to transmit reduces. Moreover, the mean time until any node attempts to transmit reduces for the smaller value of retry limit R_{max}. This is because of the fact that as a node discards a data packet after a fewer re-transmission attempts, the contention window is reset to its initial size during transmission of a new data packet and causes the node to choose relatively smaller backoff value. Thus, the mean value of backoff that the node takes reduces with a smaller retry limit which leads to an increase in the mean attempt rate of the node and causes a reduction of mean time until any node attempts in the shared medium.

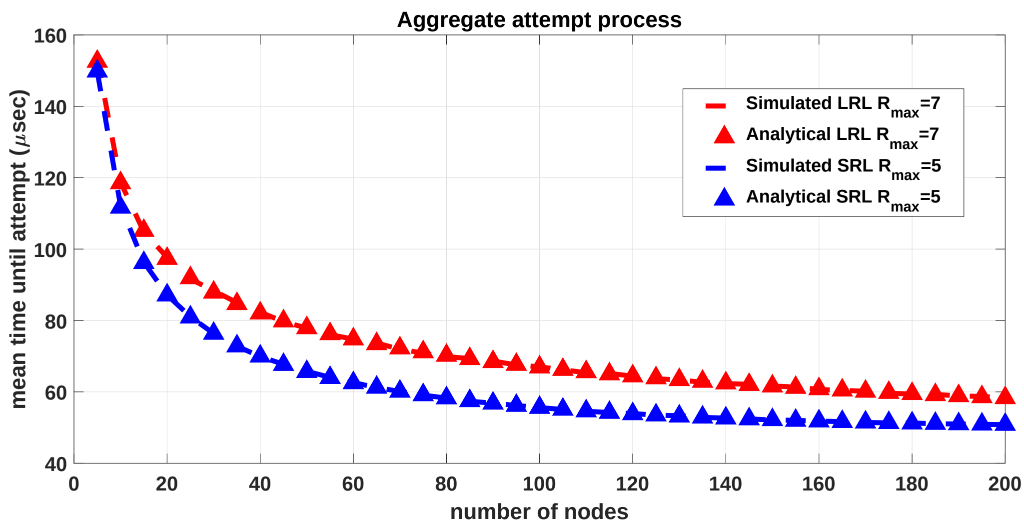

Figure 5.7: Mean waiting time until an attempt for a given number of nodes where each node has a fixed probability of transmission.

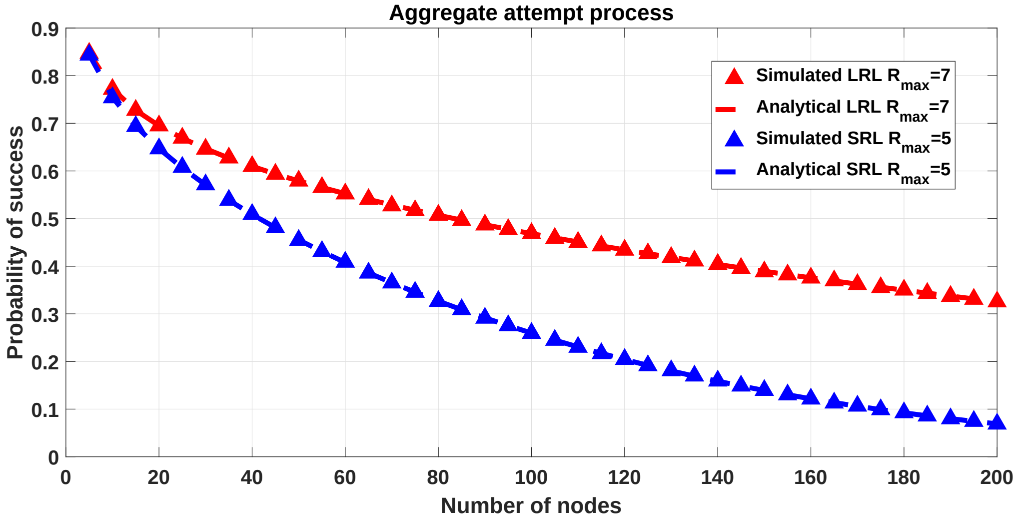

Figure 5.8: Probability of success for a given number of nodes in a group with a given probability of transmission for each node.

Fig. 5.8 shows the probability of success for a given number of nodes in a group and the size of the initial contention window $W_0 = 16$ for both SRL ($R_{max} = 5$), and LRL ($R_{max} = 7$). It is clear from the Fig. 5.8 that the success probability through the shared medium decreases as the number of contending nodes increases.

Moreover, it can be seen that the success probability reduces for SRL as compared to LRL. This is because when a data packet is discarded after fewer re-transmission attempts, the attempt to transmit a new data packet is made after a fewer backoff slots and this leads to an increased value of mean attempt rate of the nodes. Hence, the probability of success reduces when the attempt rate of the contending nodes increases. Again, the results obtained in simulations match quite well with the analytical ones.

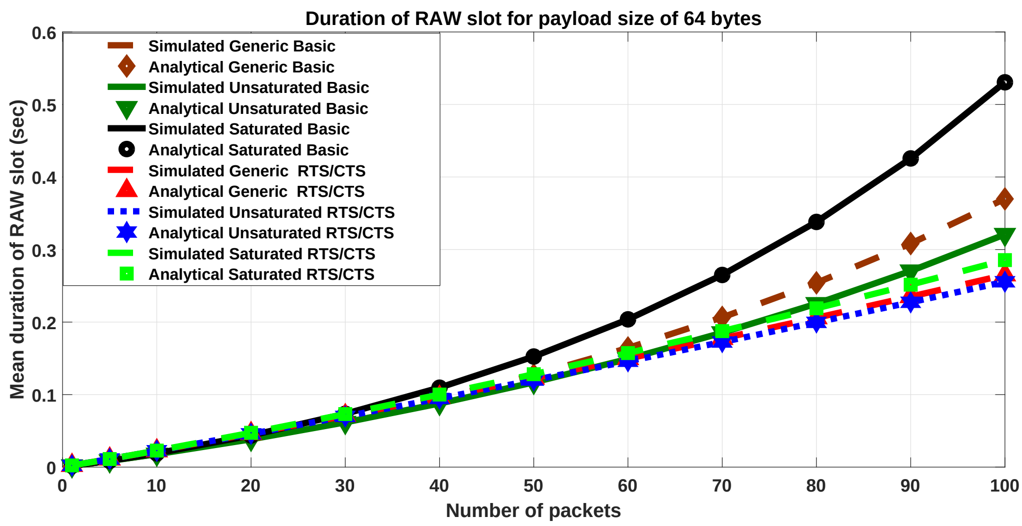

Figure 5.9: Mean time duration to transmit n data packets using the basic access and the RTS/CTS mechanism for saturated, unsaturated and generic modes for a payload of 64 bytes.

5.4.2 Mean time to transmit m data packets

Considering that there are n nodes in the group which attempt to transmit in the RAW slot according to either the basic access mechanism where each node can make as much as $R_{max} = 5$ transmission attempts, or the RTS/CTS access mechanism where each node can make as many as $R_{max} = 7$ transmission attempts, Fig. 5.9 shows the mean duration of time required to transmit n data packets for saturated, unsaturated, and generic modes with $\alpha_i = 0.3$. In this case, the size of payload is chosen as either of 64 bytes (short data packet). The Monte Carlo simulations carried out for finding the mean duration of time match well with the analytical model presented in section 5.3.2.

It can be seen that the RTS/CTS access mechanism takes a shorter time to transmit a given number of data packets compared to the basic access mechanism. This is because the time wasted in the event of collision in the RTS/CTS mechanism is less than that in the basic access mechanism. Moreover, it is evident that when nodes

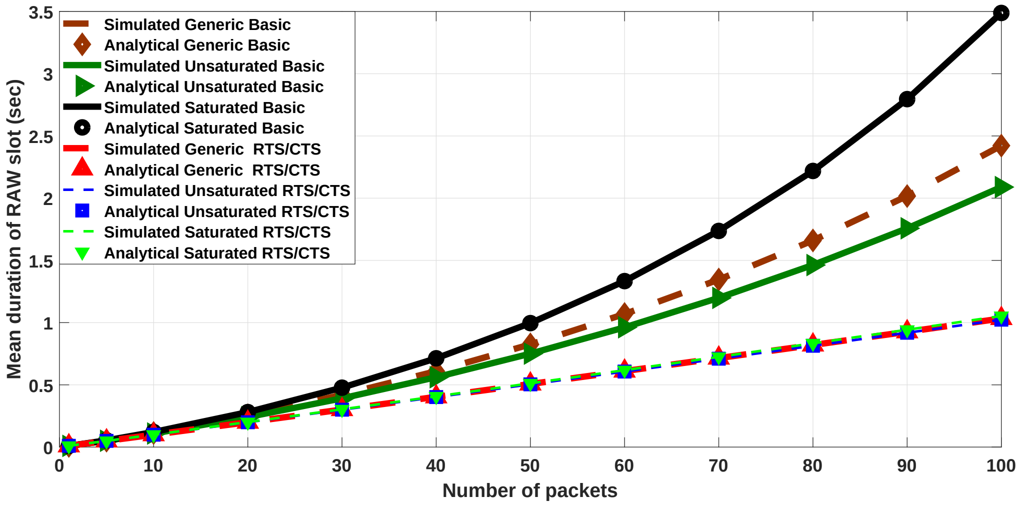

Figure 5.10: Mean time duration to transmit n data packets using basic access and RTS/CTS mechanism for saturated, unsaturated and generic modes for a payload of 1024 bytes.

are in unsaturated mode, the mean time required to transmit a given number of data packet is less than that in the saturated and generic mode because the number of contending nodes decreases with each successful transmission of data packets, and the mean time duration is highest in the case of saturated mode where the collision probability always remains constant because all the nodes in the group keep contending during the entire duration of the RAW slot. The time duration in the case of generic mode depends on the value of α_i, i.e., as α_i approaches unity, the likelihood of a node having another data packet in its buffer increases, which causes it to remain in contention for the medium access for longer, resulting in more chances of collision. Hence, the duration of the RAW slot is the highest for the saturated case, i.e., when $\alpha_i = 1$ for all the nodes in the group assigned to the RAW slot. Fig. 5.10 shows the mean time duration required to transmit n data packets for saturated, unsaturated, and generic modes with $\alpha_i = 0.3$ when the size of data packet is 1024 bytes.

The comparison of Fig. 5.9 and 5.10 shows that it requires more time to transmit data packets with the payload size 1024 as compared to when the payload size is 64 bytes, and the trends related to basic access mechanism and RTS/CTS are somewhat similar in both cases. When the size of data packet is quite large and the RTS/CTS access mechanism is used, the size of the RTS/CTS transmitted during collision is quite small as compared to the size of data packet to be transmitted successfully, and hence the amount of time wasted during collision of data packets becomes negligibly small as compared to the time utilized in transmitting the data packet successfully. This leads to almost same mean time duration for the RAW slot in saturated, unsaturated and generic modes.

The comparison of the basic and the RTS/CTS access schemes under saturated, unsaturated and generic modes shows that the mean time duration required to transmit a given number of data packets is almost same when a reduced number of nodes are contending for the medium access. This is owing to the fact that the likelihood of collision becomes meager when the number of contending nodes is less, and even though the RTS/CTS mechanism wastes a smaller duration of time in the event of collision, the overall time wasted during collision becomes almost comparable in both basic and RTS/CTS access. This is because the likelihood of collision is approaches zero as the number of of nodes become smaller are contending for medium access. Also, time spent during the transmission of RTS/CTS actually proves to be an additional overhead in the event of successful data packet transmission which is not present when the basic access scheme is used.

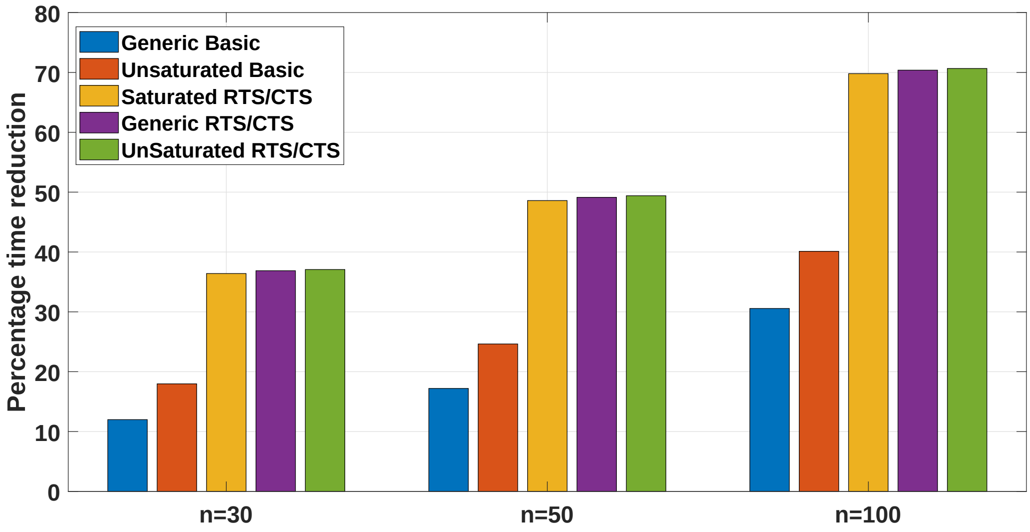

Figure 5.11: Percentage of time reduced when different schemes are used as compared to the basic access scheme in saturated mode for a 1024 byte payload.

However, as the number of contending nodes increases, the basic access scheme requires more time to transmit a given number of data packets when compared to the RTS/CTS mechanism as more time is wasted in transmission of the actual payload in the basic access scheme in the event of collision as compared to the RTS/CTS,. Also, with an increased probability of collisions in the case of a large number of nodes, the time wasted because of collisions in basic access schemes becomes more pronounced.

Moreover, it can be observed that the time consumed in the saturation mode is quite large as compared to the cases when nodes are considered in either unsaturated mode or generic modes. This is because of the fact that the number of contending nodes remains constant in saturated mode during the entire duration of RAW slot and hence, more collisions happen during the RAW slots in the saturated mode as compared to the other modes.

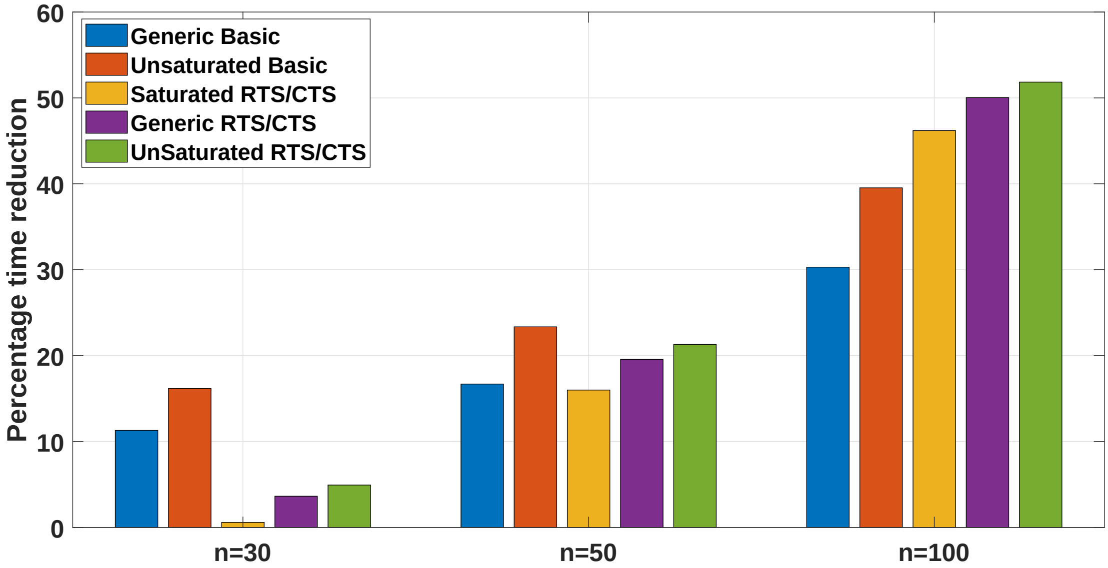

Figure 5.12: Percentage of time reduced when different schemes are used as compared to the basic access scheme in saturated mode for a 64 byte payload.

Fig. 5.11 shows the percentage of time reduced by using the different access schemes with saturated, unsaturated and generic modes when compared to the basic access scheme with nodes in saturated mode for a payload of 1024 bytes. It can be observed that the time is reduced significantly when the traffic pattern is considered as unsaturated. As the number of nodes in the group contending for the medium access becomes large, the basic access scheme takes more time to transmit data as compared to the RTS/CTS scheme. Therefore, the best access scheme in terms of time consumption for transmission of a given number of data packets for a large number of contending nodes in a RAW slot is the RTS/CTS scheme with nodes considered in unsaturated mode. However, it can be seen that RTS/CTS with generic mode, a realistic and generalized implementation of IoT scenario, is somewhat closer to the best scheme for a large number of contending nodes in a RAW slot.

Fig. 5.12 shows the percentage of time reduced by using the different access schemes with saturated, unsaturated and generic modes when compared to the basic access scheme with nodes in saturation mode for a payload of 64 bytes. It can be seen that the duration of the RAW slot required to transmit a give number of data packets is reduced significantly when nodes in unsaturated mode are accessing the medium using the RTS/CTS mechanism as compared to the other access mechanisms.

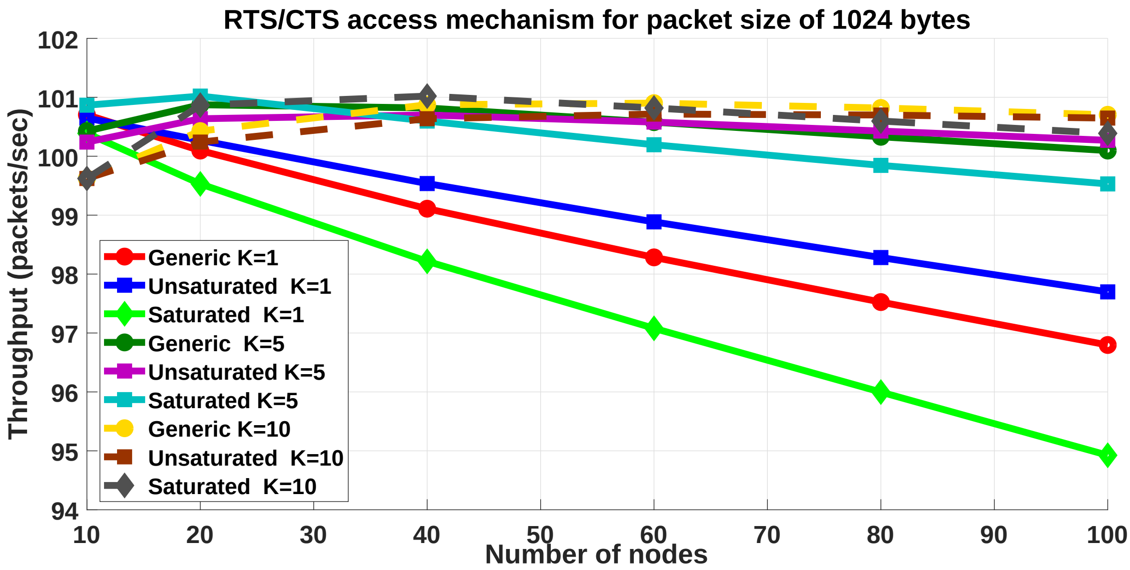

Figure 5.13: Throughput when RTS/CTS scheme is used for saturated, unsaturated, and generic modes with a payload of 1024 bytes and nodes are divided into K groups.

C): Throughput

Fig. 5.13 shows the throughput for the entire beacon interval when nodes are considered to access the medium according to the RTS/CTS scheme, have a packet size of 1024 bytes and are divided into K groups. The throughput is found in terms of packets per seconds where it is evident that the throughput is improved when nodes are divided into groups as compared to the case when nodes contend with each other without grouping ($K = 1$). The nodes in the group are in saturated, unsaturated, or generic modes and the throughput in the case of the saturated mode is smaller when compared with unsaturated and generic modes because the likelihood of collision during the entire duration of RAW slot is more in the case of saturated mode.

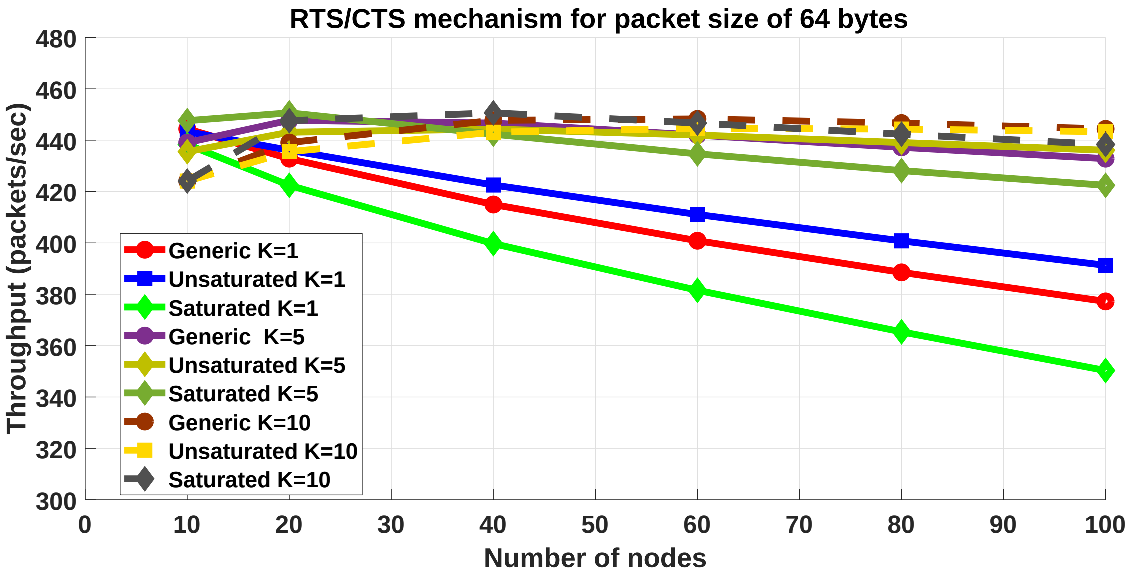

Figure 5.14: Throughput when RTS/CTS scheme is used for saturated, unsaturated, and generic mode with a payload of 64 bytes and nodes are divided into K groups.

Similarly, Fig. 5.14 shows the throughput when nodes are divided into K groups, use RTS/CTS access scheme and transmit data packets of 64 bytes.

Fig. 5.15 shows the throughput for the entire beacon interval when nodes are divided into K groups, use a basic access mechanism, and have a packet size of 1024 bytes. In case when nodes are divided into $K = 10$ groups, the throughput is better as compared to when no grouping is done ($K = 1$).

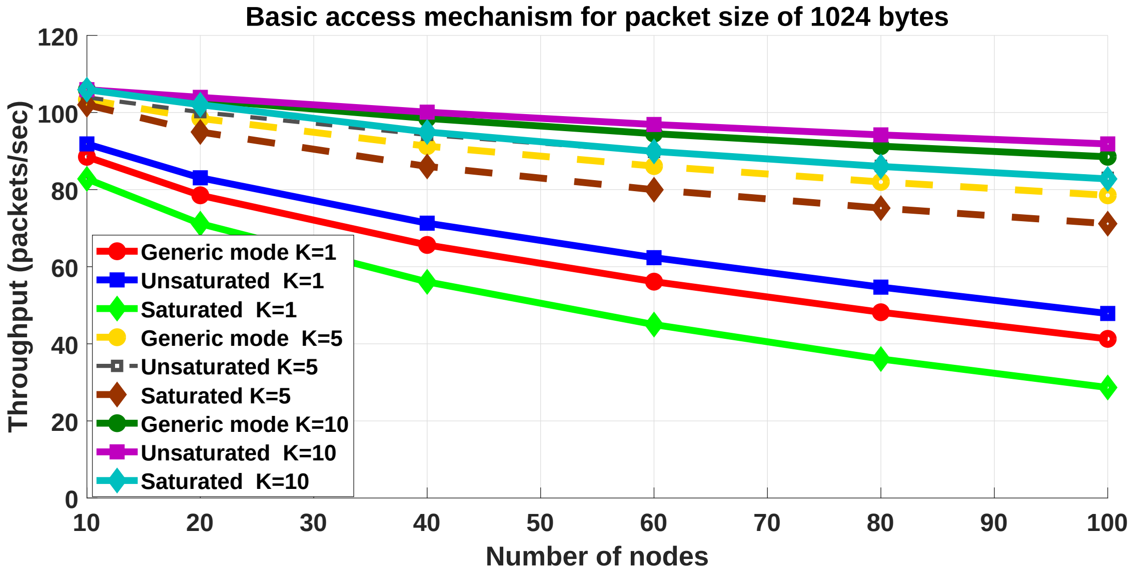

Figure 5.15: Throughput when basic access scheme is used for saturated, unsaturated, and generic mode with a payload of 1024 bytes and nodes are divided into K groups.

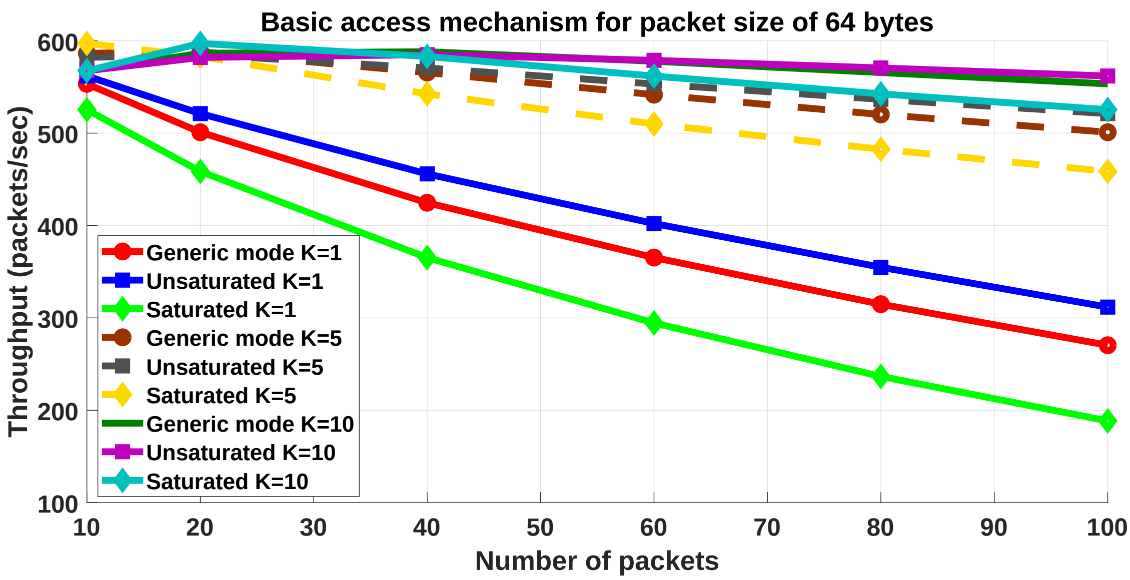

Figure 5.16: Throughput when basic access scheme is used for saturated, unsaturated, and generic mode with a payload of 1024 bytes and nodes are divided into K groups.

Similarly, Fig. 5.16 shows the throughput when nodes are divided into K groups, use the basic access scheme and can transmit data packets of 64 bytes. It is evident from these results that the use of the RTS/CTS access mechanism and RAW slots in the beacon interval yields better throughput for a large number of nodes – a scenario typical in IoT deployment.

5.5 Conclusions

One salient feature of IEEE 802.11ah is RAW which allows nodes to be grouped together and only the nodes within the group contend with each other for the medium access by using the prevalent DCF techniques. The duration of a RAW slot for a given number of nodes is calculated by an analytical models presented in this paper. Based on the number of data packets a node can transmit, there can be three modes, i.e., each node in the group has only one data packet; each node in the group has an infinite number of data packets and always has a data packet available for transmission; and each node in the group has a finite number of data packets and may receive another one from the upper layer. In this chapter, we present a model which first analyses the attempt process of all the nodes in a group contending for the medium access in a RAW slot.

Given the probabilities of collision and success during the attempt process, our model evaluates the time duration required for transmission of a given number of data packets in a RAW slot according to different modes of the nodes in the group assigned to the RAW slot. In the RTS/CTS mechanism, the time consumed in the collision of packets is less than the time to transmit the data packet successfully for a large number of contending nodes. For IoT, the most common scenario is a node having either a single data packet or a finite number of data packets, and it transmits short data packets. Analysis of the system model developed in this chapter, is quite useful for performance evaluation of RAW with nodes having any number of data packets, using either the basic access mechanism or the RTS/CTS access scheme. It is observed that that the use of the RTS/CTS access mechanism for a larger number of nodes in the groups by considering nodes either in unsaturated or generic modes yields better performance in terms of throughput as compared to the other modes.

Chapter 6

Throughput Enhancement of 2 Sub frame based RAW

In this chapter, we present a two-subframe based RAW model where the nodes are organized into groups and are assigned to a RAW slot in one of the subframes according to the size of the groups. We use Bianchi's DTMC model to model the contention among nodes in each RAW slot, find the throughput of the RAW frame and compare it with the conventional one-frame RAW model in this chapter.

6.1 Introduction

As described previously, the existing contention based MAC protocols are expected to face performance degradation with a dense deployment of IoT and M2M nodes in large scale wireless networks. It becomes a challenging task to deal with extremely high contention, high number of collisions and re-transmission attempts in an ultra-dense IoT network. One simple approach to deal with the contention issue is adopted in a Multi-user polling Controlled Channel Access (MCCA) – a deterministic channel access in Wi-Fi Mesh networks where each node is assigned a specific time interval for transmission. Although such a collision-free approach results in an increased throughput, it also results in an extremely high latency under normal traffic and can hardly be used for communication of thousands of nodes with unpredictable traffic. Schemes such as increasing the size of the contention window have been shown to be impractical as they produce an extraordinary idle duration time and thus reduce the overall efficiency. Another approach is to group the nodes through a clustering scheme which exploits the space dimension of the radio resource [99]-[100]. However, cluster formation requires location information which may not be possible to obtain when the nodes are densely deployed. A better way to limit contentions in the IoT scenario, where traffic most of the times is quite sporadic and nodes are densely deployed, is by reducing the number of nodes contending for channel access at any given time by dividing the nodes into several groups and assigning each group a specific time interval. The nodes are allowed to transmit only in their assigned time intervals. In this way, only the nodes within a group can contend with each other at a given time. This idea was adopted by the IEEE 802.11ah Task Group in its latest draft [73] where only a group of nodes is allowed to transmit in a restricted interval called the RAW slot where the nodes contend with each other according to the legacy DCF procedure described in chapter 2.

6.1.1 Related Work

The work presented in [47], [46] provides an analytical model for finding the throughput when nodes are grouped together either uniformly or randomly but considers that the duration of the RAW slot is the same for each group. The throughput degrades significantly with increase in the number of nodes contending in a RAW slot [61]. In a conventional model, the duration of the RAW slot is kept the same for groups of different sizes. However in order to enhance the throughput, we suggest that the duration of a RAW slot should be adjusted according to the size of the group, i.e., the duration should be smaller for a larger sized group.

6.1.2 Our Contribution

In this chapter, we consider the most simplified scenario of a uniform grouping scheme where the groups can have two different sizes and propose a model where entire RAW frame is divided into two sub-frames. The groups are organized into sets on the basis of their sizes and each set of groups is assigned to a sub-frame. In our model, we choose the RAW slot duration in each sub-frame according to the size of group, i.e., a large sized group is assigned a relatively smaller RAW slot duration. In this way, the overall throughput performance is shown to have improved in our proposed model as compared to the conventional model. Our proposed model can be further extended to multi sub-frame structure where groups are organized on the basis of their sizes and equal sized groups are assigned to one sub-frame.

6.2 System Model

In this chapter, we assume that there is a network of N total nodes indexed from n_1, n_2, n_3 .....n_N. We assume the all the nodes attempt to transmit their packets in the wireless medium towards the Access Point (AP). The nodes are considered to be in saturation mode, i.e., all nodes have data packet readily available in their buffers. We also assume that there are no hidden nodes in the network and that the channel is in ideal condition where there are no communication errors. These nodes are divided

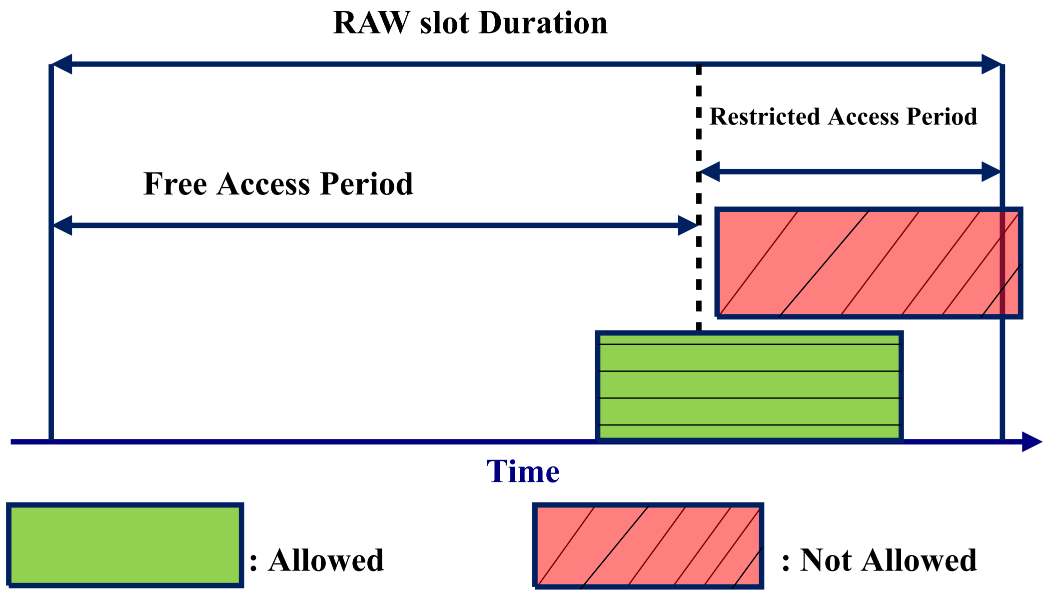

Figure 6.1: RAW slot duration with a Cross Slot Boundary.

among K groups indexed $g_1,\ g_2\ ,....g_K$ according to a uniform assignment policy. We assume that each group is assigned to a RAW slot by the AP. The nodes in a group contend with each other according to the normal DCF procedure. According to this procedure, each node having a data packet for transmission monitors the channel until it is found idle for the DCF Inter Frame Spacing (DIFS), $\ell_d = (T_{DIFS}/\tau)$ mini-slots where T_{DIFS} is the time duration of DIFS and τ is the time duration of one mini-slot. The node then enters a backoff stage and chooses a value randomly in the range of $(0,\ W_0 - 1)$ where W_0 is the size of intitial contention window. The node adopts a truncated binary exponential backoff scheme according to which the size of contention window gets doubled on each successive collision until it reaches CW_{max} and is then reset to W_0 when a packet is successfully transmitted.

The backoff counter decrements by 1 after every mini-slot. As the value of the counter reduces to zero, the node starts its transmission. We assume that the size of a data packet is constant for each node and if T_{DATA} is the time duration required for

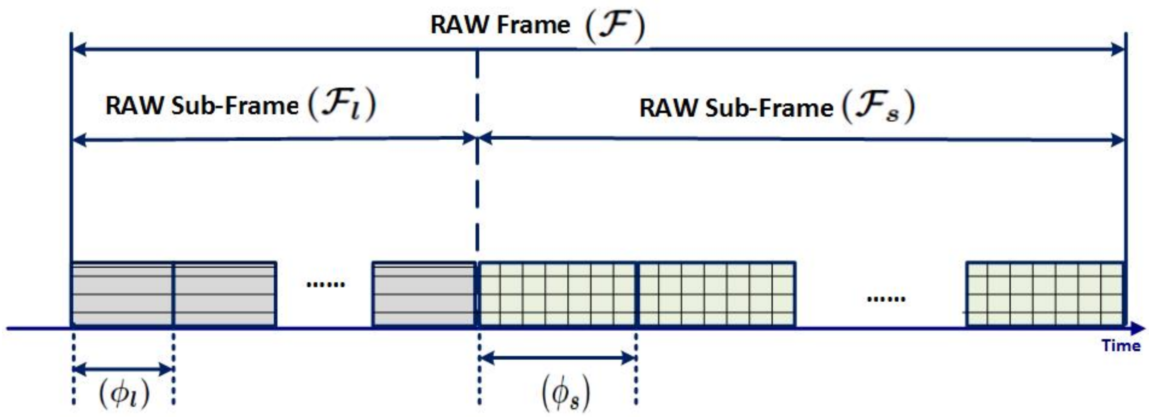

Figure 6.2: Distribution of the RAW frame according to our proposed model.

the transmission of a data packet, then the number of mini-slots needed to transmit a data packet is given as $\ell_p = (T_{DATA}/\tau)$. We also assume that any node in a group cannot cross its assigned RAW slot boundary during transmission of a data packet, i.e., nodes are restricted to start transmission during the Restricted Access Period given as $\ell_r = \ell_p + \ell_d$. If ϕ' is the duration of one RAW slot, the nodes can start their transmission only during the Free Access Period (FAP) given as $\phi = \phi' - \ell_r$. In our proposed model, a RAW frame $\mathcal{F}$ is composed of two sub-frames denoted by $\mathcal{F}_l$ and $\mathcal{F}_s$. The sub-frame $\mathcal{F}_l$ consists of $|\Omega_l|$ RAW slots and $\mathcal{F}_s$ contains $|\Omega_s|$ RAW slots. Let the duration of the RAW slot in sub-frame $\mathcal{F}_l$ and $\mathcal{F}_s$ be ϕ_l and ϕ_s respectively, then the total duration of entire RAW frame is

$$\mathcal{F} = |\Omega_l|\, \phi_l + |\Omega_s|\, \phi_s \tag{6.1}$$

where ϕ_l is the duration of each RAW slot corresponding to $\lfloor N/K \rfloor + 1$ and ϕ_s is the duration of each RAW slot for $\lfloor N/K \rfloor$ nodes. The Fig. 6.2 shows the distribution of a RAW frame according to our proposed model.

6.3 Throughput Analysis

Let $|g_i|$ be the number of nodes contending for medium access in the i^{th} RAW slot. Assuming that each packet collides with a constant and independent probability p_c, let $s(t)$ be a backoff stage that represents the number of consecutive attempts made by the node before successful transmission of a data packet at any time t, and $b(t)$ represents the value of backoff counter of a given node [61]. Then, the medium access of a node is modeled by the use of DTMC where a node is represented by a bi-dimensional process $\{s(t), b(t)\}$ as shown in Fig. 6.3. According to the DCF procedure, the backoff counter $b(t)$ decrements at the start of every mini-slot τ until it becomes 0, then

$$Pr\{s(t+\tau) = i, b(t+\tau) = k-1|s(t) = i, b(t) = k\} = 1, \quad \forall k \in (0, W_0-1)\,, i \in (0, u).$$

The transmission is done when the backoff counter decrements to 0. When the transmission becomes successful, a new packet initializes from backoff stage 0 whose backoff counter is chosen uniformly in the range $(0\ ,\ W_0 - 1)$. Thus

$$Pr\{s(t+\tau) = 0, b(t+\tau) = k|s(t) = i, b(t) = 0\} = (1-p_c)/W_0 \qquad \forall k \in (0, W_0-1),\ i \in (0, u).$$

At any backoff stage i, when the backoff counter decrements to 0 and the transmission is unsuccessful, the backoff counter chooses a value uniformly in the range $(0\ , W_i - 1)$ because the backoff stage becomes $i + 1$, and therefore

$$Pr\{s(t+\tau) = i+1, b(t+\tau) = k|s(t) = i, b(t) = 0\} = p_c/W_i, \qquad \forall k \in (0, W_i-1)\,, i \in (0, u).$$

If the packet faces consecutive failures and the backoff stage reaches at its maximum value u , then it does not increase any further, and so

$$Pr\{s(t+\tau) = u, b(t+\tau) = k|s(t) = u, b(t) = 0\} = p_c/W_u, \qquad \forall k \in (0,\, W_u - 1).$$

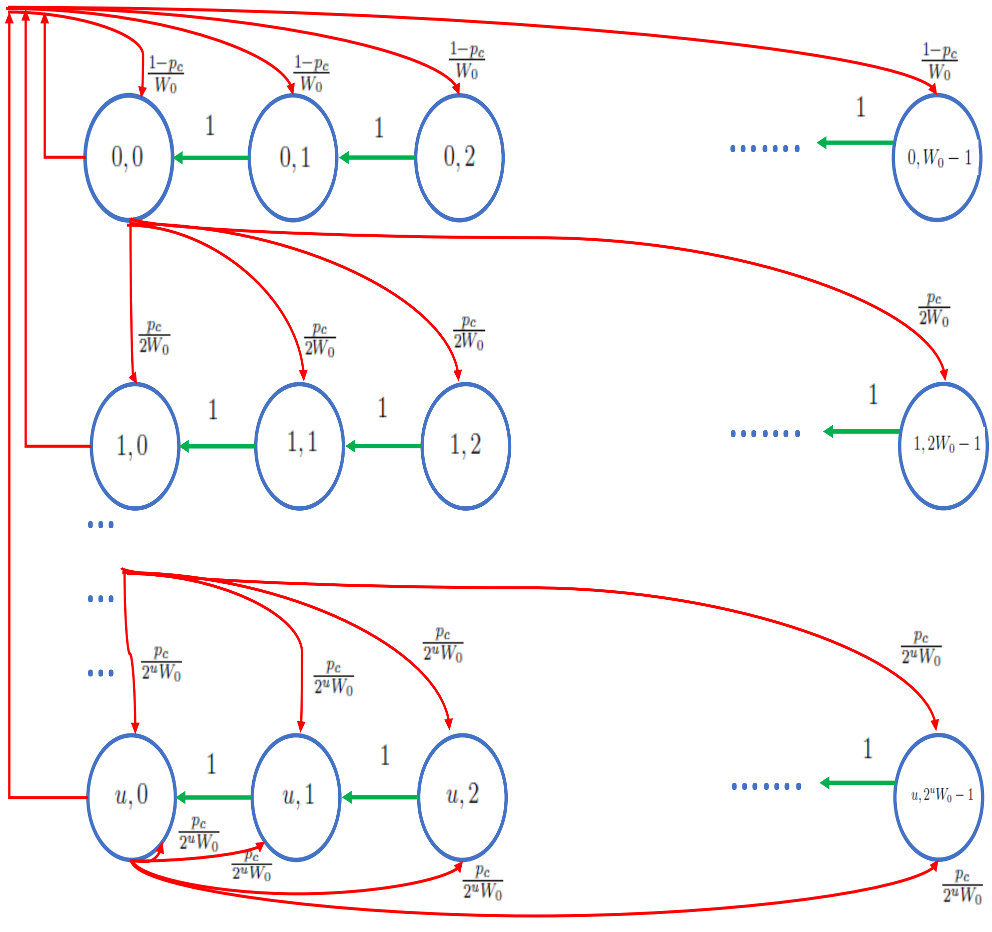

Figure 6.3: Bianchi's DTMC with one step transition probabilities

Let $\pi_{i,k} = \lim_{t\to\infty} Pr\{s(t) = i, b(t) = k\}$ $k \in (0,\, W_i - 1)$, $i \in (0,\, u)$ be the stationary distribution of DTMC. Then, we get

$$\pi_{i,k} = \frac{W_i - k}{W_i}\pi_{i,0} \qquad k \in (0,\ W_i - 1) \quad , i \in (0,\ u).$$

By imposing normalization conditions of DTMC and simplifying, $\pi_{0,0}$ is obtained as

$$\pi_{0,0} = \frac{2(1-2p_c)(1-p_c)}{(1-2p_c)(W_0+1) + p_c W_0(1-(2p_c)^u)}.$$

Since the transmission is only possible when the backoff counter is equal to zero, irrespective of the backoff stage, then the transmission probability p_t is found as

$$p_t = \sum_{i=0}^{m} \pi_{i,0} = \frac{\pi_{0,0}}{1-p_c},$$

which is simplied to

$$p_t = \frac{2(1-2p_c)}{(1-2p_c)(W_0+1) + p_c W_0(1-(2p_c)^u)}, \tag{6.2}$$

where u is the maximum number of retransmission attempts and W_0 is the size of the initial contention window. Here p_t depends on the probability of collision p_c. In each RAW slot where there are $|g_i|$ nodes, a node encounters a collision during transmission of its packet if at least one of the remaining $|g_i|-1$ nodes also start their transmission. Since each of the remaining stations also transmit with probability p_t, so we get

$$p_c = 1 - (1-p_t)^{|g_i|-1}. \tag{6.3}$$

Eq. (6.2) and Eq. (6.3) form a non-linear system having two unknowns p_t and p_c which can be solved by use of numerical methods. A node transmits successfully if exactly one node is transmitting out of $|g_i|$ nodes, provided at least one node is transmitting, then using Eq. (6.2) and Eq. (6.3), the probability of success p_s is found as [61]

$$p_s(|g_i|) = \frac{|g_i|\, p_t(1-p_t)^{|g_i|-1}}{1-(1-p_t)^{|g_i|}}. \tag{6.4}$$

Now we find the saturation throughput, i.e., the throughput when all the nodes are in saturation mode, of a single RAW slot as in [46].

Let ψ_i be a random variable representing the number of backoff slots a node i waits before starting its transmission, then we assume that it follows a geometric distribution.

$$Pr\{\psi_i = j\} = p_t(1-p_t)^{j-1} \tag{6.5}$$

Let $\triangle_b$ be a random variable representing the number of backoff slots between two consecutive transmissions and it is the minimum value of backoff slots of all $|g_i|$nodes and is equal to the minimum of $\psi_1, \psi_2, ..., \psi_{|g_i|}$.

Theorem 6.1

If $\triangle_b = min(\psi_1, \psi_2, ...\psi_{|g_i|})$, where ψ_i is a random variable representing which follows a geometric distribution, then

$$Pr\{\triangle_b = j|\,|g_i|\} = p'_t(1-p'_t)^{j-1} \quad , \quad (j \geq 1), \tag{6.6}$$

where $p'_t = 1-(1-p_t)^{j-1}$. Proof: see Appendix 6A.

Theorem 6.2

Let the random variable $\triangle_{b,m}$ denote the minimum number of backoff slots among nodes before the start of transmission. If $Y_m = \sum_{m'=1}^{m} \triangle_{b,m'}$, then the CDF of Y_m is

$$F_{Y_m}(z) = \sum_{z'=m}^{z} \binom{z'-1}{m-1} p_t'(1-p_t')^{z'-m} \quad . \tag{6.7}$$

Proof: see Appendix 6B.

Let M be a random variables representing the number of of transmissions initiated during FAP (ϕ'). There will be no transmissions initiated within FAP, i.e., (M=0) when $\triangle_{b,1} > \phi'$, i.e

$$Pr\{M=0\} = Pr\{Y_1 > \phi'\} = 1 - F_{Y_1}(\phi'). \tag{6.8}$$

However, there will be at least one transmission initiated within FAP when $\triangle_{b,1} < \phi'$. There will be exactly m transmissions initiated within FAP when $\sum_{m'=1}^{m} \triangle_{b,m} \leq (\phi' - (m-1)\ell_p' - \ell_d - 1)$ and $\sum_{m'=1}^{m+1} \triangle_{b,m} > (\phi' - m\ell_p' - \ell_d - 1)$. Hence, the probability that there will be exactly m transmissions initiated within FAP ϕ' is

$$Pr\{M=m\} = F_{Y_m}(\phi' - (m-1)\ell_p' - \ell_d - 1) - F_{Y_{m+1}}(\phi' - m\ell_p' - \ell_d - 1). \tag{6.9}$$

Let $M_U(\phi)$ denote the maximum number of transmissions that can be initiated within a RAW slot (ϕ),where

$$M_U(\phi) = \left\lfloor \frac{\phi'}{(1+\ell_p')} \right\rfloor + I_{\{\phi' > [\ell_d + 1 + \left\lfloor \frac{\phi}{1+\ell_p'}(\ell_p'+1) \right\rfloor]\}} \ , \tag{6.10}$$

where $I_{\{x>0\}}$ is an indicator function which is equal to 1 when $x > 0$ is true and zero otherwise. The mean number of transmissions within one RAW slot of duration ϕ mini-slots where $|g_i|$ nodes are contending for the medium access is

$$E_M(\phi, |g_i|) = \sum_{m'=0}^{M_U(\phi')} m'\ Pr\{M = m'\}\ . \tag{6.11}$$

Then the saturation throughput of the i^{th} RAW slot denoted by$\mathit{\Gamma}_i$ is [46]

$$\mathit{\Gamma}_i(\phi, |g_i|) = \frac{\ell_p}{\phi}\ p_s(|g_i|)\ E_M(\phi, |g_i|). \tag{6.12}$$

In a conventional model where all K RAW slots are assumed to have equal number of nodes and the duration of each RAW slot (ϕ) is kept the same, then the mean number of transmissions for the entire RAW frame is $KE_M(\phi, |g_i|)$. Then, the throughput of the entire RAW frame in conventional model, denoted by $\mathit{\Gamma}_{\mathcal{F}c}$ [46]

$$\mathit{\Gamma}_{\mathcal{F}c}(K\phi, |g_i|) = \frac{K\ell_p}{\phi}\ p_s(|g_i|)\ E_{M_i}(\phi, |g_i|). \tag{6.13}$$

According to our proposed model, the RAW frame is divided into two sub-frames $\mathcal{F}_l$ and $\mathcal{F}_s$. Let ϕ_l be the duration of each of the $|\mathit{\Omega}_l|$ RAW slots in sub-frame $\mathcal{F}_l$, then the throughput in sub-frame $\mathcal{F}_l$, denoted by $\mathit{\Gamma}_{\mathcal{F}_l}$ is found as

$$\mathit{\Gamma}_{\mathcal{F}_l} = \frac{|\mathit{\Omega}_l|\ \ell_p}{\phi_l}\ p_s(\lfloor N/K \rfloor + 1)\ E_M(\phi_l, \lfloor N/K \rfloor + 1)$$

Similarly, the throughput in sub-frame $\mathcal{F}_s$ consisting of $|\mathit{\Omega}_l|$ RAW slots , each of duration ϕ_s, denoted by $\mathit{\Gamma}_{\mathcal{F}_s}$ is found as

$$\mathit{\Gamma}_{\mathcal{F}_s} = \frac{|\mathit{\Omega}_s|\ \ell_p}{\phi_s}\ p_s(\lfloor N/K \rfloor)\ E_M(\phi_s, \lfloor N/K \rfloor). \tag{6.14}$$

Then, the net throughput of the entire RAW frame in our proposed model denoted by $\mathit{\Gamma}_{\mathcal{F}p}$ is

$$\mathit{\Gamma}_{\mathcal{F}p} = \mathit{\Gamma}_{\mathcal{F}_l} + \mathit{\Gamma}_{\mathcal{F}_s} \tag{6.15}$$

where the values of ϕ_s and ϕ_l are selected by AP according to the size of the group.

Table 6.1: List of parameters used in simulations.

Parameter	Value
Mini-slot duration, τ	52 μsec
T_{SIFS}	160 μsec
T_{DIFS}	SIFS+2*Slot duration
MAC Header	272 bits
PHY Header	112 bits
ACK	112 bits
Initial contention window, W_0	16
CW_{max}	1024
Maximum Retry Limit RL	7

6.4 Performance Evaluation and Analysis

In this chapter, simulations are performed according to Physical layer (PHY) and MAC layer parameters of IEEE 802.11 and its amendment draft IEEE 802.11ah as listed in Table 6.1[47],[51],[60]. Due to the traffic model of the IoT network, we set the payload size to be quite a small value, i.e., 512 bits. The date rate is set at 1 Mbps. Therefore, the duration of one packet transmission including SIFS and the ACK transmission duration is kept at 1.1 msec.

In our simulations, we consider 64 groups and the size of each group is $|g_i| \in \{4, 8\}$ nodes and we set the RAW slot duration at 500 msec. To validate our model, we compare the cumulative distribution function (CDF) of the number of transmissions in simulation with analytical analysis in Section 5.3.

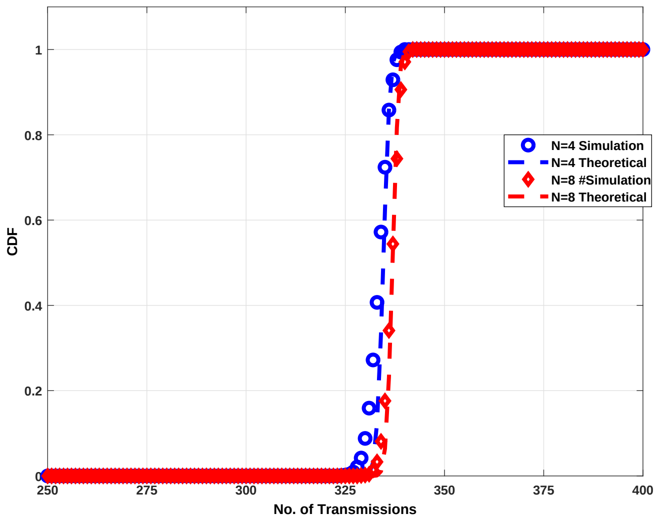

Figure 6.4: CDF of number of transmissions.

As Fig. 6.4 shows the simulated and analytical results match each other. Fig. 6.5 shows the saturation throughput for different numbers of nodes according to the normal DCF procedure following CSMA/CA when nodes are divided into K groups where $K \in \{1, 2, 4\}$. It is evident from the figure that the saturation throughput degrades with increase in the number of nodes per group. This is because contention among nodes causes collisions and failures and causes the throughput to reduce. Therefore, the throughput can be improved by keeping more and more groups, thus reducing the number of nodes per group. Fig. 6.6 shows that the throughput fluctuates with the duration of the RAW slot as some of the mini-slots are wasted. This is because all the nodes restrain from starting to transmit in the restricted access period of a RAW slot.

The figure 6.6 shows that the saturation throughput reaches its maximum possible value when the duration of the RAW mini-slot is very large. This is owing to the fact that the restricted access period becomes negligible as compared to FAP when the total RAW slot duration is very large. However, the duration of the RAW slot cannot be set at very large as it produces prolonged delays for other groups. It can be observed that for a given number of nodes, the throughput becomes approximately equal to the maximum possible value at some small duration of the RAW slot. Such values are called "optimal values of RAW slot duration" for each node.

Fig 6.7 shows the throughput improvement using the proposed model as compared to the conventional one when a network of nodes is considered from 120 to 320 nodes. According to our proposed model, AP chooses the optimal values of the RAW slot according to the size of group. This minimizes the restricted access interval and reduces the duration of those RAW slots which provide reduced throughput. In this way, the throughput in our proposed model is shown to have enhanced as compared to the conventional method.

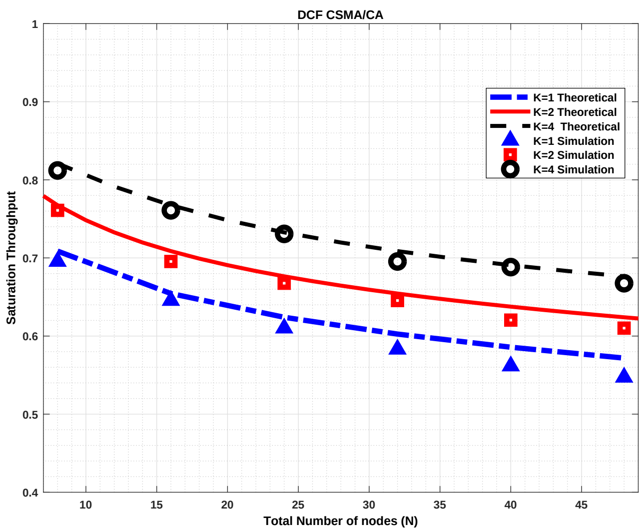

Figure 6.5: Saturation throughput for basic accesss CSMA/CA with different numbers of groups.

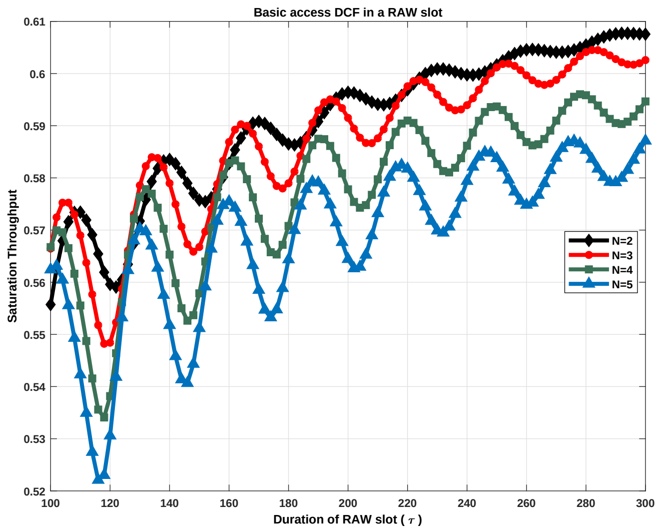

Figure 6.6: Saturation throughput for different values of RAW slot duration.

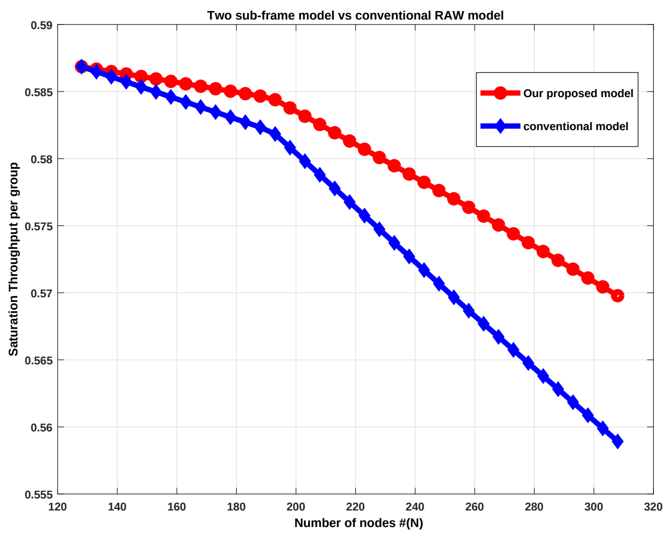

Figure 6.7: Comparison of the proposed model with the conventional model.

6.5 Conclusions

In this chapter, we have proposed a novel approach to enhance saturation throughput for a uniform grouping scheme in IEEE 802.11ah when transmission after the RAW slot boundary is not allowed. The analytical model is used to find the throughput of an entire RAW frame and the conventional RAW slot allocation scheme is compared with the two sub-frame model. The simulation results show that there is significant improvement in throughput using our proposed two sub-frame model as compared to the conventional one. This is because the duration of the RAW slot in a two sub-frame model is kept according to the size of group being assigned to each RAW slot. The proposed model can also be used in a decentralized grouping scheme by splitting the RAW frame into multi sub-frames RAW and allocating an optimal RAW slot duration for each sub-frame. Throughput can be enhanced when groups of equal size are allocated to the sub-frame having RAW slot duration.

6.6 Appendix

6.6.1 Appendix 6A

Let $\psi_1, \psi_2, ...\psi_{|g_i|}$ be independent and identically distributed (i.i.d) random variables, each following a geometric distribution with probability p_t and let $\triangle_b = min(\psi_1, \psi_2, ...\psi_{|g_i|})$. Then we have $\triangle_b \geq j$ only when $\{\psi_1 \geq j \;\&\; \psi_2 \geq j \& \;....\& \;\psi_{|g_i|} \geq j\}$. Since $Pr\{\psi_i \geq j\} = (1-p_t)^{j-1}$, it follows that

$$Pr\{\Delta_b \geq j\} = (1-p_t)^{|g_i|(j-1)}. \tag{6.16}$$

Therefore, we get

$$\begin{aligned} Pr\{\Delta_b = j\} &= Pr\{\Delta_b \geq j\} - Pr\{\Delta_b \geq j+1\}, \\ &= (1-p_t)^{|g_i|(j-1)} - (1-p_t)^{|g_i|j}, \end{aligned} \tag{6.17}$$

which can be written as

$$Pr\{\Delta_b = j\} = p'_t(1-p'_t)^{j-1}, \tag{6.18}$$

where $p'_t = 1-(1-p_t)^{|g_i|}$.

6.6.2 Appendix 6B

Let Y_m be a random variable such that $Y_m = \sum_{m'=1}^{m} \triangle_{b,m'}$ where $\triangle_{b,1}, \triangle_{b,1}, ...\triangle_{b,m}$ are independent and identically distributed geometric random variables with parameter p'_t such that

$$Pr\{\triangle_{b,i} = j\} = p'_t(1 - p_t)^{j-1}, \tag{6.19}$$

then Y_m is a negative binomial random variable with parameters m and p'_t given as

$$Pr\{Y_m = k\} = \begin{pmatrix} z' - 1 \\ m - 1 \end{pmatrix} p'_t(1 - p'_t)^{z'-m} \tag{6.20}$$

Since the minimum value of each random variable $\triangle_{b,1}, \triangle_{b,1}, ...\triangle_{b,m}$ is 1, then the minimum value of Y_m is m. Therefore, the CDF of Y_m is

$$F_{Y_m}(z) = \sum_{z'=m}^{z} \begin{pmatrix} z' - 1 \\ m - 1 \end{pmatrix} p'_t(1 - p'_t)^{z'-m} \tag{6.21}$$

Chapter 7

Conclusions and Future Work

In this chapter, a review of the main results of this thesis is presented and important conclusions are highlighted. Moreover, the possible extension and future directions of this work are also presented here.

7.1 Conclusions

In this thesis, we presented new aspects of the design of ultra-dense IoT deployments with minimal resources.and addressed the problem of minimization of various components of latency in the IoT deployments. By borrowing tools from stochastic theory, we developed comprehensive frameworks for various solutions to reduce latency and analyzed the performance of an IoT network for these solutions.

Low latency solutions have gained significant importance for a wide range of applications of daily usage. There are various scenarios where latency becomes critical in IoT, e.g., delay sensitive applications such as remote health-care and medical intervention, assisted driving and transport services, entertainment, content delivery and gaming, and automation of industry. Moreover, the development of low-latency solutions has become quite critical in the design space of IoT where most of the devices are resource-constrained in terms of energy availability, storage capacity, processing and transmission power, etc. In short, latency minimization has become necessary for a widespread adoption of IoT in our society.

The research work that is presented in this thesis can be summarized in the following two main subsections.

7.1.1 Minimizing Propagation and Processing Latency

This research topic is addressed in the thesis in two chapters, i.e., chapter 3 and chapter 4. In chapter 3, we employed tools from queuing theory to estimate the traffic load and waiting times for the newly arriving data packets in the buffer of the devices in IoT, fog and cloud layer. For a comprehensive framework, we considered a realistic scenario where the devices in boththe IoT and Fog layer have a limited storage capacity, computational power and transmission capability and developed the queuing system. We developed a strategy to offload tasks from IoT to devices either in the fog layer or the cloud servers. To determine the capabilities of fog nodes, we developed a strategy to use service discovery protocol before offloading tasks in FL. We showed that both propagation and processing latency is reduced when tasks are offloaded to the nodes which are located close to the end user and have sufficient resources to process data. To achieve this end, we used service discovery protocols to estimate the capability of devices before actually offloading the task data to them.

In chapter 4, we presented a framework for selection of new fog node among many candidates in the vicinity of end users that could enhance the processing capability of the fog network. We used tools from order statistics to rank the candidates according to their processing capabilities and developed a strategy to find the selection criteria of the candidates when they appear for inspection during the selection stage. We presented a strategy when a recruiter fog node (already present in the network) starts the selection process and acquires a candidate after comparison of its processing capability with the ones already examined during the process. We showed that the overall capability of the fog network is improved by acquisition of new fog nodes according to our proposed selection strategy, and thus the propagation and processing latency of in the IFC paradigm is reduced significantly.

7.1.2 Minimizing Network Latency

This aspect of latency reduction is addressed in the thesis in two chapters, i.e., chapter 5 and chapter 6. In chapter 5, we organized the nodes into groups according to the uniform grouping scheme and assigned those groups to different RAW slots. In this chapter, we employed a Markov model, determined the behavior of nodes when they use the DCF backoff procedure to transmit data packets, and evaluated the throughput of the network when the duration of the RAW slot is chosen according to the size of the group assigned to it. By use of the RAW mechanism where the nodes are divided into groups, the collision among the packets is reduced significantly and hence, the delays incurred during the re-transmission of data packets (because of long back-off waiting duration) reduce to a great extent.

In chapter 6, we addressed the reduction of network latency by employing RAW and reducing contention among the group of nodes. We developed frameworks for the scenarios when nodes are in saturation, unsaturated and generic modes where the node may get a data packet from the upper layer and evaluated the duration of RAW slot for each mode. For this purpose, we employed the mean value analysis to model the attempt process of nodes in the RAW slot, and used Markov chain models to estimate the mean of the duration to transmit a given number of data packets.

By estimation of the duration of RAW slots, the beacon interval is evaluated in IEEE 802.11ah for different numbers of groups for different access schemes such as basic access and RTS/CTS access mechanisms. It is found that the time wasted during collision of packets is much greater in the basic access scheme as compared to the RTS/CTS scheme. For IoT, the most common scenario is a node having either a single data packet or a finite number of data packets, and transmits short data packets. Hence, the use of the RTS/CTS access mechanism yields better performance in IEEE 802.11ah for a wide range of IoT applications.

7.2 Future Work

In the previous chapters, we developed some possible solutions for the reduction of various components of latency such as propagation, processing, networking delay, etc., that may become critical for the design of future IoT applications. As a result, we identified many open issues as possible future directions for research.

Joint Optimization of Task Offloading and Service Discovery Protocol

In chapter 3, for the sake of simplifying the analysis, we assumed that the tasks are offloaded from IL to the FL after discovering the capabilities of the fog nodes at a constant frequency. However, the SDP may result in high computational complexity and huge reporting overhead as all fog nodes report their resources to the centralized entity, especially when the protocol is executed too frequently. On the other hand, execution of SDP after a long period of time may result in overloading of the fog nodes and may lead to prolonged delays in IoT deployments with intense traffic data. Therefore, an optimization of the execution is one of the possible avenues for future research. Since fog nodes typically have finite computational and processing resources, optimization of resource allocation in the design of diverse nature of IoT applications with heterogeneous user demands poses another challenge in task offloading. Therefore, a comprehensive model to jointly optimize resource allocation and task offloading along with a light-weight service discovery protocol with an optimized execution frequency can give a better performance insight and prove to be more effective for reduction of latency. Moreover, the task data generated by IoT devices proposed in our system model is assumed to have a fixed payload size which may be extended in future to incorporate heterogeneous traffic scenarios with variable payload size and data rate.

Selection strategy for fog nodes

In chapter 4, we developed a strategy for selection of a new fog node dynamically when FaaS is utilized by assuming that the recruiter node can examine N candidates during the selection process. However, in reality, there may be a random number of candidate fog nodes which depends on different types of environments. The development of an analytical model for selection of a fog node without any prior knowledge of the total number of candidates may be a possible extension of research. Moreover, in our proposed selection strategy in chapter 4, the criteria of selection changes at two different points, i.e., select a best candidate encountered between Γ_1 and Γ_2, and second-best after Γ_2 candidates are examined by the recruiter node. The selection strategy can be extended to change criteria of selection at k different points, i.e., select a best candidate among the nodes encountered between Γ_1 to

$\Gamma_{2,}$ select second-best examined between Γ_2 to Γ_3, and so on, such that k^{th} best is selected after examination of Γ_k candidate nodes, and then finding the optimum points of $\Gamma_1, \Gamma_2, ..., \Gamma_k$.

Estimation of parameters of IEEE 802.11ah

In chapter 5 and 6, we developed an analytical model of the RAW and estimated the mean duration of time to transmit a given number of data packets in each of the RAW slots considering that each node transmits a data packet of fixed length. A further extension to this work is to develop an analytical model for heterogeneous traffic scenarios with variable payload size and data rate. Moreover, the optimization of parameters of the IEEE 802.11ah standard such as finding the number of RAW slots in a beacon interval, and determination of number of nodes in each group being allocated to the RAW slot to jointly optimize different performance metrics such as latency, throughput and energy efficiency in IoT deployments, opens up new avenues of future research.

References

[1] Statistica, “Internet of Things - number of connected devices worldwide 2015-2025 [Internet],” https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/, [Online; accessed 04-Nov-2019].

[2] E. Dresden and N. Vodafone, “A choice of future m2m access technologies for mobile network operators,” *Cellular IoT White Paper*, 2014.

[3] Forbes, “How Many Things Are Currently Connected To The "Internet of Things" (IoT)?” https://www.forbes.com/sites/quora/2013/01/07/how-many-things-are-currently-connected-to-the-internet-of-things-iot/, [Online; accessed 05-Nov-2019].

[4] S. C. Association, “EE: Average uk smart home will have 50 connected devices by 2023,” https://smartcitiesassociation.org/index.php/media-corner/news/169-ee-average-uk-smart-home-will-have-50-connected-devices-by-2023/, [Online; accessed 05-Nov-2019].

[5] A. Yastrebova, R. Kirichek, Y. Koucheryavy, A. Borodin, and A. Koucheryavy, “Future networks 2030: Architecture & requirements,” in *10th IEEE International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, 2018, pp. 1–8.

[6] J. P. Shim, M. Avital, A. Dennis, O. Sheng, M. Rossi, C. Sorensen, and A. French, “Internet of things: Opportunities and challenges to business, society, and IS research,” *38th International Conference on Information Systems (ICIS), South Korea*, 2017.

[7] E. Nygren, R. K. Sitaraman, and J. Sun, “The akamai network: a platform

for high-performance internet applications," *ACM SIGOPS Operating Systems Review*, vol. 44, no. 3, pp. 2–19, 2010.

[8] B. Challacombe, L. Kavoussi, and P. Dasgupta, "Trans-oceanic telerobotic surgery," *BJU international*, vol. 92, no. 7, pp. 678–680, 2003.

[9] R. J. Anderson and M. W. Spong, "Bilateral control of teleoperators with time delay," *IEEE Transactions on Automatic control*, vol. 34, no. 5, pp. 494–501, 1989.

[10] I. Parvez, A. Rahmati, I. Guvenc, A. I. Sarwat, and H. Dai, "A survey on low latency towards 5G: RAN, core network and caching solutions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3098–3130, 2018.

[11] M. A. Lema, A. Laya, T. Mahmoodi, M. Cuevas, J. Sachs, J. Markendahl, and M. Dohler, "Business case and technology analysis for 5G low latency applications," *IEEE Access*, vol. 5, pp. 5917–5935, 2017.

[12] A. Varghese and D. Tandur, "Wireless requirements and challenges in industry 4.0," in *IEEE International Conference on Contemporary Computing and Informatics (IC3I)*, 2014, pp. 634–638.

[13] F. E. Froehlich and A. Kent, *The Froehlich/Kent Encyclopedia of Telecommunications: Volume 17-Television Technology.* CRC Press, 1998, vol. 17.

[14] R. Ramaswamy, N. Weng, and T. Wolf, "Characterizing network processing delay," in *IEEE Global Telecommunications Conference, 2004. GLOBECOM'04.*, vol. 3, 2004, pp. 1629–1634.

[15] S. M. Babu, A. J. Lakshmi, and B. T. Rao, "A study on cloud based Internet of Things: CloudIoT," in *IEEE Global conference on communication technologies (GCCT)*, 2015, pp. 60–65.

[16] A. Botta, W. De Donato, V. Persico, and A. Pescapé, "Integration of cloud

computing and Internet of Things: a survey," *Future generation computer systems*, vol. 56, pp. 684–700, 2016.

[17] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE communications surveys & tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[18] L. Hou, S. Zhao, X. Xiong, K. Zheng, P. Chatzimisios, M. S. Hossain, and W. Xiang, "Internet of Things cloud: Architecture and implementation," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 32–39, 2016.

[19] M. Aazam and E.-N. Huh, "Fog computing and smart gateway based communication for cloud of things," in *IEEE International Conference on Future Internet of Things and Cloud*, 2014, pp. 464–470.

[20] M. Mukherjee, L. Shu, and D. Wang, "Survey of fog computing: Fundamental, network applications, and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 1826–1857, 2018.

[21] M. Aazam and E.-N. Huh, "Fog computing micro datacenter based dynamic resource estimation and pricing model for iot," in *29th IEEE International Conference on Advanced Information Networking and Applications*, 2015, pp. 687–694.

[22] M. Muntjir, M. Rahul, and H. A. Alhumyani, "An analysis of Internet of Things (IoT): Novel architectures, modern applications, security aspects and future scope with latest case studies," *Int. J. Eng. Res. Technol*, vol. 6, pp. 422–447, 2017.

[23] Y. Ai, M. Peng, and K. Zhang, "Edge computing technologies for internet of

things: a primer," *Digital Communications and Networks*, vol. 4, no. 2, pp. 77–86, 2018.

[24] "The Internet of Things: Extend the cloud to where the things are," *Cisco White Paper*, 2015.

[25] Y. Liu, J. E. Fieldsend, and G. Min, "A framework of fog computing: Architecture, challenges, and optimization," *IEEE Access*, vol. 5, pp. 25 445–25 454, 2017.

[26] A. V. Dastjerdi, H. Gupta, R. N. Calheiros, S. K. Ghosh, and R. Buyya, "Fog computing: Principles, architectures, and applications," in *Internet of Things*. Elsevier, 2016, pp. 61–75.

[27] O. Skarlat, S. Schulte, M. Borkowski, and P. Leitner, "Resource provisioning for iot services in the fog," in *9th IEEE International conference on service-oriented computing and applications (SOCA)*, 2016, pp. 32–39.

[28] A. Yousefpour, G. Ishigaki, and J. P. Jue, "Fog computing: Towards minimizing delay in the Internet of Things," in *IEEE International Conference on Edge Computing (EDGE)*, 2017, pp. 17–24.

[29] V. Gazis, A. Leonardi, K. Mathioudakis, K. Sasloglou, P. Kikiras, and R. Sudhaakar, "Components of fog computing in an Industrial Internet of Things context," in *12th Annual IEEE International Conference on Sensing, Communication, and Networking-Workshops (SECON Workshops)*, 2015, pp. 1–6.

[30] C. Vallati, A. Virdis, E. Mingozzi, and G. Stea, "Exploiting LTE D2D communications in M2M fog platforms: Deployment and practical issues," in *IEEE 2nd World Forum on Internet of Things (WF-IoT)*, 2015, pp. 585–590.

[31] T. Nishio, R. Shinkuma, T. Takahashi, and N. B. Mandayam, "Service-oriented heterogeneous resource sharing for optimizing service latency in mo-

bile cloud," in *Proceedings of the first ACM International workshop on Mobile cloud computing & networking*, 2013, pp. 19–26.

[32] R. Beraldi, A. Mtibaa, and H. Alnuweiri, "Cooperative load balancing scheme for edge computing resources," in *2nd IEEE International Conference on Fog and Mobile Edge Computing (FMEC)*, 2017, pp. 94–100.

[33] C. Fricker, F. Guillemin, P. Robert, and G. Thompson, "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)*, vol. 1, no. 4, p. 16, 2016.

[34] A. Yousefpour, G. Ishigaki, R. Gour, and J. P. Jue, "On reducing iot service delay via fog offloading," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 998–1010, 2018.

[35] G. Lee, W. Saad, and M. Bennis, "An online secretary framework for fog network formation with minimal latency," in *IEEE International Conference on Communications (ICC)*, 2017, pp. 1–6.

[36] R. Kleinberg, "A multiple-choice secretary algorithm with applications to online auctions," in *Proceedings of 16th annual ACM-SIAM symposium on Discrete algorithms*, 2005, pp. 630–631.

[37] G. Lee, W. Saad, and M. Bennis, "An online optimization framework for distributed fog network formation with minimal latency," *arXiv preprint arXiv:1710.05239*, 2017.

[38] T. S. Ferguson *et al.*, "Who solved the secretary problem?„" *Statistical science*, vol. 4, no. 3, pp. 282–289, 1989.

[39] P. Blanc, J. P. Borthagaray, D. Kohen, and M. Mereb, "Secretary problem with quality-based payoff," *arXiv preprint arXiv:1605.06478*, 2016.

[40] J. P. Gilbert and F. Mosteller, "Recognizing the maximum of a sequence," in *Selected Papers of Frederick Mosteller*, 2006, pp. 355–398.

[41] C.-F. Liu, M. Bennis, and H. V. Poor, "Latency and reliability-aware task offloading and resource allocation for mobile edge computing," in *IEEE Globecom Workshops (GC Wkshps)*, 2017, pp. 1–7.

[42] H. A. Alameddine, S. Sharafeddine, S. Sebbah, S. Ayoubi, and C. Assi, "Dynamic task offloading and scheduling for low-latency iot services in multi-access edge computing," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 3, pp. 668–682, 2019.

[43] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, 2015.

[44] C. Zhang, H. Zhao, and S. Deng, "A density-based offloading strategy for iot devices in edge computing systems," *IEEE Access*, vol. 6, pp. 73 520–73 530, 2018.

[45] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 4, pp. 974–983, 2014.

[46] L. Zheng, M. Ni, L. Cai, J. Pan, C. Ghosh, and K. Doppler, "Performance analysis of group-synchronized DCF for dense IEEE 802.11 networks," *IEEE Transactions on Wireless Communications*, vol. 13, no. 11, pp. 6180–6192, 2014.

[47] L. Zheng, L. Cai, J. Pan, and M. Ni, "Performance analysis of grouping strategy for dense IEEE 802.11 networks," in *IEEE Global Communications (GLOBECOM) Conference*, 2013, pp. 219–224.

[48] Y. Tay and K. C. Chua, “A capacity analysis for the IEEE 802.11 MAC protocol,” *Wireless networks*, vol. 7, no. 2, pp. 159–171, 2001.

[49] E. Khorov, A. Krotov, and A. Lyakhov, “Modelling machine type communication in IEEE 802.11 ah networks,” in *IEEE International Conference on Communication Workshop (ICCW)*, 2015, pp. 1149–1154.

[50] E. Khorov, A. Krotov, A. Lyakhov, R. Yusupov, M. Condoluci, M. Dohler, and I. Akyildiz, “Enabling the Internet of Things with Wi-Fi HaLow – performance evaluation of the restricted access window,” *IEEE Access*, vol. 7, pp. 127 402–127 415, 2019.

[51] L. Zheng, M. Ni, L. Cai, J. Pan, C. Ghosh, and K. Doppler, “Performance analysis of group-synchronized dcf for dense IEEE 802.11 networks,” *IEEE Transactions on Wireless Communications*, vol. 13, no. 11, pp. 6180–6192, 2014.

[52] O. Raeesi, J. Pirskanen, A. Hazmi, T. Levanen, and M. Valkama, “Performance evaluation of IEEE 802.11 ah and its restricted access window mechanism,” in *IEEE International Conference on Communications Workshops (ICC)*, 2014, pp. 460–466.

[53] M. Z. Ali, J. Mišić, and V. B. Mišić, “Efficiency of restricted access window scheme of IEEE 802.11 ah under non-ideal channel condition,” in *IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, 2018, pp. 251–256.

[54] L. Tian, J. Famaey, and S. Latré, “Evaluation of the IEEE 802.11 ah restricted access window mechanism for dense iot networks,” in *17th IEEE International*

symposium on a world of wireless, mobile and multimedia networks (WoWMoM), 2016, pp. 1–9.

[55] A. Šljivo, D. Kerkhove, L. Tian, J. Famaey, A. Munteanu, I. Moerman, J. Hoebeke, and E. De Poorter, "Performance evaluation of IEEE 802.11 ah networks with high-throughput bidirectional traffic," *Sensors*, vol. 18, no. 2, p. 325, 2018.

[56] A. Hazmi, B. Badihi, A. Larmo, J. Torsner, M. Valkama *et al.*, "Performance analysis of IoT-enabling IEEE 802.11 ah technology and its raw mechanism with non-cross slot boundary holding schemes," in *IEEE 16th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2015, pp. 1–6.

[57] M. Mahesh and V. Harigovindan, "Restricted access window-based novel service differentiation scheme for group-synchronized DCF," *IEEE Communications Letters*, vol. 23, no. 5, pp. 900–903, 2019.

[58] O. Raeesi, J. Pirskanen, A. Hazmi, J. Talvitie, and M. Valkama, "Performance enhancement and evaluation of IEEE 802.11ah multi-access point network using restricted access window mechanism," in *IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2014, pp. 287–293.

[59] S. M. Soares and M. M. Carvalho, "Throughput analytical modeling of IEEE 802.11 ah wireless networks," in *16th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, 2019, pp. 1–4.

[60] N. Nawaz, M. Hafeez, S. Zaidi, D. McLernon, and M. Ghogho, "Throughput enhancement of restricted access window for uniform grouping scheme in IEEE 802.11 ah," in *IEEE International Conference on Communications (ICC)*, 2017, pp. 1–7.

[61] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE Journal on selected areas in communications*, vol. 18, no. 3, pp. 535–547, 2000.

[62] C. Hu, H. Kim, J. C. Hou, D. Chi *et al.*, "Provisioning quality controlled medium access in ultrawideband-operated WPANs," in *INFOCOM 2006: 25th IEEE International Conference on Computer Communications*, 2006.

[63] J. W. Robinson and T. S. Randhawa, "Saturation throughput analysis of IEEE 802.11e enhanced distributed coordination function," *IEEE Journal on selected areas in communications*, vol. 22, no. 5, pp. 917–928, 2004.

[64] F. Daneshgaran, M. Laddomada, F. Mesiti, and M. Mondin, "Unsaturated throughput analysis of IEEE 802.11 in presence of non ideal transmission channel and capture effects," *IEEE Transactions on Wireless Communications*, vol. 7, no. 4, pp. 1276–1286, 2008.

[65] Y. S. Liaw, A. Dadej, and A. Jayasuriya, "Performance analysis of IEEE 802.11 DCF under limited load," in *IEEE Asia-Pacific Conference on Communications*, 2005, pp. 759–763.

[66] P. Luong, T. M. Nguyen, and L. B. Le, "Throughput analysis for coexisting IEEE 802.15. 4 and 802.11 networks under unsaturated traffic," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, p. 127, 2016.

[67] C. Y. Jung, H. Y. Hwang, D. K. Sung, and G. U. Hwang, "Enhanced markov chain model and throughput analysis of the slotted csma/ca for IEEE 802.15. 4 under unsaturated traffic conditions," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 1, pp. 473–478, 2008.

[68] L. Tian, E. Khorov, S. Latré, and J. Famaey, "Real-time station grouping

under dynamic traffic for IEEE 802.11 ah," *Sensors*, vol. 17, no. 7, p. 1559, 2017.

[69] R. Khanduri and S. Rattan, "Performance comparison analysis between IEEE 802.11 a/b/g/n standards," *International Journal of Computer Applications*, vol. 78, no. 1, pp. 13–20, 2013.

[70] G. R. Hiertz, D. Denteneer, L. Stibor, Y. Zang, X. P. Costa, and B. Walke, "The IEEE 802.11 universe," *IEEE Communications Magazine*, vol. 48, no. 1, pp. 62–70, 2010.

[71] R. Van Nee, "Breaking the gigabit-per-second barrier with 802.11 ac," *IEEE Wireless Communications*, vol. 18, no. 2, pp. 4–4, 2011.

[72] E. Perahia, C. Cordeiro, M. Park, and L. L. Yang, "IEEE 802.11 ad: Defining the next generation multi-gbps wi-fi," in *2010 7th IEEE consumer communications and networking conference*, 2010, pp. 1–5.

[73] I. 802.11ah Task Group, "TGah Functional Requirements and Evalu- ation Methodology Rev. 5," http://Available:http://www.ieee802.org/11/Reports/tgah_update.htm/, [Online; accessed 05-Nov-2018].

[74] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A comparative study of LPWAN technologies for large-scale IoT deployment," *ICT express*, vol. 5, no. 1, pp. 1–7, 2019.

[75] A. Hazmi, J. Rinne, and M. Valkama, "Feasibility study of IEEE 802.11 ah radio technology for IoT and M2M use cases," in *IEEE Globecom Workshops (GC Wkshps)*, 2012, pp. 1687–1692.

[76] R. D. Vegt, "Potential compromise for 802.11 ah use case document," *IEEE 802.11-11/0457r0*, 2011.

[77] A. Bel, T. Adame, B. Bellalta, J. Barcelo, J. Gonzalez, and M. Oliver, "CAS-

based channel access protocol for IEEE 802.11 ah WLANs," in *20th European Wireless Conference*, 2014, pp. 1–6.

[78] W. Sun, M. Choi, and S. Choi, "IEEE 802.11 ah: A long range 802.11 WLAN at sub 1 GHz," *Journal of ICT Standardization*, vol. 1, no. 1, pp. 83–108, 2013.

[79] N. Daneshfar, "Performance enhancement mechanism of IEEE 802.11 ah machine communication system," Master's thesis, 2015.

[80] Y. Zhou, H. Wang, S. Zheng, and Z. Z. Lei, "Advances in IEEE 802.11 ah standardization for machine-type communications in sub-1ghz WLAN," in *IEEE International Conference on Communications (ICC) Workshops*, 2013, pp. 1269–1273.

[81] J. S. Reigadas, A. Martinez-Fernandez, J. Ramos-Lopez, and J. Seoane-Pascual, "Modeling and optimizing IEEE 802.11 DCF for long-distance links," *IEEE Transactions on Mobile Computing*, vol. 9, no. 6, pp. 881–896, 2010.

[82] G. Bianchi, L. Fratta, and M. Oliveri, "Performance evaluation and enhancement of the CSMA/CA MAC protocol for 802.11 wireless LANs," in *Seventh IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, vol. 2, 1996, pp. 392–396.

[83] Y. Xiao, "An analysis for differentiated services in IEEE 802.11 and IEEE 802.11 e wireless LANs," in *24th IEEE International Conference on Distributed Computing Systems*, 2004, pp. 32–39.

[84] M. N. Sadiku, S. M. Musa, and O. D. Momoh, "Cloud computing: opportunities and challenges," *IEEE potentials*, vol. 33, no. 1, pp. 34–36, 2014.

[85] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in

the Internet of Things," in *Proceedings of the 1st edition of the MCC workshop on Mobile cloud computing*, 2012, pp. 13–16.

[86] M. Firdhous, O. Ghazali, and S. Hassan, "Fog computing: Will it be the future of cloud computing?" The Third International Conference on Informatics & Applications (ICIA2014), 2014.

[87] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854–864, 2016.

[88] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, "Fog computing: A platform for Internet of Things and analytics," in *Big data and Internet of Things: A roadmap for smart environments*, 2014, pp. 169–186.

[89] V. Sundarapandian, *Probability, statistics and queuing theory.* PHI Learning Pvt. Ltd., 2009.

[90] M. Zukerman, "Introduction to queueing theory and stochastic teletraffic models," *arXiv preprint arXiv:1307.2968*, 2013.

[91] F. Pisani, V. Martins do Rosario, and E. Borin, "Fog vs. cloud computing: should I stay or should I go?" *Future Internet*, vol. 11, no. 2, p. 34, 2019.

[92] J. P. Royston, "Algorithm as 177: Expected normal order statistics (exact and approximate)," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 31, no. 2, pp. 161–165, 1982. [Online]. Available: http://www.jstor.org/stable/2347982

[93] L. Zheng and L. Cai, "A distributed demand response control strategy using Lyapunov optimization," *IEEE Transactions on Smart Grid*, vol. 5, no. 4, pp. 2075–2083, 2014.

[94] K. Zhou, J. Pan, and L. Cai, "Optimal combined heat and power system

scheduling in smart grid," in *Proceedings IEEE INFOCOM*, 2014, pp. 2831–2839.

[95] A. Sehgal, V. Perelman, S. Kuryla, and J. Schonwalder, "Management of resource constrained devices in the Internet of Things," *IEEE Communications Magazine*, vol. 50, no. 12, 2012.

[96] A. Kumar, E. Altman, D. Miorandi, and M. Goyal, "New insights from a fixed-point analysis of single cell IEEE 802.11 WLANs," *IEEE/ACM Transactions on Networking (TON)*, vol. 15, no. 3, pp. 588–601, 2007.

[97] I. W. Group *et al.*, "IEEE Standard for Information Technology–Telecommunications and information exchange between systems–Local and metropolitan area networks–Specific requirements–Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 6: Wireless Access in Vehicular Environments," *IEEE Std*, vol. 802, no. 11, 2010.

[98] J. Berg, "The IEEE 802.11 standardization its history, specifications, implementations, and future," Technical Report GMU-TCOM-TR-8, George Mason University, Fairfax, VA, USA, Tech. Rep., 2011.

[99] T.-C. Hou and T.-J. Tsai, "A access-based clustering protocol for multihop wireless ad hoc networks," *IEEE journal on selected areas in communications*, vol. 19, no. 7, pp. 1201–1210, 2001.

[100] C. R. Lin and M. Gerla, "Adaptive clustering for mobile wireless networks," *IEEE Journal on Selected areas in Communications*, vol. 15, no. 7, pp. 1265–1275, 1997.