CO-EXISTENCE OF LTE-UNLICENSED AND WI-FI: OPTIMIZATION, GAME THEORY AND Q-LEARNING APPROACHES

BTECH PROJECT REPORT

submitted in partial fulfilment of the requirements

for the degree of

BACHELOR OF TECHNOLOGY

in

ELECTRICAL ENGINEERING

by

CHAITANYA AMBALLA EE15B001

Supervisor(s)

Dr. K P Naveen

भारतीय प्रौद्योगिकी संस्थान तिरुपति

TIRUPATI

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY TIRUPATI

MAY 2019

DECLARATION

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/-data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Place: Tirupati
Date: 19-05-2019

Signature
Chaitanya Amballa
EE15B001

BONA FIDE CERTIFICATE

This is to certify that the thesis titled **CO-EXISTENCE OF LTE-UNLICENSED AND WI-FI: OPTIMIZATION, GAME THEORY AND Q-LEARNING APPROACHES** , submitted by **Chaitanya Amballa**, to the Indian Institute of Technology, Tirupati, for the award of the degree of **Bachelor of Technology**, is a bona fide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Place: Tirupati
Date: 19-05-2019

Dr K P Naveen
Guide
Assistant Professor
Department of Electrical Engineering
IIT Tirupati - 517506

ACKNOWLEDGEMENTS

First of all, I would wish to thank my parents for their continuous support and encouragement throughout my project.

I wholeheartedly respect and thank my guide **Dr. K P Naveeen** for providing me an opportunity to work with him, and giving me all the support and guidance even in his busy schedules by managing some time for me. I am also grateful to him for lending his laptop to continue my research when my laptop was completely down. I am extremely privileged to have got a guidance like this all along till the completion of my project.

I also owe my deep gratitude to **Dr. Rama Krishna Gorthi**, who took a keen interest on my project, from the beginning till the end by providing all the necessary motivation along with some information regarding various deep learning frameworks required for developing a good model in my project.

I would like to offer my special thanks to Prasanna who has been with me all the time and provided me moral and emotional support. I am also grateful to my other family members and friends who have supported me along the way.

And finally, last but by no means least, also to everyone in IIT Tirupati. It was great sharing the most important and beautiful part of my life with you all during the last four years.

Thank you for all your encouragement!

ABSTRACT

KEYWORDS: LTE-U ; Wi-Fi; Co-existence; Optimization; Game Theory; ε(epsilon) Nash equilibrium ; Q-Learning.

With the increase in the number of mobile devices and the requirements for high data rates and throughput, allocating the resources in a non-efficient manner leads to so called "Spectrum Scarcity problem", where the limited available spectrum is not enough to satisfy the demand. Hence, to overcome this issue, researchers have proposed the LTE-Unlicensed, a new way of accessing the spectrum in which LTE cellular operators are allowed to use the unlicensed bands, for instance the 5GHz ISM band. However, intervention of LTE in the unlicensed bands may severely degrade the performance of the networks that are already deployed, mainly Wi-Fi. To have a fair coexistence between LTE and Wi-Fi, in this paper, we propose a model which ensures that Wi-Fi network's performance is not severely degraded when LTE is introduced and allowed to use the unlicensed spectrum. We first propose a system model and then formulate an optimization problem; we seek to derive optimal coexistence policies. We also introduce a game theoretic view for the same and characterize the solution in terms of Nash equilibrium policies. In addition to the traditional way of forming a system model and solving the optimization problem, we also introduce a Q-learning problem, a model free reinforcement learning approach. We first propose a Markov Decision based model where the LTE and WiFi stations are modeled as M/M/1 queues. We then train the model using Q-learning. Finally we demonstrate the efficiency of the Q-learned policy over a few heuristic algorithms.

Though out the thesis we discuss various methods that can be applied to enhance the coexistence between LTE-Unlicensed and Wi-Fi; the performance for each of the model is observed. The thesis comprises 6 chapters and are organised as follows. In Chapter 1, we introduced about the LTE-Unlicensed and discussed in detail the aim and scope of the problem. In chapter 2 we briefly talked about various works in the literature that are proposed till now for the fair coexistence between LTE and Wi-Fi. We also

mentioned about how our model differs from the existing methods and the advantages. Then we gave a brief overview on the works done in Q-learning on the coexistence problem. In chapter 3, we formulated the system model, optimization framework, game theoretic framework and solved for the respective solutions. Then, in chapter 4, we gave an introduction to the reinforcement learning and Q-learning, developed a simple system model and made the network train by updating the Q matrix. In chapter 5, we discussed about the simulation set up, and the results obtained along with some important observations. We conclude the paper with chapter 6 by giving a brief summary along with some future works that can be done to enhance the model.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

3GPP	Third Generation Partnership Program
ABSF	Almost Blank Subframe
BS	Base Station
IITTP	Indian Institute of Technology, Tirupati
LBT	Listen Before Talk
LTE	Long Term Evolution
MDP	Markov Decision Process
POA	Price of Anarchy
LTE-U	Long Term Evolution Unlicensed
Wi-Fi	Wireless Fidelity
Wi-Fi AP	Wi-Fi Access Point

NOTATION

α	learning rate
β	Air time matrix
γ	Discount factor
ε	exploration constant

CHAPTER 1

INTRODUCTION

1.1 Introduction to LTE Unlicensed

To cope with the increasing scarcity of spectrum resources, and demand for high data rates and throughput, researchers have been working to extend LTE cellular systems to unlicensed bands, leading to so-called LTE unlicensed (LTE-U). In other words, LTE in unlicensed spectrum has been proposed to allow cellular network operators to offload some of their data traffic by accessing the unlicensed 5 GHz frequency band. It is not developed to replace existing LTE network-based connections but developed to coexist with LTE and Wi-Fi networks. The main purpose is to offload the traffic from existing LTE networks as well as traffic from existing Wi-Fi hotspots. LTE-U would allow cellphone carriers to boost coverage in their cellular networks, by using the unlicensed 5 GHz band already populated by Wi-Fi devices. The specifications of LTE-U are outlined in 3GPP release-10/11/12. However, this extension is by no means straightforward, primarily because the radio resource management schemes used by LTE and by systems already deployed in unlicensed bands are incompatible. It is also known that bringing up LTE in the unlicensed bands disrupts the throughput of the networks that are already deployed in the higher frequency bands. To avoid interference, there is a need for fair coexistence between LTE-U and Wi-Fi with a better spectrum efficiency. Once such method to overcome the interference is dividing the time slots in a frame into two parts for each of the LTE and Wi-Fi based on the load requirement. The figure 1.1 shown below represents an almost blank sub frame in which the **D** represents Data, i.e. LTE can transmit the data where as **B** represents Blank, i.e. Wi-Fi has the transmission opportunity in this time. In other words, we give some amount of time in a frame to LTE and it will use this time for its transmissions, and the remaining time will be used by the WI-Fi AP for its transmissions following the contention based protocols. But dividing the time slots has to be done in such a way that both LTE and Wi-Fi coexist along with giving a fair amount of spectrum time allocation to Wi-Fi. So, the optimal number of slots for LTE transmission and the time for which Wi-Fi can be transmitted in a given time should be found based on the data rates requested by the users using LTE and Wi-Fi access points. With the advancements in artificial intelligence and deep learning techniques, it is possible to learn the optimal number of LTE transmission slots in a given period over time. So, we introduced a model free learning known as Q-learning to the problem and learned the optimal air time for LTE and Wi-Fi. If this could be achievable, both LTE-U and Wi-Fi can coexist with better deliverable speeds and data rates.

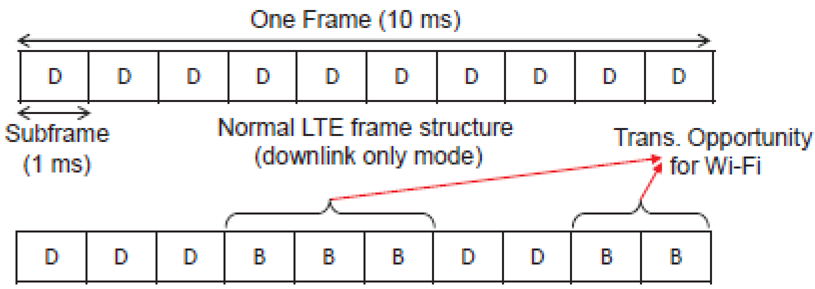

Figure 1.1: Almost blank sub-frame

1.2 Aim of the Project

Since LTE-U is being used in the unlicensed spectrum, there is a requirement to provide fair coexistence with other technologies working in the unlicensed spectrum. Since WiFi too operates at 5GHz, use of LTE unlicensed at 5GHz will cause interference which one would not likely to have. So, there is a need to fairly coexist both the LTE-U and Wi-Fi with a better spectrum efficiency. Without a properly defined fairness criterion for spectrum sharing, Wi-Fi networks may get completely stalled if the co-located LTE networks selfishly offload too much traffic to the unlicensed bands. So, using the concept of almost blank sub frames, the aim is to determine the number of slots in which LTE-U transmission can occur and the time slots for which Wi-Fi can be accessed. Also, to verify any model free learning algorithm for example **Q-learning** can be applied since the transition probabilities are not known in a partially observable Markov chain so that no theoretical model is required to decide the optimal time slots.

1.3 Sub problems to be addressed

1. In order to determine the number of optimal slots for LTE-U, there is a need to detect in how much time you need to perform this detection, i.e what is the time frame in which the decision has to be taken. Is it fixed or can it vary?

2. Also, after determining the individual air times for all the LTE's, scheduling those air times is also an important task.

1.4 Main Deliverables

Better allocation and utilization of spectrum, better data rates for the users of both LTE and Wi-Fi based on their utilities.

1.5 Contributions

We formulated a System Model which is mostly similar to a a real time scenario and obtained an optimization framework with some solid constraints for both Wi-Fi and LTE(will be discussed in the further sections), which will provide us the optimal solution. Using the same system model, we also gave a game theoretic view to the problem. This comes from the idea that not all the base stations belong the same service provider, so they end up playing a game which would maximize their own profit which might end up in reducing the efficiency of the spectrum allocated. In addition to this work, we also constructed a Q-learning problem based on M/M1 Queueing theory and made the network learn the optimal time slots for LTE and Wi-Fi transmissions. With this, the network dynamically allocates a fraction of frame time for LTE and Wi-Fi such that both co-exist fairly.

CHAPTER 2

Literature Review

In this chapter we gave a brief overview on the works carried out in the coexistence of LTE-U and Wi-Fi. Till date, there is still no widely accepted coexistence scheme to enable spectrally-efficient and fair spectrum sharing between LTE and Wi-Fi, with the exception of a few recent research efforts. Huawei and Qualcomm proposed to deploy LTE-U in only partially unlicensed bands and areas with sparse Wi-Fi deployments. The limitation of this approach is that it may lead to under utilization of spectrum resources compared to exploiting the whole unlicensed band. The tutorial paper by Guan and Melodia (2016) served us as a base idea for the problem. Guan and Melodia proposed a cognitive coexistence scheme to enable spectrum sharing between LTE-U and Wi-Fi. The scheme they designed, jointly determine dynamic channel selection, carrier aggregation and fractional spectrum access for LTE-U networks, while guaranteeing fair spectrum access for Wi-Fi based on a newly designed cross-technology fairness criterion. They limited their work to coexisting both of them in a single channel giving a minimum amount of time for Wi-Fi transmission, but didn't work on the optimal time for which LTE needs to transmit. In Almeida *et al.* (2013), the authors proposed to enable coexistence of LTE-U and Wi-Fi by taking advantage of the so-called Almost Blank Subframe (ABSF), a time domain multiplexing feature in 3GPP Rel. 10 [1]. The challenge there is how to find the optimal operating point between the air time given up by LTE and the throughput achievable by the Wi-Fi networks. Our model overcomes this issue by proposing the optimal time for the LTE and Wi-Fi transmission.

In Cano *et al.* (2016), the authors discussed various benefits and coexistence issues. Carrier aggregation, Wi-Fi scheduling, channel selection were discussed in detail. The authors in Wang *et al.* (2017) conducted a survey of the coexistence of LTE and Wi-Fi in the unlicensed spectrum .They considered various deployment scenarios and discussed about them in detail. In Nihtilä *et al.* (2013), a detailed performance analysis of LTE and Wi-Fi when co-existing on a shared channel is explained. In Ratasuk *et al.* (2012), the authors proposed a listen-before-talk scheme by enabling carrier sensing at each LTE PeNB. While the scheme can enable fair coexistence between U-LTE and Wi-Fi, it results in spectrum under utilization because of the carrier sense operation in the listen-before-talk. But our model does not require a listen-before-talk scheme(since we propose to use LTE-U which does not require an LBT) and estimates the air time for both LTE and Wi-Fi based on the required data rates. Sagari in Sagari (2014), proposed an inter-network coordination architecture to enable dynamic interference management between coexisting LTE-U and Wi-Fi networks. In Yun and Qiu (2015), the authors proposed an LTE-U,Wi-Fi coexistence scheme by allowing LTE and WiFi to transmit together and decode the interfered signals. The disadvantage of this model is the physical layer protocol stack has to be redesigned for both LTE and Wi-Fi which may not be practically possible.

Currently, there were no machine learning based approaches to allocate the time slots apart from a few works. A detailed survey about various machine learning models that can be used in Communications, mainly regarding LTE and Wi-Fi is done in Chen *et al.* (2017). Using machine learning approaches for learning the number of slots allotted for both LTE-U and Wi-Fi transmission through time is very intuitive and realizable because the time slots will not differ much from the previous slot to the present slot and also they depend heavily on the previous decisions. One advantage by using learning techniques is that the time for the estimation will be very fast when compared to solving the traditional optimization or game theoretic problems. Using Neural networks or Deep learning techniques in any field require huge data sets for training. This may not be possible since we may not acquire real time data sets. Along with the data sets, the algorithm requires the ground truth to learn from the data while training. In our case there can be no particular ground truth. In other words we are suggesting the ground truth. So our problem becomes more of regression kind of problem rather than traditional classification. One such way to make a network in such a condition is by using a variant of Reinforcement learning where actions of the agent will depend on the rewards and previous actions. In this section, we discuss various related works on the co-existence of LTE and WiFi using various machine learning techniques, mainly the works related to Q-learning, a model free reinforcement learning approach.

The authors in Su *et al.* (2018) discussed a coexistence algorithm in multi channel case based on Q-learning. Their algorithm takes into account both the fairness and the performance of the system while optimizing the duty cycle time. They designed a joint utility function of the system through-put and fairness for the coexistence scenario, and proposed a coexistence algorithm based on Independent and Joint Q-learning algorithm.In Abinader *et al.* (2014), the authors proposed the basic framework of the cooperative coexistence algorithm,which describes the general flow of collaborative coexistence algorithms. In Rupasinghe and Güvenç (2015), a new algorithm based on duty cycle adaptive algorithm is proposed, which achieves the fast convergence of optimal duty cycle by Q-learning. In another work by Su *et al.* (2018*a*), the authors proposed an algorithm that guarantees the fairness between the two systems while improving the system throughput. They've presented a system model and made use of the SINR for calculating the throughput. They have a set of 4 predefined states which are a tuple of throughput and fairness. Their rewards are based on the though-put obtained and a minimum though-put set. The difference between this work and our work is that, we've calculated the optimal duty cycle time even though we did not take SINR into account. In Cano and Leith (2015) the authors proposed a fair proportion of resource allocation algorithm in which the algorithm requires that the average channel occupancy time of the Wi-Fi device be equal to the average channel occupancy time of the LTE-U device to ensuring a fair coexistence between LTE-U and Wi-Fi. Recently Su *et al.* (2018*b*), a coexistence algorithm for allocating idleslots by LTE according to a predetermined duty cycle is proposed in a heterogeneous network. In Pedro *et al.* (2019), the authors proposed a new solution "DM-CSAT" for LTE-U, Wi-Fi coexistence based on Q-learning. They modeled the problem as a Markov decision process, and the Q-learning solution for finding the best duty cycle time is based on the Bellman's equation. They have evaluated the performance of the proposed solution for different traffic load scenarios using the ns-3 simulator.

CHAPTER 3

Optimization and Game Theory Frameworks

3.1 System model

System model always plays a key role in the performance of any model in communications. The way you develop your model and solve for the solution decides the performance. So, we started with a simple case and then moved into a network similar to a real time scenario. We took a simple case where there are only two LTE pico cell base stations and 2 Wi-Fi access points in a network and constructed a system model while imposing the necessary constraints and solved for the optimum solution by hard coding the equations. Then we moved into the very general case of L LTE's and W Wi-Fi access points where any number of LTE's can be connected to any number of Wi-Fi's if they are well inside the range of the Wi-Fi AP. Since the number of Wi-Fi's and LTE BS's are variable, we cannot always hard code the utility functions and solve for the optimum, instead we have to find a way to automate these equations. Once we have the system model in place, we start developing the optimization problem and solve for the solution. Below are the steps followed for framing the system model.

3.1.1 Deployment of Wi-Fi access points and LTE base stations

In a general real-time scenario, Wi-Fi AP's and LTE BS's are randomly present in a defined area. So, we have deployed L LTE BS's and W Wi-Fi AP's randomly in a 100m* 100m square region.

3.1.2 Channel Assignment

Assigning channels or spectrum bands to radio interfaces for communication is termed as Channel assignment. Since we are deploying AP's randomly, we need to take care of the channel assignment. We have to allocate each of the Wi-Fi AP's a particular channel to operate on. But we should keep in mind that no two adjacent Wi-Fi AP's should not be allocated a same channel which may result in interference. So, we have to find the adjacency matrix of Wi-Fi AP's which tells us which Wi-Fi AP is in the vicinity of the other AP's. This problem simply correlates to the well-known graph coloring problem where you've to make sure that no two adjacent states in a map are of no color. Unfortunately, there is no efficient algorithm available for coloring a graph with minimum number of colors as the problem is a known NP Complete problem. Here is what we did. Start by assigning the first AP to channel 1, and for each AP from AP 2 to W, check what are all the used channels for each AP based on its adjacency matrix row, and pick any of the non-used channels randomly. If all the channels are used, assign it a new channel. Alternatively, channel assignment can be also done by first fixing the number of channels to which Wi-Fi AP's are to be assigned instead of assigning a new one once all the available ones are assigned to some channel. But deciding how many channels to fix in the first case is also important. Even if you decide and fixed the number of channels, there is a catch here. If the number of Wi-Fi AP's are too high and they are very near, i.e. suppose there is a network such that say 6 AP's are connected to each other(everyone in the network is connected with every other one), and you fix the number of channels to say 4, then there is no chance of allocating all the AP's in 4 channels. The number of channels has to be at least 6 in this case. So, there is no way round except you have to do the channel assignment based on the procedure mentioned above. i.e. allocating new channels if all the channels present until then are used by some other Wi-Fi AP's. In the next page, we discussed the channel assignment algorithm briefly along with the pseudo code.

Below is the crux of the algorithm implemented for channel assignment

Pseudocode 1: Channel assignment

```
Data: W Wi−Fi AP′s
Result: Vector of assigned channels to all AP′s
Initialization: channels(i) = -1 ∀ i = 1,2...W
A(i, j) = 0 ∀ i, j = 1,2...W
Intermediate: A_{W*W}, A is the adjacency matrix of the Wi−Fi network
for i=1 to W do
    for j=1+i to W do
        if dist btw AP's i,j ≤ rW then
            A(i, j) = 1
channels(1) = 1
for i=2 to W do
    for j=1 to W do
        if A(i,j) == 1 then
            if channels(j) ≠ -1 then
                used(channels(j)) = 1
    for k=W to 1 in steps of -1 do
        if used(k) == 1 then
            break
    temp = used(from 1 to k)
    mat = find(temp == 0)
    if length(mat) == 0 then
        ch = k+1
    else
        ran = a random num which takes a max value length(mat)
        ch = mat(ran);
    channels(i) = ch
    re-initialize array "used" to "0"
```

3.1.3 Describing the system model

Let U_i be the net utility of LTE user $i (i = 1,2,\dots L)$. Let u_i be the utility functions for LTE user i. Let c_j be the cost due to presence of Wi-Fi $j (j = 1,2,3\dots W)$. Let $\beta_\ell = (\beta_{\ell 1}, \beta_{\ell 2}, ..., \beta_{\ell,C})$ be the air time available for LTE user ℓ, where C is the number of frequency channels allotted. i.e. $\beta_1 = (\beta_{11}, \beta_{12}, \dots \beta_{1C})$ is the air time available for LTE_1 where β_{11} is the air time allocated for LTE_1 when it uses Wi-Fi channel 1. Similarly, β_{1C} is the air time allocated for LTE_1 when it uses Wi-Fi channel C. So, the objective is to find the β_{L*C} matrix. Now, the net utility for user i is the difference between the utility of the i^{th} LTE user and the costs due to those Wi-Fi channels to which this particular i^{th} LTE can be connected. So, the net utility is given by:

$$U_\ell(\beta_1,\beta_2...\beta_L) = u_\ell(\beta_1,\beta_2...\beta_L) - \sum_{j=1}^{W} c_j(\beta_1,\beta_2...\beta_L) \times adjmat(i,j) \tag{3.1}$$

where $adjmat$ is the adjacency matrix of the network and $adjmat(i,j)$==1 if Wi-Fi AP j is in the range of LTE i.

Another interpretation for the cost due to the Wi-Fi AP is the log utilities for the AP's with the time left by the LTE BS's. So, instead of subtracting the costs we add those utilities of those AP's which are in the vicinity of that particular LTE.

Figure 3.1 shows a general deployment of a network consisting of 4 LTE BS and 4 Wi-Fi AP's

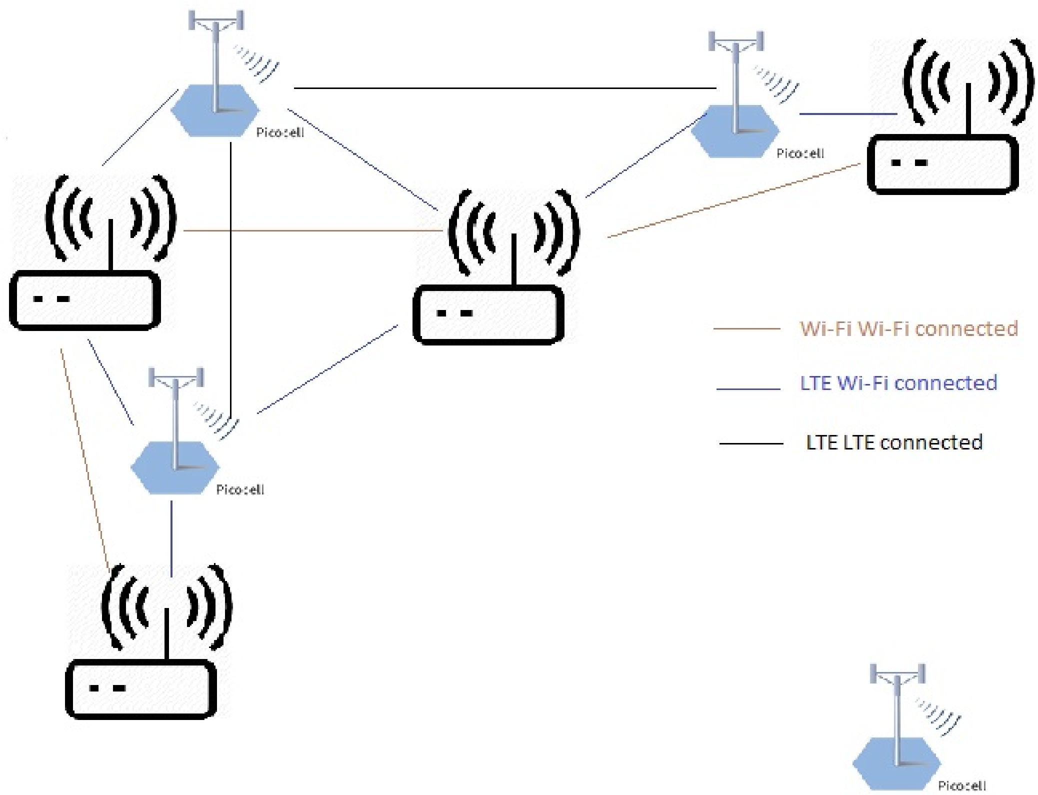

Figure 3.1: General Deployment

The red lines in the figure 3.1 depicts that one Wi-Fi AP is well within the other AP's range. The black line tells that one LTE BS is with in the ragne of the ohter LTE BS. The blue line tells that the LTE BS is with in the range of the WI-Fi AP and hence can use the chanel used by the AP for a fractional amount of time.

3.2 Optimization framework

Once we have the system model in place, we can start optimizing our network to find the solution. In our case the objective is simple,i.e. to maximize the sum payoff(net utility) of the all the L LTE Pico Base stations by giving fair amount of time for Wi-Fi. But we have to define the constraints before trying to maximize net utility. We have derived mainly 3 group of constraints given any network, the Wi-Fi constraints, the LTE constraints(will be discussed in the next session), and the general constraints. Adhering to these constraints, we will try to maximize the sum net utility of all the L LTEs.

$$Max \sum_{i=1}^{L} U_i(\beta_1, \beta_2, \ldots \beta_L) \tag{3.2}$$

$$\begin{aligned} \text{subject to: } & \text{Wi-Fi constraints} \\ & \text{LTE constraints} \\ & \beta_{ij} \geq 0 \,\forall \text{ i,j} \end{aligned}$$

3.3 Constraints

3.3.1 Wi-Fi Constraints

The main objective of the optimization framework is to solve for the β matrix which is of the dimensions $L \times C$ while maximizing the sum utilities. But the optimization has to be done under some constraints. All LTE BS's which will be using a particular channel shouldn't use all the available air time because the AP which will be present in that particular channel will not be using any air time in that channel. Let us consider a simple example and understand the motivation behind the Wi-Fi constraints.

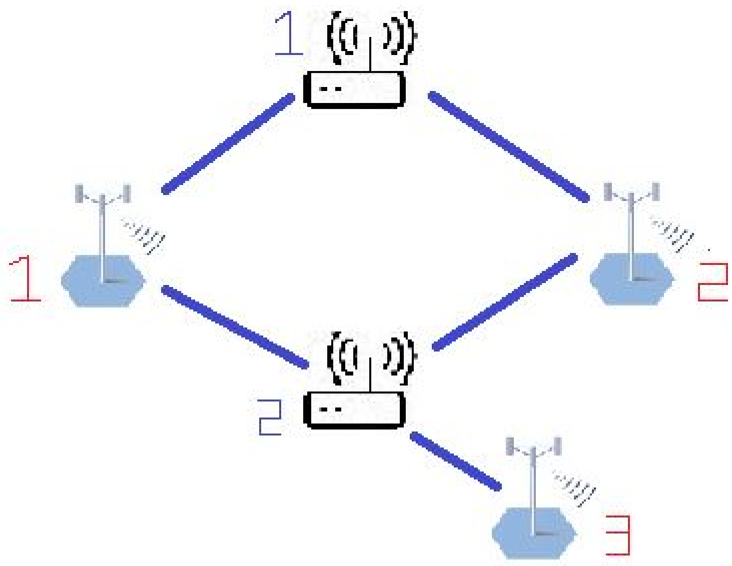

Figure 3.2: Wi-fi Deployment

Let us assume a network consisting of 2 Wi-Fi AP's and 3 LTE BS's as shown in the above figure 3.2. Since we are maximizing the utility of LTE, we will stand at the each of the LTE and try to look at the constraints. It is straight forward that each of the LTE should not consume the whole channel since there should be some time to be reserved for WiFi. But wait! Let's make the things more interesting. Instead of looking the problem in LTE's point of view, stand at each of the Wi-Fi and think what the constraints might be. Each of the WiFi should make sure that all the LTE's using that AP's channel, the air-times of all of those combined should not be more than 1. So, for each of the Wi-Fi AP's, look at the LTE BS's which can be connected to this particular AP, find the Wi-Fi channel of that AP and the sum of all those LTE's air-times while using this particular channel should be less than or equal to one. Let us consider a simple example. Let Wi-Fi number 2 is using channel 1. Let LTE BS numbers 1,3,5 are connected to this Wi-Fi 2. Then $\beta_{11} + \beta_{31} + \beta_{51} \leq 1$ will be the Wi-Fi constraints

3.3.2 LTE Constraints

The LTE constraints are a little involved than the Wi-Fi constraints. Before even moving into the LTE constraints let us consider a simple case and examine it so that we can get an idea on what constraints to impose. Let us consider a network with 3 LTE BS's and only one channel(say channel 1) available and let us assume they are connected as shown in the below figure 3.3.

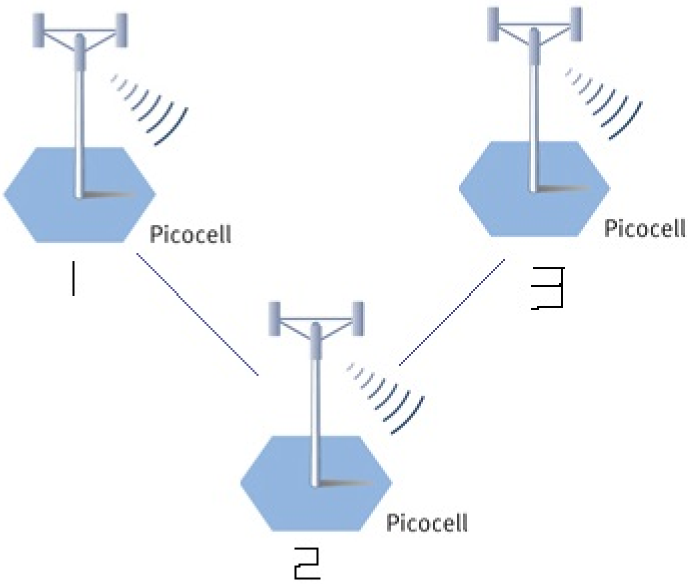

Figure 3.3: LTE deployment

LTE 2 is connected with both 1 and 3 where as LTE1 and LTE3 are not connected. Now let us write the general constraints which we call them "connectivity constraints" where for each LTE find all the LTE BS's which are in its vicinity and the sum of all of them should be less than or equal to one. In the above case, there will be 3 connectivity constraints 3.3,3.4,3.5 one for each LTE.

$$\beta_{11} + \beta_{21} \leq 1 \tag{3.3}$$

$$\beta_{11} + \beta_{21} + \beta_{31} \leq 1 \tag{3.4}$$

$$\beta_{21} + \beta_{31} \leq 1 \tag{3.5}$$

Solving for the optimum value gives all the β_{11},β_{21},$\beta_{31} = 1/3$ which means, given 1-unit amount of time, LTE 1 will transmit one third amount of time, LTE 2 will transmit another one third amount of time and the same with the LTE 3. Although this solution seems to be optimal it is not! If we have a clear look into the problem, LTE 1 and LTE 3 are not connected which means if we give first half amount of time to LTE 1 and the later half to LTE 2, then LTE 3 can also use the first half amount of time since LTE 1 is far from LTE 3 and hence no interference. So, giving 0.5 amount of time to each of the LTE's to transmit is the optimum solution. Since this is global problem, we call this the "the optimal with clique constraints". But how do we arrive at this solution for a general problem? This problem seems to be complex right, except its not! Yes, lets find all the maximal cliques for the above problem. But what is a clique? A clique is a subset of vertices of an undirected graph such that every two distinct vertices in the clique are adjacent. Simply a complete subgraph in an undirected graph is a clique. LTE 1 and 2 form a maximal clique of size 2, and similarly, LTE 2 and 3 form a maximal clique of size 2. Writing the constraints for the cliques we get equations 3.6 and 3.7, i.e.

$$\beta_{11} + \beta_{21} \leq 1 \tag{3.6}$$

$$\beta_{21} + \beta_{31} \leq 1 \tag{3.7}$$

There are only 2 cliques in the above problem. Finding the optimal value considering the above two constraints gives us the solution β_{11},β_{21},$\beta_{31} = 1/2$, which is the same solution as the optimum one. So, we have two solutions one, the solution obtained by following the "connectivity constraints" which we call "the optimal with connectivity constraints" i.e. sum of all LTE's connected to it should be less than one, and the other is the global optimal which is obtained by implementing the clique constraints. But wait how do we find the all the cliques in a network in the first place? We should keep in mind that finding all the cliques in a graph is an NP-complete problem, i.e. there is no polynomial time algorithm to find all the cliques. So, if the number of LTE BS's increases, the running time increases exponentially. However, there are some algorithms which perform way better than the brute force methods to find all the cliques in a graph. The one which we followed is Bron–Kerbosch algorithm (refer bron kerbosch (2019)) which uses recursive backtracking to find all the maximal cliques in a graph.

In order to find all the clique constraints for our problem, consider the network with only all the LTE BS's present removing all the Wi-Fi AP's. We took the base code from Jeffrey Wildman (2011), and modified it to find all the cliques present in the network using the Bron–Kerbosch algorithm. For each of the clique, multiply with the β_{L*C} matrix and every element in the matrix should be less than or equal to one since the LTE BS which form a clique can use any of the channel based on the Wi-Fi it is connected to. Incorporating both the Wi-Fi and LTE constraints the sum utility maximization problem becomes as equation 3.8.

$$Max \sum_{i=1}^{L} U_i(\beta_1, \beta_2, \ldots \beta_L) \tag{3.8}$$

$$\text{subject to : } \sum_{l=1}^{L} \beta_{l,channels(w)} \leq 1 \;\; \forall\, w = 1,2,..W$$
$$cliques' \times \beta_{l,channels(w)} \leq 1$$
$$\beta_{ij} \geq 0 \, \forall \text{ i,j}$$

3.4 Utility Functions

3.4.1 LTE Utility Functions

The utility functions for user i are assumed to be log utility functions as shown in 3.9

$$u_i = log(1 + rate_L(i) \times \sum_{j=1}^{C} \beta_{i,j} for i = 1,2,\ldots L \tag{3.9}$$

and $rate_L(i)$ is the rate vector which is predetermined based on the deployment of the network and is constant. Determining the rate vectors will be discussed later in this chapter briefly. The constraint that each air time allocated should be greater than 0 is straight forward since all the air times allocated for any of the LTE or Wi-Fi should be greater than or equal to zero. The +1 in the utility function is to ensure that the utility is always greater than 0.

3.4.2 Wi-Fi Utility Functions

The costs for the Wi-Fi are taken as quadratic costs as shown in the equation 3.10:

$$c_j = (\sum_{i=1}^{L} \beta_{adjmat(i,j)==1,channels(j)})^2 \tag{3.10}$$

Also, in other way as the LTE BS's, Wi-Fi AP's can also have log utility functions which we can maximize along with the LTE utility. But the air time for the Wi-Fi should be the sum of all the air-times of all the LTE's using that particular channel, subtracted from 1. Equation 3.11 can be used as the costs for the Wi-Fi AP.

$$c_j = log(1 + rate_W(j) \times (1 - \sum_{i=1}^{L} \beta_{adjmat(i,j)==1,channels(j)}) \tag{3.11}$$

where, $rate_W$ is a predetermined constant vector based on the network deployment and will be discussed later in this chapter.

3.5 Game Theoretic Framework

Consider a generic real time situation where different LTE base stations belong to different service providers. Every service provider wants to maximize his own profit i.e. maximizing his own utility. So, they end up playing a game. Understating how the players make their strategies if they are rational(non-cooperative) or if they are cooperative is important because it gives us an intuitive feel for the optimization problem i.e. what will be the utility for the players if the players are rational. To know how the players will play the strategies and how the game comes to an equilibrium, game theory provides us with Nash Equilibrium which gives a better insight into the game. Nash equilibrium is defined as a stable state in which each player has chosen a strategy, and no player can benefit by changing strategies while the other players keep theirs unchanged, i.e. no participant can gain by a unilateral change of strategy if the strategies of the others remain unchanged.

3.5.1 Objective

To find the Nash equilibrium, if exists, so that we can suggest the LTE base stations whether to be rational or should they try to cope up with each other for benefit of all the players.

3.5.2 Nash Equilibrium Solution

To find the Nash equilibrium solution, we have to fix the strategies of all the other players(here the LTE BS's) and we see whether a player will be benefited if he tries to deviate. So, we will start with some initial random air times for $\beta_l^0 = (\beta_{l1}^0, \beta_{l2}^0, \ldots \beta_{lC}^0)$ which satisfies the constraints, i.e. the air times are feasible and iteratively solve for convergence until the payoff while the player deviating is not more than some ε where $\varepsilon \geq 0$ and $\varepsilon \ll 1$ i.e we took,$\varepsilon = 1e-3$. Instead of solving for all the L LTE's at once, we have solved for each of the LTE simultaneously and updated the new $\beta^{(k+1)}$ every time since the solution of the complete β matrix will be feasible according to constraints we imposed.

For finding β_l^{k+1} from $\beta_1^k, \beta_2^k, \ldots \beta_L^k$,we have to maximize the utility of the l^{th} LTE varying the decision for l^{th} LTE and keeping all the other LTE's decisions fixed. But we need to again remember to impose the constraints while maximizing the net utility. Similar to the optimization problem, we have both LTE and Wi-Fi constraints. Hence the game theory problem should solve the below equation 3.12, i.e.

$$Maximize : U_l(\beta_l, \beta_{-l}^k) \tag{3.12}$$

$$\begin{aligned} \text{subject to: } & \text{wifi constraints} \\ & \text{clique constraints} \\ & \beta_{ij} \geq 0 \ \ \forall\, i, j \end{aligned}$$

where β_{-l}^k is the decision of all other LTE's except the l^{th} LTE.

3.6 Constraints

3.6.1 Wi-Fi constraints

Totally, there will be W Wi-Fi constraints. The constraints are similar to the global optimum Wi-Fi constraints except, all the β matrix is not variable. In the β matrix, we will plug in all the values from the previous iteration and making the whole row of the matrix of the LTE l for which we are trying to find the optimum utility, variable, fixing all the other utilities. We did this by performing a dot product between the new obtained matrix and the adjacency matrix of the LTE and Wi-Fi. Now in these W Wi-Fi constraints, we will consider only those constraints for which the LTE l is connected to.

3.6.2 LTE clique constraints

Similar to the clique constraints we have in the global optimum problem, we will have the constraints here also, except the problem here becomes a bit easy. Instead of looking at all the cliques, we are done if we have the cliques to which the particular LTE belongs to. The new matrix which is obtained by plugging in all the values from the previous iteration and remove the whole row of the matrix of the LTE l will be present here too. For each of the cliques, we will have to make sure that the sum of the air times of the L LTE's(which forms a cliques) including the l^{th} LTE's airtime which is the only variable every iteration, on all the channels should be less than or equal to one. These clique constraints are the optimal constraints. So, the problem is termed as "game theory optimal with clique constraints". But since computing cliques when we move to a heavily deployed network becomes very hard, we can also impose the connectivity constraints here too which we imposed in the global problem and we can term the problem as "game theory sub optimal with connectivity constraints". The constraints in the game theory sub optimal are instead of finding the cliques and then multiplying them with all the channels in the β matrix, we do the same with the first neighbors of the l^{th} LTE. Remember that for any LTE, there will be only one connectivity constraints.

For finding $\beta_l^{k+1} from \beta_1^k, \beta_2^k, \ldots \beta_L^k$,

$$Maximize : U_l(\beta_l, \beta_{-l}^k) \tag{3.13}$$

$$\text{subject to: } adjmat.\, temp_{i=1:L\ except\ l}^k$$
$$required\_cliques' \times \beta_{i=1:L\ except\ l}^k$$
$$\beta_{ij} \geq 0 \ \ \forall\ i, j$$

where adjmat is the adjacency matrix between LTE network and Wi-Fi network, temp is a $L \times W$ matrix obtained by keeping in each of the Wi-Fi column, the corresponding betas according to the assigned channel to which this LTE belongs to except for the $l^t h$ LTE where we replace the entire row by the variables we are solving .Required cliques is the clique matrix consisting of the cliques LTE l belongs to.

NOTE: We have solved for each of the LTE simultaneously and updated the new β^{k+1} every time since the solution of the complete β matrix will be feasible according to constraints we imposed.

NOTE: Stop the iterations when the payoffs of all the players is less than ε when deviating.

3.7 Computing rate vectors

3.7.1 Computing LTE rate vector

The rate vector for LTE "$rate_L$" that are used in the above sections is a vector matrix consisting of constants depending on the network deployment. Let us consider an LTE BS at location (X,Y) in the 2D space. The rate for this LTE BS depends on the LTE users that are present around this BS and are connected to this BS. For the implementation case, we considered the well known "Poisson distribution" for knowing the number of LTE users present near the LTE BS. Once the users are known, the rate for the LTE BS is calculated according to the equation 3.14.

$$rate_L(l) = \sum_{i=1}^{N} BW \times log(1 + (Pt(d(i)^{-n}) \times (R^2))/N) \tag{3.14}$$

where: BW : Bandwidth of the channel

Pt :

d(i) : distance from LTE user i to LTE BS

n : Path loss coefficient

R : exponential random variable with mean μ

N : Noise power

3.7.2 Computing Wi-Fi rate vector

The rate vector for Wi-Fi "$rate_W$" that are used in the above sections is a vector matrix consisting of constants depending on the network deployment. Let us consider an Wi-Fi AP at location (X',Y') in the 2D space. The rate for this Wi-Fi AP depends on the Wi-Fi users that are present around this AP and are connected to this AP. For the implementation case, we considered the well known "Poisson distribution" for knowing the number of Wi-Fi users present near the Wi-Fi AP. Once the users are known, the rate for the Wi-Fi AP is calculated according to the equation 3.15.

$$rate_W(w) = \sum_{i=1}^{M} BW \times log(1 + (Pt(d(i)^{-n}) \times (R^2))/N) \tag{3.15}$$

where:

BW : Bandwidth of the channel

Pt : Transmitted Power

d(i) : distance from Wi-Fi user j to Wi-Fi AP

n : Path loss coefficient

R : exponential random variable with mean μ

N : Noise power

3.8 Tools used

The tools that were used for developing the model are:

1. Matlab : All the codes for this project are written in Matlab
2. CVX : For solving the optimization problem, CVX matlab tool is used. CVX is a Matlab-based modeling system for convex optimization.

CHAPTER 4

Q-learning

4.1 Motivation and Objective

We have discussed the optimization and game theoretic frameworks in chapter 3.Our model proposes an optimal air time for transmission of LTE and WiFi. Since solving the optimization problem in real time in a really big network consisting of hundreds of nodes may take a huge time which is not realisable, using of machine learning techniques for such a scenario may boost the running time and even the performance of the model.Coming to our problem i.e. estimating the optimal duty cycle time, we do not have the data set or ground truth to perform any supervised learning or unsupervised learning. Also, the ground truth concept in our case does not make any sense since there is no particular optimal solution for any given network with any load. If they've been there, we would formulate a simple supervised learning problem and train a network that would learn the model. So, our objective is to get this air time β (now on-wards duty cycle time) though any learning technique which does not require ground truth and data sets. In this chapter we discuss one such model known as Q-learning, a model free reinforcement learning approach. We first develop a system model and then propose a learning problem and apply the Q-learning algorithm.

4.2 Introduction to Reinforcement learning

In this section we will introduce you to Q-learning, which is a part of Reinforcement Learning and then explain in detail what is the role of Q-learning in our problem statement. But, what is reinforcement learning in its first place?Machine learning is broadly divided into 3 classes, Supervised learning, Unsupervised learning and Reinforcement learning. Reinforcement Learning is one of the aspect of Machine learning in which an agent learns to behave in an environment, by performing certain actions by observing the rewards which it get from those actions. Reinforcement Learning is learning what action to take in certain situation so that the agent can maximize its numerical reward. A typical block of reinforcement learning is shown in figure 4.1

4.2.1 Elements of Reinforcement learning

1. Agent : Our player which needs to take actions
2. state : The different places in which an agent can possibly be
3. Action : The decision which the agent takes at a state
4. Policy : mapping from perceived states of the environment to actions
5. Rewards : After each action, the environment sends to the agent a single number called reward, where the objective is to maximize this reward
6. Value Function : the value of a state is the total amount of reward an agent can expect to accumulate over the future, starting from that state

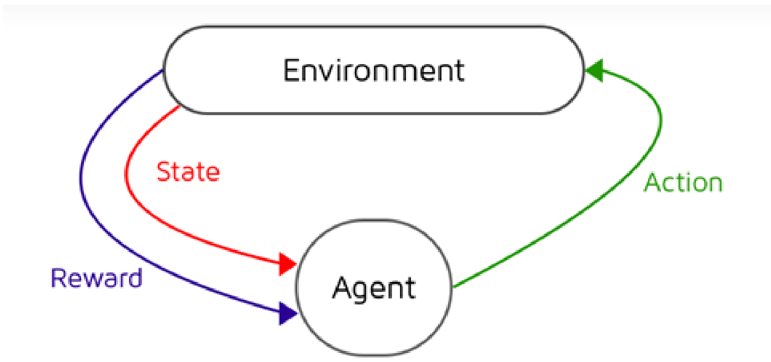

Figure 4.1: Reinforcement learning

A simple example could be a bot trying to learn how to play table tennis. We define the bot with predefined actions say forward,back hand, smash, cut. Initially the bot don't know what action to take. So, its picks up random actions. If that action is good for the corresponding state in the environment, we reward the bot with a positive value, where as if it is bad, we reward a negative value. Now the agent will learn that at the present state, if it took an action A, it gets a reward R. So, we train the network for a very long amount of time so that the agent learns the whole environment.

4.3 System Model

In this section we provide a simple system model for applying Q-learning based approach to determine the optimal duty cycle time. Let us consider a network consisting of only 1 channel along with 1 LTE BS and 1 Wi-Fi AP. Users may arrive at the LTE BS or Wi-Fi AP depending on their arrival rates. We formulate the system model by proposing a Markov Decision Process where the states, actions are derived using M/M/1 queue model. Markov Decision Process (MDP) is a mathematical framework commonly applied for modeling decision making problems in random situations.

4.3.1 M/M1/Queue

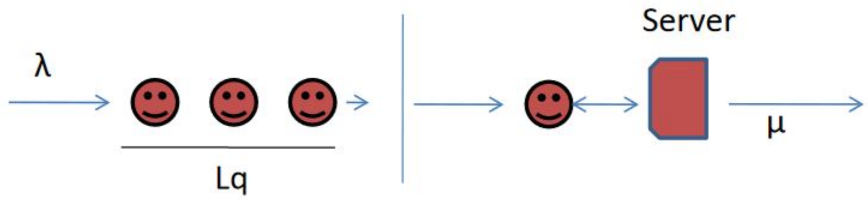

Figure 4.2: M/M/1 queue

M/M/1 queue represents the queue length in a system having a single server, where arrivals are determined by a Poisson process and job service times have an exponential distribution. The above fig 4.2 represents a M/M/1 queue. M/M/1 queue serves as the basic model in queuing theory. The features of an M/M/1 queue are depicted in the below table 4.1

Table 4.1: Table with features of M/M/1 queue

Agent	which gives the optimal solution
state	a tuple of length of both the queues along with an action
Action	Duty cycle time (say 0.3 or 0.5 of whole time)
Rewards	function which is a linear combination of the both the queue lengths
Environment	The time line from t=0 to T

4.4 Q-learning

In Q-learning, the agent, at each discrete time step t, takes one action among a predefined set of actions β, i.e the fractional duty cycle time based on the observation of its current states t and the reward r_t provided by the environment. Given the current states t the future state $t+1$ is independent of the past. Both the transition from s_t to s_{t+1} and the reward,r_{t+1}, are determined as a consequence of the action selected and the previous state. The transition probability from one state s_t to s_{t+1} depends on the current state and the action and is given by the equation 4.1:

$$p(s',r|s,a) = Pr(s_{t+1} = s', r_{t+1} = r|s_t = s, a_t = a) \tag{4.1}$$

where a represents the action taken by the agent to move from the state s to the state s', yielding a reward r. One most common way to solve an MDP is by using the famous **Bellman equation**. According to Bellman, an optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision. Interestingly to solve an MDP completely, Bellman equation requires the knowledge of availability of an explicit model completely which includes knowing all the possible state transition probabilities. But, practically, knowing all the transition probabilities is very difficult. Hence to overcome this issue, we use Q-learning which does not require an explicit model to solve the MDP. In other words, it is not required to know the state transition probabilities. **The Q-Learning algorithm is based on a Q function that is updated whenever it receives a reward from a state transition after the agent takes a certain action**. Say the agent is at time step t, the function Q_t is updated at the next decision making time i.e.end of a frame, when a reward r is observed while the agent is transitioning from state s_t to s_{t+1}. The update equation for the Q matrix is given by the equation 4.2:

$$Q_{t+1}(s_t, a_t) \leftarrow (1-\alpha)\ Q_{t+1}(s_t, a_t) + \alpha\ (r + \gamma \max_{a \in \beta} Q_t(s_{t+1}, a)) \tag{4.2}$$

where α and γ are the learning rate and the discount factor respectively. In Q-learning or to keep it more simple, in any reinforcement learning problem, one of the challenges that arise is the trade-off between exploration and exploitation. Exploitation means we move greedily, selecting the move that leads to the state with the greatest value where as exploration is occasionally selecting random moves from the other actions. Exploration is very necessary and its importance can be seen by a simple example. Let us consider a man searching for treasure in two boxes. Let box_1 gives one gold coin with 0.9 probability and box_2 gives 1000 gold coins but with only 0.1 probability. If the man continue exploiting the box_1 he will never know the hidden treasure under box_2. Instead what he should do is randomly explore the other possibilities so that he could get a better future reward. Any reinforcement model will both exploit and explore if it is really well trained. Since in the equation 4.2, we took care of the exploration part, our model can suugest the optimal duty cycle time, if well trained. Aslo, since the agent is capable to learn upon experience by exploring and exploiting, Q-Learning is highly suited for solving Markov decision problems without explicit knowledge of the transition probabilities.

For the Q-learning based coexistence algorithm, Boltzmann algorithm is used as the action policy of the agent selection. The Boltzmann algorithm is a common algorithm for balancing the accumulation of exploiting and exploring. The algorithm calculates the probability of different actions according to formula 4.3, and then select the action according to the probability

$$P_{a|s} = \frac{e^{\frac{Q(s,a)}{T}}}{\sum_{a \in A} e^{\frac{Q(s,a')}{T}}} \tag{4.3}$$

where $P_{a|s}$ is the probability that the agent will select action a in the state s, and T is the temperature value. We reduce the T value by equation 4.4, to reduce the number of explorations of the optimal policy of the agent,

$$T = \frac{T_0}{log(1+N)} \tag{4.4}$$

where T_0 is the initial temperature value, and N is the number of times that the action is selected by the agent. After calculating the probabilities of all the actions, one can pick an action based on cumulative probability graph. One more method to pick a particular action without the equations 4.3 and 4.4 is by choosing a random number and comparing it with the exploration constant(ε). We've implemented both methods.

So, by formulating the duty cycle problem into a Markov Decision Process considering the M/M/1 queue into account, we have developed our model. The remaining part of this section describes the scenario of our problem.

4.4.1 Describing the system model

Both the LTE BS and the Wi-Fi AP uses separate M/M/1 queue for its users arrival and service. Users will arrive at each of the queue according to a Poisson distribution along with their individual exponential service times. The following table 4.2 describes the elements of Q-learning in our problem.

Table 4.2: Table with elements of Q-learning

Property	Feature
Calling Population	An infinite population with independent arrivals and not influenced by the queuing system
Arrival Process	Poisson distribution of arrival rate
Queuing configuration	Single waiting line with unlimited space
Queue discipline	First come, First serve
Service Process	Exponential service time distribution

Figure 4.3: LTE queue

Figure 4.4: Wi-Fi queue

Let us consider that LTE users arrive at rate λ_L with service times following the exponential distribution with mean μ_L as shown in the figure 4.3. Also, let Wi-Fi users arrive at rate λ_W with service times following the exponential distribution with mean μ_W. Let us consider at any point of time the lengths of the queue in LTE queue is N and in the Wi-Fi queue is M as shown in the figure 4.4. So at any point the tuple (N_k, M_k) represents the queue lengths at frame time k. The agents, states, actions for the Q learning problem are defined as follows.

4.4.2 Objective

The objective is straight forward,i.e to minimize the queue lengths in both the queues. To put it more precisely,we want tp minimize the sum of the both the queue lengths. In other words, the system should serve as many LTE users and Wi-Fi users in such a way that the queue lengths of both the queues will be minimum. Since our aim is to make the queue lengths as minium as possible, equation 4.2 changes to the following equation 4.5

$$Q_{t+1}(s_t, a_t) \leftarrow (1-\alpha)\ Q_{t+1}(s_t, a_t) + \alpha\ (r + \gamma \min_{a \in \beta} Q_t(s_{t+1}, a)) \tag{4.5}$$

4.4.3 Events Description

Let us now put down all the events that may happen in a timeline so that we can decide what to do in each event.

1. flag: flag bit is a binary output i.e. flag = 1 means LTE time so all the users that are present in the LTE queue can be served by first come first serve basis and flag = 0 means Wi-Fi time, so the Wi-Fi users that has the minimum contention time and is present in the Wi-Fi queue gets the service and the process repeats. Note that Wi-Fi uses contention mechanism before allocating its uses to service.

2. LTE User generation: This event means an LTE user has arrived at a point in the time line. Once this event occurs, we will generate another random variable indicating the occurrence of next LTE user generation

3. Wi-Fi User generation: This event means an Wi-Fi user has arrived at a point in the time line. Once this event occurs, we will generate another random variable indicating the occurrence of next Wi-Fi user generation

4. LTE service time: This event describes that an LTE user has completed its service in the LTE time, so we have to remove this user from the queue.

5. Wi-Fi service time: This event describes that an Wi-Fi user has completed its service in the Wi-Fi time, so we have to remove this user from the queue.

6. Duty cycle time: This event says that the duty cycle for LTE is completed and Wi-Fi can use the channel for transmissions

7. Contention time: the contention time is the minimum of all the counters of the Wi-Fi users that are present in the queue. If this event occurs that means, some Wi-Fi user has completed its contention time so, we have to start serving the Wi-Fi user from then.

8. Frame time: If this event occurs, that implies that the frame time got completed and the agent has to take an action now for the next frame based on the current state, i.e the length of the queues and the previous action.

Algorithm 2: Q-learning algorithm

Data: $\lambda_L, \lambda_W, \mu_L, \mu_W, \beta = (0.1, 0.2, ..0.9)$

Result: actions,i.e duty cycle times at every completion of frame

1 Initialization:

2 initialise all the necessary parameters to the required values including zeros for the Q matrix and choosing a random action

3 Compute the initial state s_t

4 Calculate the probability values of agent u in the unlicensed channel m with the different action β_j in the state s_t

5 According to the equation and the equation , perform the action with the maximum probability in current state, if there are multiple identical probabilities, then randomly select one;

6 Perform action β_j, get the corresponding environmental reward value r_t, then enter the next state s_{t+1} from s_t

7 Update the corresponding action Q value of the agent

8 $t \leftarrow t+1$ jump to step 3 in the learning phase

4.4.4 Q-learning models

Once the objective is clear, we developed the model in 2 phases

1. Optimal learning phase
2. Optimal implementation phase

During the learning phase the agents learns what actions to take when the agent is in a particular state by the well known strategies of exploration and exploitation. A detailed algorithm [Algorithm 4] of the learning phase is given in the appendix chapter 2 and 3. Once the network is trained, the implementation phase can be started with the Q matrix obtained from the training phase.

NOTE: Queue and Q are two different things completely. Queue is present and introduced in the system model itself which uses first in first out way, where as Q is the name of the algorithm in which Q table is updated through time while learning.

CHAPTER 5

Results and Discussions

In this chapter we first provide the simulation results obtained along with the set up and assumptions required for the optimization and Game theoretic frameworks. Then we set up the agent in the Q-learning, trained it using the Q update method and reported the results of the network by comparing the optimal actions taken by our agent with some naive approaches.

5.1 Optimization and Game theory results

5.1.1 Simulation Set up

The following are the valid assumptions we took for setting up the experiment. We considered the utilities of the LTE to be :

$$u_i = log(1 + rate_L(i) \times \sum_{j=1}^{C} \beta_{i,j} \quad for \quad i = 1, 2, \ldots L \tag{5.1}$$

And the cost due to the presence of Wi-Fi, in other words, the utilities for the Wi-Fi are

$$c_j = log(1 + rate_W(j) \times (1 - \sum_{i=1}^{L} \beta_{adjmat(i,j)==1, channels(j)} \tag{5.2}$$

where, $rate_L$ and $rate_W$ are the constant vectors determined based on the LTE user's deployment. The rate vectors $rate_L$ and $rate_W$ are calculated by the process explained in section 3.7.1 and 3.7.2. The constants required are taken as:

BW : Bandwidth of the channel = 20

Pt : Transmitted Power

d(i) : distance from LTE user i to LTE BS

n : Path loss coefficient = 2.5

R : exponential random variable with mean μ = 10

N : Noise power = 1

M : no of users with Poisson distribution with mean = $20/10^4$

We then considered a random network deployment as shown in the figure (Fig 5.1) where there are 7 LTE BS's and 7 Wi-Fi AP's, and solved for the following 4 problems:

- i Optimal with clique constraints
- ii Optimal with connectivity constraints
- iii Game theory with clique constraints
- iv Game theory with connectivity constraints

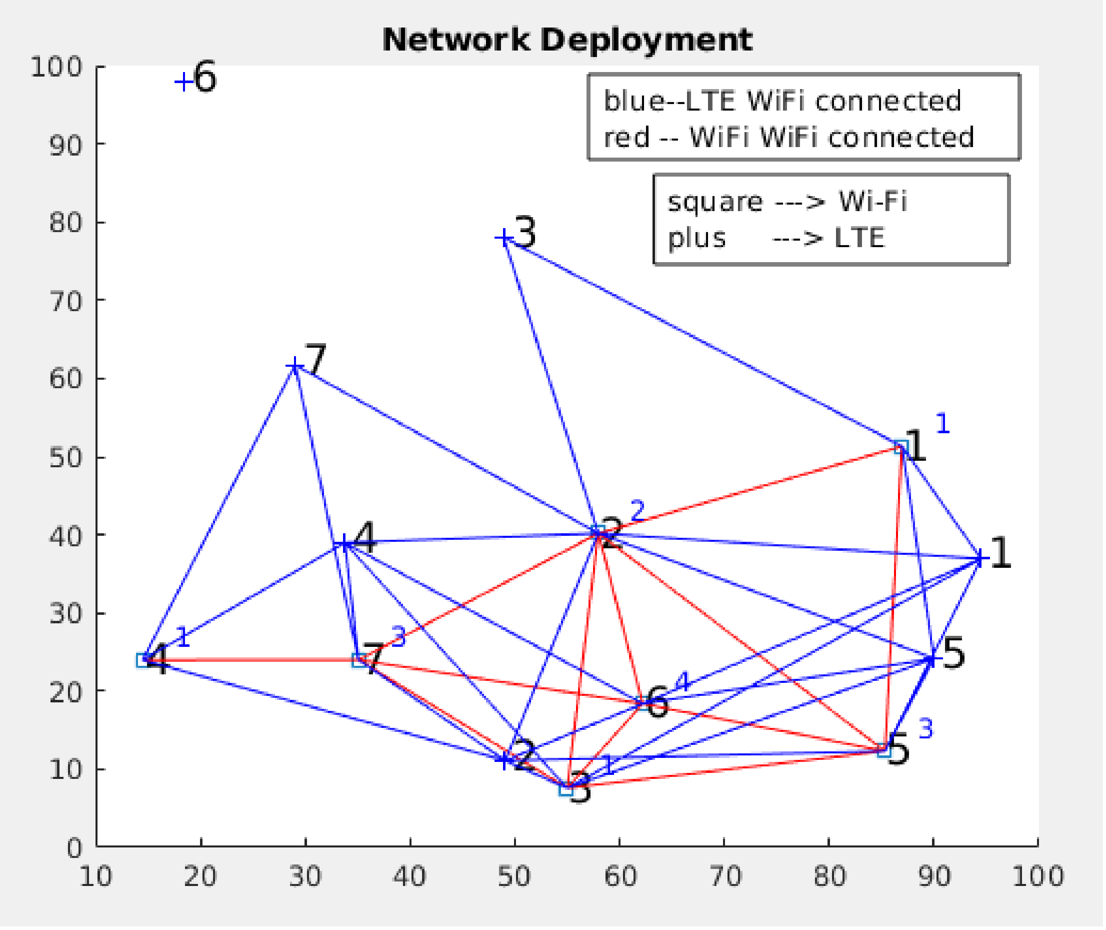

Figure 5.1: sample deployment of 7 LTE's and 7 Wi-Fi AP's

The square indicates that it is a Wi-Fi AP where as the "+" indicated that it is an LTE. All the four optimum problems are solved, and the results are depicted in the table 5.1 below.

5.1.2 Results

NOTE: All the codes are written in matlab and we ran them in "Dell latitude 3490 PC".

Table 5.1: Table with the maximum sum utilities for the deployment for the 4 sub problems

Problem	Max Sum Utility
Optimal with clique constraints	7.7552
Optimal with connectivity constraints	7.6903
Game theory with clique constraints	6.0274
Game theory with connectivity constraints	5.8117

5.1.3 Observations

One observation is that the order of the magnitude of the max sum utilities. The optimal with clique constraints has the highest sum utility since we imposed better constraints, i.e. the clique constraints. Next in the order comes the optimal with the connectivity constraints, because the connectivity constraints are heavy constraints which are not necessary but easy to compute. The search space for the optimum value becomes less in this case when compared to the clique constraints. Both the game theory solutions yielded a lesser sum utility than the global optimum because, of the players being rational trying to maximize their own utility, instead ended up getting lesser utilities. Even though social optimum suggests that cooperation will lead to a better result for all of them, they don't care much about other player's utilities. Game theory with clique constraints yield a better solution than the game theory with connectivity constraints because of the same reason mentioned above. Bigger search space is available for the clique constraints than the connectivity constraints.

While solving the Nash Equilibrium solution in the optimization framework, we obtained the solution using ε Nash Equilibrium, there will be some loss in net utility because obtaining the exact solution might take a long amount of time resulting in a unrealizable implementation. If this loss is not tolerable, that indicates that players must cooperate in order to achieve better utility rates for all of them. But if the loss variation is not too high, then we can suggest the players that they can play their own strategies without any cooperation which would be more realistic to the real-world service providers. One method which gives a measure to characterize the loss is by calculating the **"Price of Anarchy(POA)"**. The Price of Anarchy is defined as the ratio between the optimal solution and the Nash equilibrium. Generally, POA $\geq$ 1. For the deployment we considered in figure 5.1, considering 7 LTE and 7 Wi-Fi, calculating POA gives POA = 1.2866. If POA is $\gg$ then players must cooperate. If POA is near to 1, if that is tolerable, players can be rational.

To infer how the sum utilities vary, if the number of LTE's vary keeping the Wi-Fi AP's constant, we fixed the above deployment as shown in figure 5.1 and varied the LTE BS's from 7 down to 1 LTE(removing 1 LTE each time while keeping the other network constant) and calculated the sum utilities for all the 4 problems. The results are noted and a bar graph 5.2 corresponding to the values is reported below.

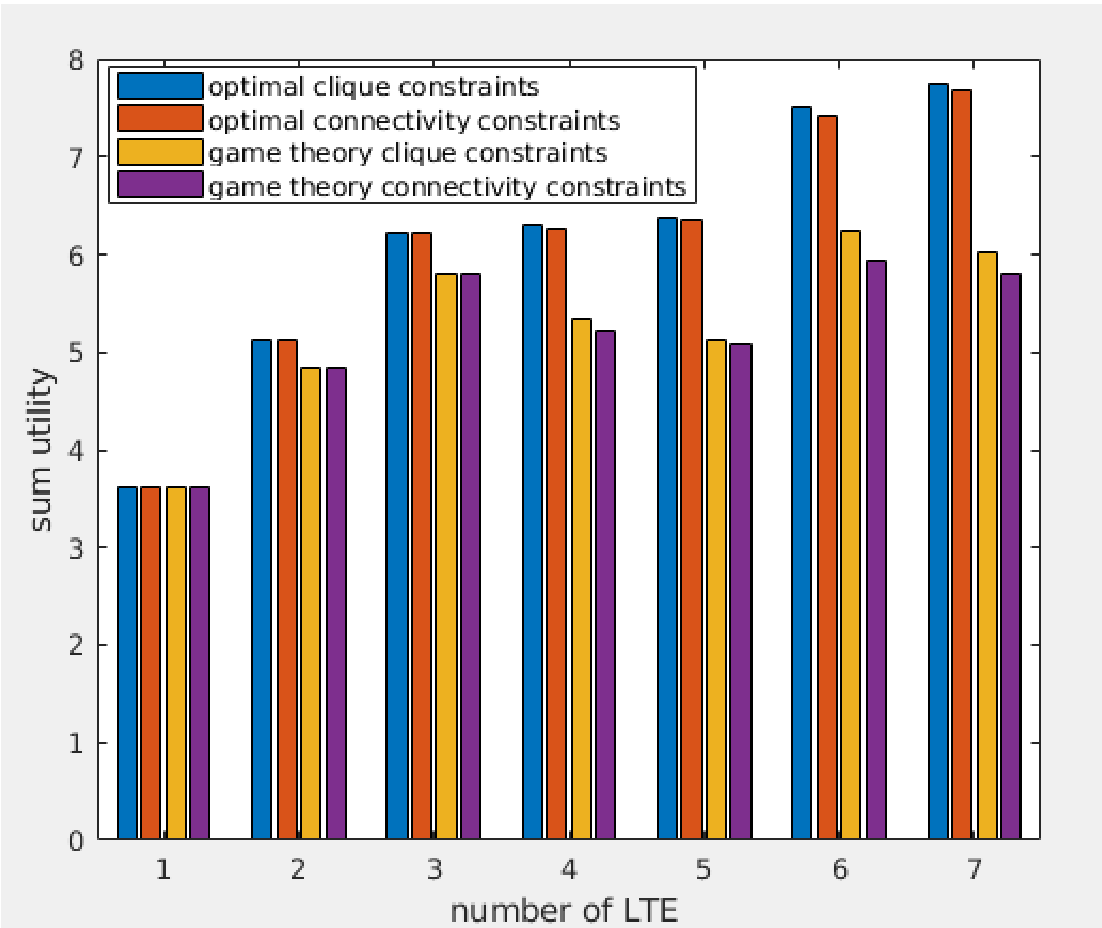

Figure 5.2: comparison of optimal vs game theoretic solutions

We can see that the order always follows, i.e. optimum with clique > optimum with connectivity > game theory with clique > game theory with connectivity. If the number of LTE is only 1, an interesting thing happens. All the 4 problems solution is same. It is because, since there is only 1 LTE, deployed in a game theoretic perspective, there are no other players to play, so the LTE will end up maximizing its own utility which is the same case in the both the optimal problems. One more observation is that, sometimes both the optimal problems solution's may be equal, it is because, the clique constraints and the connectivity constraints are same. If U^* is the optimal solution, and U^- is the solution in game theory, then, $U^- \leq U^*$

5.2 Q-learning results

5.2.1 Simulation Set up

The following are the assumptions made during the training and testing phase of the Q-learning. We considered that only a single channel is available for LTE and Wi-Fi to transmit their data. Once the observations and results are well in place we go to more number of channels which seems more appealing. LTE users and Wi-Fi users arrive at rates λ_L and λ_W with their arrival rates as μ_L and μ_W. Here are some constants:

During learning phase:

λ_L = 1

λ_W = 1

μ_L = 10

μ_W = 10

β = {0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9}   // set of actions

ε = 0.2   // exploration constant

gamma = 0.9   // discount factor

time = 1000000;   // training time

frame_time = 2;   // frame time

contention_const = 0.1; // Wi-Fi user chooses a counter time for contention from 0 to 0.1

λ = 1   //weighing constant for N,M

$\alpha = 0.9$ // learning rate

During implementation phase:

All the constant remain the same except the exploration constant ε which we set it to 0 in testing phase

$\varepsilon = 0$ // exploration constant

5.2.2 Results

NOTE: All the codes are written in matlab and we ran them in "Dell latitude 3490 PC".

Based on the constants mentioned above, we trained the agent. Then we tested it for time = 100000. We found out the cost of the network at each frame by the equation 5.3.

$$cost_k = cost_{k-1} + (N_k + \lambda\ M_k) \times \gamma^{k-1} \tag{5.3}$$

where N_k and $_k$ are the queue lengths at frame time k

We observed that the cost function as frame time increases starts saturating. For the performance analysis, we compared our trained model with 2 other naive models. They are:

1. Picking a random action every time from the set of all actions
2. Picking a constant action every time irrespective of all actions

We picked 0.5 time of the frame time as constant action every time and we computed the cost functions for the 3 models. The plot is depicted in figure 5.3

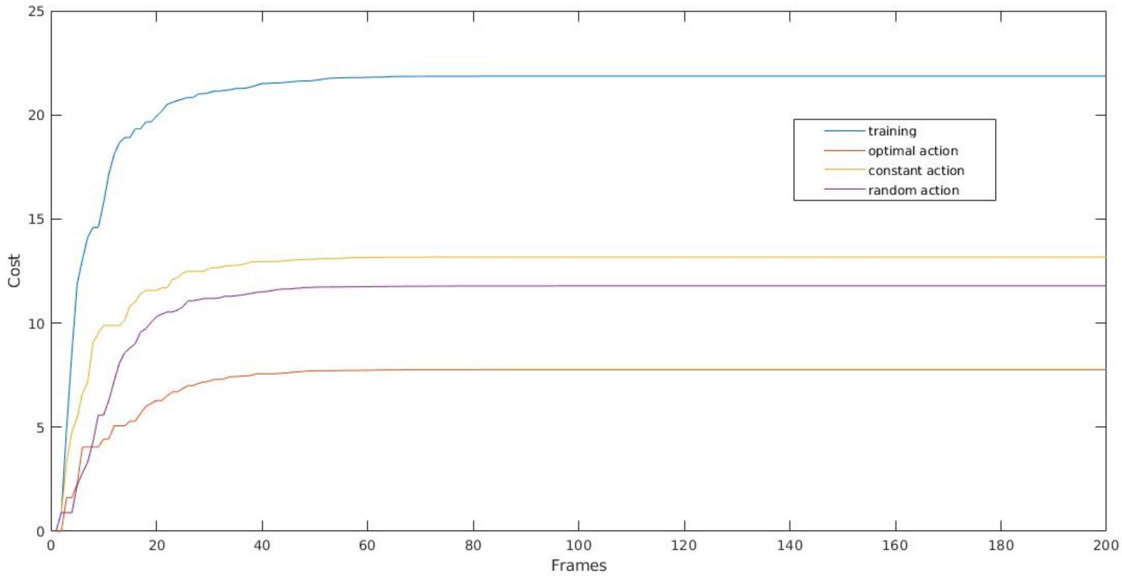

Figure 5.3: comparison of costs for optimal action, random action, constant action

5.2.3 Observations

We can observe that the optimal cost curve is the minimum among the three model,i.e. our network has learned and able to adapt dynamically the duty cycle time based on the queue lengths of the LTE and Wi-Fi. But one more interesting observation is during the training phase. The cost curve during the training phase(the curve that has the highest values) is higher than the costs of the other 2,i.e. the random action cost and the constant action cost. That means our network is performing really poor while learning phase. But in implementation, the model outperforms the other two models. Also, one more observation is that choosing a constant action here 0.5 time of frame time gives low cost than choosing a random action every time. This do really make sense because we are allocating equal amount of time for LTE and Wi-Fi when both the user generation and service times follow the same distribution.

To see how the network adapts by varying various parameters, we varied γ, and looked at the average costs for the optimal solution, random action solution and constant action solution. The plot for the same is shown in the figure 5.4

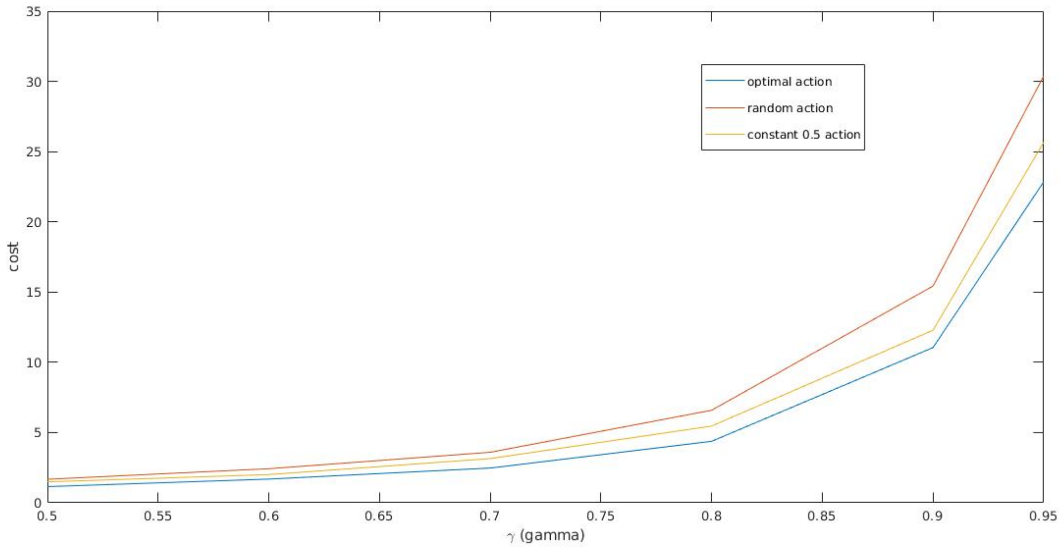

Figure 5.4: comparison of average cost for optimal action, random action, constant action while varying γ

We can see that the average cost for the optimal solution is less than the solution obtained by picking a random action and a constant action. This implies that our model is performing better than the rest of them irrespective of the value of γ. As already mentioned, picking a constant action of 0.5 will do better than picking a random action, we can observe the same in the below plot 5.4.

CHAPTER 6

Summary

6.1 Conclusions

In this chapter we conclude our work by providing a short summary of the work we have done in this project. We started with the problem of fair co-existence of LTE-U and Wi-Fi in to the interference problem incurred by LTE when used in the unlicensed bands. So, to overcome this issue, we proposed a novel method in which we divided the total time into two 2 parts and allocating each of the time to LTE and Wi-Fi respectively by taking the concept of Almost Blank Sub Frames (ABSF) into account. A system model was developed taking into consideration a real time scenario and the cost functions were formed. Then we formed the optimization problem, and solved for the optimal solution. The solutions and results are reported in chapter 5.

In addition to solving the optimization problem, we also formed a Game theoretic framework for the same problem, and solved for the Nash Equilibrium solution. The Nash equilibrium solution may not be the optimal solution but it is a more realistic solution since the players (here the LTE BS's) might not cooperate for social optimal solution by being greedy. While solving the Nash Equilibrium solution in the optimization framework, as mentioned in chapter 3, we obtained the solution using ε Nash Equilibrium, there will be some loss in net utility because obtaining the exact solution might take a long amount of time resulting in a unrealizable implementation.

Since calculating the optimal solution(estimating the fractional air time) in a very dense deployment can take a long time and with the advancements of machine learning technologies now a days, we presented a simple model which uses Q-learning, a model free reinforcement learning which does not need an explicit model to know the transition probabilities. We made use of the concept of M/M/1 queue for building the Markov Decision Process problem since any reinforcement learning problem can be modelled to an MDP. We trained our model initially updating the Q matrix, then tested the model with the final Q matrix by comparing with 2 other naive models.

6.2 Future Work

Regarding the System model, working on improving the model to a more realistic way such as including small cell base stations, etc would be of a great use because the whole performance of the model whether it is doing good or really bad depends solely on the problem of how we frame the system model. So improving these parts would be a nice start for a future work. Also, while performing the optimization, checking whether any new and better constraints can be imposed on the model to improve the system performance is one more interesting work to start with.

Also, presently, our model just gives the optimal fraction of air time for LTE and Wi-Fi, but even after getting the optimal air times, the way we schedule the air times would have a significant role in the system performance. So, scheduling the obtained air times is one major future scope to work.

In the Q learning part, present we have considered and incorporated only one channel. Since the results regarding the Q-learning are promising and satisfactorily, extending this model to more number of channel which would be more appealing could be considered as a work for future scope.

In our Q-learning model, we presently fixed the frame time to a constant, and then worked on the optimal air times for that particular frame time. But can we do something better? Can our model also learn the optimal frame time and provide it in the solution along with the fractional air time(duty cycle time)? This too would be an interesting as well as an important thing to work with.

APPENDIX A

Bron-kerbosch algorithm

Bron kerbosch algorithm is an optimized algorithm used for finding all the maximal cliques in a graph.

A **clique** in an undirected graph $G = (V, E)$ can be defined as a subset of the vertex set $C \subset V$ such that for every two vertices in C, there exists am edge connecting the two.

A **maximum clique** can be defined as the clique of the largest possible size in a graph

A **maximal clique** can be defined as the clique that cannot be extended by adding one more adjacent vertex.

Following are the sets involved and their functions while computing the cliques using the Bron-kerbosch algorithm.

R: set of vertices that construct the Maximal Clique

P: set of adjacent vertices to every vertex currently in **R**, it is possible that the vertices in **P** may be selected to set **R** for forming a maximal clique

X: set of vertices can not construct Maximal Clique since they are already in some clique

Below is pseudo code for the Bron-kerbosch algorithm with pivot (version 2)

Pseudocode 3: Bron-kerbosch algorithm for maximal cliques

```
Initialization:R= { },P={V},X={ }
Function BronKerbosch (P,R,X)
    if P ∪ X == {} then
        return R as maximal clique
    choose pivot vertex u in P ∪ X
    for each vertex v in P \ neighbors(u) do
        BronKerbosch(P ∩ neighbors(v),R ∪ v, X ∩ neighbors(v))
        P ← P \ v
        X ← X ∪ v
```

APPENDIX B

Q-learning pseudo code

The pseudo code for the learning part in the Q-learning is given below. The data, result, and the initialisation part are given below and the algorithm can be found in the next page.

Algorithm 4: Q-learning algorithm

Data: $\lambda_L, \lambda_W, \mu_L, \mu_W, \beta = (0.1, 0.2, ..0.9)$
Result: actions,i.e duty cycle times at every completion of frame

Initialization:
N = 0; M = 0
Q = zeros(100,100,len(actions)) // initializing Q matrix to 0
action_num = zeros(100,100,len(actions)) // counting the no of times a particular action is picked at a given state and prev action
eps = 0.2 //exploration constant
flag = 1 // setting flag=1 i.e LTE time
t = 0 //starting from time = 0
time = 100000 // setting training time
frame_time = 2 // setting frame time = 2
contention_const = 0.1 // for counter
lte_count = 0
wifi_count = 0
start with a random action
update action_num matrix
calculate events list

```
while t ≤ time do
    if flag == 1 then
        consider lte gen, wifi gen, lte service time, duty cycle time as the events
          and find the min event
        update t by adding min time
        if index==lte gen then
            increase lte count
            gen new lte user
            gen new service time if it is first LTE user
        else if index == wifi gen then
            increase wifi count
            gen new wifi user
            gen counter for new user
            find contention time
        else if index == lte service time then
            decrease LTE count
            if LTE count > 0 gen new service time for next LTE else make it infinity
        else if index == duty cycle time then
            make flag = 0
            find contention time if M > 0 else make it infinity
    else if flag ==0 then
        consider lte gen, wifi gen, wifi service time, contention time and frame time
          as events and fin min event
        update t by adding min time
        update events by subtracting min time
        if index==lte gen then
            increase lte count
            gen new lte user
            gen new service time if it is first LTE user
        else if index == wifi gen then
            increase wifi count
            gen new wifi user
            gen counter for new user
            find contention time
        else if index == wifi service time then
            remove the wifi from counter array which has contention times
            reduce Wi-Fi count
            if Wifi count > 0 find new contention time and make wifi service time
              infinity
            else make contention time and wifi service time as infinity
        else if index == contention time then
            update counter by subtracting contention time
            gen new wifi service time
            make contention time infinity
        else if index == frame time then
            make flag = 1
            take action based on prevN, prevM, action_num, prev_action,M,N
            update prev_action, action_num,prevN,prevM
```

APPENDIX C

Q-learning take action function

Pseudocode 5: Function for take action

Function `take action` (*prevN,prevM,Q,N,M,action_num,prev_action,*β)

α = 1 / action_num(prevN+1,prevM+1,prev_action)

γ = 0.9

find min value and index for Q(N+1,M+1) row

λ = 1

$\mathrm{Q}_{t+1}(s_t, a_t) \leftarrow (1-\alpha)\ Q_{t+1}(s_t, a_t) + \alpha\ (r + \gamma \max_{a \in \beta} Q_t(s_{t+1}, a))$

choose a random num between 0 and 1

if *rand()* < **then**

present_action = randi(1,len(β)) // exploring

else

,

β is the set of all possible actions present_action = index

REFERENCES

1. **F. M. Abinader**, **E. P. L. Almeida**, **F. S. Chaves**, **A. M. Cavalcante**, **R. D. Vieira**, **R. C. D. Paiva**, **A. M. Sobrinho**, **S. Choudhury**, **E. Tuomaala**, **K. Doppler**, and **V. A. Sousa** (2014). Enabling the coexistence of lte and wi-fi in unlicensed bands. *IEEE Communications Magazine*, **52**(11), 54–61. ISSN 0163-6804.

2. **E. Almeida**, **A. M. Cavalcante**, **R. C. D. Paiva**, **F. S. Chaves**, **F. M. Abinader**, **R. D. Vieira**, **S. Choudhury**, **E. Tuomaala**, and **K. Doppler**, Enabling lte/wifi coexistence by lte blank subframe allocation. *In 2013 IEEE International Conference on Communications (ICC)*. 2013. ISSN 1550-3607.

3. **bron kerbosch** (2019). Bron–kerbosch algorithm — Wikipedia, the free encyclopedia. URL `https://en.wikipedia.org/w/index.php?title=Bron%E2%80%93Kerbosch_algorithm&oldid=889993941`. [Online; accessed 10-Nov-2018].

4. **C. Cano** and **D. J. Leith**, Coexistence of wifi and lte in unlicensed bands: A proportional fair allocation scheme. *In 2015 IEEE International Conference on Communication Workshop (ICCW)*. 2015. ISSN 2164-7038.

5. **C. Cano**, **D. Lopez-Perez**, **H. Claussen**, and **D. J. Leith** (2016). Using lte in unlicensed bands: Potential benefits and coexistence issues. *IEEE Communications Magazine*, **54**(12), 116–123. ISSN 0163-6804.

6. **M. Chen**, **U. Challita**, **W. Saad**, **C. Yin**, and **M. Debbah** (2017). Machine learning for wireless networks with artificial intelligence: A tutorial on neural networks. *CoRR*, **abs/1710.02913**. URL `http://arxiv.org/abs/1710.02913`.

7. **Z. Guan** and **T. Melodia**, Cu-lte: Spectrally-efficient and fair coexistence between lte and wi-fi in unlicensed bands. *In IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*. 2016.

8. **Jeffrey Wildman** (2011). Bron-kerbosch maximal clique finding algorithm. URL `https://in.mathworks.com/matlabcentral/fileexchange/30413-bron-kerbosch-maximal-clique-finding-algorithm`. [Online; accessed 13-November-2018].

9. **T. Nihtilä**, **V. Tykhomyrov**, **O. Alanen**, **M. A. Uusitalo**, **A. Sorri**, **M. Moisio**, **S. Iraji**, **R. Ratasuk**, and **N. Mangalvedhe**, System performance of lte and ieee 802.11 coexisting on a shared frequency band. *In 2013 IEEE Wireless Communications and Networking Conference (WCNC)*. 2013. ISSN 1525-3511.

10. **M. d. S. Pedro**, **J. Vicente, A. de Sousa**, **J. Fuad, M. Abinader**, and **M. d. C. N. José**, Dm-csat: a lte-u/wi-fi coexistence solution based on reinforcement learning. 2019.

11. **R. Ratasuk**, **M. A. Uusitalo**, **N. Mangalvedhe**, **A. Sorri**, **S. Iraji**, **C. Wijting**, and **A. Ghosh**, License-exempt lte deployment in heterogeneous network. *In 2012 International Symposium on Wireless Communication Systems (ISWCS)*. 2012. ISSN 2154-0225.

12. **N. Rupasinghe** and **Güvenç**, Reinforcement learning for licensed-assisted access of lte in the unlicensed spectrum. *In 2015 IEEE Wireless Communications and Networking Conference (WCNC)*. 2015. ISSN 1525-3511.

13. **S. S. Sagari**, Coexistence of lte and wifi heterogeneous networks via inter network coordination. *In Proceedings of the 2014 Workshop on PhD Forum*, PhD forum '14. ACM, New York, NY, USA, 2014. ISBN 978-1-4503-2940-8. URL http://doi.acm.org/10.1145/2611166.2611167.

14. **Y. Su**, **X. Du**, **L. Huang**, **Z. Gao**, and **M. Guizani** (2018). Lte-u and wi-fi coexistence algorithm based on q-learning in multi-channel. *IEEE Access*, **6**, 13644–13652. ISSN 2169-3536.

15. **Y. Su**, **L. Huang**, **X. Du**, **A. Mohamed**, **H. Chi**, **M. Guizani**, and **I. Academy**, Q-learning-based lte-u and wifi coexistence algorithm for wireless healthcare systems. 2018*a*.

16. **Y. Su**, **L. Huang**, **S. Ke**, and **Y. Tang**, Spectrum multiplexing for lte-u and wi-fi in heterogeneous network. 2018*b*.

17. **X. Wang**, **S. Mao**, and **M. X. Gong** (2017). A survey of lte wi-fi coexistence in unlicensed bands. *GetMobile: Mobile Comp. and Comm.*, **20**(3), 17–23. ISSN 2375-0529. URL http://doi.acm.org/10.1145/3036699.3036705.

18. **S. Yun** and **L. Qiu**, Supporting wifi and lte co-existence. *In 2015 IEEE Conference on Computer Communications (INFOCOM)*. 2015. ISSN 0743-166X.