

Handover Optimisation in 4G Systems

Iain Cuthbertson

201015895

Computer & Electronic Systems

University of Strathclyde

April 2014

I hereby declare that this work has not been submitted for any other degree/course at this University or any other institution and that, except where reference is made to the work of other authors, the material presented is original and entirely the result of my own work at the University of Strathclyde under the supervision of Dr. Robert C. Atkinson.

Signature _____

Abstract

With more and more customers using mobile communications it is important for the service providers to give their customers the best Quality-of-Service (QoS) they can. Many providers have taken to improve their networks and make them more appealing to customers. One such improvement that providers can give to their customers is to improve the reliability of their network meaning that customers calls are less likely to be dropped by the network.

This dissertation explores improving the reliability of a 4G network by optimising the parameters used in handovers. The process of handover within mobile communication networks is very important and allowing for users to move around freely while still staying connected to the network. The parameters used in the handover process are the Time-to-Trigger (TTT) and Hysteresis (hys). These parameters are used to determine where a base station is better than the serving base station by enough to warrant a handover taking place. The challenge in optimising the handover parameters is that there is a fine balance that needs to be struck between calls being dropped due to a handover failing and the connection switching back and forth between two base stations, unnecessarily, wasting the network's resources. The approach taken is to use a machine learning technique known as Q-Learning to optimise the handover parameters by generating a policy that can be followed to adjust the parameters as needed.

Acknowledgements

Contents

Abstract	ii
Acknowledgements	iii
Nomenclature	1
List of Figures	3
List of Tables	4
1 Introduction	5
1.1 Project Objectives	5
1.2 Related Work	5
2 LTE	6
2.1 Network Structure	6
2.2 Self-Organising Network	7
2.3 Handover	8
2.3.1 Parameters	8
2.3.2 Procedure	10
3 Machine Learning	13
3.1 Reinforcement Learning	14
3.2 Q-Learning	15
3.2.1 Example	16
4 Simulation Design	17
4.1 Discrete Event Simulation	17
4.2 Mobility Model	18
4.3 Propagation Model	19
4.4 Simulation Testing	21

5	Handover Parameter Optimisation	22
5.1	Approach	22
5.2	Results	23
6	Future Work	24
7	Conclusions	25
	Bibliography	26
	Appendices	28
A	System Design and Evaluation	29

Nomenclature

3G	Third Generation
4G	Fourth Generation
AI	Artificial Intelligence
dB	Decibels
DES	Discrete Event Simulation
DL	Downlink
eNodeB	Evolved Node B
EPC	Evolved Packet Core
E-UTRAN	Evolved UMTS Terrestrial Radio Access Network
HSS	Home Subscriber Server
hys	Hysteresis
LTE	Long Term Evolution
MME	Mobility Management Entity
P-GW	Packet Data Network Gateway
PCell	Primary Cell
QoS	Quality-of-Service
RSS	Received Signal Strength
SCell	Secondary Cell
S-GW	Serving Gateway
TTT	Time-to-Trigger

UE	User Equipment
UL	Uplink
UMTS	Universal Mobile Telecommunications System

List of Figures

2.1	Graph of TTT values.	10
3.1	Image of Q-Learning process.	16

List of Tables

2.1	Table of the different LTE hys values.	9
2.2	Table of the different LTE TTT values.	10
2.3	Table of the different LTE Trigger types and their criteria.	11

Chapter 1

Introduction

1.1 Project Objectives

1.2 Related Work

Chapter 2

LTE

Mobile communications is on to its Fourth Generation (4G) of network infrastructure with Long Term Evolution (LTE). This network infrastructure was developed by 3GPP and is an improvement upon Universal Mobile Telecommunications System (UMTS), which is a Third Generation (3G) network. LTE has Downlink (DL) speeds of up to 300Mbit/s and Uplink (UL) speeds of 75Mbit/s with a varying bandwidth that ranges from 1.4MHz to 20MHz . The development of faster downloads speed was driven by the consumers want for better quality images, faster Internet browsing and smoother video streaming. [1]

2.1 Network Structure

The structure of the LTE network can be broken down into 3 main parts, the User Equipment (UE), the Evolved UMTS Terrestrial Radio Access Network (E-UTRAN) and the Evolved Packet Core (EPC). The UE can simply be considered as a standard mobile phone or smartphone. The purpose of the E-UTRAN is to connect a UE to the EPC and is made up of just one component, the Evolved Node B (eNodeB) or simply put base station. Figure (place holder) shows an illustration of the E-UTRAN. An UE will only communicate with one eNodeB at any time and this eNodeB is known as the Serving eNodeB. An eNodeB proves two main functions within the network; the first is to send all the radio traffic for an UE on the DL as well as receiving any traffic sent from the UE on the UL. The second function of the eNodeB is to control low-level operations such as handovers as well as provide the signalling for such operations. Due to the eNodeB's having the added complexity of controlling operations such as handovers it moves more of the processing from the core network to the edge of the network, reducing the latency of decisions being made. Every eNodeB is connected to the EPC using the S1 interface. They can also be connected to other eNodeB's by the X2 interface. This interface is mainly used for signalling and forwarding data

from a serving eNodeB to a neighbouring eNodeB during a handover.

Figure place holder for E-UTRAN image.

An illustration of an EPC can be seen in Figure (place holder). It can be seen that the EPC is made up of four main components, they are the: Mobility Management Entity (MME), Home Subscriber Server (HSS), Packet Data Network Gateway (P-GW) and Serving Gateway (S-GW). The MME provide the high-level operations for a UE such as security and managing non-radio communication data streams as well as controlling other elements within the EPC. There are very few MME's within the LTE network, generally assigned to a certain geographical region. A UE will be assigned to a single MME known as the Serving MME. The HSS is a central database that holds all the information about the network subscribers such as authentication and billing information. The P-GW is the bridge between the EPC and other packet data networks such as the Internet. The P-GW uses the SGi interface to exchange data with outside networks, such as the network operator's servers. The S-GW function is to forward data to and from the eNodeB's to P-GW, this means that the S-GW effectively acts like a router. Much like the MME a UE will be assigned to one S-GW and there will be very few within the network as a whole.

Figure place holder for EPC image. [1, 2]

2.2 Self-Organising Network

LTE, just like any other mobile network, needs to be managed. Due to LTE being more complex than its predecessors the managed of the network has also become more complex. The use of automation can simplify management of a network greatly. A technology that can be employed in LTE is that of a Self-Organising Network. This system provides three main functions, Self-configuration, Self-optimisation and Self-healing. [3, 4]

- **Self-configuration** allows for newly installed eNodeB to be configured automatically with the basic parameters for operation.
- **Self-optimisation** is a process where measurements from the UE and eNodeB along with performance measurements are used to optimise the network to make it perform better.
- **Self-healing** is designed to detect and identify failures within the network. After the self-healing system detects a failure it aims to recover from the failure without the need for human interaction with the system.

2.3 Handover

The process of handover is very important in mobile telecommunications. It involves moving the resource allocation for a mobile phone or a piece of UE from one base station to another. This process is used to provide more Quality-of-Service (QoS) to customers by allowing them to continue to use provided services even after moving out of range of the original serving base station. To keep with the QoS it is important that handovers are done fast, have little-to-no disruption to the users experience and are completed with a very high success rate. If a handover is unsuccessful it is likely that an on going call will be dropped due to there not being enough resources available on a base station, known as an eNodeB in LTE, or the received signal strength to the UE drops below a certain threshold needed to maintain the call. Handovers are stated to take roughly 0.25 seconds to complete after the decision has been made for a handover to take place [5].

2.3.1 Parameters

In LTE there are two main parameters that are used in the handover process. These parameters are the Time-to-Trigger (TTT) and Hysteresis (hys). The hys is used to define how much better the Received Signal Strength (RSS) of a neighbouring base station must be than the serving base station for a handover to be considered. The values of hys are defined in Decibels (dB) and range from 0 to 10 dB in 0.5 dB increments, this results in there being 20 different values of hys. The full range of hys values can be seen in Table 2.1.

Parameter	Value(dB)
hys	0.0
	0.5
	1.0
	1.5
	2.0
	2.5
	3.0
	3.5
	4.0
	4.5
	5.0
	5.5
	6.0
	6.5
	7.0
	7.5
	8.0
	8.5
	9.0
	9.5
	10.0

Table 2.1: Table of the different LTE hys values.

The TTT is a length of time, defined in seconds, that is used to define how long a neighbouring base station must look better than the serving base station for. There are 15 different values of TTT ranging from 0 to 5.12 seconds. Unlike with hys the TTT values do not increment linearly, instead they increment exponential with smaller increases at the lower values and bigger increases at the larger values. The full list of TTT values can be seen in Table 2.2 and a graph of how the TTT values increase can be seen in Figure 2.1.

Parameter	Value(s)
TTT	0.0
	0.04
	0.064
	0.08
	0.1
	0.128
	0.16
	0.256
	0.32
	0.48
	0.512
	0.64
	1.024
	1.280
	2.56
	5.12

Table 2.2: Table of the different LTE TTT values.

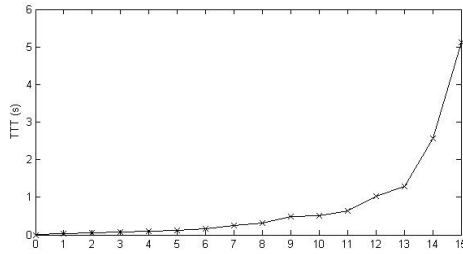


Figure 2.1: Graph of TTT values.

There are 336 different combinations of TTT and hys values. Having such a large range of combinations means that pairs of values can mean that a neighbouring eNodeB has to be better by a large hys but for a small TTT or vice-versa. This makes for an interesting dynamic for which pairs of values will work the best in any given environment.

2.3.2 Procedure

In LTE there are eight different triggers defined for initiating handovers within LTE. Table 2.3 shows different trigger events and how they are defined [6].

Event Type	Trigger Criteria
A1	Serving becomes better than a threshold.
A2	Serving becomes worse than a threshold.
A3	Neighbour becomes offset better than Primary Cell (PCell).
A4	Neighbour becomes better than threshold.
A5	PCell becomes worse than threshold1 and neighbour becomes better than threshold2.
A6	Neighbour becomes offset better than Secondary Cell (SCell).
B1	Inter RAT neighbour becomes better than threshold.
B2	PCell becomes worse than threshold1 and inter RAT neighbour becomes better than threshold2.

Table 2.3: Table of the different LTE Trigger types and their criteria.

Out of the eight triggers the A3 event is the most common and it is defined that a neighbouring eNodeB must give the UE better Received Signal Strength (RSS) by an amount defined by the hys, for a length of time defined by the TTT. An illustration of how the signals from the serving and neighbouring eNodeBs look in an A3 event can be seen in Figure (place holder). The A3 event can also be represented by the following equation:

$$RSS_{Neighbouring} + hys > RSS_{Serving} \quad (2.1)$$

Place holder for image of event A3

When a handover event is triggered a measurement report is sent from the UE to the serving eNodeB. The measurement report contains the information required for the serving eNodeB to make a decision on whether to initiate the handover or not. If the handover is triggered the serving eNodeB will signal to the neighbouring eNodeB that is to become the new serving eNodeB. If the serving eNodeB decides not to initiate the handover then no more action is taken until another handover event is triggered.

The full, high-level, procedure for a LTE handover is as follows:

1. If a neighbouring eNodeB is found to be better than the serving eNodeB a measurement report is sent by the UE to the serving eNodeB.
2. The serving eNodeB considers the information in the measurement report and decides whether or not a handover should take place.
3. If it is decided that a handover should take place then a message is sent to the neighbouring eNodeB to prepare resources for the UE.

4. Once the resources are ready for the UE the new serving eNodeB sends a message to the old eNodeB to release the resources it previously had for the UE
5. Finally a message is sent to the MME to finalise the handover process.

Chapter 3

Machine Learning

Machine learning is a form of Artificial Intelligence (AI) that involves designing and studying systems and algorithms with the ability to learn from data. This field of AI has many applications within research (such as system optimisation), products (such as image recognition) and advertising (such as adverts that use a users browsing history). There are many different paradigms that machine learning algorithms use. Algorithms can use training sets to train an algorithm to give appropriate outputs; other algorithms look for patterns in data; while others use the notion of rewards to find out if an action could be considered correct or not. [7] Three of the most popular types of machine learning algorithms are:

- **Supervised learning** is where an algorithm is trained using a training set of data. This set of data includes inputs and the known outputs for those inputs. The training set is used to fine-tune the parameters in the algorithm. The purpose of this kind of algorithm is to learn a general mapping between inputs and outputs so that the algorithm can give an accurate result for an input with an unknown output. This type of algorithm is generally used in classification systems.
- **Unsupervised learning** algorithms only know about the inputs they are given. The goal of such an algorithm is to try and find patterns or structure within the input data. Such algorithm would be given inputs and any patterns that are contained would become more and more common the more inputs the algorithm is given.
- **Reinforcement learning** uses an intelligent agent to perform actions within an environment. Any such action will yield a reward to the agent and the agent's goal is to learn about how the environment reacts to any given action. The agent then uses this knowledge to try and maximise its reward gains.

3.1 Reinforcement Learning

In reinforcement learning an intelligent agent is learning what action to do at any given time to maximise the notion of a reward. In the beginning the agent has no knowledge of what action it should take from any state within the learning environment. It must instead learn through trial and error, exploring all possible actions and finding the ones that perform the best.

The trade-off between exploration and exploitation is one of the main features of reinforcement and can greatly affect the performance of a chosen algorithm. A reinforcement learning algorithm must contemplate this trade-off of whether to exploit an action that resulted in a large reward or to explore other actions with the possibility of receiving a greater reward.

Another main feature of reinforcement learning is that the problem in question is taken into context as a whole. This is different from other types of machine learning algorithms, as they will not considered how the results of any sub-problems may affect the problem as a whole.

The basic elements required for reinforcement learning is as follows:

- A Model (M) of the environment that consists of a set of States (S) and Actions (A).
- A reward function (R).
- A value function (V).
- A policy (P).

The model of the environment is used to mimic the behaviour of the environment, such as predicting the next state and reward from a state and taken action. Models are generally used for planning by deciding what action to take while considering future rewards.

The reward function defines how good or bad an action is from a state. It is also used to define the immediate reward the agent can expect to receive. Generally a mapping between a state-action pair and a numerical value is used to define the reward that the agent would gain. The reward values are used to define the policy where the best value of state-action pair is used to define the action to take from a state.

While the reward function defines the immediate reward that can be gained from a state, the value function defines how good a state will be long-term. This difference can create possible conflicts of interest for an agent; so while its goal is to collect as much reward as possible, it has to weigh up the options of picking a state that may provide a lot of up front reward but not a lot of future reward

against a state with a lot of future reward but not a lot of immediate reward. This trade-off is similar to that of exploration versus exploit as it can define how successful a reinforcement algorithm is.

The policy is a mapping between a state and the best action to be taken from that state at any given time. Policies can be simple or complex; with a simple policy consisting of a lookup table, while more complex policies can involve search processes. In general most policies begin stochastic so that the agent can start to learn what actions are more optimal. [8]

3.2 Q-Learning

Q-Learning is a type of reinforcement learning algorithm where an agent tries to discover an optimal policy from its history of interactions from within an environment. What makes Q-Learning so powerful is that it will always learn the optimal policy for a problem regardless of the policy (which action a to take from a state s) it follows while learning as long as there is no limit on the number of times the agent can try an action. Due to this ability to always learn the optimal policy, Q-Learning is known as an Off-Policy learner. The history of interactions of an agent can be shown as a sequence of State-Action-Rewards:

$$< s_0, a_0, r_1, s_1, a_1, r_2, s_2, a_2 \dots >$$

This can be described as the agent was in State 0, did Action 0, received Reward 0 and transitioned into State 1; then did Action 1, received Reward 1 and transitioned into State 2; and so on.

The history of interactions can be treated as a sequence of experiences, with each experience being a tuple.

$$< s, a, r, s' >$$

The meaning of the tuple is that the agent was in State s , did Action a , received Reward r and transitioned in State s' . The experiences are what the agent uses to determine what the optimal action to take is at a given time.

The basic process of a Q-Learning algorithm can be seen in Figure 3.1. The general process requires that the learning agent is given a set of states, a set of actions, a discount factor γ and step size α . The agent also keeps a table of Q-Values, denoted by $Q(s, a)$ where s is a state and a is an action from that state. A Q-Value is also an average of all the experiences the agent has with a specific state-action pair. This allows for good and bad experiences to be averaged out to give a reasonable estimation of the actual value of state-action pair. The Q-Values are defined by Equation 3.1 where α is the step size which specifies how

much the new Q-Value is averaged with the old one, γ is the discount factor which specifies how much the agent considers the possible future rewards it will gain and the possible future rewards ($\max_{a'} Q(s', a')$) is the maximum of the Q-Values of all possible state-actions pairs from the action selected.

$$Q[s, a] = Q[s, a] + \alpha(r + \gamma \max_{a'} Q[s', a'] - Q[s, a]) \quad (3.1)$$

The table of Q-Values can either be initialised as empty or with some values pre-set to try and lead the agent to a specific goal state. Once the agent has initialised these parameters it observes the starting state. The starting state can either be chosen by random or be a pre-determined start state for the problem. The agent will then choose an action. Actions are chosen either stochastically or by a policy. Once an action has been chosen the agent will carry out the action and receive a reward. This reward is used to update the table of Q-Values using Equation 3.1. Finally the agent moves into the new state and repeats until termination; which can be either when the agent discovers a goal state or after a certain number of actions have been taken.

```

1: controller Q-learning(S, A, γ, α)
2:   Inputs
3:     S is a set of states
4:     A is a set of actions
5:      $\gamma$  the discount
6:      $\alpha$  is the step size
7:   Local
8:     real array Q[S, A]
9:     previous state s
10:    previous action a
11:    initialize Q[S, A] arbitrarily
12:    observe current state s
13:    repeat
14:      select and carry out an action a
15:      observe reward r and state s'
16:       $Q[s, a] \leftarrow Q[s, a] + \alpha (r + \gamma \max_{a'} Q[s', a'] - Q[s, a])$ 
17:       $s \leftarrow s'$ 
18:    until termination

```

Figure 3.1: Image of Q-Learning process.

After a Q-Learning algorithm has finished exploring the model of the environment it creates a policy. The policy is generated by searching across all actions for a state and finding the next state with the greatest value. The policy is therefore a lookup table that maps a state with the best possible next state. The policy created can then be used to solve the problem that the Q-Learning agent was exploring. [9]

3.2.1 Example

Chapter 4

Simulation Design

The simulation is a very important part of the project. It is required to provide the basic functionality of a LTE network. For simplicity the simulation was broken down into two main components; the mobile (UE) and the base station (eNodeB). Due to the project revolving around the handover process in LTE it made sense for the two main components on the simulation to be the mobile and base station as it is the mobile the triggers the measurement report and the base station that makes the decision on whether a handover should take place or not. Since the A3 event trigger is the most common I decided that it would be the only trigger implemented in the simulation to reduce the complexity within the simulation.

It was required for the mobile to be able to move freely around a group of base stations. It was decided that this movement should be random because if any machine learning algorithm can handle random movement then it should also be able to handle regimented movement. The movement that the mobile follows is defined by a Mobility Model and the choice of mobility model is explained in Section 4.2.

In wireless communications the received signal strength from a transmitter degrades the further away from the transmitter the receiver is. A propagation model can be used to define the way in which the signal strength degrades. The propagation model is very important to the simulation as it will defined how far away from a base station the mobile can be without dropping the call. A comparison and explanation of the choice of propagation model can be seen in Section 4.3.

4.1 Discrete Event Simulation

The concept of time is very important in real world simulations. A popular method for creating this concept of time is Discrete Event Simulation (DES).

It works on the basis of a scheduler where events, to be processed at a certain time, are passed to the scheduler and the scheduler then passes the event on to be processed. The way that DES creates the concept of time is by allowing events passed in the scheduler to have a time that it should be processed at, the scheduler will keep all the events that are currently still to be processed in an ordered list, the scheduler will then "jump" to the time the first event in the list is to be processed at and passes the event on to be processed. The process of "jumping" to the next time that the first event in the list is to be processed at is what gives DES the concept of time.

Another advantage to Discrete Event Simulation is that the passing of events from the scheduler not only gives a concept of time but it also allows for a message to be sent with the event to tell a different part of the simulation what to do at a given time. This message can also sent parameters for the other part of the simulation to use as well.

4.2 Mobility Model

A mobility model defines the way in which an entity will move. For the purposes of the simulation the mobility model used needed to be random in nature. After some reason it was decided that mobility model to be used in the simulation would either be the Random Direction or Random Waypoint model.

The Random Direction Model is defined as follows:

1. Select a direction randomly between 0 and 355 degrees.
2. Select a random speed to move at.
3. Select a random duration to move for.
4. Move in the selected direction at the selected speed for the selected duration.
5. Repeat until termination.

An illustration of the movement given by the Random Direction Model can be seen in Figure (placeholder).

place holder for image of random direction

The Random Waypoint Model is defined as follows:

1. Randomly select the co-ordinates for a point within the environment.
2. Select a random speed to move at.
3. Select a random length of time to pause for when the destination is reached.

4. Move towards the selected co-ordinates at the selected speed
5. Pause for the randomly selected length of time.
6. Repeat until termination.

An illustration of the movement given by the Random Waypoint Model can be seen in Figure (placeholder).

place holder for image of random waypoint

It was decided that the Random Direction Model would be used in the simulation because the Random Waypoint Model has the problem that it is possible to select the co-ordinates of a point very close to where you begin and then pause for a long period time. The possibility of that happening isn't desired within the simulation. Random Direction does not have this problem and it is also possible to set boundaries on the parameters to make sure that a minimum distance is travelled. [10]

4.3 Propagation Model

A propagation model defines how the received signal from a transmitter decays the further from the transmitter you are. There are many different models available, all with different functions and purposes. After some research three models were considered; the Okumura-Hata Model, the Egli Model and the Cost231-Hata Model.

The Okumura-Hata model is very popular for simulating transmissions in built up areas. Equations 4.1, 4.2 and 4.3 show the formulas for the model.

$$L_u = 69.55 + 26.16\log_{10}f - 13.82\log_{10}h_B - C_h + [44.9 - 6.55\log_{10}h_B]\log_{10}d \quad (4.1)$$

$$C_H = 0.8 + (1.1\log_{10}f - 0.7)h_m - 1.56\log_{10}f \quad (4.2)$$

$$C_H = \begin{cases} 8.29(\log_{10}(1.54h_M))^2 - 1.1 & \text{if } 150 \leq f \leq 200 \\ 3.2(\log_{10}(11.75h_M))^2 - 4.97 & \text{if } 200 < f \leq 1500 \end{cases} \quad (4.3)$$

Where:

- L_u is the path loss (dB).
- H_B is the height of the base station antenna 30 to 200m.
- H_R is the height of the mobile antenna 1 to 10m.
- f is the frequency of the transmission 150 to 1500MHz.

- C_H is the antenna correction factor.
- d is the distance between the base station and the mobile 1 to 20km.

The Egli Model was another model that was considered for the simulation. Equation 4.4 shows the formula for the model.

$$P_{R50} = 0.668G_B G_M \left[\frac{h_B h_M}{d^2} \right]^2 \left[\frac{40}{f} \right]^2 P_T \quad (4.4)$$

Where:

- P_{R50} is the path loss (dB).
- P_T is the power if the transmitter (W).
- G_B is the absolute gain of the base station antenna.
- G_M is the absolute gain of the mobile antenna.
- h_B is the height of the base station antenna (m).
- h_M is the height of the mobile antenna (m).
- d is the distance between the base station and the mobile (m).
- f is the frequency of the transmission up to 3000MHz.

The Cost231-Hata model is an extension of the Okumura-Hata to work for frequencies between 1.5 GHz and 2 GHz. The formulas for this model can be seen in Equations 4.5, 4.6 and 4.7.

$$L = 46.3 + 33.9 \log f - 13.82 \log h_B - a(h_R) + [44.9 - 6.55 \log h_B] \log d + C \quad (4.5)$$

$$a(h_R) = (1.1 \log f - 0.7) h_R - (1.58 \log f - 0.8) \quad (4.6)$$

$$C = \begin{cases} 0dB & \text{for medium cities and suburban areas} \\ 3dB & \text{for metropolitan areas} \end{cases} \quad (4.7)$$

Where:

- L is the path loss (dB).
- f is the frequency of the transmission 1500 to 2000MHz.
- h_B is the height of the base station antenna 30 to 200m.
- h_M is the height of the mobile antenna 1 to 10m.

- d is the distance between the base station and the mobile 1 to $20km$.
- $a(h_R)$ is the antenna correction factor.

After comparing the three models above it was decided that the Cost231-Hata Model would be the one used in the simulation due to it working with frequencies up to $2000MHz$ (which is the minimum operating frequency of LTE) unlike the Okumura-Hata model which only works up to $1500MHz$. The Cost231-Hata Model was also picked over the Egli Model because the Egli Model used more parameters that would add more complexity to the simulation. [11, 12]

4.4 Simulation Testing

Chapter 5

Handover Parameter Optimisation

5.1 Approach

The approach taken for optimising the handover parameters in LTE uses a Q-Learning algorithm based on the process given in Section 3.2. In the approach the model of the environment has a state for every combination of TTT and hys; giving a total number of states of 336. An action within the model can move to any other state that is different by one of the following changes to the handover parameters:

1. A single increase of TTT.
2. A single increase of hys.
3. A single increase of both TTT and hys.
4. A single decrease of TTT.
5. A single decrease of hys.
6. A single decrease of both TTT and hys.

Having the actions only change the parameters by one value each time not only allows for refined optimisation of the parameters but it also makes sure that no large changes can suddenly happen.

Due to the nature to the kind of problem that is being solved, the reward gained by an action is dynamic and is likely to be different each time it is taken. Rewards are based on the number of drop and ping-pongs accumulated in the simulation for current state in the environment model. The reward is given to the agent and the Q-Value for that state is updated just before agent selects the next

action to take. The agent selects a new action in discrete time steps, this allows for the simulation to run for fixed periods of time with TTT-hys pairs specified by a state in the environment model.

After the agent has been given enough time to try every action at least once the Q-Learning is terminated and a policy is generated. This policy can then be used to attempt to optimise the handover parameters by changing the TTT and hys values after a call is dropped or the connection ping-pongs between base stations.

5.2 Results

Chapter 6

Future Work

Chapter 7

Conclusions

Bibliography

- [1] C. Cox, *An introduction to LTE: LTE, LTE-advanced, SAE and 4G mobile communications*. John Wiley & Sons, 2012.
- [2] 3GPP TS 23.002 V11.6.0, *Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); LTE; Network architecture (Release 11)*, June 2013.
- [3] S. Feng and E. Seidel, “Self-organizing networks (son) in 3gpp long term evolution,” *Nomor Research GmbH, Munich, Germany*, 2008.
- [4] 3GPP TR 36.902 V9.3.1 Release 9, *LTE; Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Self-configuring and self-optimizing network (SON) use cases and solutions (Release 9)*, May 2011.
- [5] T. Jansen, I. Balan, J. Turk, I. Moerman, and T. Kurner, “Handover parameter optimization in lte self-organizing networks,” in *Vehicular Technology Conference Fall (VTC 2010-Fall), 2010 IEEE 72nd*, pp. 1–5, IEEE, 2010.
- [6] 3GPP TS 36.331 V10.7.0, *LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification (Release 10)*, November 2012.
- [7] E. Alpaydin, *Introduction to Machine Learning*. MIT press, 2 ed., 2010.
- [8] A. G. Barto and R. S. Sutton, *Reinforcement learning: An introduction*. MIT press, 1998.
- [9] D. L. Poole and A. K. Mackworth, *Artificial Intelligence: Foundations of Computational Agents*. Cambridge University Press, 2010.
- [10] R. Roy, *Handbook of Mobile Ad Hoc Networks for Mobility Models*. Springer, 2010.
- [11] J. Chebil, A. K. Lwas, M. R. Islam, and A. Zyoud, “Comparison of empirical propagation path loss models for mobile communications in the suburban

area of kuala lumpur,” in *Mechatronics (ICOM), 2011 4th International Conference On*, pp. 1–5, IEEE, 2011.

- [12] N. Shabbir, M. T. Sadiq, H. Kashif, and R. Ullah, “Comparison of radio propagation models for long term evolution (lte) network.,” *International Journal of Next-Generation Networks*, vol. 3, no. 3, 2011.

Appendices

Appendix A

System Design and Evaluation

The contents...