HANDOVER OPTIMISATION USING NEURAL NETWORKS WITHIN LTE

A THESIS
SUBMITTED TO THE DEPARTMENT OF ELECTRONIC AND ELECTRICAL ENGINEERING
OF THE UNIVERSITY OF STRATHCLYDE
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Neil Sinclair

September 2013

This thesis is the result of original research completed by the author. This work has not been submitted for any other degree at this University or any other institution.

Neil Sinclair

Abstract

Mobile communication infrastructures are getting more complex with the addition of femtocells into the network architecture. Allied with this, the increased use of smart phones add strain onto the network because of higher data requirements. Femtocells are a useful resource to reduce the demand on the macrocell layer and effective handover management is needed to transfer services to and from each base station. The importance of handover management is high within LTE and is included within a use case of Self Organizing Networks. Base stations can autonomously decide whether handover should take place and assign the values of relevant parameters. Setting relevant parameters effectively requires more delicate attention with femtocells to allow for effective and seamless handover to the macrocell. Novel approaches with small amounts of additional signal processing can be utilised to improve handover efficiency.

In this thesis, variations of Self Organising Maps have been implemented. Self Organising Maps can be used to learn the locations of the indoor environment from where handover requests have occurred and, based on previous experience, decide whether to permit or prohibit these handovers. Once the neural network has adapted to the indoor environment, handover can be optimised in different regions independently while still permitting necessary handover. The results of the investigations described within this thesis show that utilising location within the handover process is en effective way to improve handover performance within an indoor environment using an LTE femtocell.

Acknowledgements

I would like to thank Dr. Robert Atkinson and Dr. David Harle for their supervision throughout the duration of my PhD. I am grateful to both Robert and David for their support and advice as well as their willingness to talk about the ideas and direction of my PhD work. My thanks go to my supervisors for the opportunity to do my PhD within the Centre for Intelligent Dynamic Communications (CIDCOM).

During my time at CIDCOM I learnt that research is best completed with the help of others. My interaction with other researchers and the illuminating discussions that took place were a major source of inspiration to me throughout this degree. I am thankful to my colleagues at the University of Strathclyde as well as my family and friends for their help and encouragement. I would like to say a special thanks to Jakub Konka, Jorge Espi Aleman, Xavier Bellekens, and Ian Armstrong for their moral support and friendship. I would also like to thank the late Dr. John Bush for his endless help and support at the beginning of my PhD.

Contents

Nomenclature

2G	Second generation of mobile communication systems
3G	Third generation of mobile communication systems
4G	Fourth generation of mobile communication systems
ANN	Artificial Neural Network
AoA	Angle of Arrival
BIC	Bayesian Information Criterion
CAPEX	CAPital EXpenditure
DES	Discrete-Event Simulator
eNodeB	evolved NodeB
EPC	Evolved Packet Core
EPS	Evolved Packet System
eUTRAN	evolved UMTS Terrestrial Radio Access Network
GTP	GPRS Tunnelling Protocol
GW	GateWay

HeNodeB	Home-eNodeB
HPI	Handover Performance Indicator
Hys	Handover Hysteresis
LTE	Long Term Evolution
MIMO	Multiple Input Multiple Output
MME	Mobility Management Entity
NGMN	Next Generation Mobile Networks
NS2	Network Simulator 2
NS3	Network Simulator 3
OFDMA	Orthogonal Frequency-Division Multiple Access
OPEX	OPerational EXpenditure
PDN	Packet Data Network
PDN GW	Packet Data Network GateWay
PRNG	Pseudo Random Number Generator
QoS	Quality of Service
RACH	Radio Access CHannel
RAN	Radio Access Network
RAT	Radio Access Technology
RNC	Radio Network Controller

RRM	Radio Resource Management
RSRP	Reference Signal Received Power
RSS	Received Signal Strength
S-GW	Serving-Gateway
SAE	System Architecture Evolution
SC-FDMA	Single Carrier-Frequency-Division Multiple Access
SCTP	Stream Control Transport Protocol
SMS	Short Message Service
SOM	Kohonen Self Organising (Feature) Map
SON	Self Organising Network
TAC	Tracking Area Code
TTT	Time To Trigger
UE	User Equipment
UMTS	Universal Mobile Telecommunications System
UTRAN	UMTS Terrestrial Radio Access Network

List of Publications

[1] N. Sinclair, D. Harle, I.A. Glover, and R.C. Atkinson, "A kernel methods approach to reducing handover occurrences within LTE," in *18th European Wireless Conference (EW)*, Poznan, Poland, April 2012, pp. 1 –8.

[2] N. Sinclair, D. Harle, I.A. Glover, and R.C. Atkinson, "A neural networking approach for indoor to outdoor handover," in *Festival of Radio Science (FRS)*, Durham, UK, April 2012.

[3] N. Sinclair, D. Harle, I.A. Glover, J. Irvine, and R.C. Atkinson, "An advanced SOM algorithm applied to handover management within LTE," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 5, pp. 1883–1894, 2013.

[4] N. Sinclair, D. Harle, I.A. Glover, J. Irvine, and R.C. Atkinson, "Parameter optimization for LTE handover using an advanced SOM algorithm," in *77th Vehicular Technology Conference (VTC Spring)*, Dresden, Germany, June 2013.

Chapter 1

Introduction

The increasing demand by mobile subscribers for high data rates is a driver for two key aspects of the Long Term Evolution (LTE) cellular system. Firstly, the highly flexible Orthogonal Frequency-Division Multiple Access (OFDMA)-based air interface combined with Multiple Input Multiple Output (MIMO) antenna technology increases bandwidth efficiency compared to that of existing 3G systems. Secondly, LTE-based systems are expected to utilise additional femtocell base stations to meet the demand for high data rate services. Utilising both of these key aspects allows for paradigms to be use that were not previously possible. Using femtocells, capacity at the macrocell layer can be released and users can be assigned to the femtocell layer. Cellular data provision is an area of increasing concern for operators due to the rapid uptake of smart phones, and femtocells represent a significant technology in addressing these capacity concerns.

Wide scale deployment of large numbers of base stations has implications for the economic viability of a cellular system. As increasing numbers of base stations are deployed, the manual effort and hence cost to configure, optimise and maintain them becomes unsustainable. Furthermore, as the number of base stations increases so does the complexity of the system. Recent studies have shown that 70% of all voice and data traffic derives from users located indoors [1]. However, due to the penetration loss of exterior walls, these users often experience low service quality which limits them to lower bit-rate connections. LTE's integrated support for femtocell technology directly addresses the issue of penetration loss; femtocells can be deployed indoors and consequently provide the required high signal strengths. However, there is a consequent increase in management complexity.

Today's mobile networks need to be frequently re-parameterised in order to accommodate upgrades to coverage and changing traffic loads. Planning, deployment, configuration and optimisation of these network parameters requires significant expenditure from network operators as a result of the time and expertise required to maintain the network. The error prone manual tuning process may also result in non-optimal performance of the network due to the inherent delays associated with changing parameters in the entire network. This has resulted in an industrial pull from operators to introduce a degree of self management. Within LTE, the self management functionality is referred to as a Self Organizing Network (SON) [2] [3], a multi-faceted term that encompasses self-configuration, self-healing and self optimisation. SON offers LTE a plug-n-play functionality that allows both macrocells (eNodeBs) and femtocells (HeNodeBs) to first be deployed and then autonomically self-configure to the requirements of the network. A SON allows tuning of a network to be completed in a timely manner with minimum human interaction. Moreover, a SON can be deployed to optimize handover performance between neighbouring base stations, including femtocells. Of particular interest in this study is self-optimization of handover. Handover management is one of the use cases of the SON paradigm defined by the operators alliance Next Generation Mobile Networks (NGMN) and is used to optimize handover performance between neighbouring base stations, including femtocells.

1.1 Summary of Contributions

The main contributions attained from the work completed within this thesis can be summarised as follows:

- The effective software implementation of handover and neural networks while adhering to LTE specifications within LTE.
- The implementation of a proof of concept that demonstrates the performance of SOM for handover optimisation [4].
- Creation of a novel algorithm implementing k-means into a kernel SOM [5].
- Creation of a novel algorithm that incorporates X-means into the standard kernel SOM algorithm (XSOM) [6].
- Assessment of novel algorithms compared to standard LTE approaches.
- Effective parameter optimisation based on location while utilising neural networks [7].

1.2 Research Objectives

This thesis addresses the very important task of handover optimisation in an LTE environment utilising a femtocell. It is generally acknowledged that unnecessary handovers add strain on the network that could otherwise be avoided. The addition of MIMO in LTE femtocells provides the opportunity to research approaches that were not previously possible. Handovers can be successfully optimised by utilising SON in LTE systems.

The objectives of this research may be summarized as follows:

- A study into the structure and operation of the LTE infrastructure as well as the structure and technical approaches used in previous communications technologies.
- A literature survey of existing and proposed handover optimisation techniques and their use within LTE.
- An investigation into neural networking methods and how they could be applied to handover management.
- Research into the LTE specifications to understand how handover is managed within LTE systems and investigate areas for future development.
- Develop a simulation platform to develop and implement handover optimisation techniques.
- Design, validate and evaluate proposed solutions to handover management that are presented within this thesis. This involved creating prototypes of algorithms and testing their performance in an indoor scenario.

It is important to recognise that telecommunications operators require a high level of reliability while operating profitably. The more efficient their use of their network is the more profitable the business can be. Therefore, methods are needed that can use existing systems effectively and be implemented, used and maintained with minimal effort. It is also necessary to take into consideration that the communications network is increasing in complexity and size with the addition of femtocells. Algorithms that can operate independently and apply generic approaches that are scalable and can self-optimise will be key to the future of profitable telecommunications operations.

1.3 Thesis Outline

This thesis consists of seven chapters and one appendix. The earlier chapters contain some background to introduce the required concepts that are needed fully understand the approaches undertaken for the research completed within this thesis. The main technical chapters then follow and contain details of the main contributions of the work. The thesis is then concluded before the methodology of the work is described in Appendix A. An overview of each chapter follows.

Chapter 2 provides details of many aspects of handover and its implementation within LTE. The Evolved Packet System (EPS) structure before and after the addition of femtocells is described as well as details of changes from previous evolutions of the mobile communications network. The handover process is investigated and explained in this chapter. The signal flow that constructs the handover process is described as well as the handover parameters that govern when handover takes place. Within LTE, handover is within the remit of the SON paradigm and can be completed in a more autonomous manner than in previous mobile communications systems.

Within chapter 3, SON is fully explained along with its implementation in LTE. Autonomic networks and their origins will be explained and are the basis of many SON paradigms. The approaches created for use within SON are part of self-X while adhering to the use cases of SON. The SON use cases include the optimisation of handover. Any approach created for use within SON can be assessed through Handover Performance Indicators (HPIs). These facts are all discussed in detail within the chapter.

The technical chapters start with Chapter 4. The first of the technical chapters includes a proof of concept. The proof of concept involves the use of a Kohonen Self Organising (Feature) Map (SOM) and its application to an indoor environment that utilises an LTE femtocell. The results show that the number of handovers can be reduced by minimising the number of false starts to handover that take place. Handover can be successfully prohibited in regions that have bad history with handover.

A novel SOM-based algorithm was proposed within chapter 5. The proposed XSOM algorithm uses both the X-means algorithm and a kernel SOM algorithm. By using X-means within a kernel SOM algorithm, the area of the neural network is divided into Voronoi cells which allows a reduction in the level of false learning that occurs. Chapter 5 then proceeds to demonstrate the application of the novel algorithm in a femtocell environment for handover optimisation. The XSOM algorithm provides improvements over the Kohonen SOM algorithm for both the learning rate and number of handovers prohibited.

The algorithm proposed in Chapter 5 is a novel algorithm. Chapter 6 utilises this algorithm for handover parameter optimisation. Within LTE, handover parameter optimisation involves altering the Time To Trigger (TTT) and Handover Hysteresis (Hys) values depending on the specifics of each base station environment. In this chapter, handover parameters are optimised based on location within the region of the femtocell environment. Optimising handover based on location is then shown to improve the performance of the LTE system based on HPIs.

Chapter 7 concludes the thesis and brings together the aspects proposed in each of the chapters. The benefits that each of the ideas provide over simpler approaches within LTE are also stated. Approaches to further work are then explained based on the ideas proposed within this thesis.

Chapter 2

LTE Handover

2.1 Introduction

LTE is the fourth generation of mobile communications (4G) and is an evolved version of Universal Mobile Telecommunications System (3G). LTE will have downlink speeds of 100 Mbps and uplink speeds of 50 Mbps for 20 MHz spectrum but can be increased with the use of MIMO and beamforming. Spectrum usability will be flexible, using frequencies from 1.4 MHz to 20 MHz for both uplink and downlink all over the world and the spectrum will be highly utilised and as efficient as possible. The demand, by users, for better quality of picture messages, better video calls and improved internet access as well as being able to send Short Message Service (SMS) messages and be used for voice calls has prompted significant research effort in recent years. To fully utilise the improved services, smart-phones have now become more common within the mobile handset market which in turn has led to an increase in data rates required from the network.

The increasing demand by mobile subscribers for high data rates is a driver for two key aspects of the LTE of cellular systems. Firstly, the highly flexible OFDMA-based air interface (OFDMA in the downlink and Single Carrier-Frequency-Division

Multiple Access (SC-FDMA) in the uplink) combined with MIMO antenna technology increases bandwidth efficiency compared to that of existing 3G systems. Such improved efficiency allows additional services to be used within LTE that were not possible within previous generations of mobile networks. Secondly, LTE-based systems are expected to use more femtocell base stations to meet the demand for high data-rate services. Taking such an approach, users such as those located indoors can be assigned to the femtocell layer thereby freeing capacity at the macrocell layer. In this context, the femtocell serves an exclusive set of users as defined by a closed subscriber group [8].

The addition of femtocells within LTE relieves strain from the macrocell layer, increases spectral efficiency and potentially increases the data rates provided to mobile users. However, this comes at the costs of a more complex evolved UMTS Terrestrial Radio Access Network (eUTRAN) structure within the EPS leading to an increase in the number of handovers that occur within the mobile network.

2.2 Evolved Packet System Structure

The architecture of the EPS includes both the Evolved Packet Core (EPC) which includes all aspects of the core network and the eUTRAN which includes all base stations within the network. The eUTRAN [9, 10, 11] is an evolved version of the UMTS Terrestrial Radio Access Network (UTRAN), from 3G [12, 13] and the Radio Access Network (RAN) from 2G [14], that improves on previous limitations. The architecture of the eUTRAN has been designed to be as simple as possible and as scalable as possible. As a result, the eUTRAN has a flat, all-IP architecture with reduced latency and higher data rates, as defined by 3GPP. The architecture is shown in Figure 2.1.

The LTE access network is a network of base stations, evolved NodeBs (eNodeBs)

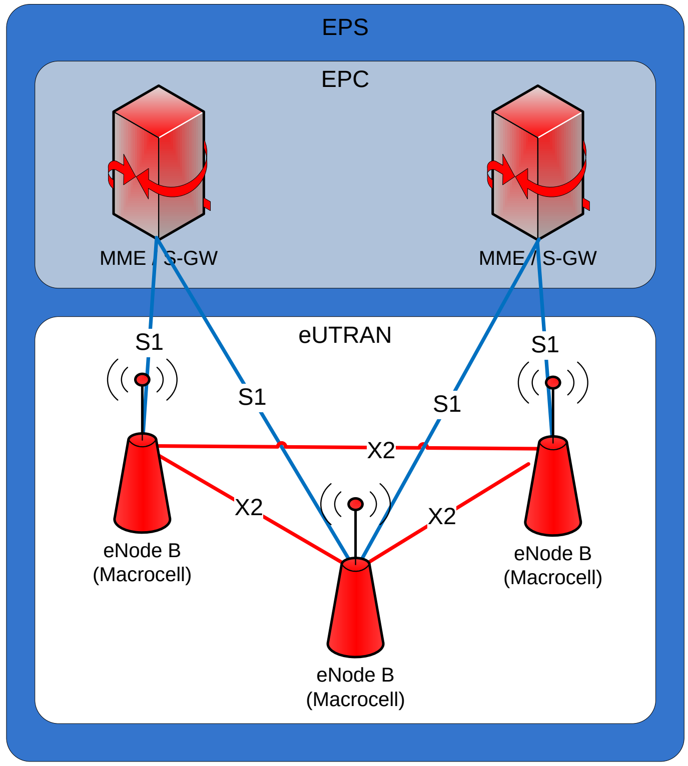

Figure 2.1: Illustration of the eUTRAN

connected to both the EPC and the User Equipment (UE), as shown in Figure 2.1. UEs are devices that connect mobile users to the mobile network. The eNodeBs are responsible for Radio Resource Management (RRM) actions including Radio Bearer Control, Radio Admission Control, Connection Mobility Control and Dynamic Resource Allocation. There is no centralised intelligent controller and the eNodeBs are normally inter-connected by the X2-interface (Figure 2.1) and towards the EPC using the S1-interface (Figure 2.2). The reason for distributing the intelligence amongst the base stations in LTE is to reduce the connection set-up time and reduce the time required for a handover. For an end-user, the connection set-up time for a real time data session is in many cases crucial, particularly in applications such as online gaming. The time for a handover is critical for real-time services where end-users tend to end calls if the handover takes too long.

The EPC contains the following elements: Mobility Management Entity (MME); Serving-GateWay (S-GW); Packet Data Network (PDN). The MME is in charge of the control plane functions required by the user and the session management protocols. The S-GW routes and forwards the user data packets; it is also a mobility anchor when handover to an older technology (*i.e.* UMTS) is being completed. The Packet Data Network GateWay (PDN GW) provides connectivity of the UE to external packet based networks; it performs actions such as policy enforcement, packet filtering, charging support, lawful interception and is also a mobility anchor when handover is being completed to a different technology (*i.e.* WiMAX).

The EPC structure is shown in Figure 2.2. The control plane carries control data through the S1-C interface that is essential for controlling network parameters and sustaining the connection between the UE and the network, *i.e.* maintains transmission resources, handover, etc. The user plane carries the user data through the network through the S1-U interface. As can be seen in Figure 2.2, the S-GW receives the user information and the MME receives the control information. The architecture of the eUTRAN can be seen to be a much flatter architecture than in the UTRAN of 3G because the RNCs have been removed making the architecture much simpler than before. The eNodeBs are now structured in a mesh networking structure. The connections between the eNodeBs are used for control messages for handover initialisation and completion as well as to provide the autonomous network functionality.

The links that connect the EPC to the eNodeB's within the eUTRAN (shown in Figure 2.1) are the S1 interfaces. The protocols used across the S1 interface are described by the S1AP [15] series of protocols. The S1 interface consists of both the user plane protocols (S1-U) [16] and control plane protocols (S1-C) [16]. S1-U uses the GPRS Tunnelling Protocol (GTP) to encapsulate the data and is the reference point between the eUTRAN and the bearer used in the user plane (S-GW) [9]. S1-C

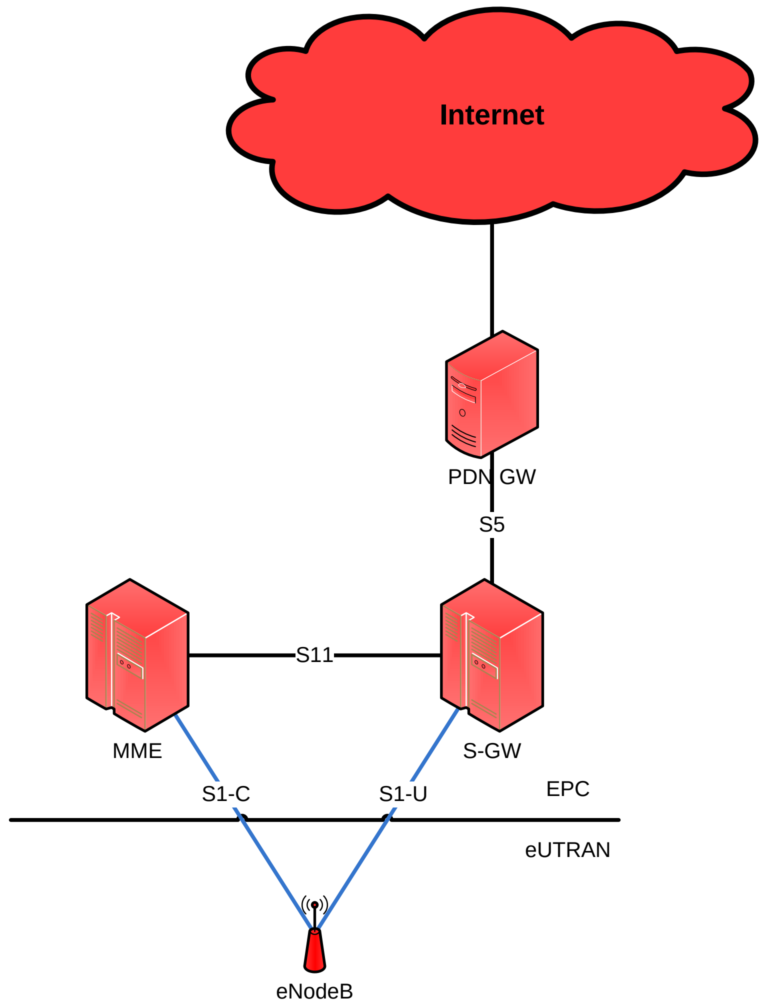

Figure 2.2: Illustration of the EPC

uses the Stream Control Transport Protocol (SCTP) [16] and is the reference point for the control plane between the eUTRAN and the MME. The functionality of the S1AP includes managing the System Architecture Evolution (SAE) bearer, providing paging of the UE, setting up the S1 link, resetting the S1 link, error reporting and mobility functions for the UE, including handover.

The links between the eNodeB's within the eUTRAN (shown in Figure 2.1) represent the X2 interfaces. The X2 interface is responsible for supporting enhanced mobility, inter-cell interference management and SON functionality [9]. The protocols used across the X2 interface are described by the X2AP [17] series of protocols. The X2 interface [16] uses the GTP to encapsulate the data and enable communication between 2 eNodeBs. SON, enhanced mobility and inter-cell interference management are the main uses of the X2 interface in LTE. Some further processes completed through this link are setting up the X2 interface, resetting the X2 interface, load management of the network resources, error reporting and eNodeB configuration. X2 can also be used to complete handover requests in a fast and efficient manner.

2.3 Handover Process

Handover is a key process within any mobile network. It ensures, as users move, that they remain connected to the network with a defined Quality of Service (QoS). Handover is an operation that occurs when the bearer for a particular user is transferred from one base station to another. Such a transfer can occur due to multiple users requesting use of the same radio resources within a given cell or be the result of a user travelling beyond the range of their current base station. The handover process should take place as seamlessly as possible to ensure the user is unaffected by it. To achieve perceptively seamless handover, the process should be completed as fast as possible and have an extremely high success rate otherwise dropped calls might occur. A dropped call is a call that ends as a result of a UE not receiving an acceptable signal strength to sustain the call. Handover takes an estimated time of 0.25 secs [18] to be completed which is an unnecessary waste of time to the network if handovers are not required.

Within LTE, all handovers have the following properties:

- Handovers should be as fast as possible. If the latency is too high then a dropped call is likely to occur resulting in a loss of connection for the user.

- Handovers should be lossless and any information received by the previous serving base station will be sent to the new base station once a handover has been competed.
- The core network does not decide whether handover should take place and is involved in handover decisions to a negligible extent.
- The UE will gather data on the current and surrounding base stations and provide it to the serving base station.
- All handovers are user-assisted but network-controlled. The serving base station controls the occurrence of handovers.

Handover has existed within all generations of widely distributed mobile networks. The handover within 2G [19] is hard handover [20] therefore it follows a "break-before-make" process. Break-before-make handover infers that the previous connection to a base station is severed before a connection to a new base station is made. When this handover takes place it is important to complete the handover as rapidly as possible in order to appear as seamless from the customers point of view. Rapid completion of handover allows for minimisation of disruption and avoidance of QoS degradation. The handover types used within 3G are not only hard handover but also include soft and softer handover as well as hard handover [20]. Soft handover is a make-before-break process that allows for a connection to a new base station (NodeB) to be established before the old one is severed between 2 different base stations. Softer handover is a special case of soft handover that adds and removes the radio links that connect a UE to an individual NodeB. Technically, softer handover is not a handover technique as it is used to improve the reception quality of the connection by using more than 1 radio link within a NodeB. Within 4G [21], handover is the same as within 2G [19]; hard handover [20].

4G handover takes place between 2 base stations, based on measurements taken from the UEs. Generally, the process for handover is as follows:

1. When another eNodeB has provided a stronger signal strength than the serving eNodeB for a defined period of time, a measurement report is triggered.

2. When a measurement report it triggered, the current eNodeB will initiate a handover to a new eNodeB.

3. The new eNodeB prepares the radio resources prior to accepting the handover.

4. Notification is sent to the current eNodeB to release the resources by the new eNodeB.

5. A path switch message is sent to the MME within the EPC to complete the handover.

So, handover mainly occurs when triggered by a measurement report being sent from the UE. Individual handover parameters are used to control the timing and likelihood of handovers. The nature and impact of these parameters are now discussed.

2.3.1 Handover Parameters

According to the LTE standards [22] for handover to occur, there are parameters that must be considered. Two tuneable parameters can be used to govern handover performance. These parameters are TTT and Hys. The choice of TTT and Hys values are pre-defined in LTE networks [22]. There are 16 valid TTT values,

The Hys value varies in 0.5 dB steps between 0 and 10 dB. Handover to a candidate base station can only be executed if that candidate provides better signal strength than the serving base station by an amount equal to, or exceeding, Hys for a duration

Table 2.1: TTT Values

Parameter	Value
TTT	0 s
	0.04 s
	0.064 s
	0.08 s
	0.1 s
	0.128 s
	0.16 s
	0.256 s
	0.32 s
	0.48 s
	0.512 s
	0.64 s
	1.024 s
	1.280 s
	2.560 s
	5.120 s

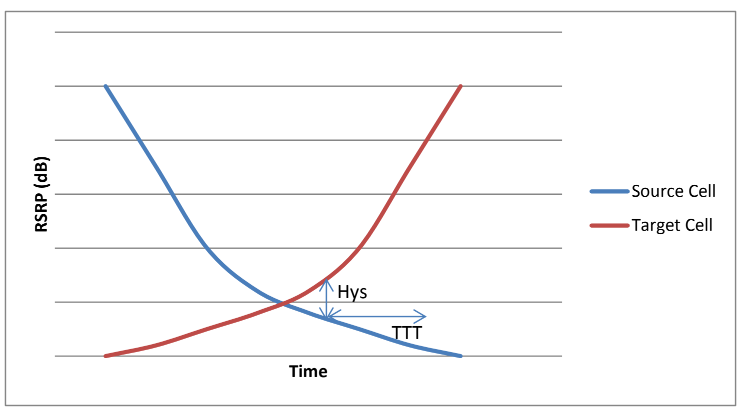

Figure 2.3: Illustration of handover parameters: TTT and Hys

equal to the TTT, as shown in Figure 2.3. Finding the optimal values for TTT and Hys is a difficult task.

Optimal handover timing is a complex task as a result of the irregular base station coverage areas coupled with the effects of shadowing and multipath propagation which gives rise to stochastic variation in Reference Signal Received Power (RSRP) [23] and signal quality. RSRP in this context is equivalent to the Received Signal Strength (RSS) of the UE in the network. This results in terminals receiving an improved RSRP from a neighbouring base station at one instant and a worse RSRP the next as a result of movement of environmental scatterers, even for stationary terminals. Such changes in RSRP can trigger unnecessary and unwanted handovers adding stress to the network.

Handover too early and handover too late metrics [24] are defined, in LTE, to capture handover performance. Handover too early [25] occurs when a handover is executed unnecessarily and handover too late occurs when a call is dropped before execution is completed. When a handover too early is detected the algorithm increases both TTT and Hys to reduce the probability of it occurring again. However, such a change increases the probability of a call being dropped. Handover too early [25] can lead to the occurrence of ping-pong handover. Ping-pong handover is when multiple handovers occur between base stations within a short time interval; these are generally unnecessary handovers. If a handover too late is detected then the algorithm decreases both TTT and Hys to reduce the probability of future calls being dropped. However, this increases the likelihood of handover too early. A balance must be sought in order to avoid both handover too early and handover too late.

Handover optimisation, within LTE, is concerned with the conflicting requirements to minimize the likelihood of dropped calls and minimize the number of unnecessary handovers. An optimisation algorithm must be used within LTE to find the best statistical balance between these undesirable events. If the parameters are optimally set there will be a reduction in the number of handover too early and handover too late occurrences which will lead to a more efficient use of resources and a reduction in the number of handovers that occur.

2.3.2 Handover Triggers

There are multiple triggers to initiate handover [22] and when any of these triggers have been activated a measurement report is sent from the UE to the eNodeB. The triggers, as defined by 3GPP, are shown in Table 2.2.

Table 2.2: Handover Triggers

Event Trigger	Trigger Criterion
A1	Serving cell becomes better than an absolute threshold
A2	Serving cell becomes worse than an absolute threshold
A3	Neighbouring cell becomes a margin of offset better than the serving cell
A4	Neighbouring cell becomes better than a threshold level
A5	Serving cell becomes worse than threshold 1 and a neighbouring cell becomes better than threshold 2
B1	Inter Radio Access Technology (RAT) neighbouring cell becomes better than threshold
B2	Serving cell becomes worse than threshold 1 and inter RAT neighbouring cell becomes better than threshold 2

Of the list, the most common trigger to take place is event A3. Within event A3, a measurement report is sent when the RSRP of a neighbouring cell exceeds that of the serving cell by a given Hys value for a period of time (TTT), as described in Equation (2.1) and Section 2.3.1.

$$\mathrm{RSRP}_{\mathrm{servingcell}} + \mathrm{Hys} < \mathrm{RSRP}_{\mathrm{neighbouringcell}} \tag{2.1}$$

Upon receipt of the measurement report, the base station decides whether to allow the handover to take place. If the handover is allowed, the eNodeB initialises the signal flow for the handover process to occur; if the handover is not allowed, no further action is taken by the network.

2.4 Macrocell Handover Process

The eUTRAN structure of 4G leads to multiple handover types being required, even when only macrocells are included within the network. When a handover occurs between an eNodeB that is connected to a specific MME to an eNodeB that is connected to the same MME, an intra MME handover takes place. However, when the eNodeB's are served by different MMEs inter MME handover takes place. Figure 2.4 depicts the infrastructure of the EPS and shows the connections that lead to the different handover types.

The handover process is described in more detail in Figures 2.5 and 2.6. Figure 2.5 shows the handover process for an inter MME handover and Figure 2.6 shows the handover process for an intra MME handover.

To enable an inter MME handover (flow shown in Figure 2.5), the mobile reports the signal quality of the network for both current and neighbouring cells to gauge whether a handover is required. The measurement report is analysed by the current eNodeB: if a handover is not required then no further action is taken; if a handover is required (*i.e.* the eNodeB decides that the mobile would be better served by another base station) then a request is made to the MME including a list of potential cells. The current MME, having decided on a target base station, sends the handover request to the target MME via the GTP which masks the data as IP data to send it through the network. The target MME passes the handover request onto the target eNodeB; this handover request is treated as a new call to the target eNodeB so a traffic channel needs to be allocated for the mobile. A message containing all the required

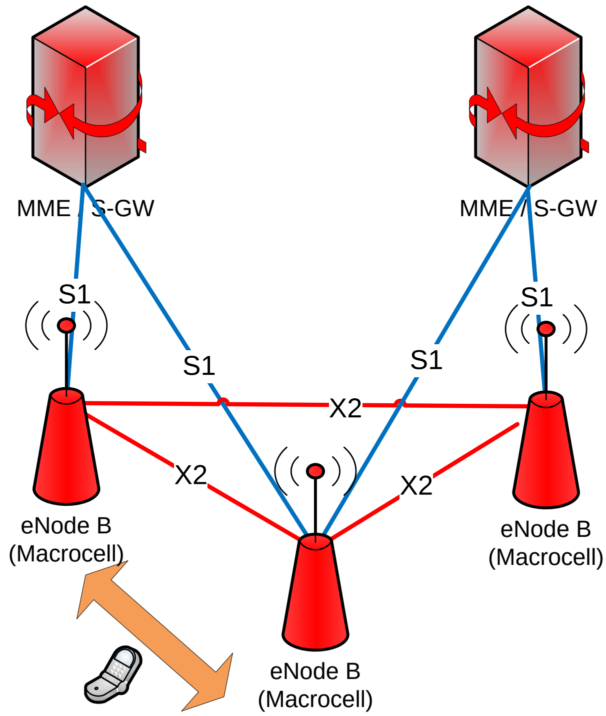

Figure 2.4: Illustration of LTE RAN

information for the mobile is generated by the eNodeB. The target MME then accepts the call by acknowledging the handover request. The current MME then delivers a handover command to the current eNodeB, a command which includes the radio resource information. The handover confirm message is then sent from the mobile to the target eNodeB to the target MME. All that remains to do is to release the call resources in the previous base station which is instigated by the previous MME sending a release command to the previous eNodeB. Handover is then complete.

To enable intra MME handover (flow shown in Figure 2.6) the mobile reports the signal quality of the network for both current and neighbouring cells to gauge whether a handover is required. The measurement report is analysed by the current eNodeB: if a handover is not required then no further action is taken; if a handover is required (*i.e.* the current eNodeB decides that the mobile would be better served by

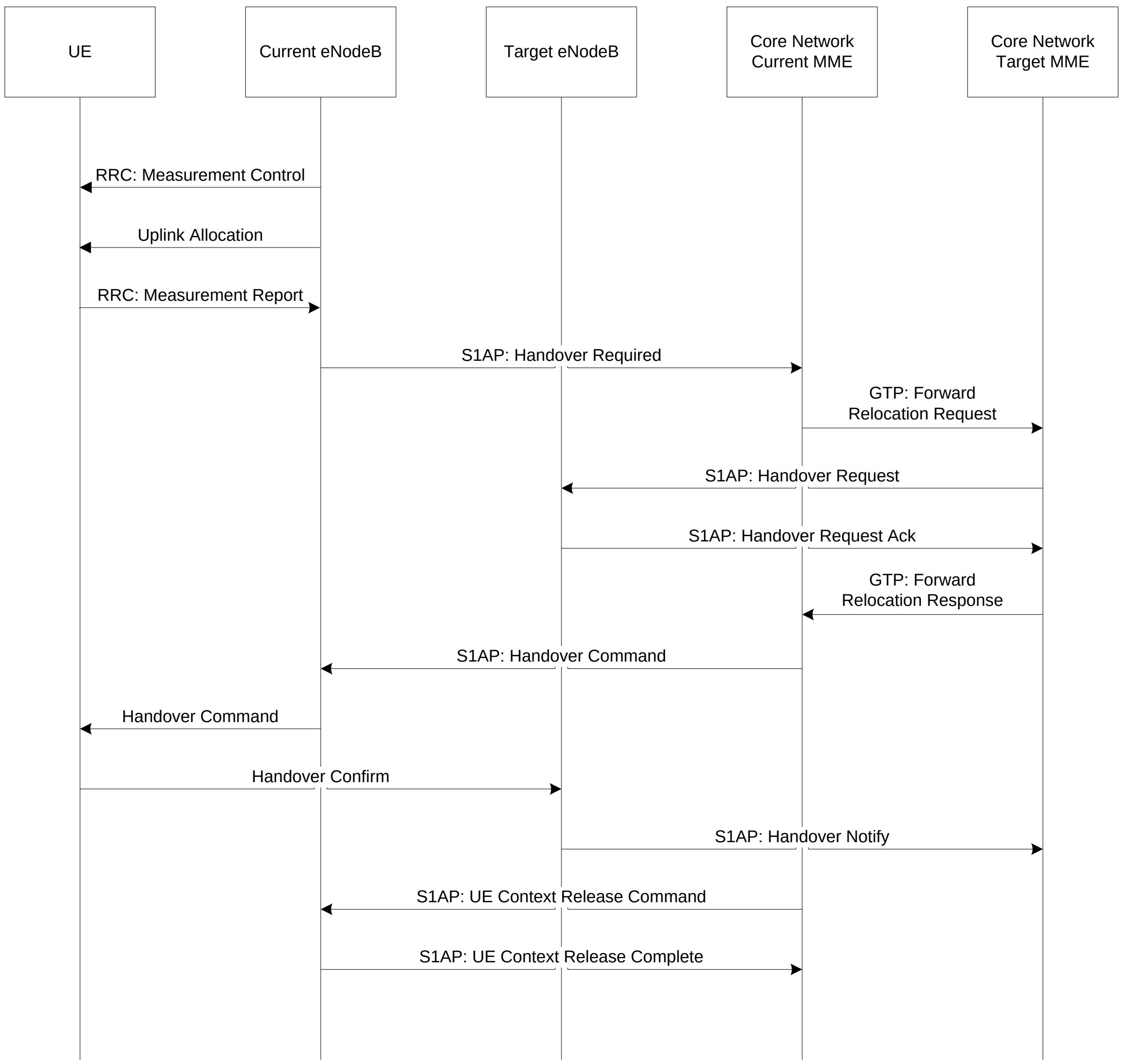

Figure 2.5: Inter-MME handover process

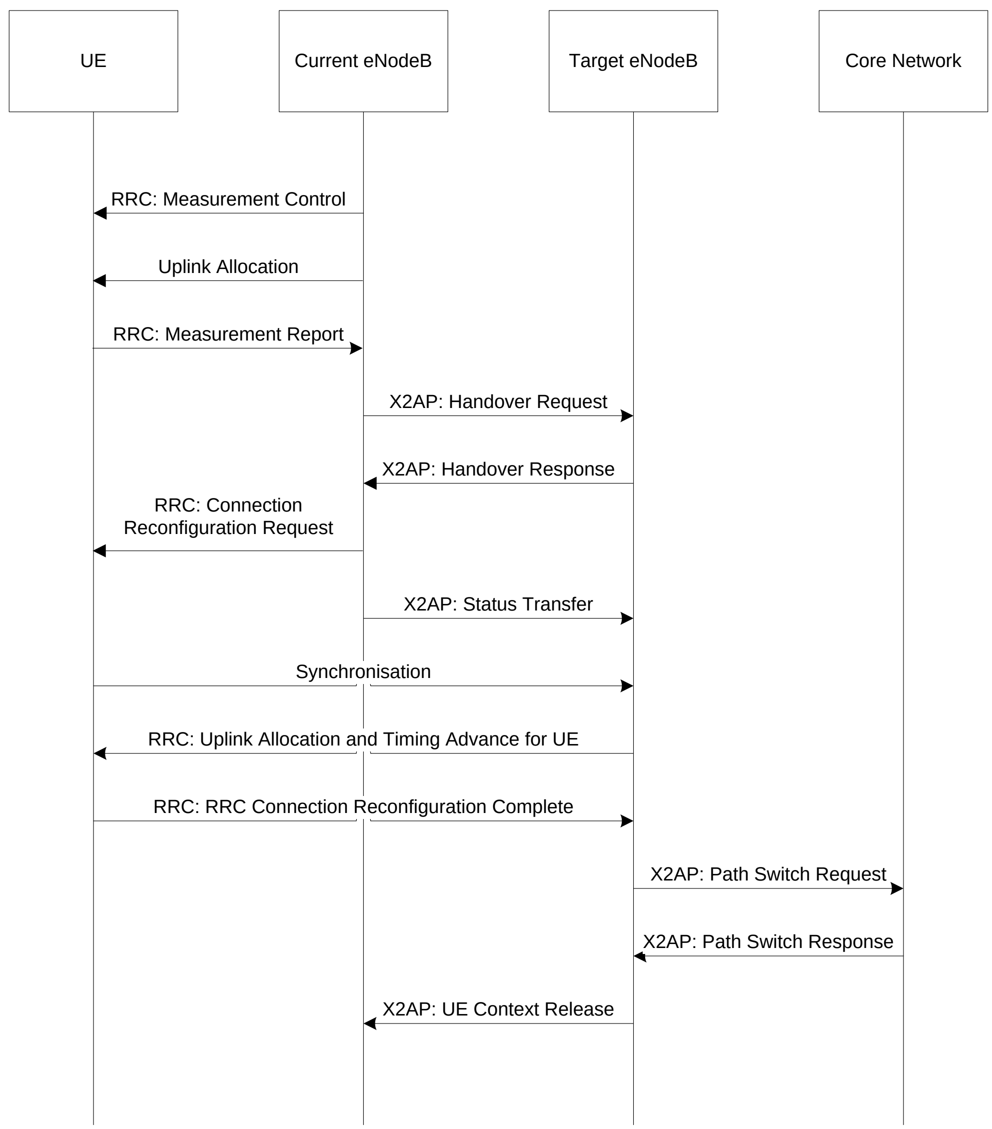

Figure 2.6: Intra-MME handover process

another base station) then a request is made to the target eNodeB directly through the X2 interface. The current eNodeB, having decided on a target base station, sends the radio resource connection reconfiguration request to the mobile to initialise the reconfiguration of the radio resources to the target eNodeB. The current eNodeB also sends a Status Transfer message to the target eNodeB to inform the target eNodeB that will be taking over the connection between the network and the mobile. The mobile then sends a synchronisation message to the target eNodeB which replies with the radio resource specifications that the mobile and the new eNodeB will utilise for the connection as well as the allocation for the traffic channel used by the mobile. At this point the exchange of radio resource physical information between the mobile and the target eNodeB is complete. A path switch message is then sent from the target eNodeB to the EPC; this message informs the EPC that the mobile will be changing eNodeB to another that will provide it with a better service. This is the only contact with the EPC. Handover is then completed with messages from the target eNodeB to the current eNodeB to release the call resources in the previous base station.

The intra MME handover is simple to implement. The decision to handover being completed without the input of the MME or any other element in the EPC simplifies the process. The intra MME handover is completed through the X2 interface; shown in Figure 2.6. The complexity of the intra MME handover (shown in Figure 2.6) can be compared to that of the inter MME handover process shown in Figure 2.5. Figures 2.5 and 2.6 illustrate the difference that the X2 interface has made to the handover process within the LTE architecture. The inter MME handover is more complex as a result of the incorporation of the EPC within the handover process. This will add to the delay associated with the handover process by increasing the number of messages sent and the distance the messages require to travel. The X2 interface improves the handover process by excluding the MME from the handover process with the exception of sending it to the UE's new routing requirements. In addition to making the handover process faster, the X2 interface also provides a route to send the packets that were sent to the previous eNodeB before the handover was finalised.

The eUTRAN structure of 4G has to incorporate femtocells into the network architecture. The change in the network architecture complicates the handover process.

2.5 Femtocells

Femtocells [26, 27, 28, 29] are cellular network access points that connect the end user via mobile devices to the EPC in a similar manner to macrocells. They can be used to guarantee mobile coverage to a small number of mobile devices within their coverage area, provide faster connection speeds and more reliable connections. Due to their low cost and the ability to support multiple simultaneous users, femtocells are ideal for use within a home or office to improve the data rates and QoS received. Femtocells have the following specifications [21]:

- Range – 20m
- Users – 2 to 6
- Backhaul – Broadband Internet connection
- Benefits – Small and low cost
- MIMO – 2×2

The use of MIMO within femtocells for LTE [30] allow applications to be completed on the femtocell that were not previously possible. The short range of femtocells leads to many benefits that cannot be achieved with macrocells. Femtocells require low power which reduces the operational cost and there will be less signal degradation from multipath propagation and attenuation, leading to an increase in the quality of connection. However, the likelihood of handover taking place is much higher and therefore the level of dropped calls increases. The latter is an important issue because network operators require high efficiency within the network and need to avoid repercussions of excessive handovers if possible.

The motivations for femtocell deployment are: improved indoor coverage, increased demand for data services that result an increase in market penetration of smart-phones and less strain on the current network architecture. Femtocells support users that would have otherwise been using a macrocell connection. Removing some of the strain from the macrocell network is required because the intensity of network traffic in the macrocell network is increasing as a result of more network intensive applications becoming available on mobile phones. The benefits of reduced network traffic to the customer include increased connection speeds and better QoS.

2.5.1 Femtocells in Mobile Communication Networks

Within the UK, femtocells that operate on the 3G network have been commercially deployed by Vodafone since Jan 2010 [31] but other operators have since followed. Vodafone has labelled the product as “Sure Signal”; it is marketed as a product that will improve the signal in the home. The business model that governs the operation of the femtocells is based on network requirements and questionnaires given to customers. The business model involves a closed subscriber group serving up to 4 users simultaneously with the customer paying the operational costs (electricity, backhaul and femtocell purchase) and using a broadband Internet connection for the backhaul network. Femtocells currently outnumber the number of macrocells deployed [32]. Within 3G, femtocells were not deployed from the first installation but were added on much later to address the demand from users for high data rates. With future communications networks (e.g. LTE), femtocells will be deployed from the first installation and have a prominent place in the network architecture [33, 34].

Femtocells are being included in the LTE network to overcome some shortcomings of previous mobile communications networks. Femtocells are useful to the network in a number of ways:

- Improve service coverage indoors where macrocells struggle.
- Offload traffic from the macrocell layer to improve the power efficiency of the network and maximise the macrocell capacity.
- Increase radio spectrum reuse.
- Increase data rates by avoiding the high penetration loss of macrocell coverage.
- The lower required transmission power Improves the “green effect” of the communications network.

The applications supported by mobile phones increasingly require higher levels of service from RANs. The network cannot sustain this trend in network requirements indefinitely because there are limitations to the capacity that the network can provide. The inclusion of femtocells within the 4G network architecture can improve network capacity and help handle the increase in network requirements. The distance between the UE and the base station (eNodeB) can be high, this creates problems that, from the UEs perspective, dramatically reduce the RSRP. Keeping the user close to the base station by using a femtocell improves the QoS for the user by improving the connection speeds for the uplink and downlink as well as improving the reliability of the connection. Such proximity also allows for a higher level of frequency reuse compared to traditional macrocell base stations.

Femtocells will be primarily used within home environments. Therefore, there will be a higher density of femtocells in dense urban environments as shown in Figure 2.7.

As can be seen in Figure 2.7, the user retains close proximity to the base station since the range of the femtocell coverage is only in the immediate are area of the

Figure 2.7: An urban environment with femtocell deployment

house. Dense urban environments can lead to a high density of femtocells which can result in issues such as access type, interference and handover.

2.5.2 Issues with Femtocell implementation into the Macrocell Network

Integrating femtocells into the RAN is simple in principle but in practice there are many aspects that affect the performance of both the macrocell and femtocell layers of mobile communication networks [21, 27]. A coveted improvement when implementing femtocells within the communications network is to increase the QoS of the connection for the users accompanied by no adverse effects on the current network. There are many implementation details that need to be considered before widescale femtocell deployment can occur effectively. These issues are discussed in the following sections:

Business Model

When deploying femtocells, the business model helps to govern the operation of the network and the equipment by defining specific business goals. It can be defined as a set of choices that effect the equipment operations and how these operations will be completed from a business perspective. No low level technical explicit decisions are made; only decisions that will effect the customer in a business sense. When creating the business model, it is important to think about who will financially benefit from the network and who will gain from the business perspective. Either the user or the network operator may gain from the use of femtocells and the issues associated with this deployment are now discussed.

The choice of access method can significantly impact the operation of the femtocell. The access method defines who can obtain a connection to each individual femtocell. If the femtocell operates a closed access mode then only users that are registered with a femtocell can connect to it; this is favourable for use within the home because customers who are paying should have control over who accesses the femtocell. Open access allows network subscribers to connect to the femtocell whenever they are within range; this is favourable in a public area as it can be used by any network subscriber. Open access femtocells are not ideal for use within a household because the home owner who is paying should always be guaranteed a connection to the femtocell. Hybrid access is defining the access type (closed or open) based on where the femtocell will be applied and who is paying rather than using a one type everywhere approach.

The running cost of femtocells is a key aspect which is strongly influenced by the access type that the femtocell adopts. Femtocells use a broadband Internet connection as their backhaul network; they use the Internet connection to transfer the data between the femtocell and the mobile communications network infrastructure. If the femtocell operates in closed access mode, then the customer will be the sole benefactor of the femtocell and therefore will most likely have to pay for the femtocell and its running costs. If the femtocell has open access then this begs the question: 'why should the customer pay all the costs of the femtocell operation since the network operator is benefiting from the customer having a femtocell that anyone on the network can use?'. As a result, using closed access simplifies the economic decisions required for implementing femtocells commercially.

RF Interference

When being integrated into RAN infrastructures femtocells can either operate on the same frequency band as the macrocell layer or use a different frequency band. If femtocells use the same frequency band as the macrocell layer then interference can occur between macrocell and femtocell. It is possible for the femtocell network to use a different frequency band in order to eliminate the frequency interference that occurs between the different base stations. However, within LTE, femtocells and macrocells must be able to coexist on the same frequency. As a result, advanced interference mitigation techniques exist [35, 36, 37] that allow both femtocells and macrocells to operate on the same frequency as each other with minimal repercussions.

Plug-n-Play Functionality

As femtocells are to be integrated into the existing communication network architecture, the network will become more complex and adaptable than previous RAN architectures. The plug-n-play functionality allows femtocells to be deployed without explicit information about the femtocell environment. This complicates the creation of an autonomous system because all radio environments are unique. As increasing numbers of base stations are deployed, the manual effort and hence cost to configure, optimise and maintain them becomes unsustainable. To simplify the configuration of the network, a plug-n-play functionality will be implemented to achieve a degree of autonomy within the network. This autonomic nature should allow each femtocell to be installed as simply as possible and be able to operate efficiently immediately after being connected to the network.

For plug-and-play functionality to be effective, the femtocell should be able to acquire knowledge of the network on installation and then use this knowledge to set up, optimise and heal whenever it is required to do so. The autonomic nature being used to obtain a plug-n-play functionality is referred to as SON and is covered in more detail in Chapter 3.

Wide scale deployment of large numbers of base stations has implications for the economic viability of a cellular system. Having plug-n-play functionality in the network will allow the network operator to reduce the setup and maintenance costs of the network. The network will be able to find issues that occur and fix them with minimal human intervention, making the network much more autonomous. Only significant issues with the network would require any input from a network technician.

Mobility

Mobility, within a communications context, is the ability to move from the coverage area of one base station to another; this transfer of bearer should be as seamless as possible. The communications market that femtocells will be used within is mobile communications; mobility issues must first be overcome. When communications terminals are wireless and moving, additional considerations have to be resolved. Issues with frequency planning and handover management have to be addressed and more advanced methods have to be developed in order to cope with the increased regularity of mobility issues. Handover is the transference of the user's connection from one base station to another; this is discussed in Section 2.3.

2.6 Femtocell Handover Process

The eUTRAN structure of 4G will include femtocells in the LTE network architecture. When Home-eNodeBs (HeNodeBs (femtocells)) are included in the network infrastructure, the structure of the eUTRAN has to change. Within the eUTRAN used for HeNodeBs, there is no X2 interface included because of the large number of femtocells and their random temporal deployment making it too complicated to implement. The eUTRAN architecture used for femtocells is shown in Figure 2.8.

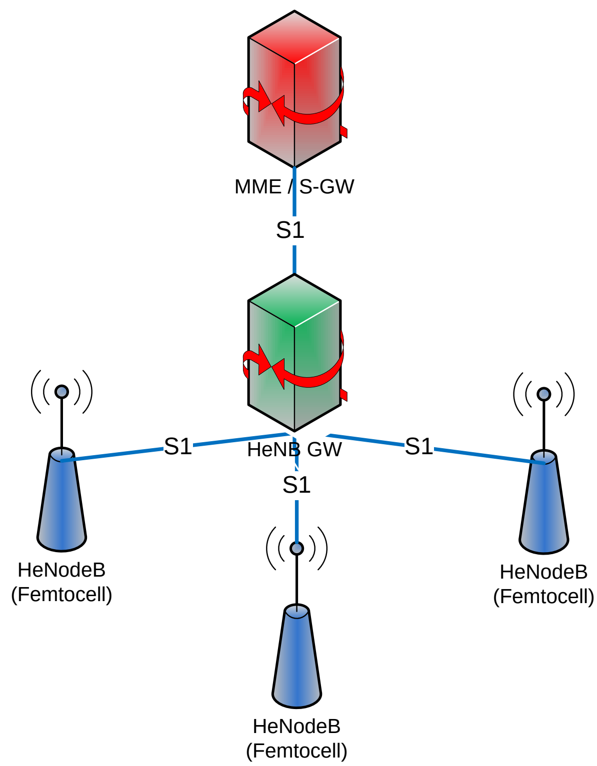

Figure 2.8: Illustration of LTE eUTRAN using femtocells only

As shown in Figure 2.8, there is now a networking element called a HeNodeB GateWay (GW) included in the architecture. The HeNodeB GW is assigned an eNodeB ID and is viewed by the MME as being an eNodeB; it is responsible for allocating identities for each of the HeNodeBs within its control and stores them in a list. These ID numbers along with unique Tracking Area Codes (TAC) are assigned to all elements in the architecture and are used to track the elements in the architecture. When ID's for the base stations (femtocells and macrocells) are stored within neighbour lists, base station type will also be stored. The HeNodeB GW acts as a concentrator to limit the amount of S1 interfaces required by the MME and S-GW.

The addition of femtocells will also change types and levels of handover that will be carried out by the communications system. The additional handover processes that are required include:

- femtocell to femtocell
- macrocell to femtocell
- femtocell to macrocell

Each of these handover types are similar because all are part of the same overall eUTRAN structure. When the structures of the eUTRAN involved with both eNodeBs and HeNodeBs are combined, the architecture of the eUTRAN becomes more complicated as shown in Figure 2.12.

The different handover types will now be described. When femtocells are involved, all handovers resemble that of inter-MME handover between 2 macrocells because there is no X2 interface.

For femtocell to femtocell handover as depicted in Figure 2.10, the route of the request to initiate handover must traverse the MME since there is no X2 link. Based on the macrocell to macrocell handover within the complete LTE architecture an example signal flow diagram can be generated that shows how the femtocell to femtocell handover is accomplished. The handover message flow for a femtocell to femtocell handover is shown in Figure 2.10 and uses the same process as the macrocell to

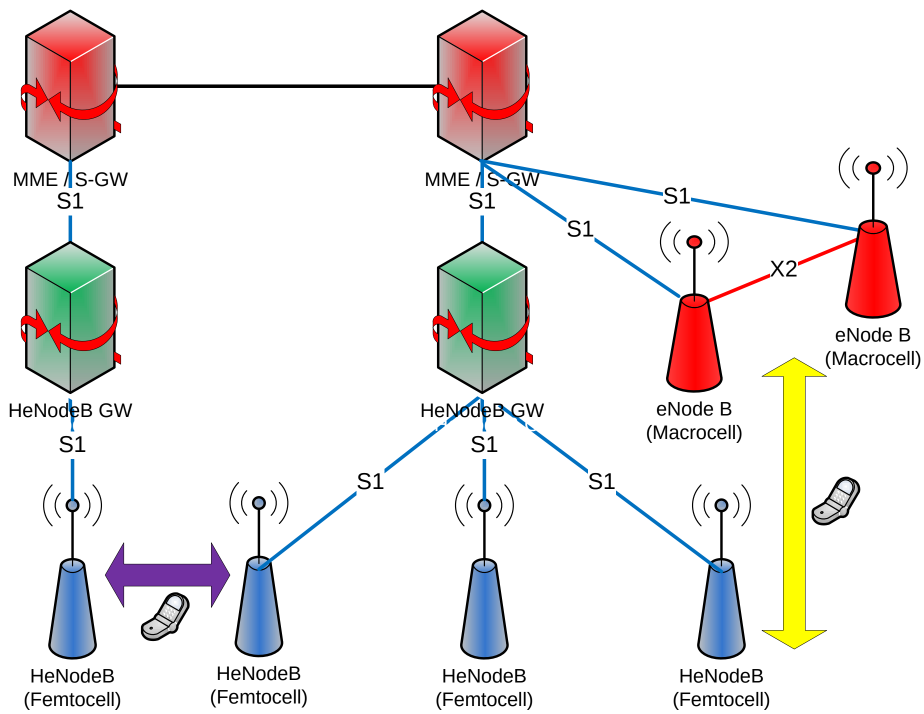

Figure 2.9: Illustration of a complete eUTRAN

macrocell handover; this means that the decisions are still made by the same elements as before with the exception that there is now potentially an intermediate network element between the HeNodeB and the MME (HeNodeB GW).

When handover is required from a macrocell to a femtocell as depicted in Figure 2.11, the macrocell will know that it is transferring information to a femtocell from the TAC and the eNodeB ID that has been reported from the UE; the TAC and eNodeB ID's are stored in a neighbour list. Using the information from the UE, the HeNodeB GW is identified and a signal is sent to the MME which then relays the signal to the appropriate HeNodeB. Now that the handover has been initiated, the handover can proceed as before by the HeNodeB setting up the radio resources then notifying the eNodeB to release the resources to the HeNodeB and a path switch message being sent to the MME. The signalling used for this handover process is shown in Figure

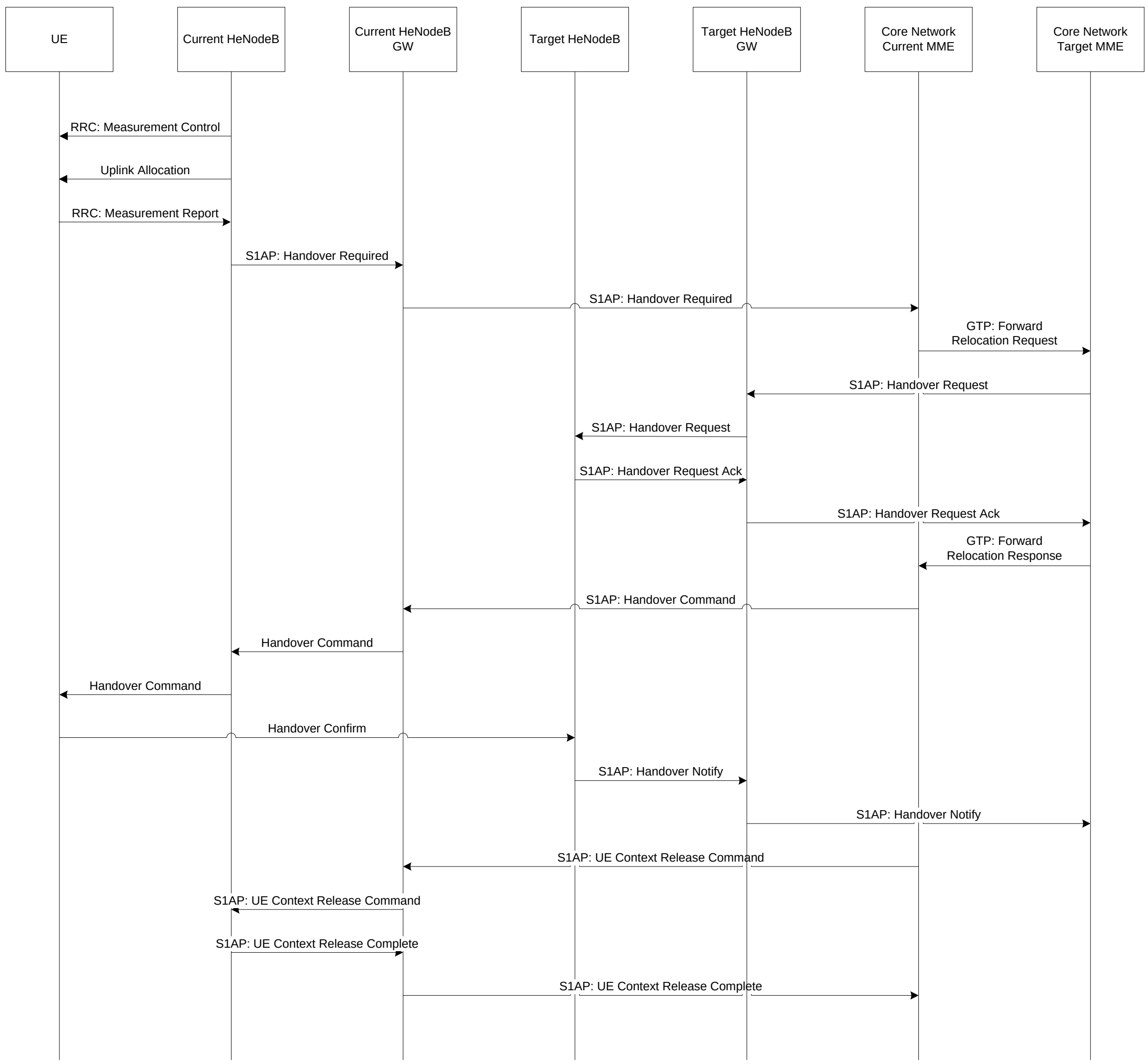

Figure 2.10: Femtocell to femtocell handover process

2.11.

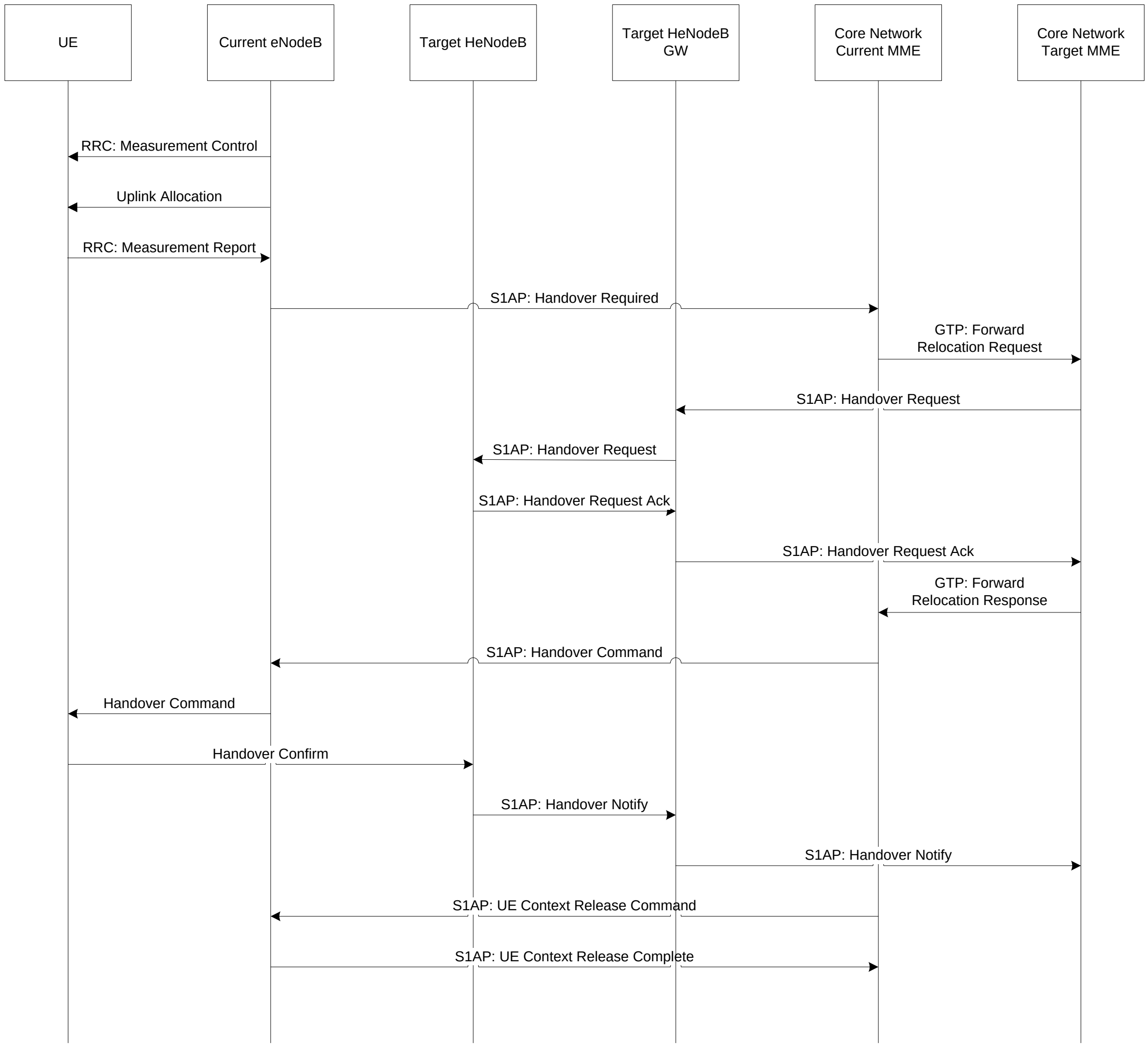

Figure 2.11: Macrocell to femtocell handover process

When a handover is required from a femtocell to a macrocell as depicted in Figure 2.12, the chain of network components involved in the initiation message is similar to the macrocell to femtocell handover because the signal has to go via the MME. When notified by the UE that a handover should take place the femtocell knows that it is going to handover to a macrocell from the TAC and the HeNodeB ID reported by the UE; the TAC and eNodeB ID's are stored in a neighbour list. The TAC and eNodeB ID's are maintained by the EPC, the network will have knowledge of the type of base station pertaining to each ID number; the HeNodeB GWs will have a list of all the femtocells within their control in the form of their HeNodeB ID number. The eNodeB GW then notifies the MME who then forwards the information to the eNodeB that will take over operation of the UE.

To enable a handover (message flow shown in Figures 2.10 to 2.12) the mobile reports the signal quality of the network for both current and neighbouring cells to gauge whether a handover is required. The measurement report is analysed by the current base station. If a handover is not required, then no further action is taken but if a handover is required, (*i.e.* the base station decides that the mobile would be better served by another base station) then a request is made to the current MME. The current MME, having decided on a target base station, sends the handover request to the target MME via the GTP. The target MME passes the handover request to the target base station, this handover request is treated as a new call to the target base station so a traffic channel needs to be allocated for the mobile. A message containing all the required information for the mobile is generated by the current MME and sent to the mobile via the handover command messages. This handover command is then confirmed by the mobile; this confirmation initiates a stream of commands confirming that the mobile is about to change its network connectivity point. The current MME then delivers a context release command to the current base station; this command includes the release of the radio resource information. The handover is then completed by the release of the call resources by the previous base station and confirmation being sent to the previous MME.

Requiring handover information to pass through the MME is not an optimal approach because the handover process will have higher latency than if the information is directly sent through an X2 link. Within the 2nd generation (2G) of communications, the RAN required all handovers to be completed via the core network; this

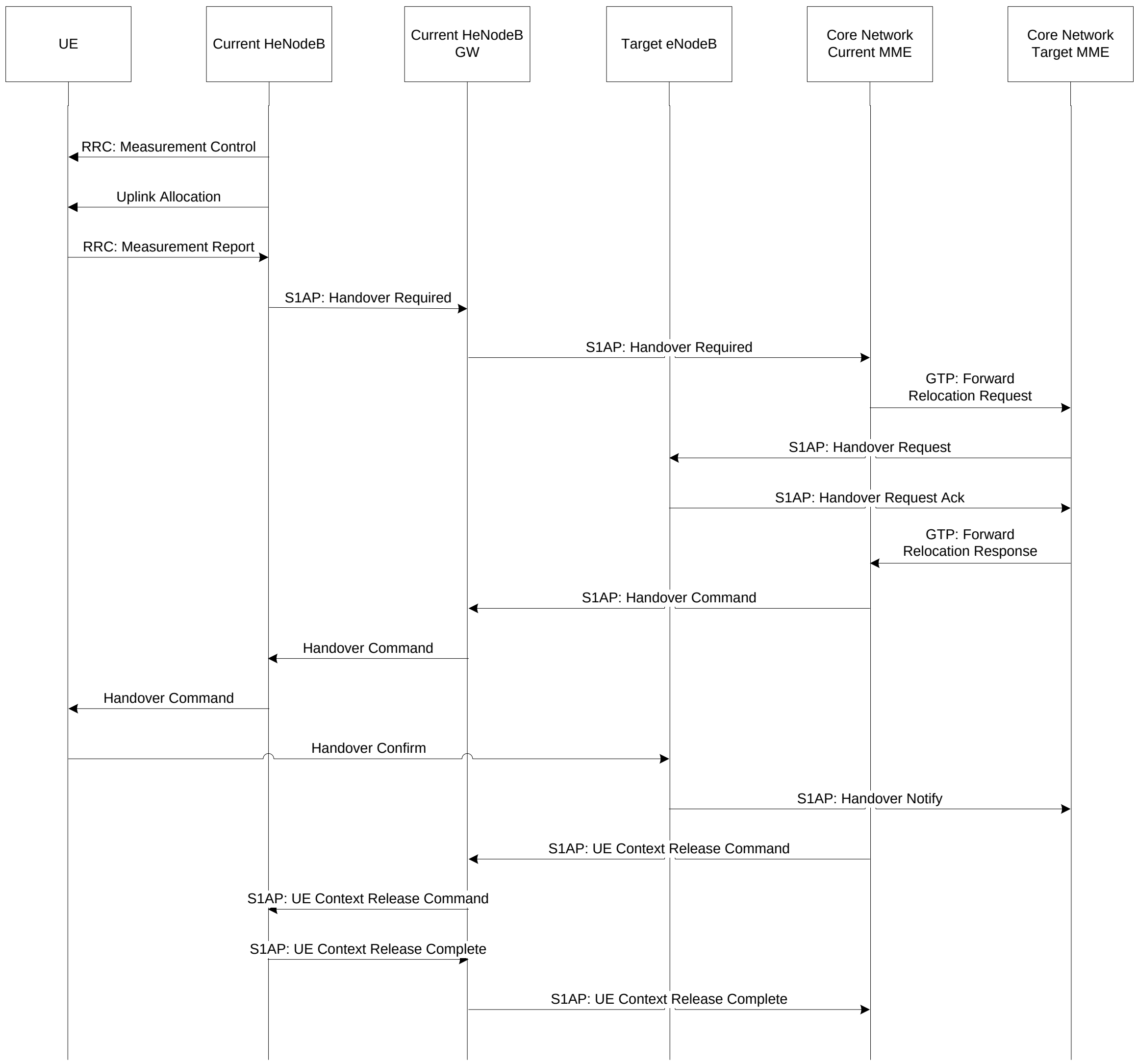

Figure 2.12: Femtocell to macrocell handover process

was changed within the 3rd generation (3G) by including direct links between the RNCs through which the handover could be executed. From an external perspective, it could be perceived that the design decisions have forced a devolution of the mobile communications network architecture but not through choice. The level of deployment of femtocells could be extremely high and also sporadic due to the customers (not the network provider) deciding when and where the femtocells will be. This renders any other design choices for the eUTRAN impractical and unworkable since additional links would make the entire network highly complex and completely unmanageable.

2.7 Summary

The eUTRAN of LTE is the next step of evolution within the mobile communications infrastructure. The addition of femtocells into the eUTRAN will potentially add stress onto the LTE network by increasing the occurrence of handovers. Additional handovers add strain to the network through the consumption of radio channels (RACH) and fixed links; through additional processing load in admission control, bearer setting and path switching; and have the potential to degrade the QoS of ongoing connections [38].

This chapter has discussed the addition of femtocells into the eUTRAN as well as the issues with implementing them into the current mobile communications architecture. To be successful in the network femtocells will have to be implemented with a form of autonomic networking in order to configure and run optimally. 3GPP have defined SON within LTE standards to implement autonomic networking concepts for mobile communications operations, including mobility robustness.

Chapter 3

Self Organising Networks

3.1 Introduction

The rapid proliferation of smart-phones and increased data rates demanded by subscribers have led to LTE using femtocells and picocells to meet future traffic requirements. The addition of so many base stations will require a more efficient network generally and more efficient handover management specifically. Effective handover management minimises the amount of handover too early and handover too late thus creating a more efficient number of handovers triggered. SON will be used to operate and optimise the system to realise this increased efficiency. Base stations within the network will be able to configure their radio parameters automatically with minimal human interaction using 3 facets: self-configuration, self-optimisation and self-healing.

3.2 Autonomic Networking

Autonomic networking [39, 40] applies the principles of autonomic computing to a networking environment. The term Autonomic Computing refers to computing systems having the ability to self manage and react to unpredictable events while hiding the complexities of the system to the end user. The inspiration for autonomic computing is the manner by which the nervous system regulates the operations of biological organisms [41]. This methodology was first applied by IBM in 2001 to handle the increasing complexities of managing computing systems without any external input. The autonomic paradigm is one in which time-consuming and error-prone tasks are undertaken by self managing components, leaving human administrators free to concentrate on high-level policies. Autonomic systems are thus provided with high level policies that are used to govern how the system will adapt and optimise to unforeseen changes. These policies state what the system should aim to do, not how it should be completed; the latter is the role of the autonomic element. Autonomic networks use the same paradigms created for autonomic computing systems but apply these ideas to network management [42]. The fundamental structure of autonomic systems is a control loop, represented in Figure 3.1.

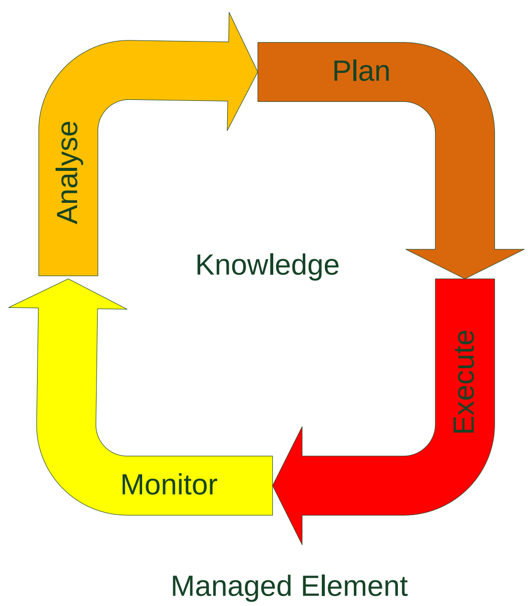

Figure 3.1: Autonomic loop

The four stages involved in any autonomic system: Monitor; Analyse; Plan; Execute, as depicted in Figure 3.1. This is a useful structure for any self optimising system because the inherent feedback supports an autonomically adjusting management system. The four stages of the control loop constitute the fundamental building blocks of the autonomic system and are structured as follows:

- The Monitor phase is concerned with the acquisition, collation and filtering of data concerning the managed element or its environment, this constitutes the input to the autonomic system.
- The Analyse phase examines the data and determines potential actions to be taken to optimise the performance of the system or take corrective action.
- The Plan phase uses the conclusions of the Analyse phase to decide which specific actions should be taken to reconfigure or optimise the managed element.
- The Execute phase translates the planned actions into a sequence of technology-specific commands and implements them.

Making a computing system autonomous increases the opportunities for cost reductions as well as reducing human contact during both the setup of devices and the period of operation. This new computing architecture uses a degree of self-management to facilitate adaptation to unforeseen events occurring within the system. Many of the principles used within autonomic computing can be used within autonomic networking.

Next generation communication networks will be more dynamic, heterogeneous and larger in scale. Conversely, this added complexity could make the network less reliable and more prone to errors that require to be healed. With these new advances in the network it will be increasingly hard to manage therefore a more autonomous system is coveted by network operators.

3.3 Self Organising Networks

Next Generation Mobile Networks (NGMN) created the foundations for the initial use of SON [2, 3, 25, 26, 43, 44] in 2006. 3GPP later adopted this idea for use within LTE. SON is still in development for use within LTE but there are specific use cases defined to govern SON operations. SON allows the network to be more dynamic and optimised than with 2G and 3G systems [25].

SON uses the principles of autonomic networking (Section 3.2) to create a network that is capable of adapting without any human interaction. Due to elements such as spectral efficiency and required traffic, there will be a large increase in the number of base stations within the network making autonomic networking essential in managing the network infrastructure. Femtocells are deployed by customers as well as operators. Such deployment exhibits random temporal and geographical characterisation which in turn leads to an increase in network complexity and management. To facilitate the ad-hoc nature of network architectures, automatic adjustments to packet routing with the eUTRAN is required.

If no autonomic networking is implemented, the time required from technicians to optimise and fix the femtocells could be significant and potentially unsustainable as such deployment increases. Without an autonomic approach, the femtocell's network view will be rendered obsolete shortly after initial deployment and therefore its network use will not be optimal. Currently, within 3G femtocell implementation, plug-n-play functionality does exist commercially. However, significant improvements are required to the auto configuration of new base stations along with the automatic reconfiguration and optimisation of the femtocells themselves to allow them to heal and optimise.

3.3.1 Self-X

The self-management used within a complex networking or computing scenario is known as Self-X and is used to provide some autonomy (*i.e.* configure, optimise and heal) within networking or computing architectures. The concept of Self-X dramatically reduces the input required from system administrators and permits the system to govern itself. In order to achieve self-management to cope with unforeseen changes in the network [42], the system should be dynamic which is achieved using the 3 paradigms depicted in Figure 3.2.

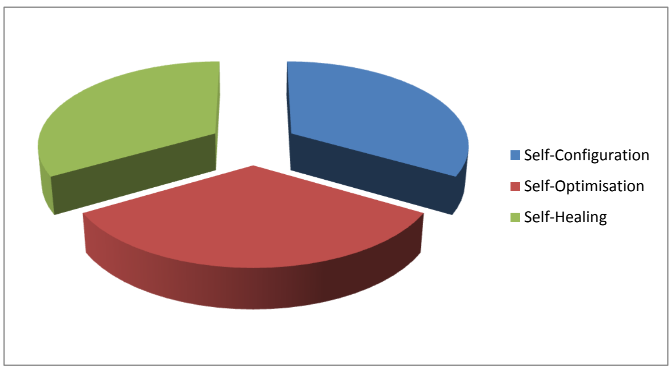

Figure 3.2: The elements of Self-X

Self-Configuration

As new network elements become more complicated to install, configure and integrate with other pre-existing systems, it can be a challenging task to achieve effective network operation. As a result of network complexity, configuration is normally undertaken by a skilled technician which represents a significant cost for the operator.

Self configuration [11, 26] is the process where newly deployed femtocells are configured autonomously. A mechanism is desired during the pre-operational phases of network elements to aid with its planning and deployment. The basic configuration involves creating the logical associations with the network as well as implement the necessary security concepts and handover parameters required for efficient and secure operation within the network. The idea is that when a network element is introduced into a network it should be able to incorporate into the system without any adverse effects to the existing components. If there are any adverse effects to the network as a whole then the network element can reconfigure and recover but, ideally, a newly deployed network component would avoid this from happening.

Self-Optimisation

Active components within a network should be monitored to ensure efficient and effective operation. Optimising networking components and systems can be time consuming and impractical if there are many components in the system, e.g. a cellular network. The complexity inherent in network monitoring results from the large quantity of changeable parameters.

Self-optimisation [11, 26] allows the network components to improve their performance with minimal human input. This can be done by monitoring communications quality and then proactively identifying better ways to complete network operations, *i.e.* changing frequency, handover, etc, while reducing cost and/or improving performance. A typical example of this is, optimisation of the neighbour list stored by the nodes within a network (*i.e.* the list should be up to date, correct and in the most efficient order).

Self-Healing

During normal operation errors or changes can occur within a network that can reduce its performance. Trying to debug and fix issues in a large network can be a difficult task and can take many teams of technicians to solve. When problems happen within consumer networks (*i.e.* Internet Service Provider networks), customers can be affected; the longer the problem takes to fix the more unhappy customers become.

Self-healing [11] is not limited solely to diagnosis and applying solutions for hardware and software issues; it can also be used to detect such issues. When a problem is detected the network element will check log files to diagnose the problem then search for software patches that will solve the problem. Once the software patches are installed, the system will be retested to ensure that the repair was effective. Most issues can be resolved using these processes, only in cases of critical network failure should a network operator have to intervene.

Self-X is useful for the management of highly complex systems. With the addition of femtocells into the LTE infrastructure, the mobile communications network becomes too complex for traditional configuration, optimisation and healing methods. SON functionality minimises the life cycle cost of running a network by eliminating manual configuration and optimisation of the network in all base stations, including femtocells. SON should reduce the unit cost and retail price of wireless data services by reducing the labour required within the communications network.

3.3.2 Expenditure Reduction

Today's mobile networks need to be frequently re-parameterised in order to accommodate updates to coverage and changing traffic loads. Planning, deployment, configuration and optimisation of these network parameters all require significant expenditure from network operators as a consequence of the time and expertise required to maintain the network. The error prone manual-tuning process may also result in non-optimal performance of the network. Changing parameters in the entire network has inherent delays. Self-healing and self-optimising architectures can be used to maintain optimal performance by altering network parameters under interference and overload conditions to mitigate any faults as they occur.

The highly dynamic and complex network structure associated with future communications networks has resulted in an industrial pull from operators to lower both the CAPital EXpenditure (CAPEX) and the OPerational EXpenditure (OPEX) by introducing a degree of self management [45]. CAPEX within the network can be defined as the equipment and setup costs of the network. These costs include those associated with equipment, services and site construction. OPEX within the network is the costs incurred from every day operation of the network. These costs are a significantly large proportion of those incurred by a network and include items such as maintenance, technical support, repairing equipment and energy consumption. Lower energy consumption is a major driver for SON and OPEX reduction. Reducing both CAPEX and OPEX while network complexity continues to increase is a difficult task that can be facilitated through the use of SON [2]. SONs permit tuning of a network to be completed in a timely manner with minimal human interaction.

Approximately 17% of wireless CAPEX is spent on engineering and installation services [46]: This can be mitigated with the use of self-configuration and self-optimisation within SON. Also, 24% of wireless revenue is used for OPEX services like network operation and maintenance, training, power, etc [46]. The use of SON can not only decrease the costs of the network but can also aid energy saving by reducing the power consumed by the equipment used within the network.

Energy is a significant proportion of the expenses of any cellular network and the ability to reduce this is coveted by all network operators. A high level of savings can be gained from utilising the varying load that occurs within the network. These savings can be attained by turning off resources or entering a low power mode when they are not expected to be used, *i.e.* turning off specific base stations during the night. In order to be successful, base stations will have to coordinate their operations. More savings can be achieved by utilising intelligent power saving methods in all base stations within the network, especially femtocells. To reduce the costs of the network (both CAPEX and OPEX) 3GPP have defined a set of use cases that outline the areas in which SON can be implemented.

3.3.3 Use Cases

There are use cases that govern the permitted applications of SON. SON is responsible for a wide variety of network operations ranging from energy saving to handover management. There are 9 use cases for SON within LTE [11, 25]:

- Coverage and Capacity Optimisation
- Energy Savings
- Interference Reduction
- Automated Configuration of Physical Cell Identity
- Mobility Robustness Optimisation
- Mobility Load Balancing Optimisation
- RACH Optimisation
- Automatic Neighbour Relation Function
- Inter-Cell Interference Coordination

Coverage and Capacity Optimisation

Providing optimal coverage throughout the entirety of the mobile communications network can be difficult. Each base station in operation has a limited transmission range that affects the coverage of the network. By increasing the transmission range of each base station, the overall capacity of the network is reduced. Ideally the entire capacity of the network will be constantly in use with no users left unconnected; this is a difficult task. As a result, the requirement for high QoS and good coverage in all sections of the network for UE's in both idle and active mode is a key aspiration. Providing both optimal capacity and full coverage requires a trade-off. If there is a lack of either capacity or coverage then the service received by UE's will be sub-standard. This use case allows coverage and capacity problems to be automatically found through eNodeB and UE measurements and solved with the use of autonomic networking.

Energy Savings

In all aspects of engineering energy saving is now a high priority. Cuts in the energy used by the mobile communication network is a requirement for all network providers in order to reduce the level of OPEX. Ideally, the capacity offered by the network would fit perfectly with the number of UE's currently connected to the network. Absolute efficiency is hard to achieve because the network is highly changeable. SON can help by continually updating the network, according to the current requirements.

Interference Reduction

In order to deploy the network and optimise performance, interference has to be omitted as much as possible. Capacity can be improved by reducing the interference by turning off cells that are not required by the network; if a cell is turned on then it can cause unnecessary interference within the network. Femtocells represent a potentially large interference problem in the network if advanced interference mitigation techniques are not deployed.

Automated Configuration of Physical Cell Identity

When new physical cell ID's are needed, they should be automatically configured to reduce the manual intervention involved within the LTE eUTRAN. Different cells may have to use the same physical cell ID. In order to avoid these physical cell ID's being in the same area, autonomy should be included in the allocation procedure.

Mobility Robustness Optimisation

Mobility represents a key challenge as individuals move in a seemingly random manner making the network management required for effective operations of the mobile device more complex. Random movements can cause spurious handover related actions which can lead to an inefficient use of network resources. In order to avoid any repercussions, handover operations can be altered to optimise network performance. After initial deployment, the process of manually tuning handover operations can be a time consuming task that is generally too expensive to perform. This use case allows for handover operations to be updated constantly throughout the network during the entire period of its operation.

Mobility Load Balancing Optimisation

It is the combination of the network dimensions (number of users, number of base stations, etc), coverage area and mobility of the users that make efficient operation a challenge. SON can mitigate, to some degree, imbalances in load and avoid congestion and under utilisation by distributing users more evenly through the network.

RACH Optimisation

Radio Access CHannel (RACH) is an uplink channel that is used for initial access and uplink synchronisation. Uplink resources can be reserved for the RACH but the quantity of reserved resources should be as accurate as possible to avoid inefficiencies within the network. This is difficult to do manually. When set correctly, short call setup, short handover delays, high call setup success and high handover success can be achieved.

Automatic Neighbour Relation Function

The automatic neighbour relation is responsible for finding new neighbours to the current base station and adding them to the neighbour relation table while removing those no longer active. The increase in size and complexity of the architecture make this a more complex task in modern networks.

Inter-Cell Interference Coordination

Inter-cell interference is a major issue within any mobile communications network. By utilising SON, interference may be reduced in both the uplink and downlink by deploying advanced algorithms that exploit the available resources in the related base stations.

3.3.4 Handover Optimisation

In currently deployed mobile communications networks, manual tuning is the only method for optimising handover parameters. Due to the expense of optimising network parameters, handover parameters are generally only set on initialisation into the network and not afterwards. Infrequent updating of handover parameters leads to sub-standard operation. Mobility Robustness Optimisation is the use case that uses SON to automatically and continuously tune handover parameters with minimal or no human interaction. Optimising handover operations concerns both the self-configuration and the self-optimisation elements of the SON functionality. Moreover, a SON can be deployed to optimise handover performance between neighbouring base stations [18] [47] [48], including femtocells.

Poorly configured handover parameters can impede the performance of the network by causing handover ping-pongs, handover failures and radio link failures. Handover failures are errors that have occurred during the handover process; these are usually invisible to the user. Radio link failures cause the connection perceived to the user to be disrupted or disconnected. Therefore, the main objective of mobility robustness optimisation is to avoid repercussions of badly set parameters that can severely degrade the performance of the network. The cause of handover issues can be categorised into handover too late, handover too early and handover to a wrong cell. Handover too early and handover too late were explained in Section 2.3.1. Handover to a wrong cell [24] occurs when the handover process completes a handover to a cell in which it is not meant to. This is characterised by a radio link failure immediately after a successful handover operation followed by the UE re-establishing a connection to previous cell.

Handover operations are time consuming and costly to the network operator therefore reducing the number of unnecessary and failed handovers is an important objective. Handover parameter optimisation should aim to reduce the number of handovers by detecting any scenario that is caused by incorrect handover configuration and re-configuring the relevant parameters. Parameter optimisation results in reduced numbers of handover too early, handover too late, handover to a wrong cell occurrences as well as reduced inefficiency in the network resources.

Given the plug-n-play requirement for SON, it must be assumed that nobody will be in a position to provide any *a-priori* knowledge concerning the radio environment where a femtocell may be located. Importantly, this influences the design of any autonomous system because every radio environment is unique. Factors such as the placement of a femtocell; the architecture of the building including building materials; the furniture in the building; and the number and location of external macrocell base stations all result in difficulties in pre-programming each femtocell with likely handover locations. Within the macrocell layer, due to factors such as building infrastructure, land layout and ever changing conditions like traffic in cities each cell is unique. As a result, any algorithm used within base stations for handover optimisation must be able to fully configure and optimise itself to the environment in which it resides without any prior information of that environment. To confirm that an optimisation technique has been successful, handover performance metrics are required that can quantify any improvements made.

Handover Performance Metrics

When evaluating handover in an LTE system, there is a requirement to define suitable metrics to represent the network performance. One approach to evaluating network performance is the use of HPIs [18]. There are HPIs for handover failure, dropped calls and ping-pong.

The handover failure ratio ($\mathrm{HPI}_{\mathrm{fail}}$) is defined as:

$$\mathrm{HPI}_{\mathrm{fail}} = \frac{N_{\mathrm{fail}}}{N_{\mathrm{Total}}} \tag{3.1}$$

The dropped calls ratio ($\mathrm{HPI}_{\mathrm{drop}}$) is defined as:

$$\mathrm{HPI}_{\mathrm{drop}} = \frac{N_{\mathrm{Hdrop}}}{N_{\mathrm{Total}}} \tag{3.2}$$

The ping-pong handover ratio ($\mathrm{HPI_{pp}}$) is defined as:

$$\mathrm{HPI_{pp}} = \frac{N_{\mathrm{HPP}}}{N_{\mathrm{Total}}} \tag{3.3}$$

Here, N_{fail} represents the number of handover failures, N_{Hdrop} is the number of dropped calls, N_{HPP} is the number of handover ping-pongs and N_{Total} is the total number of handovers triggered. These metrics, shown in Equations (3.1), (3.2) and (3.3), provide a measure of the performance of the algorithm and are usually expressed as a percentage. Ideally, the number of failed handovers, dropped calls and handover ping-pong occurrences would be zero and hence so would $\mathrm{HPI_{fail}}$, $\mathrm{HPI_{drop}}$ and $\mathrm{HPI_{pp}}$. Unfortunately, practical systems are not ideal and do not operate optimally. In a practical system handover failures, dropped calls and handover ping-pong will occur within the network but the number of such occurrences can be reduced to minimise their effect. Machine learning can be useful for a continuously adapting system in order to optimise how each networking element interacts and optimises handover parameters. Successful use of machine learning could lead to a reduction in the HPIs and, hence, lead to an improvement as compared to standard LTE performance. Neural networks is the machine learning technique used within the work in this thesis as explained in Section 4.2 and shown in later chapters.

3.4 Summary

SON is a promising solution to the problems of future mobile communications networks. Issues such as robustness, performance, energy efficiency, complete and continuous coverage, increased system capacity and network management and optimisation can be improved by the use of an autonomous system, SON. Improving on these concepts will help reduce the OPEX and CAPEX of the network.

Mobility Robustness is one of the use cases of SON. In traditional networks handover parameters had to be adjusted manually. As a result of the costs involved, the handover parameters were rarely adjusted after initial deployment. Within the mobile communications infrastructure and the inclusion of femtocells, mobility robustness must become more autonomous. With the use of SON, handover can be continuously and autonomously tuned to optimise performance.

The following chapters describe the work undertaken to increase handover efficiency within an indoor environment utilising LTE femtocells. Each chapter will cover necessary theory relating to each chapter specifically followed by a formal explanation of the proposed enhancement and finally performance measures for each enhancement.

Chapter 4

Handover Prohibition

4.1 Introduction

With the current trend of deploying femtocells to service indoor users [49], handover between the indoor (femtocell) and outdoor (macrocell) environments now becomes a pertinent issue. The mobile terminal must be able to reliably and seamlessly handover as the user leaves their home or office. In order to achieve this goal, handover optimisation can be used to balance two key conflicting demands: minimising the probability of dropped calls, whilst minimising unnecessary handovers. A system that can achieve this efficiently can be included within the remit of SON in LTE systems [2]. Within this chapter, an approach is explained that aims to reduce all unnecessary handovers within an indoor environment subject to maintaining a limit on dropped calls within an LTE system. Ideally there will be zero dropped calls after convergence.

To achieve the facets of SON, intelligent machine learning concepts can be implemented. The main constraint for a SON system for femtocells is that the specifics of each femtocell environment is unknown when it is deployed. The problem of using position to optimise handover whilst adhering to this constraint limits the potential learning approaches. Neural networks have low complexity, self-learning, adaptable to multiple environments and robust to noise making them a good choice for use within SON in the femtocell environment. Neural networks have the ability to learn an environment and use this knowledge to create a general solution to the given problem. Autonomic networking is the basis of SON, proposed by 3GPP within LTE, that can utilise neural networks. The assumptions being made for this approach are that the position of the user can be accurately detected and that the regions in the radio environment that represent the window and the door are not overlapping and can be determined.

Since Mobility Robustness/Handover Optimisation is one of the use cases of SON defined by NGMN, much work has been undertaken in the area of handover optimisation. No handover prohibition work has been completed until now but other handover optimisation strategies have been investigated. Jansen *et al.* [18] presented a successful parameter optimisation algorithm that involved adjusting the parameters based on the resulting key performance indicators (Handover Failure ratio, Ping-pong handover ratio and call dropping ratio). Kitagawa *et al.* [47] presented results showing that handover can be optimised by altering the handover margin to avoid poor handover performance in a speed varying scenario of 3km/h to 300km/h. Yang *et al.* [50] conducted research on altering the handover message flow in femtocell to macrocell handover to improve its performance. Becvar *et al.* [51] proposed utilising the users distance from a Femtocell to determine whether to alter the Hys. Becvar's work assumes that when the user is close to the femtocell, handover is not required but handover is needed when the users are further away. The novelty of the work presented in this chapter is that handover optimisation is completed on a location specific basis rather than for the entire base station, in a realistic indoor scenario.

A key advantage of LTE femtocells over competitor technologies, such as Wi-Fi, is the ability to support high quality voice traffic. The ability to support seamless handover (and hence retain high voice quality) between indoor and outdoor coverage areas represents a unique selling point of 4G technologies. In order to exploit the potential advantages offered, a handover mechanism has to be adopted that provides a delicate balance between being too timid or too aggressive. A mechanism that is too timid may result in calls being dropped and the detection of a handover too late. Unnecessary handovers place additional demands on the network: through consumption of radio channels and fixed links; through additional processing load in admission control, bearer setting and path switching; and have the potential to degrade the QoS of ongoing connections [38].

Consider an active and mobile user within an indoor environment. When the mobile terminal approaches, and passes through, an exterior door (as shown in Figure 4.1) it will detect an increase in the RSRP [23] from an externally located macrocell. As a consequence, a measurement report will be transmitted from the mobile terminal to the femtocell base station, informing the femtocell that a macrocellular base station has been detected and is a candidate for handover. The femtocell will use the measurement report to initiate handover to the macrocell if required. Now, consider the situation where an active mobile terminal approaches a large window with low penetration loss, as shown in Figure 4.2. The increase in RSRP from the macrocell will cause a measurement report to be transmitted from the mobile terminal to the femtocell and subsequently invoke a handover, as in the previous example. However, as the mobile terminal continues to move past the window, the relatively high received signal level from the macrocell is likely to decline rapidly and thus trigger another measurement report form the mobile terminal to the macrocell, indicating that a better signal can be obtained from the femtocell. This will invoke a second handover, in quick succession, from the macrocell back to the femtocell. Clearly, the second example represents a scenario where unnecessary handover has occurred. The goal of the algorithm presented in this chapter is to identify indoor regions where

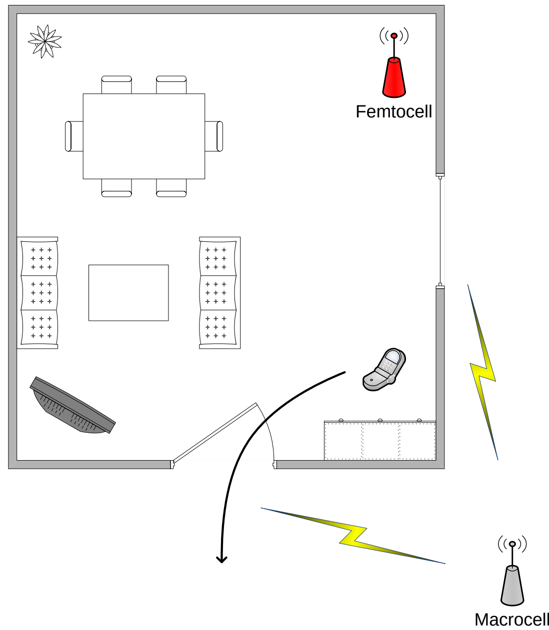

Figure 4.1: Movement scenario 1

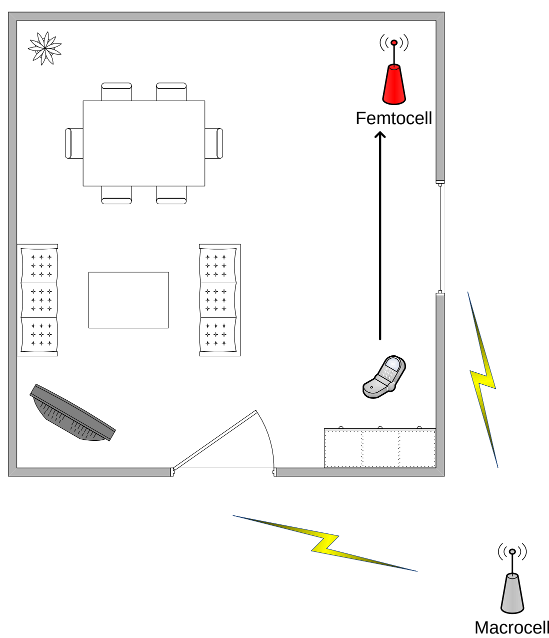

Figure 4.2: Movement scenario 2

handover to macrocell base stations should be permitted and regions where handover should be prohibited. Three principal regions are defined:

1. Areas of low signal strength from the macrocell. In these regions, a measurement report will not be generated and therefore the proposed algorithm need not consider them. For this reason such areas can be regarded as null zones.

2. Areas of high signal strength from the macrocell where few unnecessary handovers occur. These regions are referred to as permission zones since handover to the external base station will be beneficial. It is believed that these zones will coincide with architectural features such as external doors.

3. Areas of high signal strength from the macrocell where many unnecessary handovers occur. These regions are referred to as prohibition zones since handover to the external base station should be suppressed because it is likely that a second handover in the opposite direction will soon follow. These regions will be consistent with architectural features such as windows and glass exterior walls. In a practical system, the number of times handover can be requested when the user moves into this area can be limited by means of the 'reportamount parameter [22]. The level of signalling required for each measurement report is minimal in comparison to each handover that is completed. Two sets of handovers are required for each handover ping-pong.

The zones are depicted in Figure 4.3 to help gain an understanding of the areas within a room, as explained above.

Within LTE, there are a number of tunable parameters [22], among which TTT and Hys are of most interest when optimising the handover process. However, tuning these parameters can be challenging as changes can incur adverse effects. Increasing these parameters reduces the likelihood of unnecessary handovers but also increases

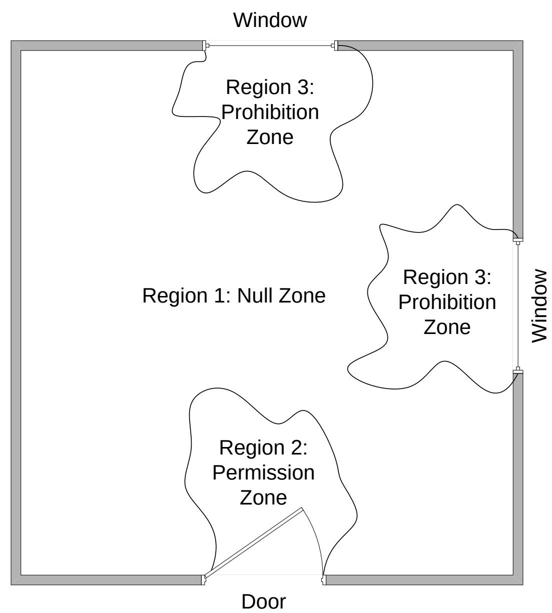

Figure 4.3: Handover zones

the probability of dropped calls; decreasing the parameters has the opposite effect. The scenario considered within this chapter presents the situation where the Hys and TTT are increasing for every unnecessary handover to such an extent that an active call will be dropped when it genuinely requires to handover to a macrocellular base station.

The problem under investigation in this chapter is how to facilitate handover to the macrocell layer whilst minimising unnecessary handovers. Reducing the number of unnecessary handovers increases the energy efficiency of the femtocell resulting from lower signalling within the network and more efficient use of the network resources. To facilitate the handover algorithm, positional information is incorporated in order to optimise the handover decision locally and minimise any adverse effects of parameter alterations (for an entire cell). For clarity, it should be noted that the positional information used in this algorithm is the location of regions within the radio environment in which handover occurs and not the true physical location of the user. However, there may be strong correlation between both of these forms of location. The problem is complicated by the fact that every building has a unique radio environment which is a function of: the femtocell base station placement, the architecture of the building (including building materials), and the number and location of external macrocell base stations. Therefore, each building will have a unique topography of permission and prohibition zones. Given the economic drivers for autonomic approaches, the femtocell base station cannot be assumed to be pre-programmed with this information in advance. Rather, the femtocell should be able to configure and optimise its performance with experience. In terms of the types of machine learning approaches that can be applied to this problem, supervised learning strategies are therefore not applicable. The SOM(discussed in Section 4.3) is a type of neural network that is particularly useful in this context by continually mapping regions where either successful or unnecessary handover has occurred and using this information to identify the periphery of the permission and prohibition zones.

4.2 Neural Networks

Neural networks (sometimes referred to as Artificial Neural Networks (ANNs)) [52, 53, 54, 55] are a type of biologically inspired intelligent system that attempts to mimic neural processes that occur within the brain. Neural networks can be used to solve a variety of problems within applications that require a form of clustering, pattern recognition or prediction.

Research on neural networks and their application for telecommunications management for many different purposes has been undertaken. A few examples of the applications will now be described. Gardner *et al* [56] utilised a SOM within the wired communications domain to detect fault scenarios in an SDH-based environment. Akoush *et al* [57] present a successful example of using a novel hybrid Bayesian neural network model to predict user locations within cellular networks. Espi *et al* [58] implemented a Hopfield neural network approach to network selection for multihomed hosts. The research has shown that neural networks can provide many improvements to the field of communication network management. However, operators are disinclined to implement these changes due to a lack of control over the outcomes and the ability to trace why the outcome was chosen. Intelligent network management will be needed in future networks to compensate for the additional size and complexity of the network. This intelligence requires a generic approach to compensate for the variety of environments that the devices will be deployed within.

Intelligent systems are useful within a specific environment/scenario but not generic enough to solve a problem in all environments/scenarios. The benefit of adopting neural networks is that the technique is flexible and the underlying principles can be generalised and then applied to a range of environments. Neural networks are useful as generic processes that can be applied to any situation and dynamically adapt and optimise themselves to the environment deployed within. The generality of the processes is derived from biological inspiration.

4.2.1 Biological Inspiration

A neuron [52, 59] is a biological cell that takes inputs, processes the information and outputs a result. Each cell is comprised of a soma (cell body), axons and dendrites, as shown in Figure 4.4. As the receivers of signals from other neurons, the dendrites represent the input path to the cell. There are many dendrites connected to each soma. The soma has a nucleus that processes all signals received by the dendrites. The axon is the output of the neuron and allows signals to be passed down its length.

There is only one axon connected to each soma and the axon can be considerably longer than any other component of the neuron. The axon ends in smaller branches called terminal branches which eventually lead to the dendrites of another neuron. In this way, the neurons form a network.

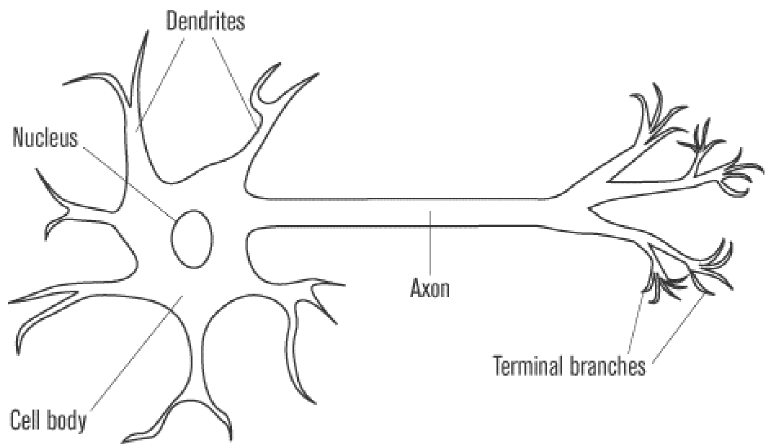

Figure 4.4: Structure of a biological neuron [60]

A network of neurons providing a significant level of processing enables the ability of the human brain to make such complex and rapid decisions. For the high level of computing processes that are completed within networks, mimicking the processes of the human brain is a powerful approach. The brain has the ability to:

- accomplish high levels of parallel processing
- generalise ideas and apply them to multiple applications
- adapt to changing scenarios
- be highly fault tolerant

The aim of neural networks is to provide these abilities to computing systems.

4.2.2 Artificial Neurons

Neural networks are biologically inspired algorithms that allow computing systems to artificially represent biological neural processes that occur within the human brain. Each neural network is a mathematical model that processes information in a manner that shares properties with processes of the human brain. Each neuron within the brain has the ability to accept an input, process the data and output a result; artificial neurons work in a similar manner (Figure 4.5). Many inputs can be provided to the neuron but only one output can be released. However, this output can be used to manipulate multiple elements within the applied environment.

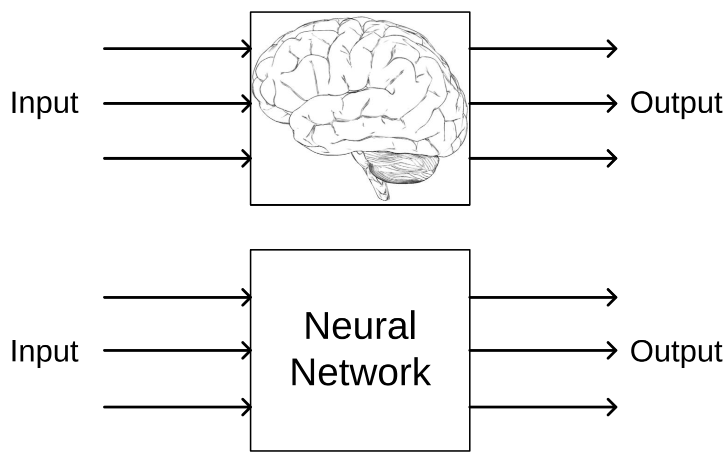

Figure 4.5: Artificial neural networks and the brain

A neural network is a network of simple processing elements that combine to achieve complex global behaviour that is determined by the structure and aim of the network. Each neural network is an adaptive system that allows its structure to change based on current requirements to solve a specific problem. In essence, neural networks learn by example.

As depicted in Figure 4.6, an artificial neuron has many inputs and a single output. The structure of an artificial neuron is similar to that of a biological neuron (Figure

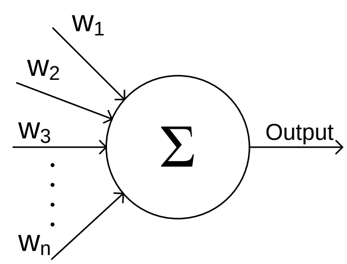

Figure 4.6: An artificial neuron

4.4). The inputs can be compared to dendrites, the centre is similar to the soma, and the axon is comparable to the output of the artificial neuron. In many neural networks the output of one neuron is fed into the input of another, creating a highly complicated network architecture.

4.2.3 Applications

The range of potential applications for ANNs is wide. ANNs are very useful in applications where a task is too impractical to be completed by hand. Situations where the resulting outputs of a task can be highly dynamic can be difficult and time consuming to be completed manually. The main application areas for ANNs are clustering, pattern recognition/classification and prediction.

Clustering is the process of finding a structure in a collection of unlabelled data. Organising objects into groups whose elements are similar in some manner is the aim of any clustering algorithm. A cluster is a collection of elements that are similar to each other and dissimilar to the elements within other clusters. The decision of how to organise data into a series of clusters can be both difficult to accomplish and hard to verify.

Pattern recognition is the process of assigning labels to given input values. Classification is a type of pattern recognition that assigns each input value to one of a given set of classes. Pattern recognition and classifying each input can be a difficult task. The use of ANNs for this task eliminates user input and autonomously completes the process.

Prediction is the process of forecasting a future value, usually based on information gathered from the past and current states. The ability to learn from experience makes ANNs ideal for use to predict the future. Highly dynamic systems can be difficult to predict future results but ANNs have the capability to detect trends in the data and predict an outcome.

4.2.4 Architectures

The ANN is a highly complicated interconnection of simple processing units. The structure of the neurons and the way in which they are linked together indicates the types of neural networks that can be used. The structure also has a strong relation to the type of learning used and more specifically the learning algorithm used. Generally, there are 2 different types of network architectures: feedforward networks and recurrent networks [59, 61].

Feedforward networks allow a one directional flow of information through the network. The structure of the neurons is into that of layers: input layer, hidden layers and an output layer, an example is shown in Figure 4.7. Information flows along connected pathways from the input layer of neurons, through any hidden layers to the output layer. There is no feedback involved thus, the output does not affect any other layer within the network.

There are 2 types of feedforward neural networks: single layer and multilayer. Within single-layer feedforward neural networks there are only input and output layers. Within multilayer feedforward neural networks there are hidden layers that are not seen directly from either inputs or outputs of the network.

Recurrent networks are different to feedforward networks because they include at

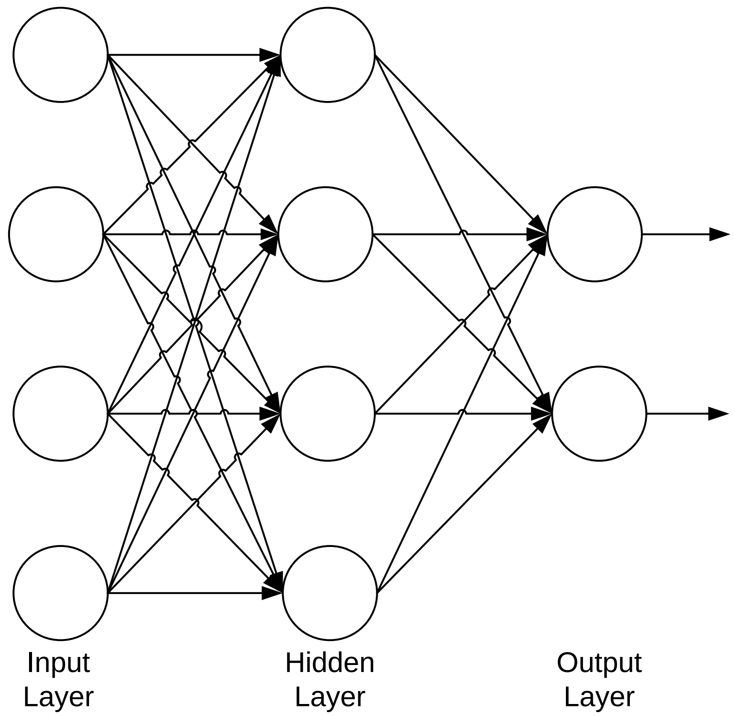

Figure 4.7: An example of a multilayer feedforward neural network

least one feedback loop within the network structure. An example is shown in Figure 4.8. The feedback can either lead back to another layer in the network or to the input of the same neuron. The addition of feedback makes recurrent networks more adaptable than feedforward networks and enables them to "learn" from experience more accurately. When feedback has been received by a neuron, the inputs are modified which then changes the state of the network.

A SOM consists of an input layer, a weight layer and an output layer, as depicted in Figure 4.9. Each input is connected to all the weights within the weight layer and the output layer is a product of both the inputs and the weights. Associated with each neuron is a weight within the weight layer. A SOM can be regarded as a special case of a feed forward neural network.

Different network architectures require different types of learning. Learning occurs

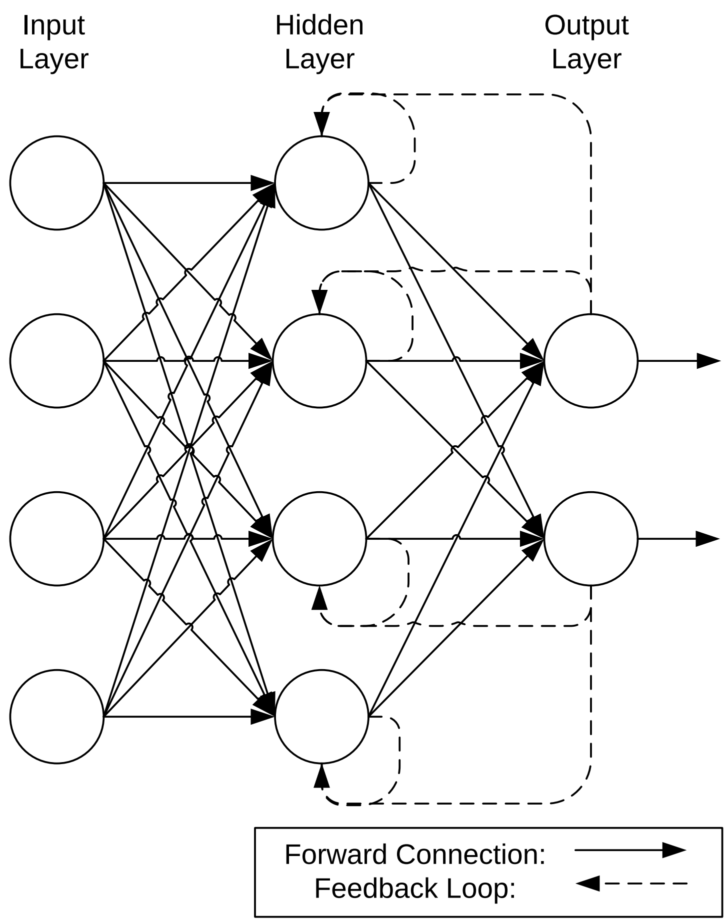

Figure 4.8: An example of a recurrent neural network

via updating the neurons within the network, the connections between them and the architecture of the network as a whole. This is accomplished by updating the weights within the neural network.

4.2.5 Learning Methods

Learning is a fundamental benefit of many intelligent systems [54, 59]. By enabling the ANN to adapt over time, the performance of the ANN should improve over time by iteratively updating the weights within the network. The ability to "learn" from

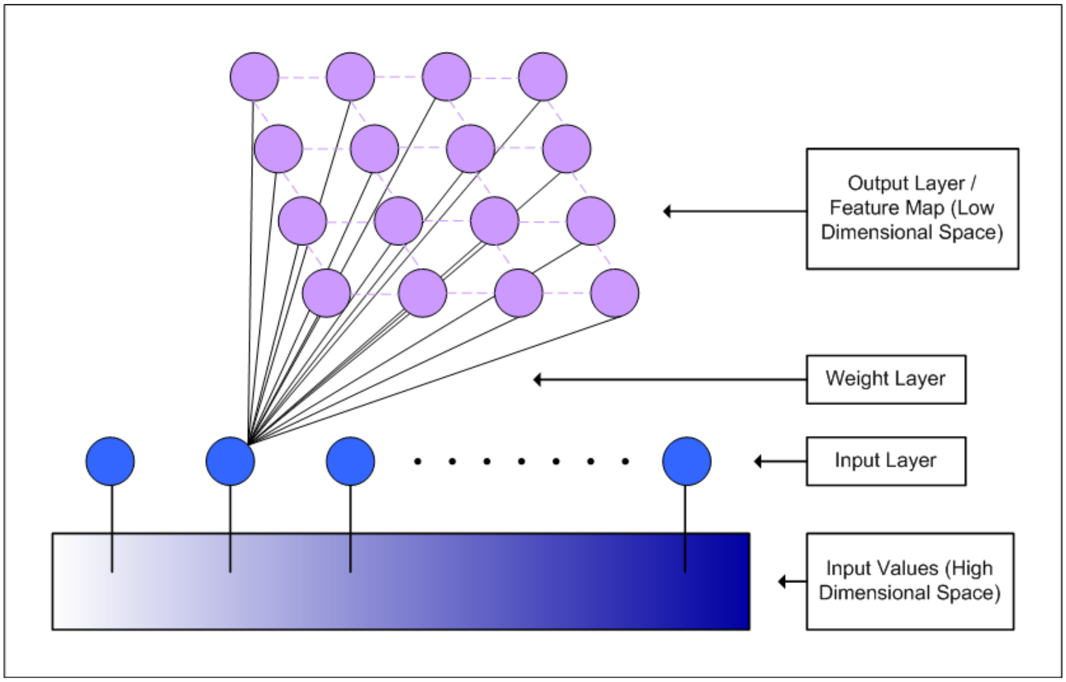

Figure 4.9: SOM structure

experience makes ANNs suitable for a range of applications where other 'intelligent' schemes have limited effectiveness.

In order to complete any learning process there are prerequisites. To select an appropriate learning type from supervised, unsupervised or reinforcement, an understanding of the environment that the neural network will be deployed along with the inputs to the network is required. Different environments necessitate different techniques from the ANN with regards to the appropriate learning mechanism. There are 3 main learning paradigms that are used with ANNs:

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

A supervised learning system requires to be explicitly "taught" during a given training period. During this training period, the distribution of the training data should match that of the problem set. For each input, the output is checked to ensure correctness. If the output is wrong then an error has incurred within the process and should be investigated. The ANN can then be modified accordingly.

An unsupervised learning system does not require any explicit "teaching" in order to operate. However, this means that the ANN is not provided with any knowledge of the outputs that it should produce for future arbitrary inputs. The network will "learn" on its own using past experience. With unsupervised networks, high accuracy of the input to output correlation is not guaranteed but they can be deployed in a plug-n-play fashion with no operator input. A SOM is a type of unsupervised learning neural network.

A reinforcement learning system is a form of supervised learning where no output is provided to the ANN, only whether the output is correct or wrong. The correct actions are discovered by exploration of the allowed outputs and exploitation of the output that yields the highest reward for the given input. Depending on how correct or wrong the output is, a reward is provided to the network; the aim is, obviously, to maximise the rewards gained.

The decision on which type of learning to use within an environment can be a difficult task. Within the work here, an unsupervised learning approach has been chosen. Unsupervised learning has been chosen because the learning of the weights for each neuron has been detached from the learning of the communications system. As such, a SOM is a useful approach for retaining knowledge of the the locations that handover is more likely to take place within the radio environment. However, different scenarios have different demands and thus the appropriateness for each learning technique varies (*i.e.* a SOM is not suitable for all applications). Reinforcement learning could also be used for this task and successful handover, handover too early and handover too late could be used to define good or bad operation of the neural network. Evaluation and understanding of the application for the ANN is key to its success.

The learning utilised within the optimisation of handover is separated into two parts. The first part of the learning is the use of a SOM to retain knowledge of the locations that handover can occur within the radio environment and control which neurons learn from each input. The second part of the learning is for a thresholding system to be put in place. The thresholding system is used to prohibit handover when the user location is mapped to a neuron that has a history of unnecessary handovers. When an unnecessary handover occurs, the number of unnecessary handovers is incremented for each neuron learning from that input. Once this value is higher than a threshold, handovers are prohibited. This two stage learning process is useful for optimising handover within an indoor environment and the use of a SOM is the key component for utilising position to optimise handover.

4.3 Self Organising Map

4.3.1 Theory

The SOM [62, 59] was devised by Teuvo Kohonen in 1982 as a type of unsupervised neural network that creates a low dimensional, discretised representation of an input space and uses this to determine clusters of neurons. It can be considered as an abstract mathematical model of a topographic mapping that occurs within the cerebral cortex. It differs from other neural networks in that it uses only the input and the configuration of the network to generate the output: there is no neighbourhood function to preserve the topological properties of the input or oracle to govern the results.

A SOM is an effective tool for data visualisation due to its ability to view high dimensional data in low dimensions; in this way, it also produces a similarity graph of the input data on a low dimensional display. As the SOM thereby compresses information while preserving the original data, it can also be thought of as creating abstractions of the initial data. Generally, the input will be of high dimensions and the neurons and output are arranged in either a one or two dimensional lattice. Generally, a two dimensional lattice is preferred; the case in this work. Through the application of SOMs, non linear statistical relationships are converted into simple geometric ones. The lattice can be regarded as a special case of a feed forward neural network with a single computation layer. Within SOM, all neurons are connected to all inputs and, learn in a cooperative manner based on distance from the input. More technically, SOMs are a special type of a recursive regression process where only a limited subset of neurons adapt at each step. In order to retain knowledge of past events, each neuron has the ability to learn and remember past events which is a valuable trait in any autonomic managed element. A SOM algorithm consists of four phases which describe the learning process: initialisation, competition, cooperation, and synaptic adaptation.

- Initialisation: The weights within the network are uniformly distributed within the region of the network. The initial values for the parameters are set here.
- Competition: Each time an input to the network is detected, the weights within the network are compared to the input. The most similar weight to the input is deemed the winner
- Cooperation: All weights within the region of the winning weight (calculated based on a monotonically decreasing sphere of influence) are updated based on their distance from the winning weight; this corresponds to the neurons learning from the input

- Synaptic Adaptation: The parameters that govern the learning of the neural network are updated. This ensures that the system will tend to a solution and not learn indefinitely

The autonomic managed element (discussed in Section 3.2) can be accomplished using 4 stages: Monitor; Analyse; Plan; Execute. The Monitor phase of the SON algorithm is comprised of determining the location of the user by detecting where a handover measurement report has been triggered. The Analyse phase of the algorithm is based on a Kohonen SOM and enables the femtocell to learn the locations of the propagation environment that correspond to both permission and prohibition zones. Next, the Plan phase takes this information and decides on an appropriate response; *i.e.* to permit or prohibit the requested handover. This phase also creates a profile of locations in which handover may take place. Finally, the Execute phase translates the decision from the Plan phase into LTE specific commands and permits or prohibits the handover request.

4.3.2 Mathematical Approach

The Kohonen SOM is particularly useful for detecting clusters within data. Here it is used to perform location fingerprinting based on RSRP and Angle of Arrival (AoA). The four phases of SOM: initialisation, competition, cooperation, and synaptic adaptation will now be described in detail.

Initialisation

Initialisation of the SOM network is concerned with pre-setting the individual weight values of each neuron in the lattice as shown in Figure 4.9. The initial values are drawn from a uniform distribution within the area of the lattice (the propagation region of the femtocell), as shown in Figure 4.10.

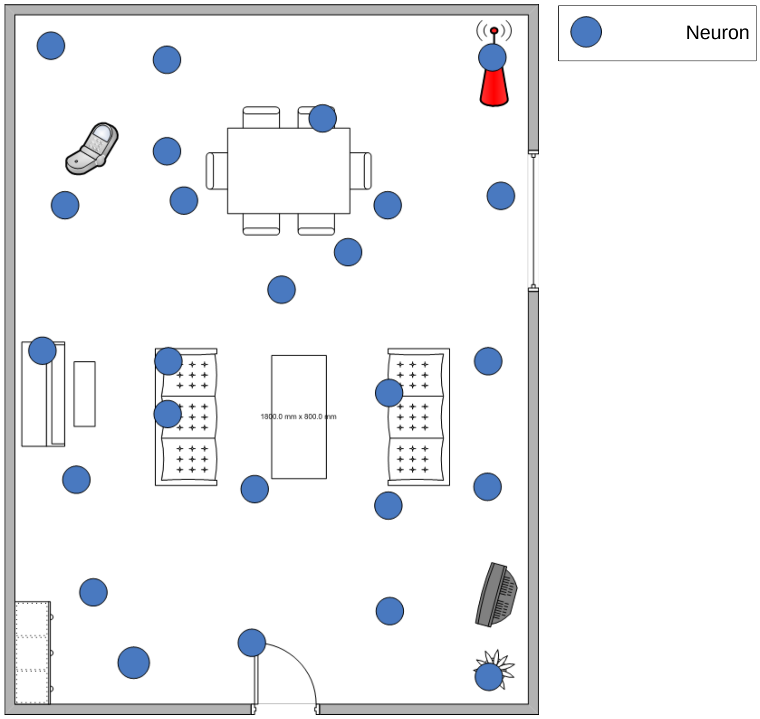

Figure 4.10: Initialisation stage of SOM

Due to the application proposed, the initialisation of the weight positions in the SOM algorithm must be completed randomly but, even, initially unordered vectors will become ordered as the algorithm progresses. Each neuron input will be associated with a weight and this represents the geographical location obtained using the RSRP and AoA from the mobile terminal at the time the measurement report was generated. Each input will be associated with a weight within the higher dimensional feature space. Since the algorithm is unknown, the neurons within the network will require time to learn the algorithm and cannot be initialised in specific places within the network (*i.e.* prohibition and permission zones).

Competition

The next step of the process is for the inputs to be applied to the algorithm. During normal operation this would occur every time a mobile terminal generates a measurement report. Since each input is connected to each neuron, the input and weight vectors have the same dimensions. The representation for an a-dimensional input (user location) is defined in Equation (4.1) and the weight vector associated with each neuron in the lattice is defined in Equation (4.2).

$$\mathbf{x} = [x_1, x_2, ..., x_a]^T,\ \mathbf{x} \in \mathbb{R}^a \tag{4.1}$$

$$\mathbf{w}_j = [w_{j1}, w_{j2}, ..., w_{ja}]^T,\ j = 1, 2, ..., l,\ \mathbf{w}_j \in \mathbb{R}^a \tag{4.2}$$

Here, l is the total number of neurons in the network.

Given that there is no activation function, the output of each neuron will be a combination of both the input and weight vectors. From a geometrical perspective the winning neuron is calculated using Euclidean distance, therefore, the shorter the Euclidean distance, the closer the weight vector is to the input vector. The competitive aspect of this algorithm is that the neuron whose weight vector provides the best match to the input vector will produce the lowest output and will be selected as being the winning neuron, as shown in Figure 4.11.

If the index of the winning neuron is denoted by $i(\mathbf{x})$ within the lattice $\mathcal{L}$ (denoting the grid of neurons in the weight space) then the winner is given by Equation (4.3).

$$i(\mathbf{x}) = \arg\min_j \|\mathbf{x} - \mathbf{w}_j\|,\ j \in \mathcal{L} \tag{4.3}$$

Once the winner has been selected as the closest match to the input, it can be utilised by the next stage of the SOM algorithm.

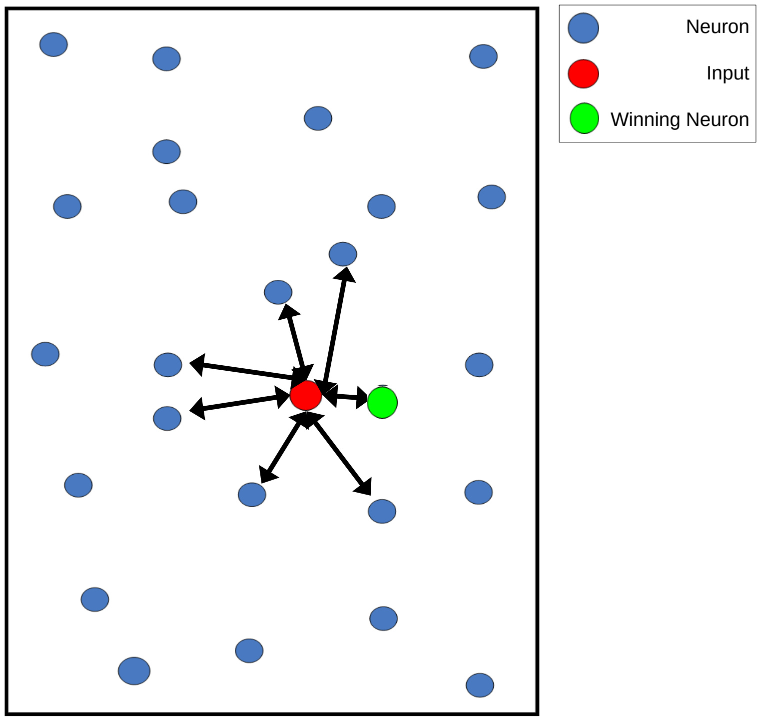

Figure 4.11: Competition stage of SOM

Cooperation

Once the winner for a given input vector has been selected, the weights of the neurons within the winner's sphere of influence are updated, as depicted in Figure 4.12. This constitutes a cooperative learning process since, unlike other competitive learning strategies, it is not just the winning neuron that has its weight values modified. This group learning strategy permits the network to converge more quickly and accurately compared to the case where only the winner would modify its weights.

The sphere of influence is governed by a neighbourhood function which determines how many of the winner's neighbours can undergo learning, and also the degree to which they will learn. Within the sphere of influence, neighbours closer to the winning neuron will have their weights updated by a greater amount than those located further

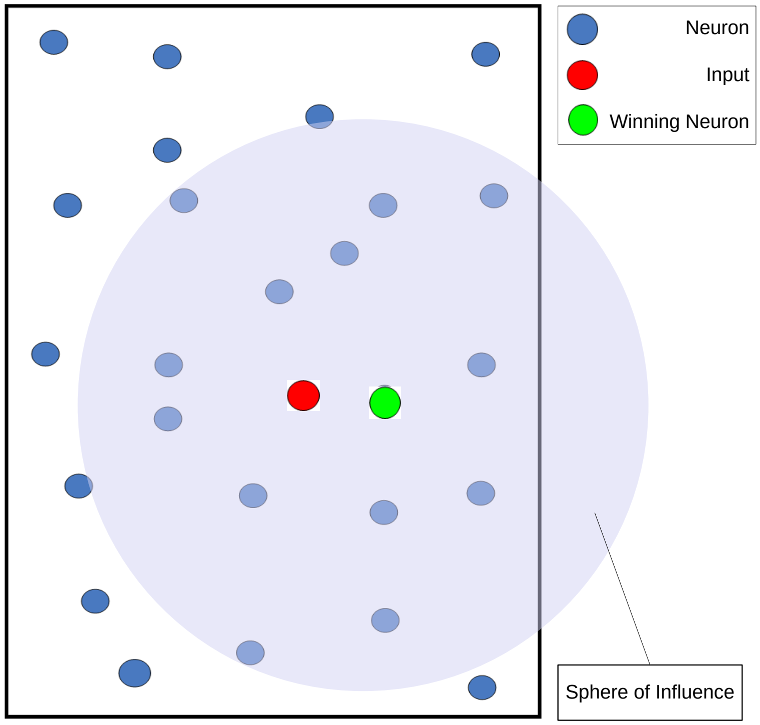

Figure 4.12: Cooperation stage of SOM

away. In order to achieve this, a distance metric between neurons in the lattice is required, where the distance between two neurons e and f is given by Equation (4.4).

$$d_{f,e} = \|\mathbf{r}_f - \mathbf{r}_e\| \tag{4.4}$$

Here, $\mathbf{r}_e$ and $\mathbf{r}_f$ are the locations of neurons e and f in the lattice respectively.

The neighbourhood function should decay monotonically with distance from the winner. Furthermore, the neighbourhood function should be the maximum at the winner ($d_{f,e} = 0$) and decay to zero as $d_{f,e} \to \infty$. A popular choice for the neighbourhood function which satisfies these requirements is the Gaussian function as shown in Equation (4.5), and it is this function that is adopted in this work.

$$h_{f,e} = \exp\left(-\frac{d_{f,e}^2}{2\sigma^2}\right), \; e, f \in \mathcal{L} \tag{4.5}$$

The parameter σ defines the width of the Gaussian function. In essence σ determines the size of the sphere of influence around the winning neuron. Special consideration needs to be made with the size of the neighbourhood function at initialisation. If the neighbourhood is initialised to be smaller than the region of the map then, the map will not be ordered globally resulting in weights positioned in incorrect locations and potentially increasing the vector quantisation error. When using a Kohonen SOM, the size of the sphere of influence (*i.e.* σ) is reduced over time ($h_{f,e} \to 0$ as $t \to \infty$); in practice this translates to number of iterations ($h_{f,e} \to 0$ when $n \to \infty$), n. The width of the neighbourhood function can be made to decay with time by making σ decay with time. Here, we realise this by assigning an exponential decay to σ as shown in Equation (4.6):

$$\sigma(n) = \sigma_0 \exp\left(-\frac{n}{\tau_1}\right) \tag{4.6}$$

n is the iteration number, σ_0 is the initial value and τ_1 is a temporal decay time constant chosen by the designer.

By incorporating temporal decay, Equation (4.5) can now be re-written as Equation (4.7).

$$h_{f,e}(n) = \exp\left(-\frac{d_{f,e}^2}{2\sigma^2(n)}\right) \tag{4.7}$$

Synaptic Adaptation

The adaptation process is concerned with the execution of the weight update procedure for all neurons within the sphere of influence of the winner. This involves utilising not only the sphere of influence but also a learning rate. Generally $g(y_j)$

represents the rate of learning and is a positive scalar function of neuron j's output. An appropriate choice for this function is given by Equation (4.8):

$$g(y_f) = \eta y_f \tag{4.8}$$

Parameter η is the learning rate. In practice the learning rate also decays with time (or iterations), as shown in Figure 4.13; therefore, it is a decreasing function as shown in Equation (4.9):

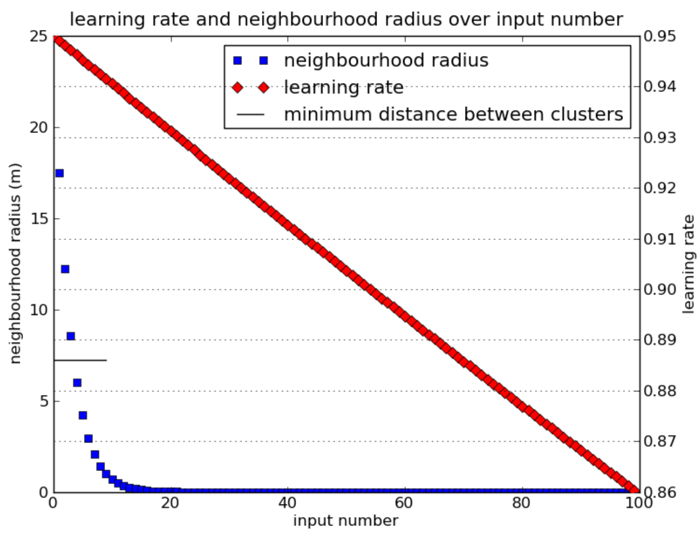

Figure 4.13: Decay of learning rate and neighbourhood function

$$\eta(n) = \eta_0 \exp\left(-\frac{n}{\tau_2}\right) \tag{4.9}$$

η_0 is the initial value and τ_2 is a second time constant. The augmented Hebbian weight update equation can be written as shown in Equation (4.10):

$$\Delta \mathbf{w}_f = \eta y_f \mathbf{x} - \eta y_f \mathbf{w}_f \tag{4.10}$$

By setting $y_f = h_{f,e(\mathbf{x})}$ the weight update equation can be written as shown in Equation (4.11).

$$\Delta \mathbf{w}_f = \eta h_{f,e(\mathbf{x})} \left(\mathbf{x} - \mathbf{w}_f \right) \tag{4.11}$$

Thus, the weights for neuron j within the sphere of influence of the winner are updated iteratively according to the rule given by Equation (4.12) as shown in Figure 4.14.

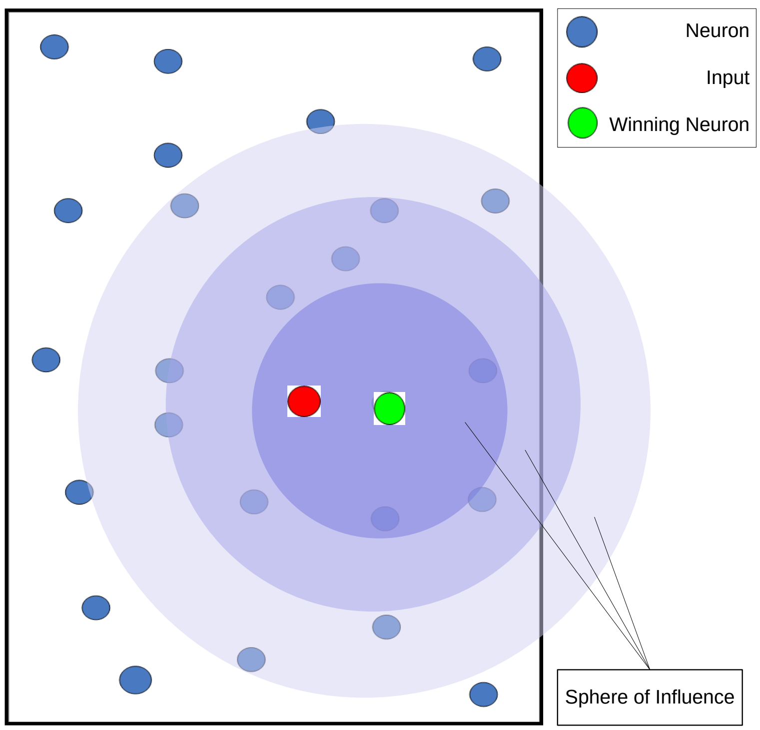

Figure 4.14: Synaptic adaptation stage of SOM

$$\mathbf{w}_f(n+1) = \mathbf{w}_f(n) + \eta(n) h_{f,e(\mathbf{x})}(n) \left(\mathbf{x}(n) - \mathbf{w}_f(n) \right) \tag{4.12}$$

Over time the neurons are continually updated based on (4.12). The parameters of the neurons adapt towards the optimal locations based on the learning rate and the neighbourhood function. Over time, both the learning rate and the neighbourhood function decrease (Figure 4.14) and become very low until no other neuron is updated other than the winning neuron. Once this happens, the locations of both permission and prohibition zones have been identified.

4.4 Simulation Model and Results

Within NS3, a model has been created that simulates a user walking around a given area. The femtocell detects the location of the user when a handover is requested (Monitor stage), analyses the requirement for handover using the SOM algorithm (Analyse stage), plans whether to allow or suppress the handover (Plan stage) and executes the decision in a way that adheres to the requirements of handover in LTE (Execute stage). The results can then be recorded and stored in order to evaluate the effectiveness of the algorithm.

Within the simulation environment, parameters have been set to estimate the performance of a real-world LTE system and assess the improvement that the proposed algorithm will have. Within case studies one and two, the parameters have been set for use within a large room that could represent a hall or small business (*i.e.* a coffee shop or small office). The random walk mobility model allows for a random change in direction after a defined period of time or distance travelled as shown in Figure 4.15. A Random Direction mobility model has also been used that allows for a random direction to be chosen whenever the user meets a wall, as shown in Figure 4.16. The characteristics of the user within the mobility models have been modified for instances that the user enters both prohibition and permission zones to mimic the behavior of people within these areas in a physical environment. The mobility of the user at a Permission zone has been altered to allow the mobile user to walk through the door when a prohibition zone has been entered. Also, when a mobile user walks within the region of a Prohibition zone, the mobility of the user will mostly continue past the window but occasionally will pause by the window for a short duration. A single-slope, distance based propagation model has been created that defines the propagation characteristics perceived by the terminal based on its location within the propagation environment. The RSRP of both a single macrocell and a single femtocell are used to determine the requirement for handover. Both the propagation model and the mobility model are used, effectively, to highlight the potential performance of the algorithm.

In order to demonstrate the effectiveness of the algorithm, case studies will be considered that represent typical performance. Case study one incorporates one prohibition zone and one permission zone and scenario two incorporates two prohibition zones and one permissive zone. The simulation details summarised in Table 4.1 are common to case studies one and two. Case study one involves the use of a Random Walk mobility model and case study two utilises the Random Direction mobility model. These case studies will be presented together and they demonstrate that the choice of model does not effect the generality of the results because the algorithm works in an event-based manner and not in a temporal manner.

When running the model, it adheres to the handover triggering process defined for LTE but the SOM algorithm becomes part of the decision process utilised by the femtocell. The location of the user when the handover trigger occurs represents the Monitoring stage of the autonomic element. The locations that handovers can occur within the simulation environment are where the macrocell RSRP is greater than the femtocell RSRP. The handovers triggered can be femtocell to macrocell or macrocell to femtocell handovers, and are shown in Figures 4.19 to 4.26.

Once the Monitor stage of the autonomic element has taken place, the Analyse stage occurs. Here, the SOM algorithm is utilised and the location of the user is considered in analysing whether handover should be allowed or prohibited. The handover

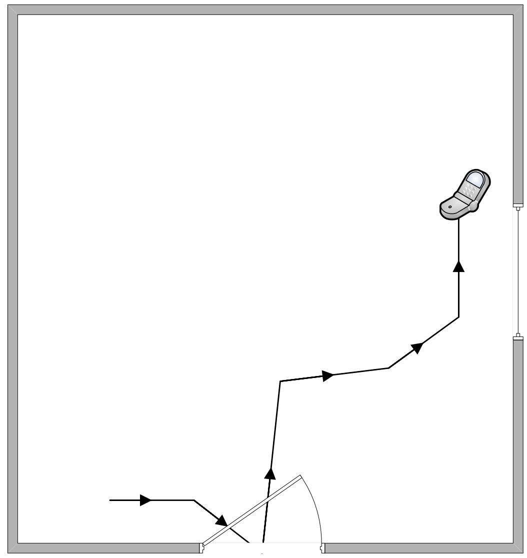

Figure 4.15: Movements of a user governed by a random walk mobility model

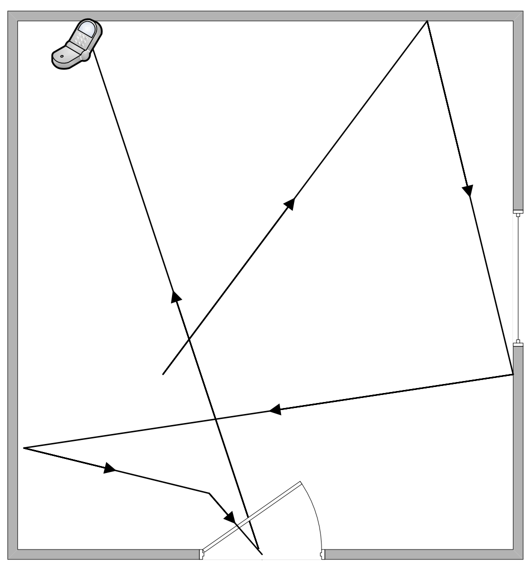

Figure 4.16: Movements of a user governed by a random direction mobility model

Table 4.1: Simulation Details

Parameter	Value
Simulation dimensions	40 m × 40 m
Room dimensions	40 m × 32 m
Exit area	40 m × 8 m
No. of mobile terminals	1
Direction change time	1.0 sec
Movement speed	2 - 4 m/sec
Initial position	centre
Mobility model	random walk or random direction
Propagation model	single-slope
Hys	5 dB
TTT	320 ms
Error	0 m
Neurons	100

regions and whether they are doors (permission zones) or windows (prohibition zones) are shown in Figures 4.17 to 4.18. Using the information gathered at the Analyse phase, the Plan phase then decides whether to allow or suppress the handover request. The decision as to whether to allow or deny the request, as well as the location of the user at the time of the request is shown in Figures 4.19 to 4.26. Each figure depicts the locations of 100 suppressed handovers and the equivalent set of handovers that have been permitted. Note that these are co-incident with the locations of the window and door. Within Figures 4.19 to 4.26, it can be seen that the regions that the handovers are permitted and prohibited are initially undefined, leading to handovers being permitted everywhere. Such behavior is because the algorithm is initially timid and non-restrictive at initialisation and is required for the algorithm to learn while not disrupting the success of required handovers within the LTE system. Later in the simulation, the parameters of each neuron within the neural network have resulted in clusters which are then used to permit/prohibit handovers. The neurons within the SOM neural network have the ability to retain knowledge of previous successful and

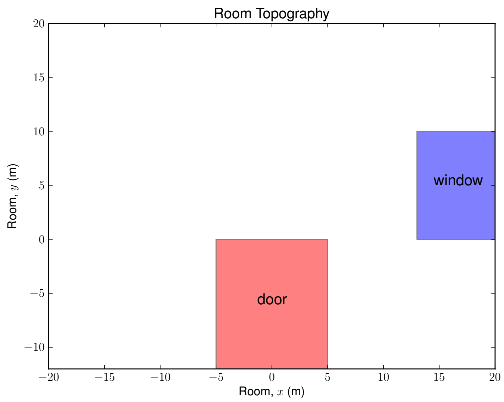

Figure 4.17: Case 1: Room topography

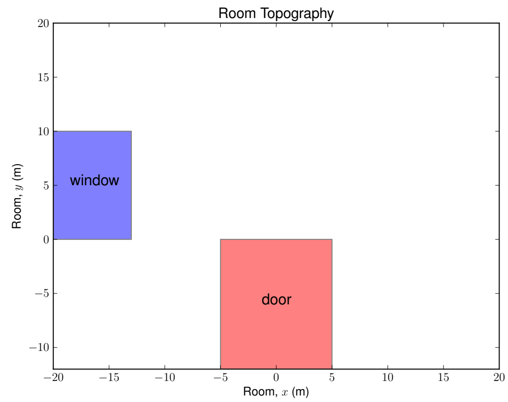

Figure 4.18: Case 2: Room topography

unsuccessful (unnecessary) handovers and optimise the handover performance over time. This results in handover only being allowed within the permission zone and only suppressed within the prohibition zone. It should be noted that the algorithm is given no prior information regarding the location of the windows, doors or where handovers will occur. This knowledge is gained by trial and error.

The learning that takes place is of an unsupervised nature. The algorithm first gathers information about its environment before optimising the handovers that occur. At initialisation, all handovers are allowed to occur in order to preserve the success of required handovers within the indoor environment. Once handovers occur, it can detect instances where ping-pong handover take place and aim, over a period of time, to prohibit handover in this region. This process constitutes the learning time of the algorithm. The longer the run time, the greater the level of handovers that can be prohibited until the neurons converge. Figures 4.19 to 4.26 diagrammatically show what handovers were permitted and prohibited as they occurred over time within a simulation run. Each figure depicts one hundred prohibited handovers and the corresponding permitted handovers during this time. Figures 4.27 and 4.28 show the percentage of unnecessary handovers that were not prohibited over handover

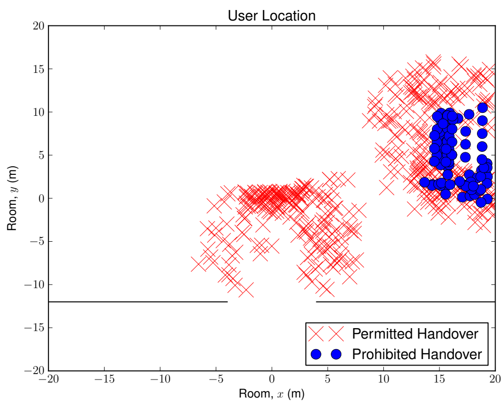

Figure 4.19: Case Study 1: Handover prohibitions 0 to 99

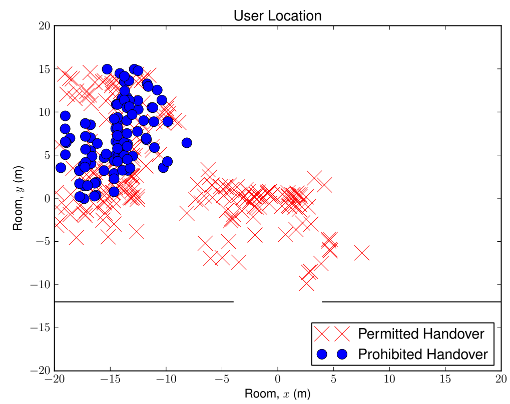

Figure 4.20: Case Study 2: Handover prohibitions 0 to 99

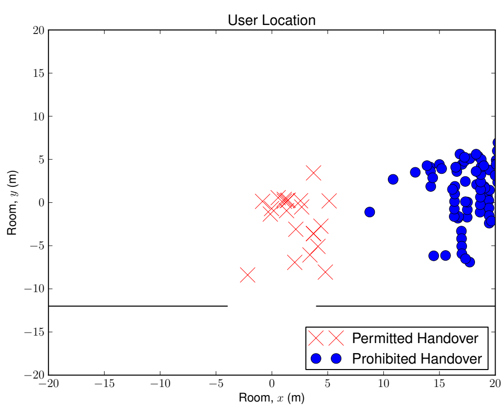

Figure 4.21: Case 1: Handover prohibitions 300 to 399

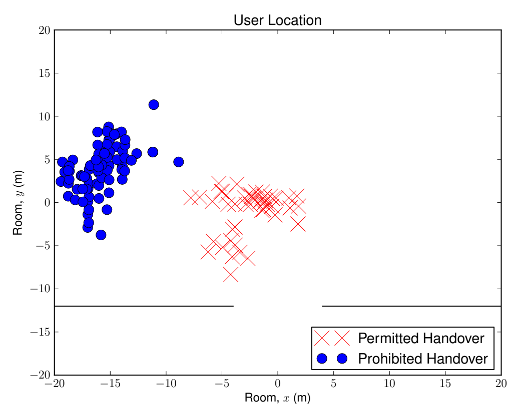

Figure 4.22: Case 2: Handover prohibitions 300 to 399

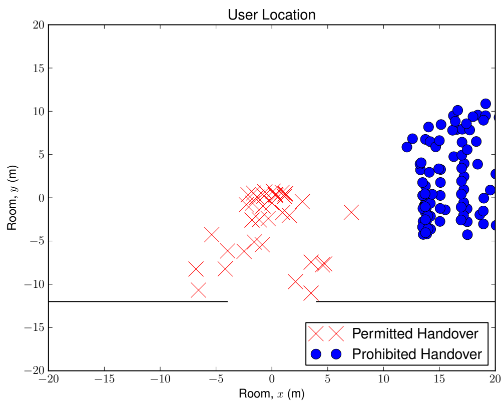

Figure 4.23: Case 1: Handover prohibitions 600 to 699

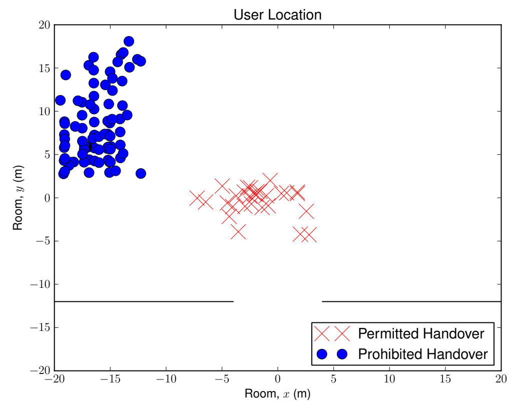

Figure 4.24: Case 2: Handover prohibitions 600 to 699

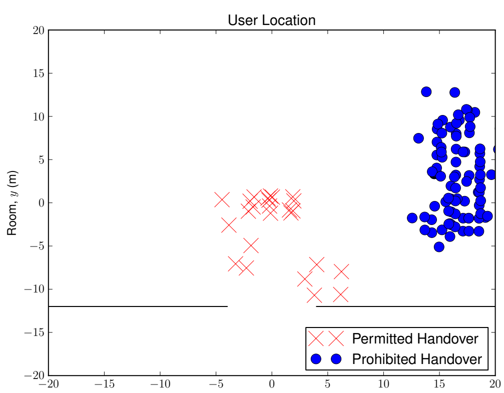

Figure 4.25: Case 1: Handover prohibitions 900 to 999

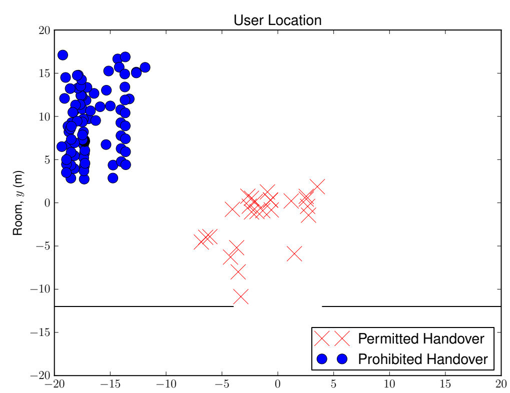

Figure 4.26: Case 2: Handover prohibitions 900 to 999

iterations (these represent a form of time).

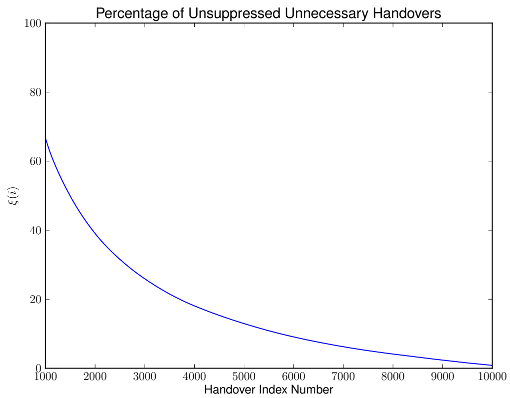

Figure 4.27: Case study 1: Percentage of Unsuppressed Unnecessary Handovers for SOM at the expense of no dropped calls if the handover index is sufficiently large

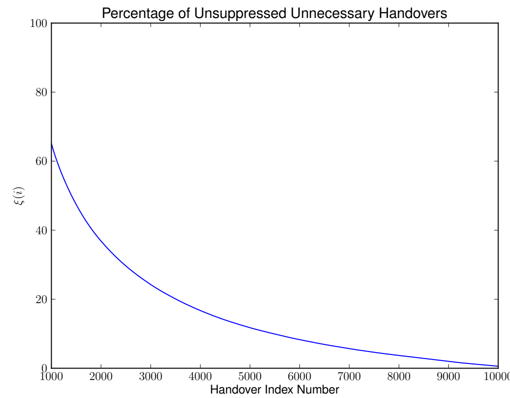

Figure 4.28: Case study 2: Percentage of Unsuppressed Unnecessary Handovers for SOM at the expense of no dropped calls if the handover index is sufficiently large

$\xi(i) = (h)/X$ where h is the number of unneccessary handovers at handover index i during the last X handovers. Within this thesis X is 1000. $\xi(i)$ is achieved at no dropped calls if the handover index is sufficiently large as can be seen from Figures 4.25 and 4.26. Figures 4.27 and 4.28 reveal two aspects of the proposed neural networking algorithm applied to handover management. The transient response of the curve represents the inherent learning curve that occurs. The faster the transient response, the quicker the algorithm learns its environment, which leads to a more efficient level of unnecessary handovers. Note that, even a system with a longer transient response is still more efficient than the simple LTE system because prohibited handovers still occur. As a result, the faster the system learns, the better the performance. The second aspect of Figures 4.27 and 4.28 to observe is the steady state response. This represents the level of unnecessary handovers that not have been inhibited by the algorithm in an LTE system. Figures 4.27 and 4.28 diagrammatically show that implementing SOM into SON for handover management can improve the level of handovers that take place by reducing the number of unnecessary handovers that occur. By comparing the modified LTE graphs in Figures 4.27 and 4.28 it can be seen that the choice of mobility model does not effect the generality of the results because the algorithm works in an event-based manner and not in a temporal manner. These figures use a moving average to remove any influence of the initial performance of the algorithm.

The choice of mobility model has been investigated and proven to not effect the generality of the results because the algorithm works in an event-based manner and not in a temporal manner. The algorithm has also been implemented in an indoor scenario that more closely represents a domestic environment (*i.e.* a living room). In order to simulate a living room environment, the room size has been reduced, as shown in Table 4.2. Other changes involve decreasing the speed of the user to adhere to the different behavior of users in their home and changing the TTT and Hys to test the system under different circumstances. The modified random walk mobility model (Figure 4.15) has been used within this environment because it more closely represents the movement of a user within a home than the modified random direction mobility model (shown in Figure 4.16). Similar to case studies one and two, a propagation model has been incorporated as a single-slope propagation model to alter the RSRP of the femtocell as the mobile terminal navigates through the simulation environment. The RSRP of both a single macrocell and a single femtocell are used to determine the requirement for handover. The algorithm is generic in nature and has the ability to, not only, adapt to different mobility and propagation models but also to varying numbers of prohibition and permission zones.

The algorithms effectiveness in detecting the number of clusters within the femtocell environment as well as which clusters correspond to both permission and prohibition zones will now be investigated. Typical performance will be demonstrated using 2 additional case studies. Case study three incorporates one prohibition zone and one

Table 4.2: Simulation Details

Parameter	Value
Simulation dimensions	7 m × 9 m
Room dimensions	7 m × 7 m
Exit area	2 m × 7 m
No. of mobile terminals	1
Direction change time	1.0 sec
Movement speed	1 - 3 m/sec
Initial position	centre
Mobility model	random walk
Propagation model	single-slope
Hys	5 dB
TTT	320 ms
Error	0 m
Neurons	100

permission zone (similar to the previous scenarios) and case study four incorporates two prohibition zones and one permissive zone. The simulation details summarised in Table 4.2 are common to case studies three and four. These case studies demonstrate how well the algorithm adapts to the number and location of clusters within the propagation region of the femtocell and show that the number of handovers can be dramatically reduced in an autonomous manner.

Just as it was in case studies one and two, the SOM algorithm is the basis of the autonomic element within SON. This allows the femtocell to decide if a handover should be permitted or prohibited based on handover success or failure in that area of the propagation environment. The handover regions and whether they are doors (permission zones) or windows (prohibition zones) are shown in Figures 4.29 to 4.30. The locations that handover has been permitted or prohibited in snapshots of algorithm operation within the simulation environment are shown in Figures 4.31 to 4.38. Each figure depicts the locations of 100 suppressed handovers and the equivalent set of handovers that have been permitted. Note that these are co-incident with the locations of the permission and prohibition zones. Just as with case studies one and two, the algorithm allows handover to be non-restrictive at initialisation and prohibits handovers as it learns the environment. The algorithm successfully detects where to prohibit and permit handover without the requirement of knowing how many permission and prohibition zones there are.

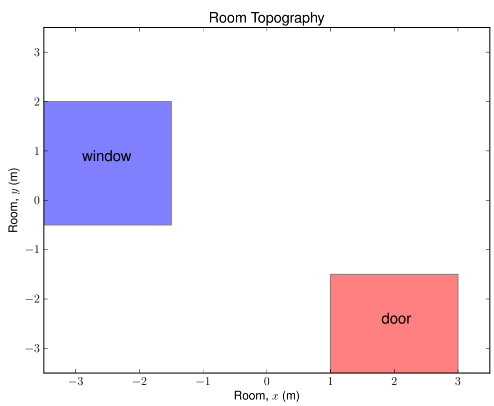

Figure 4.29: Case 3: Room topography

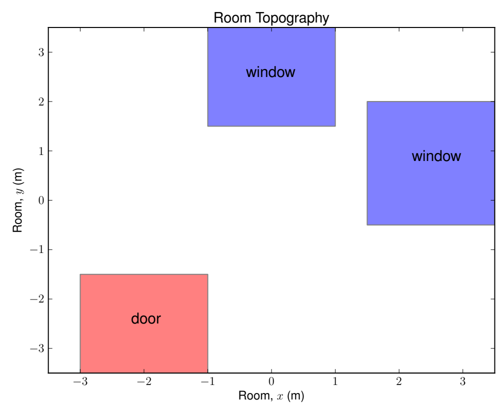

Figure 4.30: Case 4: Room topography

The process of learning the environment has an inherent learning curve associated with it. The algorithm learns by gaining information about its environment. Any degree of learning achieved by the algorithm results in a reduction in the number of unnecessary handovers that occurs. The rate of learning and the percentage of unsuppressed unnecessary handovers is dependent on the environment being simulated and the mobility path generated. However, Figures 4.39 and 4.40 show that the number of zones within the environment does not dramatically alter the performance of the algorithm.

Figures 4.39 and 4.40 were generated using a moving average. They reveal that the transient response and the steady state response of the algorithm in both scenarios is very similar. These results are similar because the algorithm's ability to learn is not affected by the number of prohibition or permission zones within the

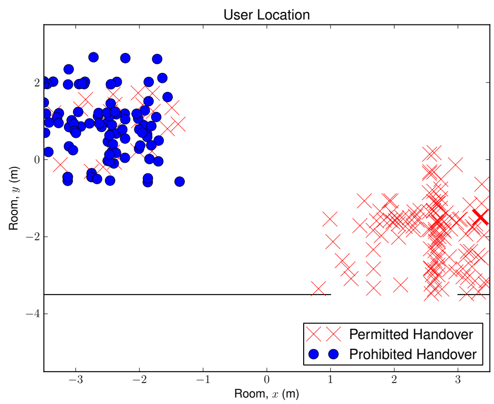

Figure 4.31: Case study 3: Handover prohibitions 0 to 99

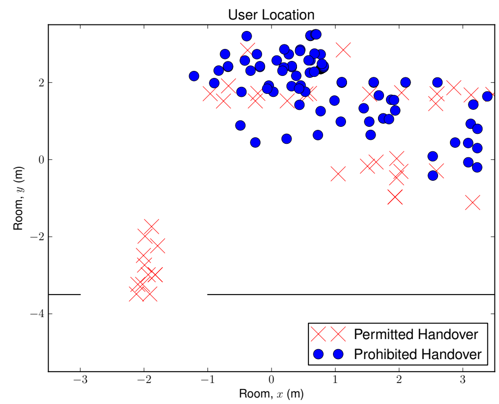

Figure 4.32: Case study 4: Handover prohibitions 0 to 99

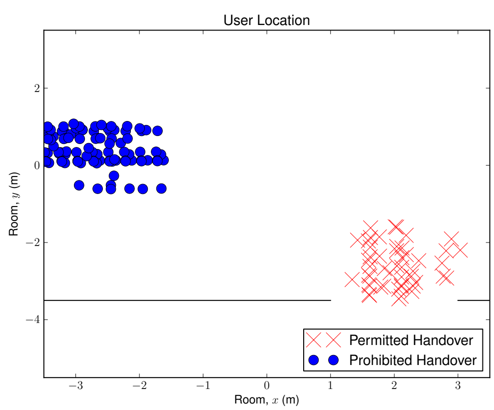

Figure 4.33: Case study 3: Handover prohibitions 300 to 399

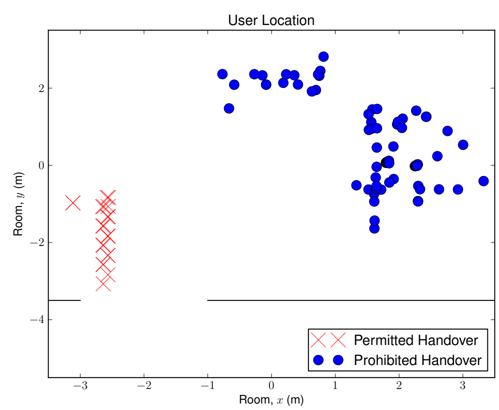

Figure 4.34: Case study 4: Handover prohibitions 300 to 399

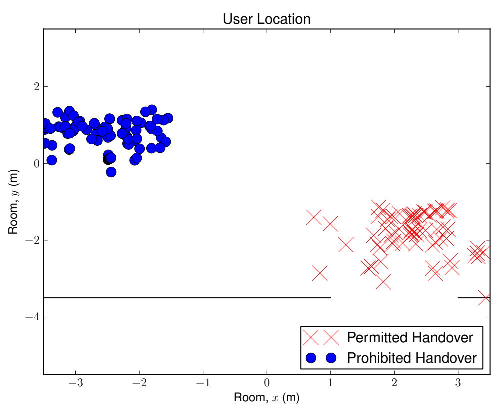

Figure 4.35: Case study 3: Handover prohibitions 600 to 699

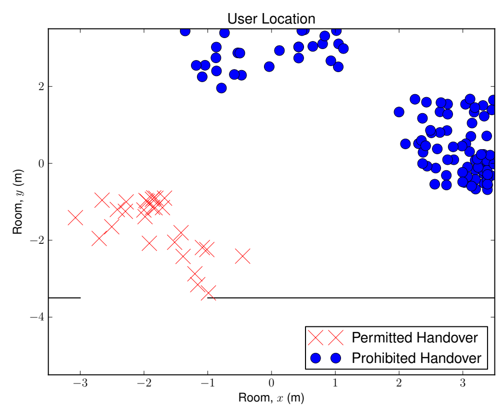

Figure 4.36: Case study 4: Handover prohibitions 600 to 699

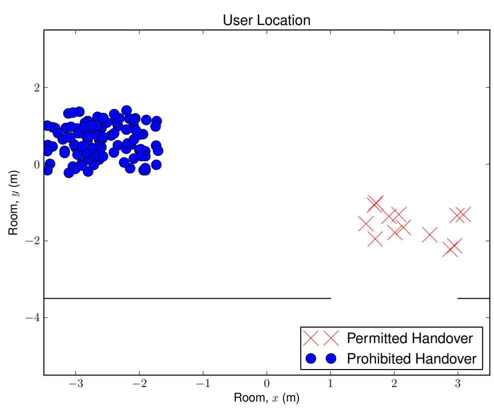

Figure 4.37: Case study 3: Handover prohibitions 900 to 999

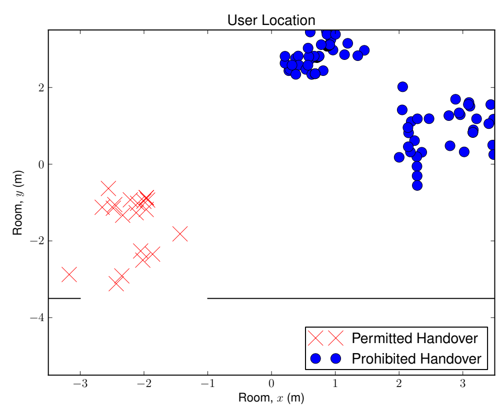

Figure 4.38: Case study 4: Handover prohibitions 900 to 999

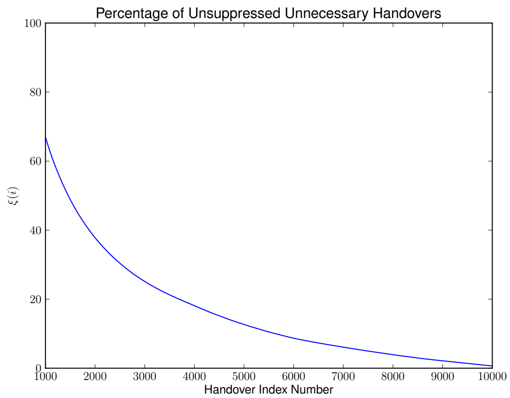

Figure 4.39: Case study 3: Percentage of Unsuppressed Unnecessary Handovers for SOM at the expense of no dropped calls if the handover index is sufficiently large

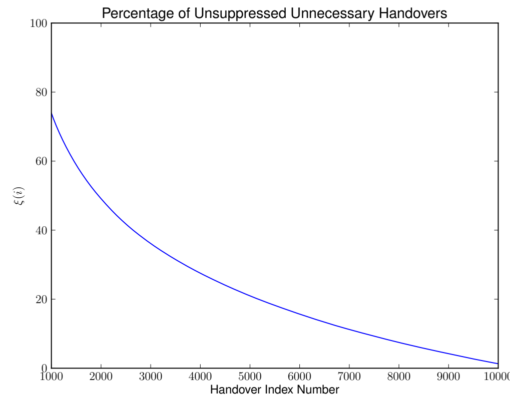

Figure 4.40: Case study 4: Percentage of Unsuppressed Unnecessary Handovers for SOM at the expense of no dropped calls if the handover index is sufficiently large

simulation environment. Figures 4.39 and 4.40 depict the percentage of unsuppressed unnecessary handovers when the algorithm is in operation. Implementing SOM into SON for handover management can improve the level of handovers that take place. This is achieved by prohibiting unnecessary handovers at the expense of no additional dropped calls when the handover index is sufficiently large as shown in Figures 4.37 and 4.38. By comparing Figures 4.39 and 4.40 it can be seen that the number of clusters in the simulation environment has only a minor effect on the results.

Since indoor radio environments are inherently complex due to scatter from clutter, the effect of a position estimation error (of up to 3m) has been investigated for all case studies. This error does not significantly impact the learning rate or the accuracy of the algorithm. Neural networks are generally insensitive to error because the inaccurate movement of the neurons during learning will, in effect, cancel each other out.

4.5 Summary and Conclusions

In this chapter, a self-optimising algorithm has been proposed. The self optimising algorithm utilises a SOM to improve handover efficiency while adhering to the requirements of SON within LTE systems. It has been shown that by using the location of the user as an input to the SOM, the femtocell can optimise the handover scenario within an indoor environment. The algorithm used requires no information about the deployed environment and is able to use handover experience to classify regions within the radio environment as prohibition or permission zones.

The algorithm operates using the autonomous control loop. It monitors the situation by observing the location of the user at the point of handover triggers. This location is then used within the Analyse stage as the input to the SOM which then plans and implements a decision to allow or suppress each handover occurrence. Section 4.4 shows the algorithm at the input and the output of the SOM algorithm. The result is that handovers are inhibited in regions that have a history of unnecessary handovers without being prohibitive to handover occurrences in regions that handover is genuinely required to sustain a connection.

Once the location of the user is detected, the algorithm can reduce the level of handovers that take place by identifying areas that have a history of unnecessary handovers in an autonomous manner. The advantage of using this algorithm within SON is that it becomes more flexible with regards to the femtocell being able to adapt to its environment autonomically and improve handover efficiency in a fast and efficient manner. The simulation results show that the optimisation algorithm improves network performance significantly by reducing the number of unnecessary handovers that take place compared to a standard LTE system. This learning takes place while maintaining the constraint of an acceptable number of dropped calls.

Chapter 5

Improved Handover Prohibition

The work covered in Chapter 4 has investigated the feasibility of using a Kohonen SOM to optimise the occurrence of handovers within an LTE system. The next step in this investigation is to improve the performance by increasing the rate of learning.

5.1 Introduction

As a consequence of the rapid uptake of smart-phones, demand for Internet access on mobile handsets continues to increase towards what has been termed the "data explosion" [63]. Furthermore, studies have shown that 70 % of all voice and data traffic is attributable to users located indoors [1]. However, due to the high penetration loss of exterior walls, they often experience relatively poor service quality, limiting them to low bit-rate connections. Since high-bit rate services are in greater demand by users located indoors, femtocells introduce a convenient means of providing high data-rates to those subscribers by relieving some of the strain on the macrocell layer. However, inefficient handover usage can be expensive to the network operators and should be reduced. The model under investigation here is a method to reduce the number of handovers that occur in a faster manner than the Kohonen SOM algorithm.

By improving the learning rate that occurs, there will be less unnecessary handovers and hence ping-pong handovers which will reduce the strain on the network through consumption of radio channels (RACH) and fixed links; through additional processing load in admission control, bearer setting and path switching [38]. The sooner the algorithm can adapt, the earlier the resource/cost savings can be made. This motivates investigation into improved learning rates.

To determine the level of performance improvement obtained by increasing the learning rate, identical scenarios to those in Chapter 4 were used. The scenarios describe why tuning handover parameters can be ineffective or, indeed, counter productive. In the first scenario, as the mobile terminal approaches and passes through an external door (as shown in Figure 4.1) handover takes place and is required. In the second scenario, an active mobile terminal approaches a large window with low penetration loss, as shown in Figure 4.2. Handover at windows potentially result in handover ping-pong which is costly for the network operator and unnecessary for the user.

Unnecessary handovers may have negative consequences for future handover performance since they will cause an increase in the Hys and TTT parameters (assuming a parameter adaption mechanism has been implemented). By increasing the parameters, future handover has been made more conservative. Modifying the parameters in this fashion may subsequently prove disastrous when the terminal leaves the building at some future time as described in the first scenario: the handover response may become so conservative that the call will be dropped before handover is executed. Note, there are occasions whereby an active mobile terminal approaches and pauses by a large window. Under such a circumstance, handover to the macrocell base station is unlikely to generate an unnecessary handover; nonetheless, it would be preferable to avoid such an eventuality in order to keep closed subscriber group traffic assigned to the femtocell where possible. The aim of the algorithm presented in this chapter is to identify indoor regions where handover to external base stations should be permitted and regions where handover should be suppressed. In order to detect this, three principal regions are defined the same as in Chapter 4 depicted in Figure 4.3.

The problem under investigation in this chapter is how to facilitate handover to the macrocell layer in a timely fashion whilst minimising unnecessary handovers. Reducing the number of unnecessary handovers increases the energy efficiency of the femtocell. It results from lower signalling within the network and more efficient use of the network resources. To facilitate an improved handover algorithm, positional information is incorporated into the algorithm in order to optimise the handover decision locally and minimise any adverse effects of parameter alterations (for an entire cell). For clarity, it should be noted that the positional information used in this algorithm is the location of regions within the radio environment in which handover occurs and not the true physical location of the user. However, there may be a strong correlation between both of these forms of location.

For a building of arbitrary shape and construction, an algorithm is required that can optimise handover performance. To realise such an objective, the direction finding capability of MIMO systems is exploited to provide a profile of locations (or more correctly regions in the radio environment) where handover is genuinely required (permission zones) and those where unnecessary handovers are likely to occur (prohibition zones). The kernel SOM is particularly useful in this context by continually mapping regions where either successful or unnecessary handovers have occurred, and using this information to identify the periphery of the permission and prohibition zones.

5.2 Kernel Self Organising Map using X-means

5.2.1 Theory

The Kernel SOM [59, 64, 65] was proposed by MacDonald *et al.* in 2000 as a modified version of the unsupervised neural network, the Kohonen SOM. Similarly to the Kohonen SOM, the kernel SOM creates a low dimensional, discretised representation of the input space using an unsupervised group learning approach. Unlike the Kohonen SOM, the kernel SOM uses kernel methods to calculate distances within the algorithm.

The set of methods known as kernel methods map data non-linearly to a high dimensional feature space and allow for linear operations to be performed on the data. The use of linear operations in a space gives the computational simplicity of linear methods with the representational advantages of non-linear methods. By applying this theory to a SOM, it is possible to get increased detail at the points of interest. More specifically, the distance between the weights and the inputs can be calculated in a space and the resulting vector quantisation error will be reduced.

Within this chapter, a modified kernel SOM is used. The kernel SOM algorithm has been altered to include the use of X-means. The addition of X-means into the SOM allows the neural networking model to learn faster because of a reduction in the level of false learning and is a novel adaptation of the original SOM and kernel SOM algorithms. False learning occurs when a weight within the network is updated in an incorrect manner. The advanced algorithm is composed of the following stages:

- Initialisation: the weights within the SOM are uniformly distributed within the region of the network. Practically this corresponds to the propagation region of the femtocell. The parameters required for the SOM are initialised here.

- Competition: when an input is received by the algorithm (the location of the user), the weights within the network compete to identify the neuron that is most similar to the input. This results in a form of vector quantisation.

- X-means: this stage is added into the traditional Kohonen SOM algorithm. It allows for a Voronoi cell diagram to be created with each resulting cluster being a different cell in the diagram.

- Cooperation: each weight within the network is updated if it is within the region of the winning node (calculated based on a monotonically decreasing sphere of influence) and in the same cell of the Voronoi diagram (calculated using X-means). This allows for group learning to occur in a more efficient manner than with the traditional algorithm.

- Synaptic Adaptation: each neuron and each of the parameters are updated, tend to a solution and do not learn indefinitely.

The XSOM algorithm is used in this model to determine the areas of the permission and prohibition zones based on an estimate of distance based on the RSRP and AoA of the measurement report using the autonomic control loop that is present in all autonomous systems. The femtocell locating where a handover trigger is transmitted from is the Monitor phase. The XSOM algorithm constitutes the Analyse phase with the input being the location of the user and allows the femtocell to learn the regions of the radio environment that handover is likely to take place in. The Plan phase decides if the current handover falls within a permission or prohibition zone, and the Execute phase prohibits or permits the handover within the LTE network. This learning is completed in a group-based manner to allow faster convergence of the neurons within the network. The convergence of the neurons into accurate locations minimises the error inherent in the vector quantisation based algorithm, SOM. For clarity, the X-means algorithm will be explained in Section 5.2.2 followed by a mathematical explanation of the XSOM algorithm in Section 5.2.3.

5.2.2 Mathematical Approach: X-means

X-means [66] is a data clustering method that autonomously splits all the elements within its network into the correct number of Voronoi cells: it is an advanced form of the k-means algorithm with no user input required. The disadvantage of k-means is that it requires the number of clusters, *i.e.* k, to be known in advance. In situations where k cannot be known *a priori*, k-means is not ideal. As a result, either k can be preset to a default value, or a more advanced algorithm should be used that has the ability to detect the number of clusters autonomously.

Advanced forms of the k-means algorithm exist that attempt to solve its limitations. Min *et al.* [67] presented work that used genetic algorithms to detect the optimal location for the initial cluster centres; this does not solve the problem of an unknown number of clusters. Hamerly *et al.* [68] produced work that tests data for a Gaussian distribution and increases the number of clusters until all clusters demonstrate a Gaussian distribution. The limitation in this work is that to create a cluster the data has to follow a Gaussian distribution. Tseng *et al.* [69] presented work on using a genetic algorithm to detect the number of clusters using the logic that some members of each cluster may not be close to that cluster; this should not be the case with the application in this thesis.

X-means has been chosen as the advanced form of the k-means algorithm that will be implemented within a kernel SOM algorithm. In many applications it is not possible to know the number of clusters that best suits the data set (*i.e.* the number of clusters should ideally equal the number of windows and doors). X-means has the ability to scale well computationally and detect the number of clusters. This makes it an ideal choice for any application that requires the ability to automatically handle an arbitrary number of clusters (*i.e.* arbitrary number of prohibition and permission zones) and is particularly well suited to situations where no prior knowledge of the radio environment and hence the number of clusters is available. This removes the need for human intervention during the initialisation of the femtocell base station and is consistent with the requirement for plug-n-play functionality as discussed in Chapter 3. However, there is a requirement for the range that the number of clusters (k) will fall within. That is,

$$k_{\min} \leq k \leq k_{\max}, k \in \mathbb{N} \setminus \{0\} \tag{5.1}$$

$k_{\min}$ to $k_{\max}$ is the range of k which will be calculated using this algorithm. This range can be a default set of values for every femtocell. The number of clusters (k) directly alters the Voronoi cell diagram that is the result of the X-means algorithm since the number of cells is the number of clusters. Voronoi diagrams are made up of the number of clusters defined for the data provided, Figure 5.1 shows an example.

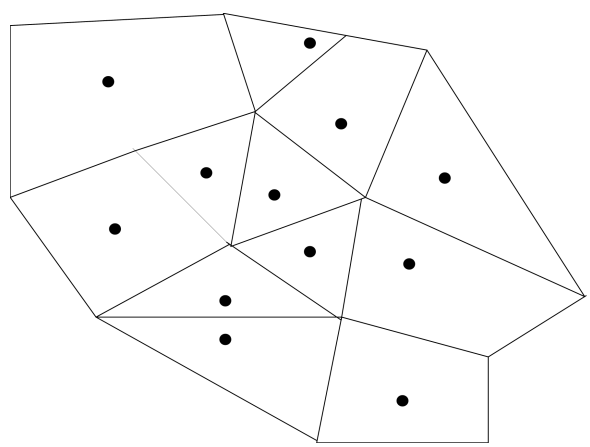

Figure 5.1: A general Voronoi diagram

The inclusion of X-means within the kernel SOM can detect the required number of clusters within the Voronoi diagram and promotes faster convergence times. The improvement in the convergence time is achieved through a reduction in the level of false learning within the system. To achieve this reduction, only the weights in the same Voronoi cell as the input learn from this input. X-means operates after each iteration of the k-means algorithm by making local decisions about whether to split each Voronoi cell in half to better fit the data. This allows the algorithm to start by using $k_{\min}$, increment as required and finish by using any value within the range (shown in Inequality (5.1)), that best fits the data. The algorithm can be split into the following stages:

1. The partitioning is completed by, initially, allocating $k_{\min}$ centroids randomly within the area of the network.

2. Each weight can then be allocated to its nearest centroid using Equation (5.2) where m denotes the centroid, c the index, $q(\mathbf{w})$ the index of the winning centroid and z the index of k-means until convergence.

$$q(\mathbf{w})_z = \arg\max_{c} \|\mathbf{w}_j - \mathbf{m}_{cz}\|,\ j \in \mathcal{L},\ c \in [0, k] \tag{5.2}$$

   This results in the generation of Voronoi cells.

3. Now that each weight has been allocated to its corresponding centroid, the centroid must be updated using Equation (5.3).

$$\mathbf{m}_{cz} = \frac{1}{R_{cz}} \sum_{j=1}^{R_{cz}} \mathbf{w}_j \tag{5.3}$$

   where, R_c is the number of weights allocated to mean c. Each new centroid location ($\mathbf{m}_c$) is the mean value of all the allocated weights.

4. Steps 2 and 3 are repeated until convergence of the centroid and allocated weights has been achieved.

5. Now that the weights within the network have been successfully allocated to their nearest centroids and the centroids have been calculated, the number of

centroids can be updated. The algorithm works by splitting each of the centroids into two centroids. Determining whether this split is valid is facilitated by the Bayesian Information Criterion (BIC). The BIC scoring operates by using posterior probabilities to score the models. To approximate the posteriors, up to normalisation, Equation (5.4) is used.

$$\mathrm{BIC}(M_s) = \hat{l}_s(D) - \frac{p_s}{2} \cdot \log R \tag{5.4}$$

Here, $\hat{l}_s(D)$ is the log-likelihood of the data taken at the maximum likelihood point, p_s is the number of parameters in M_s and R is the number of weights in data set D. The maximum likelihood estimate for the variance is calculated using Equation (5.5).

$$\hat{\sigma}^2 = \frac{1}{R-k} \sum_i (\mathbf{x}_i - \mathbf{m}_{q(\mathbf{w})})^2 \tag{5.5}$$

where k is the current number of centroids being used in the X-means algorithm and i is the input index. The log-likelihood of the data points that belong to centroid m_c ($\hat{l}_s(D_c)$) and including the maximum likelihood estimates, yields Equation (5.6).

$$\begin{aligned} \hat{l}_s(D_c) = & -\frac{R_c}{2}\log(2\pi) - \frac{R_c \cdot M}{2}\log(\hat{\sigma}^2) \\ & -\frac{R_c - k}{2} + R_c \log R_c - R_c \log R \end{aligned} \tag{5.6}$$

Within this equation R_c is the number of weights allocated to m_c. The number of parameters p_s is the sum of $k-1$ class probabilities, $M \cdot k$ centroid coordinates, and one variance estimate, as shown in Equation(5.7).

$$p_s = (k - 1) + (M \cdot k) + k \tag{5.7}$$

The number of clusters, k, is increased based on the resultant BIC score until either the solution has converged or the condition stated in Equation (5.1) is violated. Convergence is validated by comparing the BIC score of the final network to the BIC score of the initial solution.

This algorithm can be used in conjunction with a SOM in order to create a neural networking algorithm that is similar to a SOM but with a faster convergence rate.

5.2.3 Mathematical Approach: XSOM Algorithm

We will now present the XSOM algorithm [6] which is a novel enhanced form of our previous work with the SOM algorithm [4] [5]. It converges faster than the SOM algorithm and has an additional stage of the algorithm as well as a different distance metric. The advanced XSOM algorithm consists of the four phases of a kernel SOM: initialisation, competition, cooperation, and synaptic adaptation. Within this algorithm there is also an additional stage: X-means. Many aspects of the algorithm have been explained within Chapter 4 and will not be repeated in this chapter.

Initialisation

Initialisation of the SOM network presets the individual weight values of each neuron in the lattice to values drawn from a uniform distribution. The initial weight values for this work will be distributed within the propagation region of the femtocell. This operates in the same manner as with the Kohonen SOM algorithm described in Section 4.3.2.

Competition

The next step of the process is for inputs to be applied to the algorithm. Under operational conditions this would occur every time a mobile terminal generates a measurement report. Since each input is connected to each neuron, the input and weight vectors have the same dimensions. The representation for an a-dimensional input is defined in Equation (4.1) and the weight vector associated with each neuron in the lattice is defined in Equation (4.2). The input refers to the users location and happens whenever a handover is triggered in the region of the femtocell.

Within the kernel SOM, Euclidean distance is replaced with the kernel trick. Each coordinate from the input and weight space maps to an element within the feature space; transforming the data into a set of points in a Euclidean space. This conversion takes place using a kernel function that allows more detail at the points of interest which reduces the vector quantisation error. The distance between the input and the weights for any kernel SOM or SOM can be determined by many methods (usually the inner product or Euclidean Distance); in this case, using the kernel trick [70].

The kernel trick allows for the computation of a dot product in a high dimensional feature space using simple functions defined on pairs of input patterns. This allows for a non-linear mapping to the feature space which gives more detail at the points of interest. The mapping of $\mathbf{x}$ to $\phi((x))$ can be implicitly carried out with no knowledge of ϕ. This means that only knowledge of the inputs, the weights and the kernel function ($K(\cdot, \cdot)$) is required. By using the kernel trick rather than Euclidean distance, the resulting reduction in the vector quantisation error increases the convergence rate of the network. The distance in terms of the kernel function is shown in Equation (5.8). The mapping to the feature space is completed using a kernel such that $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$ where $\phi(\mathbf{x})$ is the function that maps the data onto the feature space.

$$\begin{aligned}\|\mathbf{x} - \mathbf{w}_j\|^2 &= \|\phi(\mathbf{x}) - \phi(\mathbf{w}_j)\|^2 \\ &= K(\mathbf{x}, \mathbf{x}) + K(\mathbf{w}_j, \mathbf{w}_j) - 2K(\mathbf{x}, \mathbf{w}_j)\end{aligned} \tag{5.8}$$

A Gaussian Kernel function is used as shown in Equation (5.9).

$$K(\mathbf{x}, \mathbf{w}_j) = \exp\left(\frac{-\|\mathbf{x} - \mathbf{w}_j\|}{2\sigma^2}\right) \tag{5.9}$$

Once the winner, the closest match to the input, has been selected as it can be utilised by the cooperation stage of the algorithm.

X-means

X-means partitions the area of the kernel SOM algorithm into a defined number of clusters within the range of the allowed number of clusters, shown in Equation (5.1) (explained in Section 5.2.2). The algorithm estimates the correct number of clusters based on the dataset it is given by splitting each of the clusters individually when required. The number of the clusters used and the method for splitting clusters when required is explained in Section 5.2.2. The number of clusters is directly related to the number of cells in the Voronoi diagram which should be the same as the number of windows and doors in the implemented environment, as shown in Figure 5.2. This is completed autonomically.

The input to the kernel SOM algorithm has been assigned to the closest neuron within the Competition stage of the algorithm. This neuron will be assigned to a cluster along with the other neurons within the lattice. The assigned cluster number will be utilised within the cooperation stage of the algorithm.

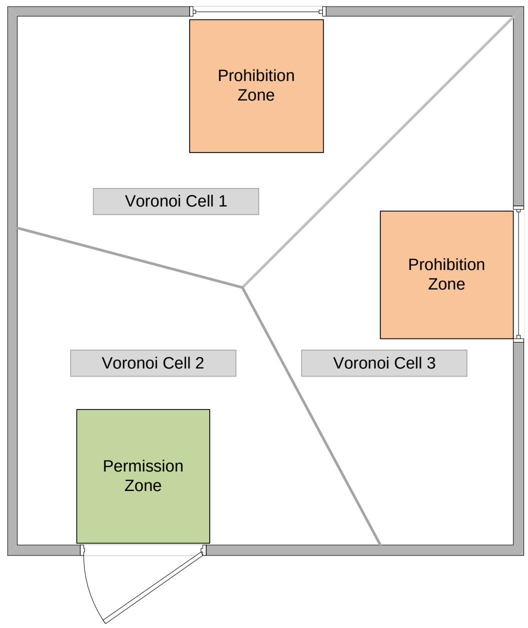

Figure 5.2: Voronoi cells within a room

Cooperation

Once the winner for a given input vector has been selected and the weight has been assigned to its closest centroid, the weights of the neurons within the winner's sphere of influence are updated if they are linked to the same centroid as the winner. The degree to which each neuron learns depends on the distance from the winning neuron. The distance is calculated using kernel methods, in a similar manner to Equations (5.8) and (5.9).

This constitutes a cooperative learning process since, unlike other competitive learning strategies, it is not just the winning neuron that has its weight values modified. This group learning strategy permits the network to converge more rapidly and accurately compared to the case where only the winner would modify its weights. Adding X-means into this algorithm allows only the weights that are in the same cluster as the winner to be updated which improves the accuracy of the weight locations. As a result of the learning process used in all SOMs, the weights become the product of the system learning its environment and will converge to the areas that the inputs occur, *i.e.* the locations of the handover triggers in the femtocell environment. The cooperation stage of the algorithm operates principally in the same way as in the Kohonen SOM with the additional requirement of the cluster centre and winning node being within the same region of the environment and the distance being calculated using kernel methods.

Synaptic Adaptation

The adaptation process is concerned with the execution of the weight update procedure for all neurons within the sphere of influence of the winner. This involves utilising not only the sphere of influence but a learning rate too. When the neurons have been continuously updated over a period of time the locations of the neurons will converge to optimal location as a result of the learning rate becoming very low and the neighbourhood no longer updating any nodes other than the winner. Once this happens, the locations of both permission and prohibition zones have been identified. The process performed here is the same as with the Kohonen SOM, described in Section 4.3.2.

5.3 Simulation Modelling and Results

To evaluate the effectiveness of the novel neural networking algorithm, a simulation model was created within NS3. This allowed both the evaluation of the advanced kernel SOM algorithm (described in Section 5.2) and comparison to the Kohonen SOM algorithm described in Chapter 4. To evaluate the proposed algorithm, the simulation model was used to apply it to handover suppression within SON in LTE in a similar manner as was explained in Section 4. Specifically, the algorithm is used to detect the regions within the environment that correspond to both Permission and Prohibition zones and allow or suppress handover occurrences accordingly. To evaluate the algorithm, scenarios have been modelled that incorporate multiple numbers of permission and prohibition zones in a small room that would represent usage in the living room of domestic environment. To show the performance of the applied algorithm, the level of HPIs were evaluated along with the percentage of unsuppressed unnecesary handovers. The simulation details that were common to all case studies are shown in Table 5.1.

Table 5.1: Simulation Details

Parameter	Value
Simulation dimensions	7 m × 9 m
Room dimensions	7 m × 7 m
Exit area	2 m × 7 m
No. of mobile terminals	1
Direction change time	1.0 sec
Movement speed	1 - 3 m/sec
Initial position	centre
Mobility model	random walk
Propagation model	single-slope
Hys	5 dB
TTT	320 ms
Error	0 m
Neurons	100

Within an LTE system, when the RSRP of a base station other than the serving base station (detected by the mobile terminal) is higher than the serving base station by a Hys value for the TTT period a measurement report is generated. This measurement report then initiates the handover process and can be considered as a handover trigger. In an LTE system that utilises the proposed algorithm, handover is then permitted or prohibited based on previous experience of handovers in the radio region that the handover trigger was generated. To facilitate such decisions, the regions of the propagation environment that relate to prohibition and permission zones must first be determined from the regions of the radio environment that handover triggers occur. The initial setup of the femtocell allows convergence of the neurons within the SOM to the locations of the radio environment where handover may take place by using the location of the handover triggers as the input to the neural network. Each weight within the neural network has the ability to retain knowledge of previous handovers within that area. Specifically, the regions within the radio environment where unnecessary handovers can occur will be detected and reduced over time. By reducing the occurrences of unnecessary handovers within the femtocell environment, the number of handovers that occur have been optimised. The performance of the system and the number of handovers prohibited is linked to X-means and its ability to accurately detect the number of clusters in the environment. Four case studies that demonstrate the ability of the novel XSOM algorithm followed by X-means and its ability to detect the correct number of clusters then the resilience of XSOM to location error will now be discussed

5.3.1 Case Studies

Four case studies now will be presented to fully explain and discuss the merits of the XSOM algorithm in a femtocell environment. Case studies three and four from Chapter 4 have been repeated to illustrate the typical performance of the algorithm and are the basis of case studies one and two. Case studies one and two demonstrate the algorithm's effectiveness in detecting clusters and learning the specific environment. The first case study incorporates one prohibition zone and one permission zone. The second case study incorporates two prohibition zones and a single permission zone. Case studies three and four will then illustrate the performance of the algorithm when the number of clusters is not correctly detected. The simulation details summarised in Table 5.1 are common to all case studies. Each of the learning curves have been generated using 30 parallel simulation runs to provide an ensemble average.

The algorithm deployment is part of an autonomic control loop. Firstly, the Monitor phase takes place that requires the detection of handover triggers and the location in which they take place. This information is then the input to the XSOM algorithm that constitutes the Analyse phase. Once the XSOM has taken place, the Plan phase uses the vector quantised output and the previous experience of handover in that area to decide whether to permit or prohibit the handover. The Implementation phase of the autonomous control loop then translates this into technology specific commands and practically speaking would control the output of the handover request.

The algorithm's ability to detect the location of the user is an important element that affects the performance of the algorithm. The regions that handovers might take place and whether they correspond to the region of windows (prohibition zones) or doors (permission zones) are shown in Figures 5.3 to 5.4. The algorithm must be able to effectively identify and distinguish between each type of zone. Once initialised the femtocell is conservative so all handovers will occur until there is enough experience to begin prohibiting handovers in the prohibition zones and permitting handover solely in the permission zones. Figures 5.5 and 5.6 diagrammatically represent the first one hundred prohibited handovers and the permitted handovers that occur during this time.

Figures 5.7 to 5.8 diagrammatically represent later snapshots of one hundred prohibited handovers and the permitted handovers that occur during this period of time. The latter snapshots differ from the first one hundred handovers since the femtocell has already learnt the environment. Within the first one hundred handover prohibitions there is a period that handover is allowed everywhere and is then prohibited in the region of prohibition zones alone. The later snapshots do not suffer from the same system phenomenon. These figures show that the femtocell is learning. It

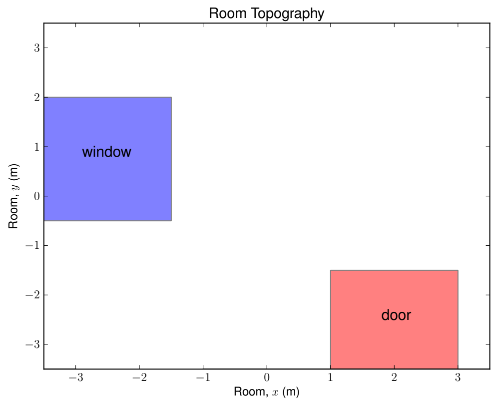

Figure 5.3: Case 1: Room topography

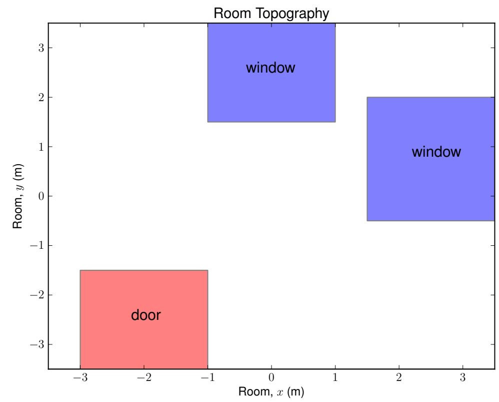

Figure 5.4: Case 2: Room topography

should be noted that the algorithm is not given any prior information regarding the location of the prohibition or permission zones. This knowledge is gained through an unsupervised learning approach.

The inherent learning curve required for the algorithm to operate is shown in Figures 5.9 and 5.10. As already mentioned, the femtocell starts non-restrictive and allows handovers to occur everywhere as normal. An initial value for unnecessary handovers is used for each neuron in the network and scalar reinforcement is carried out that eventually indicates whether the terminal state is a state that should be avoided (prohibition zone). As the femtocell learns its environment, handovers are prohibited with increasing regularity and the system eventually converges by prohibiting all unnecessary handovers. Comparing the trend from Figures 5.5 to 5.8, to Figures 5.9 and 5.10 it can be seen that after a period of time the femtocell will learn the environment in which it is deployed in an unsupervised nature with no external interaction. However, there is a trade-off between fast learning of the environment (leading to rapid performance improvement) and adaptability. Once the algorithm has fully adapted to a radio environment, changes in that radio environment (e.g.

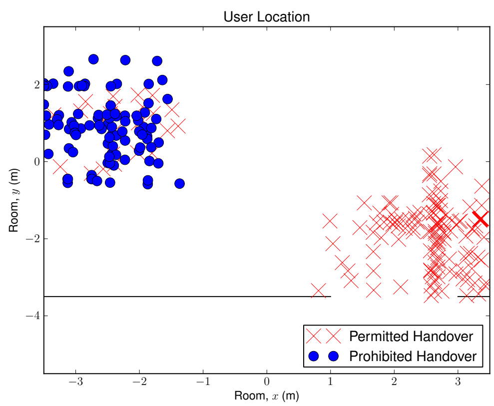

Figure 5.5: Case study 1: Suppressions 1 to 100

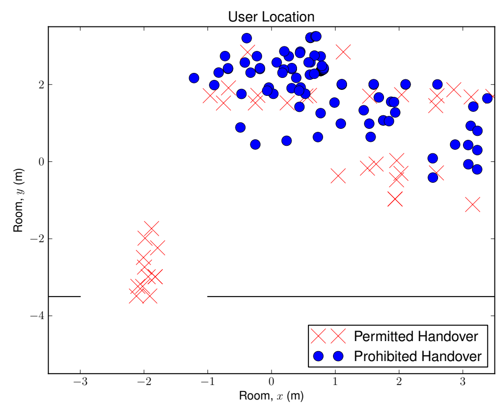

Figure 5.6: Case study 2: Suppressions 1 to 100

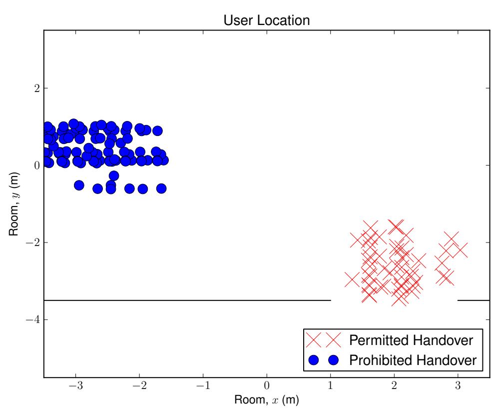

Figure 5.7: Case study 1: Suppressions 301 to 400

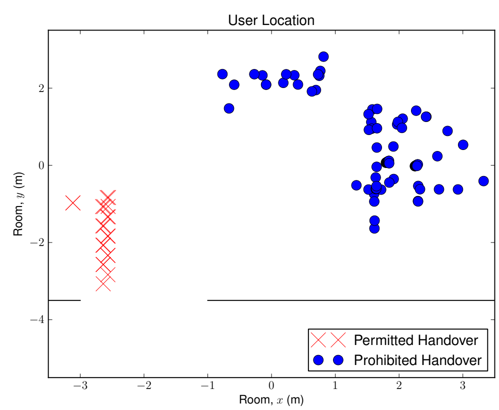

Figure 5.8: Case study 2: Suppressions 301 to 400

changes in the location of furniture and other reflectors) will lead to suboptimal performance. This eventuality can be overcome by a simple manual reset.

In Chapter 4 it was shown that the SOM algorithm can be used to prohibit and permit handover requests using the SOM algorithm. Since the aim of this chapter is to create a novel technique that results in a faster learning curve the results of the XSOM algorithm will be compared to the Kohonen SOM algorithm. Figures 5.9 and 5.10 compare the learning algorithms of both algorithms; using this, a comparison on learning rate can be made.

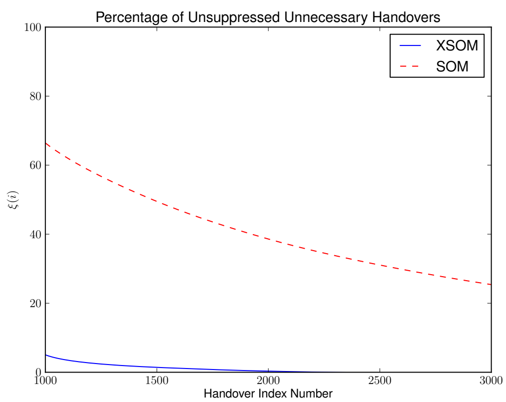

Figure 5.9: Case study 1: Percentage of Unsuppressed Unnecessary Handovers for SOM and XSOM at the expense of no dropped calls if the handover index is sufficiently large

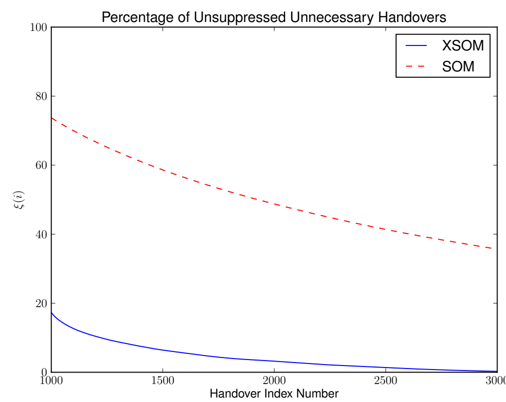

Figure 5.10: Case study 2: Percentage of Unsuppressed Unnecessary Handovers for SOM and XSOM at the expense of no dropped calls if the handover index is sufficiently large

As can be seen by comparing the learning curves of both the Kohonen SOM algorithm and the improved XSOM algorithm, shown in Figures 5.9 and 5.10, the kernel SOM provides an improved performance. The curves in in Figures 5.9 and 5.10 are governed by $\xi(i) = (h)/X$ where h is the number of unneccessary handovers at handover index i during the last X handovers. $\xi(i)$ is achieved at no dropped calls if the handover index is sufficiently large as can be seen from Figures 5.7 and 5.8. By comparing the transient response of Figures 5.9 and 5.10 it can be seen that the convergence time is better with the XSOM algorithm compared to the Kohonen SOM algorithm. A specific example of the difference in performance is that the percentage of unsuppressed unnecessary handovers after 2000 handovers is 1% for XSOM in comparison to 40% for the SOM algorithm in case 1 and 3% for XSOM in comparison to 51% for the SOM algorithm in case 2. This is a considerable improvement. The change in distance metric and the addition of X-means into the weight updating process have led to improved performance over the Kohonen SOM algorithm. The addition of X-means leads to more improvement than the use of the kernel trick due to the reduction in false learning from other zones in the environment. The graphs were generated using 30 parallel simulation runs to provide an ensemble average.

The optimised algorithm can also be compared to the standard LTE system based on HPIs. The handover ping-pong ratio for these specific scenarios are shown in Figures 5.11 and 5.12. The handover dropped call ratio for these specific scenarios are shown in Figures 5.13 and 5.14. The HPI figures include the performance of the network with and without the algorithm being proposed.

As can be seen from Figure 5.11 and 5.12, the situation includes a high level of handover ping-pong occurrences. The standard LTE system is shown to rapidly converge at a high level of handover ping-pongs. However, when the proposed algorithm is included, the number of handover ping-pongs is significantly reduced and would constantly reduce as the system continually learns the details about the environment.

The level of dropped calls within the region of the femtocell for both the modified and unmodified LTE systems within both case studies are shown in Figures 5.13 and 5.14. In the modified system, the number of dropped calls is lower than with the standard LTE system.

Case studies one and two have effectively shown that the XSOM algorithm provides an improvement to the standard Kohonen SOM algorithm. Case studies three

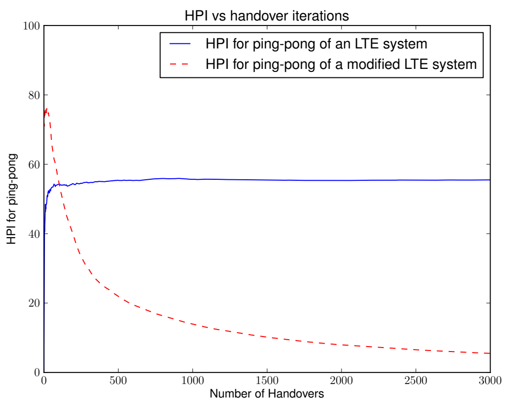

Figure 5.11: Case study 1: HPI for ping-pong handover

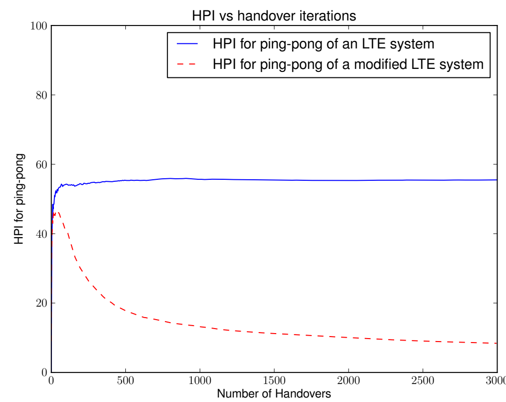

Figure 5.12: Case study 2: HPI for ping-pong handover

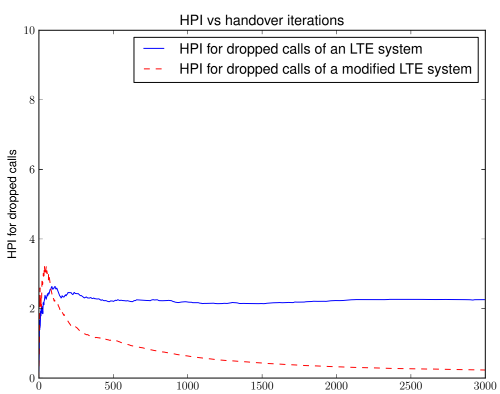

Figure 5.13: Case study 1: HPI for dropped calls

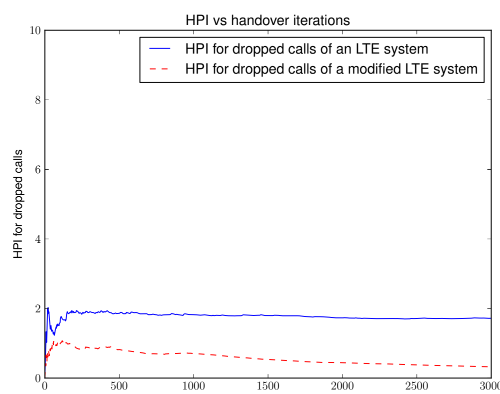

Figure 5.14: Case study 2: HPI for dropped calls

and four will demonstrate the performance of the algorithm when the number of clusters are not correctly detected. Case studies three and four follow the same structure as case study one (one prohibition zone and one permission zone). The number of clusters detected is set to 3 for case study three and set to 10 for case study four to demonstrate the performance when the number of clusters detected is slightly and very wrong.

Just as with case studies one and two, the algorithm deployment is part of an autonomous control loop that allows the location to be detected and used to decide whether to allow the handover or not. The algorithm's ability to detect the location of the user is an important element that affects the performance of the algorithm in severe cases. The algorithm must be able to effectively identify and distinguish between each type of zone, shown in Figures 5.15 and 5.16. At initialisation, the number and types of zones is unknown and all handovers are allowed. The algorithm then seeks to identify regions within the propagation environment and then to identify them as to whether handover should be permitted or prohibited. Figures 5.17 and 5.18 diagrammatically represent the first one hundred prohibited handovers and the related permitted handovers.

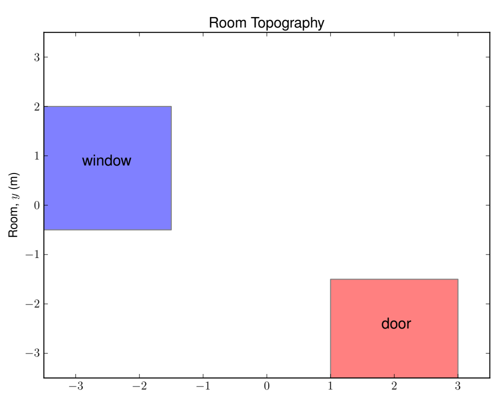

Figure 5.15: Case 3: Room topography

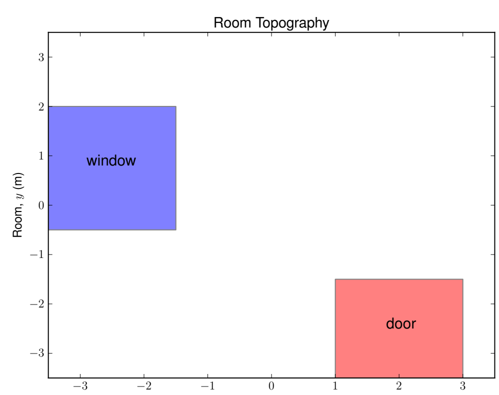

Figure 5.16: Case 4: Room topography

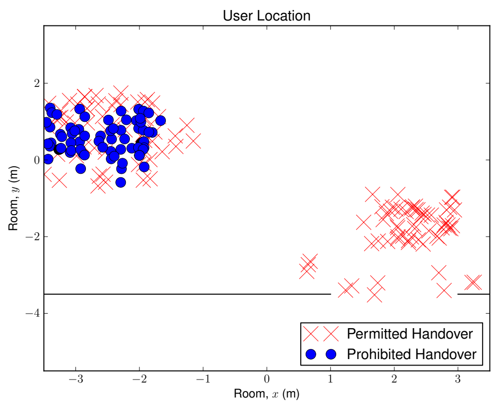

Figure 5.17: Case study 3: Suppressions 1 to 100

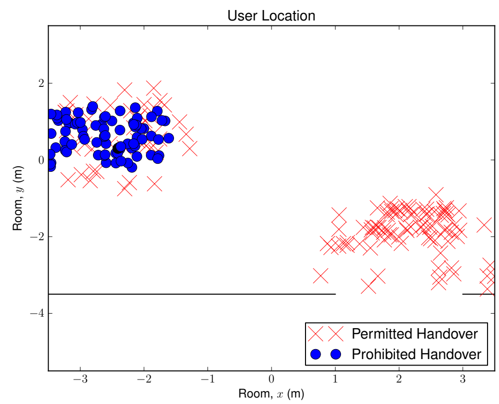

Figure 5.18: Case study 4: Suppressions 1 to 100

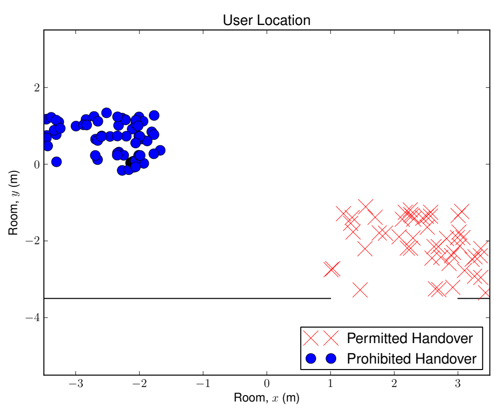

Figure 5.19: Case study 3: Suppressions 301 to 400

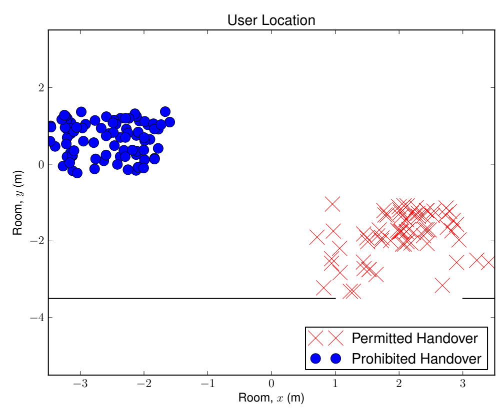

Figure 5.20: Case study 4: Suppressions 301 to 400

Figures 5.17 and 5.18 show that initially many handovers are allowed to take place in a handover regions within the environment. Figures 5.7 and 5.8 show that in later snapshots of the simulation process the handovers only occur within the permitted zones and are inhibited in the prohibited zones. Therefore, there is a learning curve inherent to the algorithm and that once the algorithm has adapted to its deployed environment, handovers can be optimised. Figures 5.17 to 5.20 are similar to that of case study one because the handover zones are the same in case studies one, three and four.

As already stated, the algorithm starts non restrictive at initialisation and then restricts the handovers that take place based on experience and location. This method has an inherent learning curve associated with it, shown in Figures 5.21 and 5.22. As the system adapts to the environment, the number of handovers prohibited increase until the steady state has been reached when the system prohibits all unnecessary handovers at the expense of no dropped calls when the handover index is high enough. Case studies one and two clearly show that the algorithm effectively optimises the number of handovers that occur in a more efficient manner than with the standard Kohonen SOM. The Kohonen SOM will be used for comparison in case studies three and four to show the change in performance with incorrect clusters.

When analysing the performance of case studies one and two in Figures 5.9 and 5.10, respectively, it was stated that there are performance of the XSOM algorithm was an improvement over the SOM algorithm. The XSOM resulted in only 1% of unsuppressed unnecessary handovers in case 1 and 3% for case 2 after 2000 handovers. Cases 3 and 4 show a slightly different result because of the inaccurate detection of clusters. By comparing the learning curve of the Kohonen SOM and the XSOM in case 3, Figure 5.21, it can be seen that there is an improved performance in comparison to the Kohonen SOM algorithm. However, the improvement in performance is not as good as with the previous case studies due to the incorrect number of clusters detected. After 2000 handovers there is 4% of unsuppressed unnecessary handovers in comparison to the 1% obtained by the utilising the correct number of clusters in case 1. This performance degradation is not very high. Case 4 shows different results than the other case studies since it has detected a very incorrect number of clusters, as shown in Figure 5.22. The learning rate has been heavily affected and so has the steady state. After 2000 handovers there are 23% of unsuppressed unnecessary handovers which is very high in comparison to the 1% obtined by using the correct number of clusters but is still better than the 40% obtained by the Kohonen SOM algorithm. Cases 3 and 4 show that even when the number of clusters has been incorrectly detected the algorithm is still able to reduce the number of handovers that occur which improves the efficiency of the system. The change in distance metric and the addition of X-means into the weight updating process have led to the change in performance over the Kohonen SOM algorithm.

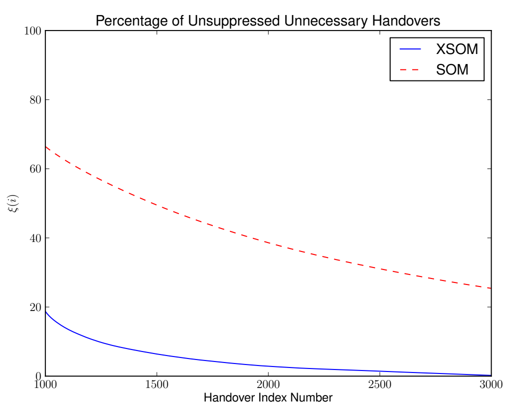

Figure 5.21: Case study 3: Percentage of Unsuppressed Unnecessary Handovers for SOM and XSOM at the expense of no dropped calls if the handover index is sufficiently large

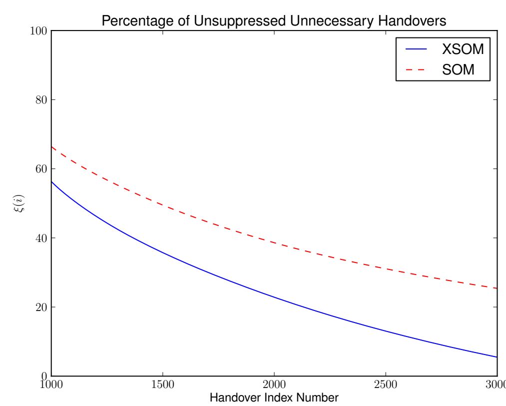

Figure 5.22: Case study 4: Percentage of Unsuppressed Unnecessary Handovers for SOM and XSOM at the expense of no dropped calls if the handover index is sufficiently large

To further evaluate the performance of the algorithm, HPIs were used. The relevant HPIs were for handover ping-pong and dropped calls. The handover ping-pong

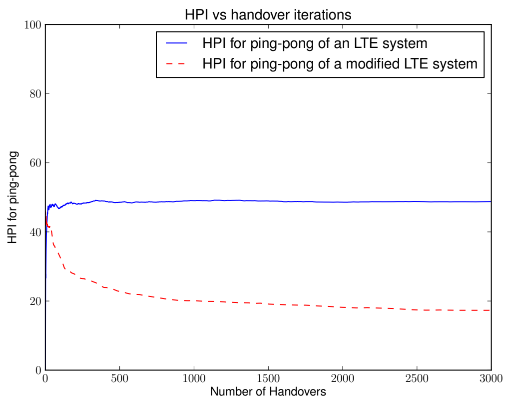

Figure 5.23: Case study 3: HPI for ping-pong handover

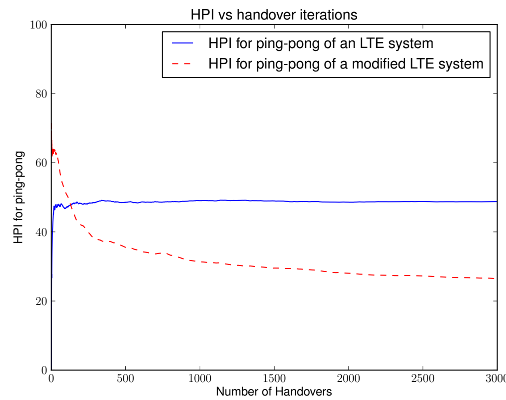

Figure 5.24: Case study 4: HPI for ping-pong handover

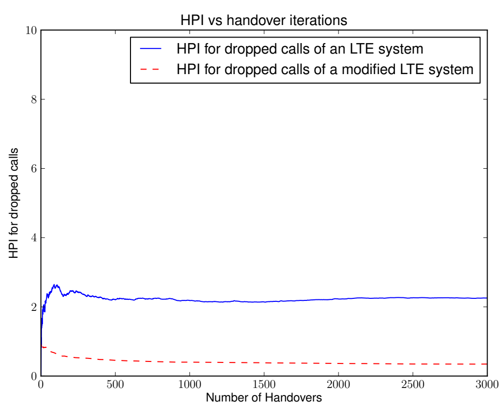

Figure 5.25: Case study 3: HPI for dropped calls

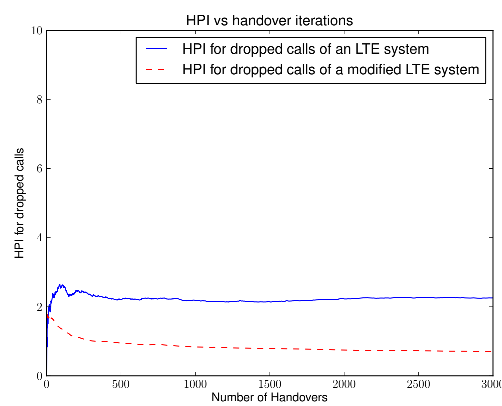

Figure 5.26: Case study 4: HPI for dropped calls

ratio for case studies three and four are shown in Figures 5.23 to 5.24. The HPI figures include the performance of the network with and without the algorithm being proposed.

Just as with case studies one and two, the scenario includes a high level of handover ping-pong occurrences to best demonstrate the effectiveness of the algorithm. The performance of the algorithm in case studies one (Figure 5.11) and two (Figure 5.12) is superior to that of case studies three (Figure 5.23) and four (Figure 5.24) because the algorithm has detected the correct number of clusters.

The level of dropped calls in case studies three and four is shown in Figures 5.25 and 5.26. The figures demonstrate the performance of the Kohonen SOM and the XSOM (with a fixed and incorrect number of clusters). The XSOM algorithm is shown to perform better than the Kohonen SOM algorithm with a considerably lower level of dropped calls occurring once the algorithm has learned the environment. The performance improvement is decreased in case studies three and four in comparison to case studies one and two but still show a considerable improvement over the Kohonen SOM algorithm. The graphs were generated using 30 parallel simulation runs to provide an ensemble average.

The case studies show that a considerable number of handover ping-pongs occur within the network. The locations of these ping-pongs have been detected and the handovers within this region have been increasingly prohibited. Once a handover trigger has been paired to its closest neuron, learning can occur to optimise handover performance. It has been shown based on learning rate and HPIs that the novel XSOM algorithm is an improvement over the Kohonen SOM algorithm. The level of prohibited handovers and the resulting convergence rate is linked to the accuracy of the X-means algorithm and its ability to effectively detect the correct number of clusters (as discussed in Section 5.3.2). Case studies three and four demonstrate that when a sub-optimal number of clusters is detected, the algorithm operates ineffectively. It should be noted that the number of handovers has decreased and any reduction in the number of handovers represents an improvement in network performance.

5.3.2 XSOM Cluster Detection Accuracy

The optimum performance of the proposed XSOM algorithm is based on the X-means algorithm and its ability to correctly estimate the required number of clusters, in this case three. When the number of clusters (permission and prohibition zones) is estimated correctly, there is a minimal level of false learning and the algorithm performs optimally.

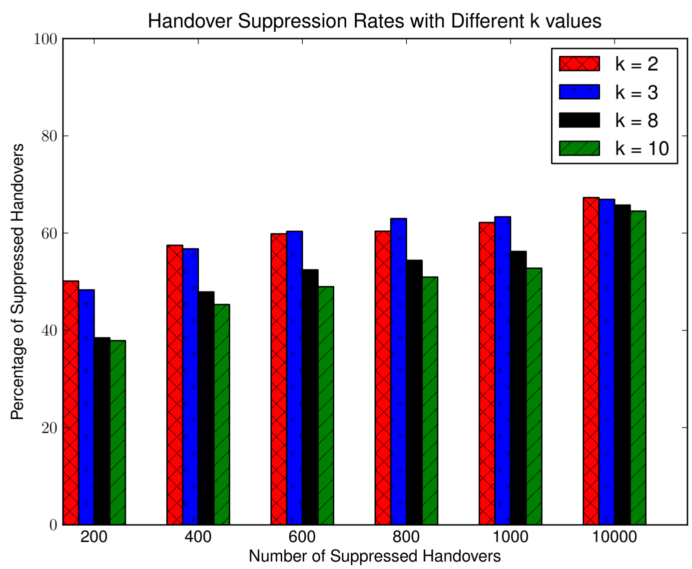

Figure 5.27: Handover suppression rate with different k values

Figure 5.27 compares different values of k for different numbers of handovers suppressed. As k approaches the correct number of clusters, the overall performance is significantly improved. However, when a non-ideal value for k is used (e.g. when k is 8 or 10), the performance of the algorithm is far from optimal. A non-optimal value for k still yields an improvement over the unmodified LTE system by reducing the overall number of handovers that occur. Thus, the use of X-means is valid within plug-n-play functionality of SON within the indoor environment, due to it being able to autonomously estimate how many handover areas there are within the region of the femtocell.

The presented case studies demonstrate the same trend as is shown in Figure 5.27. Case studies one and two demonstrate that the mechanism can effectively adapt to the number of permission and prohibition zones that occur in an autonomic fashion. Case studies three and four demonstrate what happens when the wrong number of clusters is detected.

5.3.3 Position Estimation Error

The inherent learning curve when the algorithm is operating optimally is shown in Figures 5.9 to 5.10. These figures show that error in the position of the user does greatly affect the performance of the algorithm. As already mentioned, the femtocell starts non-restrictive and allows handovers to occur everywhere as normal. As the femtocell learns its environment, handovers are prohibited with increasing regularity and the system eventually prohibits all unnecessary handovers. Ideally, the number of handovers being prohibited would be 66% and as such, the percentage of suppressed handovers should converge to this value. The limit being converged to would potentially change based on the scenario being deployed within. This convergence would be achieved when handovers are being prohibited solely within prohibition zones and solely permitted within permission zones. However, when deployed within a practical LTE environment it is not always possible to get an accurate estimation of the location within the radio environment which intuitively would have an effect on the results of the algorithm.

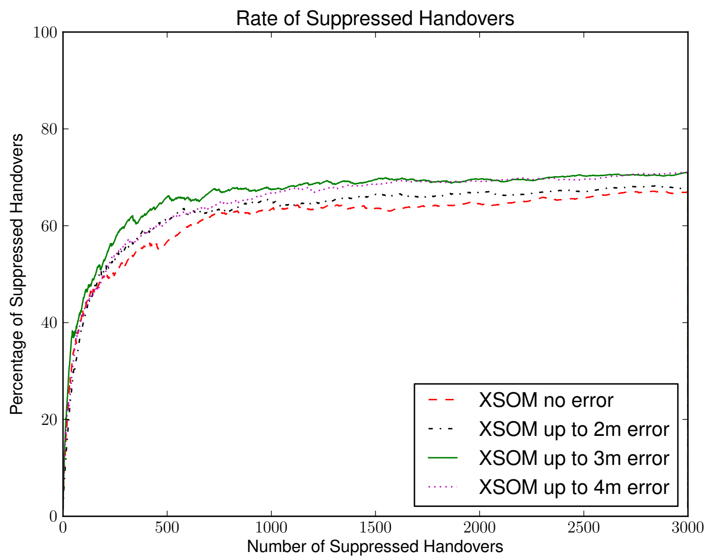

Figure 5.28: Case study 1: Handover suppression rate with and without error

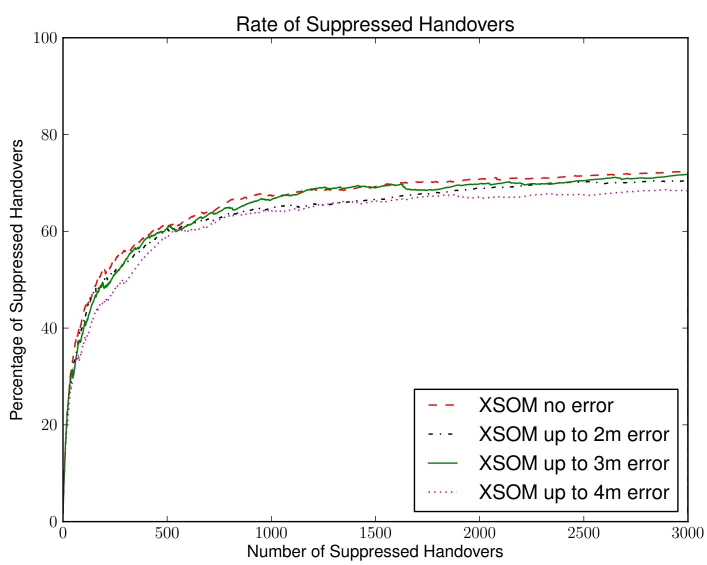

Figure 5.29: Case study 2: Handover suppression rate with and without error

In a practical environment, the position of the user would incorporate a positional error because indoor environments are inherently complex radio environments. The location error within the indoor environment is caused by the effects of clutter and other obstacles. The well-known robustness of neural networks to noise is demonstrated by the algorithm being insensitive to positional accuracy. The effect of different ranges in errors with uniform distribution of the cartesian coordinates of the user was tested. The effect of different ranges in error is depicted in Figures 5.28 and 5.29. The inaccurate movement of the neurons due to error, during learning will, in effect, cancel each other out.

5.4 Summary and Conclusions

In this chapter, an efficient algorithm to reduce unnecessary handovers in an indoor-outdoor scenario has been proposed. The self-optimising algorithm uses a novel SOM that incorporates kernel methods and X-means to improve the Kohonen SOM algorithm. It has been shown that the XSOM algorithm can take the location of the user as an input and optimise handover performance in an indoor scenario. The resultant algorithm increases the speed of learning and reduces the quantisation error that occurs in relation to the SOM algorithm (Chapter 4). The plug-n-play functionality is retained along with being within the SON paradigm. The algorithm requires no information about the deployed environment and is able to use handover experience to classify regions within the radio environment as prohibition or permission zones.

The algorithm operates by monitoring the environment and waiting for a handover trigger to serve as an input to the algorithm. Once an input has been detected, it is analysed and the algorithm plans and implements whether to prohibit or permit the handover request. The XSOM algorithm estimates the number of zones within the environment and classifies each zone as a prohibition or permission zone. In a situation where the femtocell has incorrect knowledge about the number of permission and prohibition zones, the algorithm is still an improvement over a typical LTE system. The results in Section 5.3 show that handover is prohibited in regions where handover ping-pong is likely and does not affect regions that handover is genuinely required.

When using the algorithm within the SON paradigm, it allows the femtocell to be more efficient with handover occurrences and less wasteful of network resources. The algorithm quickly adapts to its environment and allows the femtocell to be more flexible to its environment by significantly reducing the handovers that occur by removing all unnecessary handovers. In scenarios where the learning process is not optimal, an improvement over standard LTE performance is still being achieved.

Chapter 6

Handover Parameter Optimisation

The work covered in Chapter 5 has shown that the novel XSOM algorithm is an improvement over the Kohonen SOM algorithm in lowering the total level of handovers that occur. It also demonstrated the algorithm's use in prohibiting handover occurrences in areas of an indoor environment utilising an LTE femtocell that are unnecessary. The next step in the investigation is to utilise handover parameters to modify when, and by extension where, handover takes place while still using the XSOM algorithm.

6.1 Introduction

With femtocells being deployed indoors, handover between the indoor (femtocell) and outdoor (macrocell) environments becomes a very pertinent issue due to the increase in frequency of handover because of the short transmission range of femtocells. The system must be able to reliably and seamlessly handover as the mobile user leaves their home or office. In order to achieve this, handover optimization must be used to balance two key conflicting demands; minimizing the likelihood of dropped calls, and minimizing unnecessary handovers.

The LTE specifications capture the possibility of incorrect handover timing (due to poorly configured parameters) with the definition of the handover too early and handover too late metrics, as explained in Chapter 3. Handover parameters (explained in Section 2.3.1) can be used to mitigate the level of handover too early and handover too late occurrences. When a handover too early is triggered, the Hys and TTT can be increased to reduce the likelihood of unnecessary handover. However, requiring neighbouring base stations that are candidates for handover to provide a significantly superior signal strength for a longer time period will have the undesirable effect of increasing the rate at which calls will be dropped. The signal strength from a serving base station may drop below the absolute minimum required to sustain a connection before handover is completed. This possibility is captured by the LTE specifications with the definition of the handover too late metric. Thus, any algorithm that tunes the Hys and TTT must strike a delicate balance between unnecessary handover and dropped call rates.

The addition of SON in LTE has resulted in a plethora of work on handover parameter optimisation. Each work completed in parameter optimisation alters TTT and Hys using different approaches but have similar aims. Balan *et al* [71] have created a self-optimising algorithm that alters the Hys and TTT based on the current HPI performance. The HPIs used are for ping-pong, handover failure and radio link failure and include a weight to manipulate their importance. Carvalho *et al* [72] changes the Hys and TTT autonomously in reaction to the occurrence of handover ping pong. Zhang *et al* [73] presented results showing that the parameters can be optimised in reaction to the number of cell-boundary crossings and the number of handovers performed in a period. Within the work described in this chapter, the novelty of the approach is that the handover parameters are altered based on location and using a neural network (SOM). To appreciate why location is important, consider two similar scenarios involving an active indoor user engaged in a call whilst moving around the interior of a building served by an indoor femtocell.

In the first scenario, as the mobile terminal approaches, and passes through, an external door (as shown in Figure 4.1) it is likely to detect an increase in the RSRP from an externally located macrocell. As a consequence, a measurement report will be transmitted from the mobile terminal to the femtocell base station informing the femtocell that handover may be required. However, failed handovers can occur here (when a user leaves the serving area of the femtocell) if a handover mechanism is too conservative (i.e. the TTT and Hys values are too high). Such failed handovers are likely to lower the Hys and TTT parameters, making future handover decisions more aggressive.

The second scenario (as shown in Figure 4.2) is slightly different: an active mobile terminal approaches a large window (with low penetration loss). The increase in RSRP from the macrocell will cause a measurement report to be transmitted from the mobile terminal to the femtocell which may invoke a handover, as in the first scenario. However, as the mobile terminal continues to move past the window, the relatively high RSRP from the macrocell is likely to decline rapidly and thus trigger another measurement report from the mobile terminal to the macrocell, indicating that a better RSRP can be obtained from the femtocell. Such actions will invoke a second handover, in quick succession, from the macrocell back to the femtocell (ping pong handover). Unnecessary handovers may cause an increase in the Hys and TTT parameters and in doing so make future handover more conservative. Modifying the parameters in this fashion may subsequently prove disastrous when the terminal leaves the building at some future time as described in the first scenario. The handover response may become so conservative that the call will be dropped before handover is executed. Note, there are occasions whereby an active mobile terminal approaches and pauses by a large window. Under such a circumstance, handover to the macrocell base station is unlikely to generate an unnecessary handover; nonetheless, it would be preferable to avoid such an eventuality in order to keep closed subscriber group traffic assigned to the femtocell where possible.

A handover parameter optimization algorithm is required to correctly balance the effect of these scenarios. To this end, a novel XSOM algorithm is used to optimize the handover parameters. Three principle regions are defined to predict the required changes in handover parameters with the same principles as in Chapters 4 and 5:

1. Areas of low signal strength from the macrocell. In these regions, a measurement report will not be generated and therefore the proposed algorithm need not consider them. For this reason such areas can be regarded as null zones.

2. Areas of high signal strength from the macrocell where few unnecessary handovers occur. These regions are referred to as permission zones since handover to the external base station will be beneficial. It is believed that these zones will coincide with architectural features such as external doors and yield low values of TTT and Hys due to low ping pong rates.

3. Areas of high signal strength from the macrocell where many unnecessary handovers occur. These regions are referred to as prohibition zones since handover to the external base station should be suppressed because it is likely that a second handover (in the opposite direction) will soon follow. These regions will be consistent with architectural features such as windows and glass exterior walls. The TTT and Hys values are likely to be high in these regions to avoid ping pong handovers.

The zones are depicted in Figure 4.3 to help gain an understanding of the areas within a room, as explained above.

Specifically, increasing the handover parameters reduces the probability of handover too early within a prohibition zone but increases the probability of handover too late within a permission zone when the parameters are optimised globally. Any handover optimization approach must facilitate the reduction of unnecessary handovers whilst seamlessly supporting handovers within permission zones. Therefore, the LTE handover optimization parameters, by themselves, are insufficient to optimize handover within an indoor scenario. Neither the femtocell nor the macrocell base stations can differentiate between a terminal approaching a window or a door, so attempts to tune the handover parameters without distinguishing between these scenarios will result in sub-optimum performance. Crucially, the handover parameters generally operate on a cell-wide level: the values of the parameters apply to the entire cell. In this chapter we assert that the best approach to adopt under these circumstances is to allow the parameters to be tuned and optimized on a location basis. The idea is to detect the locations within the handover environment that handover ping pong and failed handovers occur and use this information to change handover parameters. The central idea is to mitigate handover requests in regions that have a history of unnecessary handovers but not in regions that handover is ultimately required.

Due to the effects of shadowing and multipath propagation, a rise of stochastic variation in RSRP and signal quality may occur. Even a stationary terminal may receive better RSRP from a neighbouring base station at one instant and a worse RSRP the next due to movement of environmental scatterers. Such changes in RSRP can trigger unnecessary and unwanted handovers adding stress to the network. The TTT and Hys parameters control the timing of a triggered handover. The TTT and Hys values are pre-defined in LTE networks [22]. There are 16 valid TTT values as shown in 2.1. The Hys value varies in 0.5 dB steps between 0 and 10 dB. An optimization algorithm must find the best values for TTT and Hys that result in a statistical balance between the occurrence of both handover too early and handover too late.

In the work described here, the autonomic system will monitor when and where unnecessary handovers and dropped calls occur between the femtocell and macrocell and seek to reduce them over time by adapting the handover parameters based on location. The direction finding capability of MIMO systems is exploited to provide a profile of locations (i.e. regions in the radio environment) where handover is required (permissive zones) and those where unnecessary handovers are likely to occur (prohibition zones). An advanced kernel SOM is used to continually map locations where successful and unnecessary handovers have occurred and use this information to identify the permissive and prohibition zones.

6.2 Kernel Self Organising Map using X-means

6.2.1 Theory

The purpose of the autonomic managed element, to be included within SON, is to optimize the handover process based on the application of an improved kernel SOM [59, 64, 65]. The improvement of the kernel SOM is the inclusion of X-means within the unsupervised neural networking algorithm, as explained in Section 5.2. Similar to previous chapters, the Monitor phase of the SON algorithm is comprised of determining the location of the user and detecting where a handover measurement report has been triggered. The Analyse phase of the algorithm is based on a kernel SOM and allows the femtocell to learn the locations of the propagation environment that correspond to both permissive and prohibition zones. Next, the Plan phase takes this information and decides on an appropriate response; i.e. to increase or decrease the handover parameters. Finally, the Execute phase translates the decision from the Plan phase into LTE specific commands. It is the Plan phase that uses an improved kernel SOM algorithm to provide the femtocell with a profile of locations in which handover may take place.

Within this chapter, a modified kernel SOM is used (fully explained in Chapter 5). The kernel SOM is particularly useful for detecting clusters within data and in this work it is used to perform location fingerprinting based on RSRP and angle of arrival. The kernel SOM algorithm has a multi-dimensional input space, a weight space of the same dimension as the input and an output space of smaller dimension. A kernel SOM is a special version of the SOM that allows for a kernel method to replace the distance measurements within the SOM. Using kernel methods for a distance metric allows for a non-linear mapping from the input space to a high dimensional feature space which results in additional detail (accuracy) at the point of interest and reduces the vector quantization error that are inherent to SOM. There are four phases which describe the learning process of the kernel SOM: initialization, competition, cooperation, and synaptic adaptation. This algorithm has been augmented with a fifth stage, X-means. The advanced algorithm is composed as follows:

- Initialisation: the weights within the SOM are uniformly distributed within the region of the network (this corresponds to the propagation region of the femtocell). The parameters required for the SOM are initialised here.

- Competition: when an input is received by the algorithm (the location of the user), the weights within the network compete to identify the neuron that is most similar to the input. This results in a form of vector quantisation.

- X-means: this stage is added into the traditional kernel SOM algorithm. It allows for a Voronoi cell diagram to be created with each resulting cluster being a different cell in the diagram as shown in Figure 5.2.

- Cooperation: each weight within the network is updated if it is within the region of the winning node (calculated based on a monotonically decreasing sphere of influence) and in the same cell of the Voronoi diagram (calculated using X-means). This allows for group learning to occur in a more efficient manner than with the traditional algorithm.

- Synaptic Adaptation: each neuron and each of the parameters are updated, tend to a solution and do not learn indefinitely.

X-means allows for the SOM to be handled as a number of Voronoi cells and allows the neural networking model to learn faster due to a reduction in the level of false learning. This is a novel adaptation of the original SOM and kernel SOM algorithms. False learning occurs when a weight within the network is updated in an incorrect manner.

The XSOM algorithm is used in this model to determine the areas of the permission and prohibition zones, based on an estimate of distance (based on the RSRP) and AoA of the measurement report, using the autonomic control loop that is present in all autonomous systems. The femtocell locating where a handover trigger is transmitted from is the Monitor phase. The XSOM algorithm constitutes the Analyse phase with the input being the location of the user and allows the femtocell to learn the regions of the radio environment that handover is likely to take place in. The Plan phase decides if the current handover falls within a permission or prohibition zone based on the current TTT and Hys, and the Execute phase prohibits or permits the handover within the LTE network. This learning is completed in a group-based manner to allow faster convergence of the neurons within the network. The convergence of the neurons into accurate locations minimises the error inherent in the vector quantisation based algorithm, SOM.

6.3 Simulation Model and Results

NS3 simulations have been used to evaluate the performance of the modified kernel SOM algorithm in optimising indoor handovers by altering the TTT and Hys. Specifically, the system detects the regions within the radio environment where unnecessary handovers or failed handovers occur and seeks to reduce these to an optimum level over time by altering the TTT and Hys. To evaluate the algorithm, scenarios have been modelled that incorporate multiple numbers of permission and prohibition zones in a small room that would represent usage in the living room of domestic environment. To show the performance of the applied algorithm, the level of HPIs were evaluated along with the suppression rate and the location of the user. The parameters of the simulation environment that were used to demonstrate the effectiveness of the parameter optimization algorithm are summarised in Table 6.1. The parameters in Table 6.1 are common to all case studies presented within this chapter.

Table 6.1: Simulation Details

Parameter	Value
Simulation dimensions	7 m $\times$ 9 m
Room dimensions	7 m $\times$ 7 m
Exit area	2 m $\times$ 7 m
No. of mobile terminals	1
Direction change time	1.0 sec
Movement speed	1 - 3 m/sec
Initial position	centre
Mobility model	random walk
Propagation model	single-slope
Initial Hys	5 dB
Initial TTT	320 ms
Error	0 m
Neurons	100

When the user moves around the simulation environment, handover triggers take place at both the regions that correspond to prohibition and permission zones. Handover triggers are detected (*i.e.* when the mobile terminal has detected another base station with a stronger RSRP (by a Hys value) for a prescribed period of time (TTT)). The algorithm provides a modification that can be made to the standard LTE system that allows handover parameter modification based on location (the users location as perceived by the femtocell when the handover was triggered) while adhering to the handover process defined in the LTE standards. In this modified LTE system, the TTT and Hys values used are specific to the location of the user and are optimized as the system learns the success or failure of handover in this region. The values for the Hys and TTT will be different for permissive and prohibition zones.

The algorithm deployment is part of an autonomic control loop. The Monitor phase detects the location of the use when a handover trigger takes place. The location of the user constitutes the input of the XSOM algorithm which is the Analyse phase of the autonomic control loop. Once the XSOM algorithm has finished, the location of the user and the previous experience of handover in that area to plan whether to change the handover parameters or not (Plan phase). The Implementation phase then translates the outcome of the Plan phase into technology specific requirements and commands. The outcome of the Implementation phase controls any alterations made to the handover parameters. The TTT and Hys parameters are increased to the next allowed value when unneccessary handover has been detected and decreased to the next available value when a dropped call is detected.

The following case studies demonstrate that the mechanism can effectively adapt to the number of permission and prohibition zones that occur in an autonomic fashion and optimise the handover parameters for these zones independently. The performance of the system is linked to X-means and its ability to accurately detect the number of clusters in the environment. The case studies that demonstrate the ability of the novel XSOM algorithm followed by X-means and its ability to detect the correct number of clusters will now be described.

6.3.1 Case Studies

Case studies will now be presented to show the benefits of the XSOM algorithm when deployed to optimise handover parameters in a femtocell environment. The four case studies presented in this chapter are the same as the case studies presented for handover prohibition using the XSOM algorithm in Chapter 5. Case studies one and two will show typical performance of the algorithm and case studies three and four demonstrate the performance of the algorithm when an incorrect number of clusters are detected. The first case study incorporates one prohibition zone and one permission zone. The second case study incorporates two prohibition zones and a single permission zone. Case studies one and two demonstrate the algorithm's ability to detect clusters and learn the simulation environment in which it is deployed within.

The location of the user at the point of a handover trigger is detected by the monitoring stage of the autonomic system. The regions of the indoor environment that handovers are likely to occur and whether they relate to windows (prohibition zones) or doors (permission zones) are shown in Figures 6.1 and 6.2. The ability of the algorithm to process the inputs and successfully detect clusters is a key element that affects the performance of the algorithm. The initial setup of the femtocell allows convergence of the neurons within the XSOM to the locations of the radio environment where handover may take place by using the location of the handover triggers as the input to the neural network. Each weight within the neural network has the ability to retain knowledge of previous handovers within that area. Specifically, the regions within the radio environment where unnecessary or failed handovers are likely occur will be detected and the handover parameters altered accordingly. By reducing the occurrences of unnecessary and failed handovers within the femtocell environment, the number of handovers that occur will be optimised. A snapshot of the locations of 100 handover triggers for each case study is shown in Figures 6.3 and 6.4. Once the location of the mobile user has been detected, the analysis phase examines the data and decides on possible actions that can be taken: increase the TTT and Hys parameters; decrease the TTT and Hys parameters; or don't change the parameters. The parameters are updated on any neuron within the sphere of influence and cluster of the winning node. The plan phase uses the data and the possible actions to decide on an appropriate process that will be used to optimize the handover scenario. Each of the learning curves have been generated using 30 parallel simulation runs to provide an ensemble average.

At initialization, the handover parameters for all the nodes within the neural network are set to default values. Handover then operates as normal detecting where handover is unnecessary (handover too early) or where handover has failed (handover too late). This information is used to optimize the TTT and Hys values for each weight within the network. When an unnecessary handover is detected, TTT and Hys are both increased to their next higher allowed values. When a handover failure has been detected, TTT and Hys are both decreased to their next lower allowed values. The handover triggers (either macrocell to femtocell or femtocell to macrocell) occur in the regions where the macrocell RSRP is greater than the femtocell RSRP. Figures 6.5 and 6.6 show the values of TTT and Figures 6.7 and 6.8 show the values of Hys, for the weights within the network after 1000 handovers. Both the TTT and Hys values are changed at the same time to show that the use of position can be used to effectively alter the handover parameters. This simple approach avoids the effects of multipath and changing mobility conditions. A better alteration algorithm would likely yield better results.

Figures 6.5 and 6.6 show the values for TTT and figures 6.7 and 6.8 show the values for Hys, for all the weights within the network after 1000 handovers have taken place. Within these figures the dotted line represents the initial value and

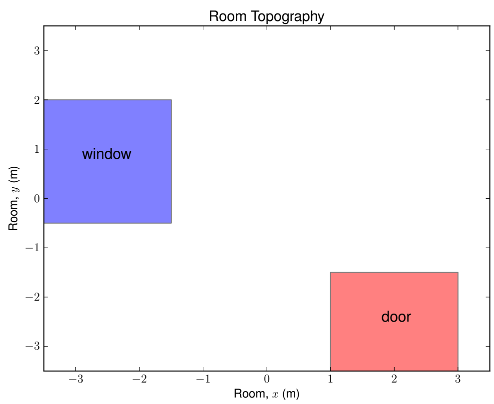

Figure 6.1: Case 1: Room topography

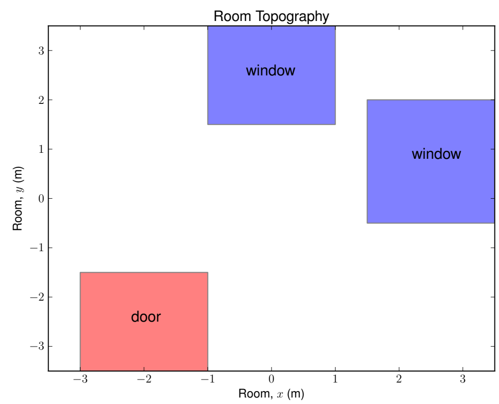

Figure 6.2: Case 2: Room topography

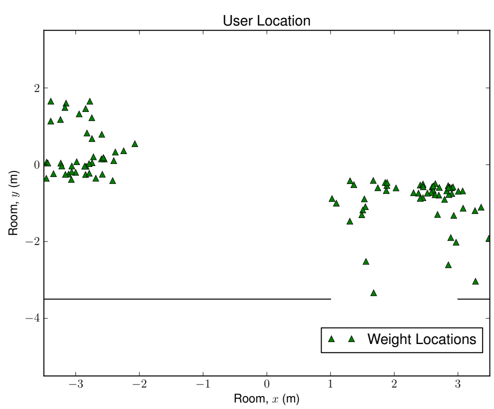

Figure 6.3: Case study 1: Handover locations

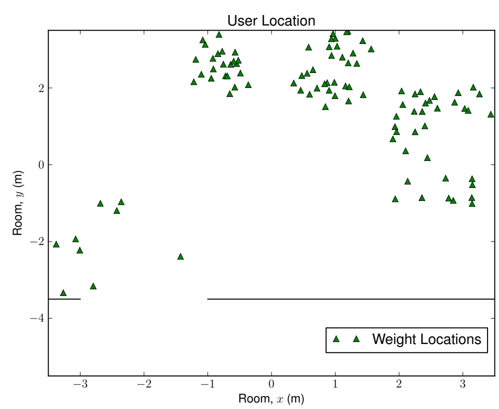

Figure 6.4: Case study 2: Handover locations

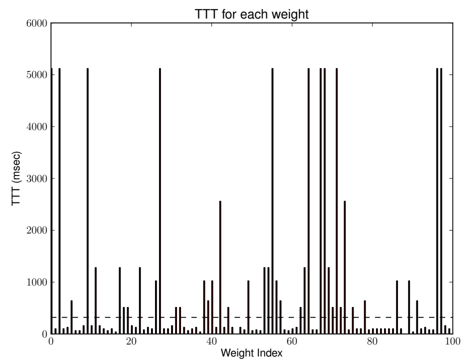

Figure 6.5: Case study 1: TTT for each weight

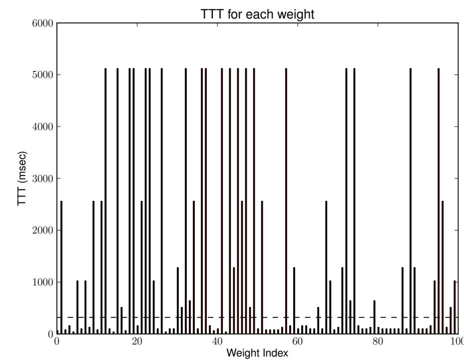

Figure 6.6: Case study 2: TTT for each weight

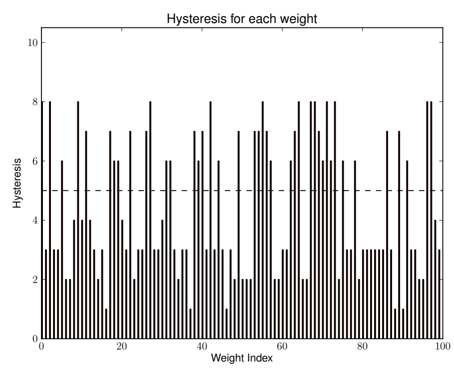

Figure 6.7: Case study 1: Hys for each weight

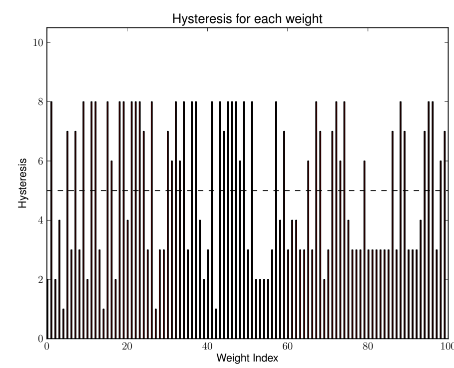

Figure 6.8: Case study 2: Hys for each weight

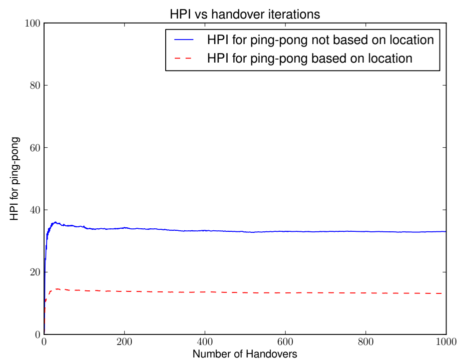

Figure 6.9: Case study 1: HPI for ping-pong handover

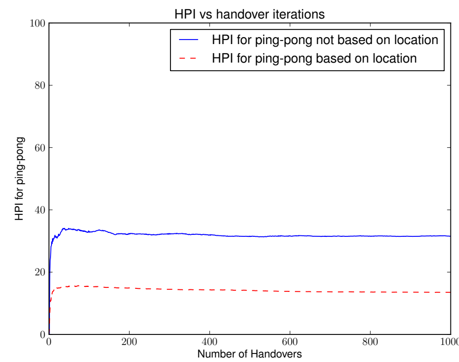

Figure 6.10: Case study 2: HPI for ping-pong handover

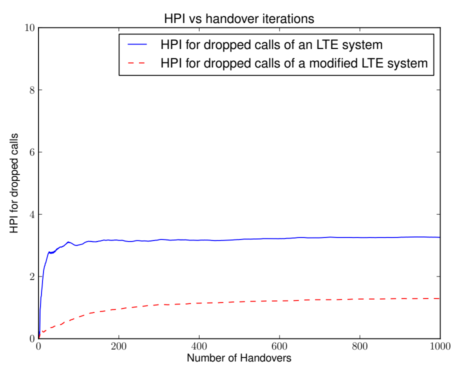

Figure 6.11: Case study 1: HPI for dropped calls

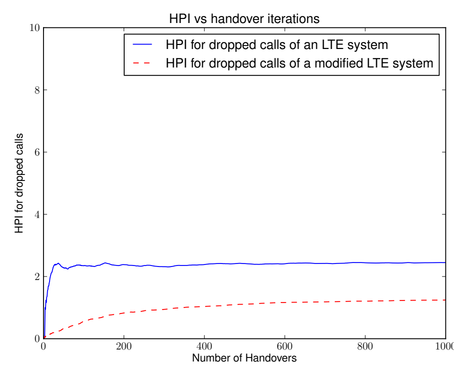

Figure 6.12: Case study 2: HPI for dropped calls

a threshold for where the weights are located. The weights with values above the horizontal dotted line are located within a prohibition zone and the weights with values below the dotted line are located within the region of a permissive zone. Due to the fine tuning of parameters, the number of handovers that take place have been optimized by reducing the level of unnecessary handovers and handover failures.

In order to demonstrate that the algorithm is an improvement on the basic LTE network, the HPIs are evaluated. In this case, the HPIs are the ping pong handover ratio and the handover dropped call ratio are of prime importance and are defined in Equations (3.3) and (3.2), respectively. The handover ping pong ratio for these specific scenarios are shown in Figures 5.11 and 5.12. The handover dropped call ratio for these specific scenarios are shown in Figures 5.13 and 5.14. The HPI figures include the performance of the network with and without the algorithm being proposed (XSOM). The handover parameter alteration for the scenario that location is not taken into consideration involves the same parameter alteration technique without location being taken into consideration (*i.e.* the alteration is done on a cell-wide level). In a standard LTE system the parameters will potentially not be altered or optimised.

$\mathrm{HPI}_{\mathrm{pp}}$ with and without the proposed optimization algorithm (XSOM) are shown in Figures 6.9 and 6.10 for the simulated scenarios. As can be seen from Figure 6.9 and 6.10, the situation includes a high level of handover ping pong occurrences. The Figures show that in case study one the situation involving the XSOM is 42.4 % of the situation when location is not taken into consideration to optimise the handover environment. Case study 2 shows a 41.9 % of no location being used to improve performance.

$\mathrm{N}_{\mathrm{Hfail}}$ with and without the proposed optimization algorithm (XSOM) are shown in Figures 6.11 and 6.12 for the simulated scenarios. Within the situations being simulated there is a minimal level of dropped calls occurring as a result of bad handover performance. Case study one shows that when location is taken into consideration N_{Hfail} is 38.2 % of the N_{Hfail} when location is taken into consideration and the proposed algorithm used. Case study 2 has a 44 % improvement when the proposed XSOM algorithm is used in conjunction with the location of the user.

The levels of the HPIs for both ping pong handover and dropped calls are better using the XSOM algorithm in conjunction with the location of the user. When the handover parameters are optimised on a cell-wide basis, handover failures and handover ping pong can occur regularly because prohibition and permission zones alter the parameters in opposing ways. When location is taken into consideration, the handover parameters can be altered for the prohibition and permission zones separately. The result of optimising the handover parameters using location results in a reduction of handover issues.

Case studies one and two have demonstrated the performance of the XSOM algorithm when it is operating ideally. Case studies three and four will then illustrate the performance of the algorithm when the number of clusters is not correctly detected. The layout of the prohibition and permission zones in case studies three and four are the same as in case study one and have one prohibition and one permission zone. This allows for a comparison to be made from the situation that the number of clusters is correctly detected to when the clusters are not correctly detected. Within case study three, the number of clusters is set to being 3 and within case study four, the number of clusters is set to 10. As a result, case studies three and four demonstrate the performance of the algorithm when the number of clusters have been marginally and majorly incorrectly detected.

As previously stated, the algorithm constitutes the main aspect of the autonomous control loop. In order to get the input to the XSOM algorithm, the location has to be correctly detected. The algorithm has to determine which locations correspond to permissive zones and which correspond to prohibition zones, shown in Figures 6.13

and 6.14. At initialisation, the algorithm spreads the weights of the neural network uniformly throughout the area of the network and has no knowledge of the topography of the deployed environment. The weights then adapt to the deployed environment and begin to optimise the handover parameters based on the region that the user is within. The weights and their ability to adapt to the environment of the femtocell allows the algorithm to "remember" and "learn" where handover issues may occur and how to avoid them by altering the handover parameters. Figures 6.15 and 6.16 diagrammatically represent the location of the user for one hundred handovers.

Figures 6.15 and 6.16 show that the location of the user has been detected successfully. Once the location of the user has been detected the Monitor phase has been completed and the Analyse phase can begin. The Analyse phase allows for the location of the user to be input into the XSOM algorithm and used to alter the handover parameters according to location and history. The Plan and implementation phase then chooses to increase the TTT and Hys parameters, decrease the TTT and Hys parameters; or not alter the parameters.

The handover parameters are initially set to default values for all the neurons within the network and are then optimised according to the handover success within regions of the propagation region of the femtocell. Handover operates as it would in the standard LTE system and the occurrence of handover ping pong and failed handovers are then used to alter TTT and Hys. When a handover ping pong (handover too early) happens the TTT and Hys are both increased for relevant weights to reduce the likelihood of it happening in that region of the propagation area, in the future. Handover failure due to a dropped call (handover too late) results in an decrease of the TTT and Hys for relevant weights within the network. Figures 6.17 and 6.18 show the values of TTT and Figures 6.19 and 6.20 show the values of Hys, respectively, for the weights within the network after 1000 handovers.

Figures 6.17 and 6.18 show the values for TTT and figures 6.19 and 6.20 show

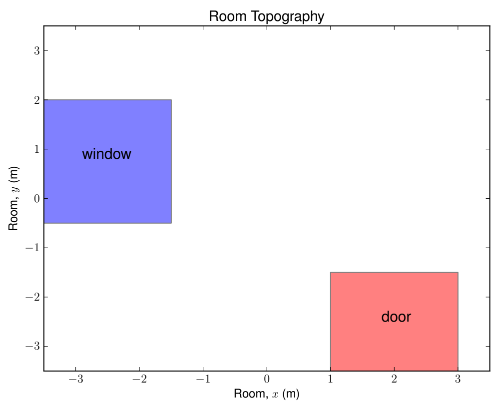

Figure 6.13: Case 3: Room topography

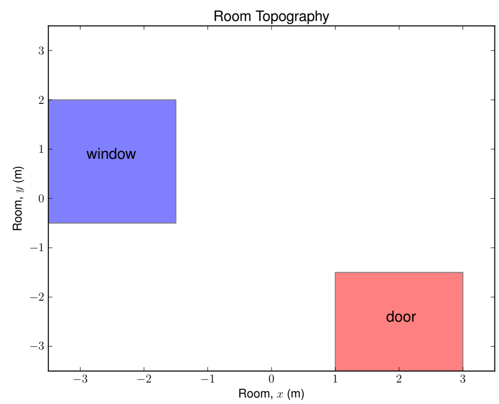

Figure 6.14: Case 4: Room topography

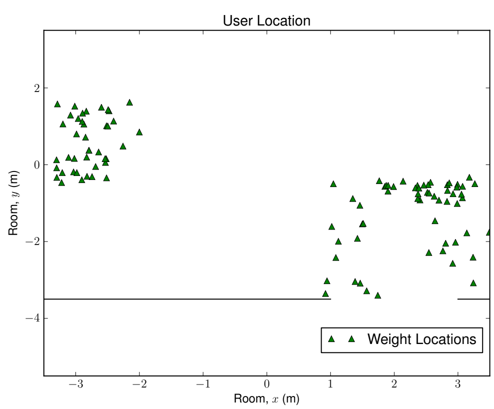

Figure 6.15: Case study 3: Handover locations

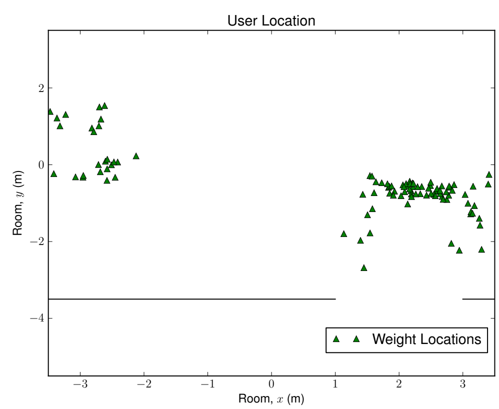

Figure 6.16: Case study 4: Handover locations

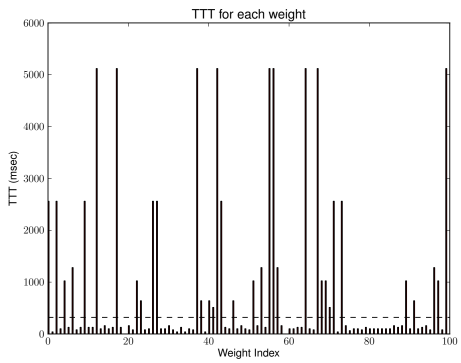

Figure 6.17: Case study 3: TTT for each weight

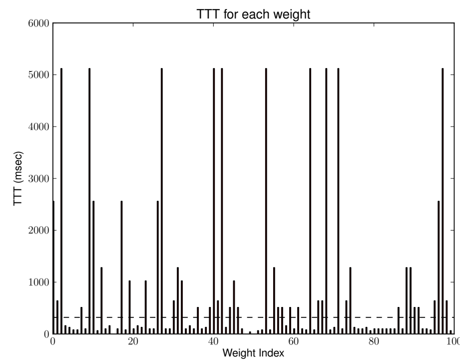

Figure 6.18: Case study 4: TTT for each weight

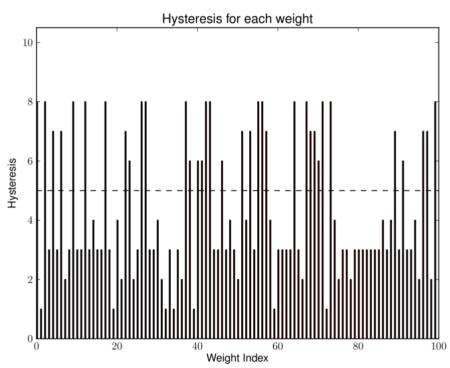

Figure 6.19: Case study 3: Hys for each weight

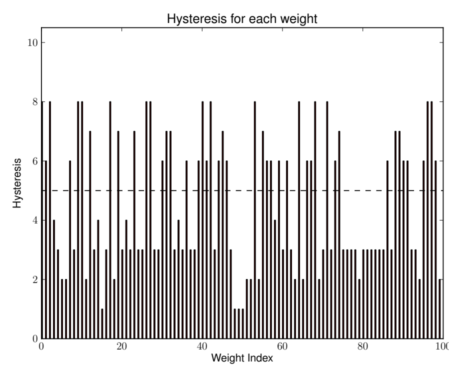

Figure 6.20: Case study 4: Hys for each weight

the values for Hys. Within these figures the dotted line represents the initial value and a threshold for where the weights are located. The weights with values above the horizontal dotted line are located within a prohibition zone and the weights with values below the dotted line are located within the region of a permissive zone. The algorithm successfully tunes handover parameters to location an optimises the handover process as a result. This optimisation leads to a reduction in handover issues by changing the parameters based on location.

To demonstrate the effectiveness of the algorithm the HPIs were calculated for both utilising location for the handover optimisation and not utilising location to modify handover parameters. Both of these parameter alteration techniques alter the parameters in the same way. However, one method alters the parameters on a cell-wide level and the other method alters the parameters of part of the propagation region of the femtocell. HPIs for both ping pong and dropped calls are of prime importance and are used to demonstrate the improvement that the XSOM algorithm can have to the operation of an LTE femtocell. The handover ping pong ratio for these specific scenarios are shown in Figures 6.21 and 6.22. The handover dropped call ratio for these specific scenarios are shown in Figures 6.23 and 6.24.

Figures 6.9 and 6.9 show the HPI for ping pong when utilising location in the parameter optimisation and when location is not used in the process. The result is that the HPI for handover ping pong in case three is 48.6 % more when not utilising location and the XSOM algorithm, in case three and 78.8 % more in case four.

Figures 6.11 to 6.12 depict N_{Hfail} with and without the XSOM algorithm that is being proposed. It can be seen that within case study three N_{Hfail} (when using location) is 48.6 % more than when not utilising location. N_{Hfail} when using location in case study four is 78.8 % more than when not utilising location. This is sensible because altering the parameters to regions that have specific handover issues will result in a reduction of poor handover performance which results in a reduction in

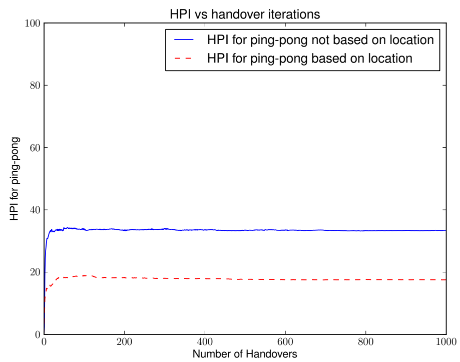

Figure 6.21: Case study 3: HPI for ping-pong handover

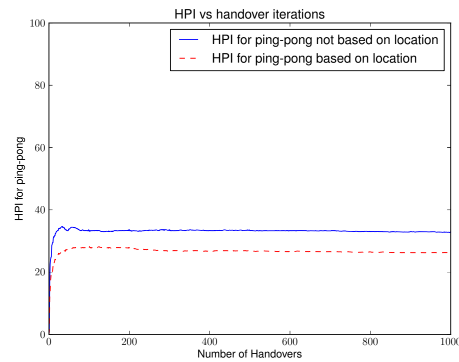

Figure 6.22: Case study 4: HPI for ping-pong handover

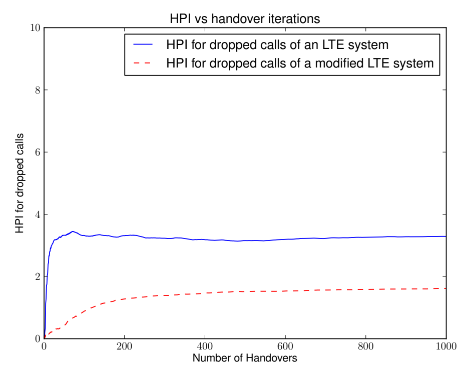

Figure 6.23: Case study 3: HPI for dropped calls

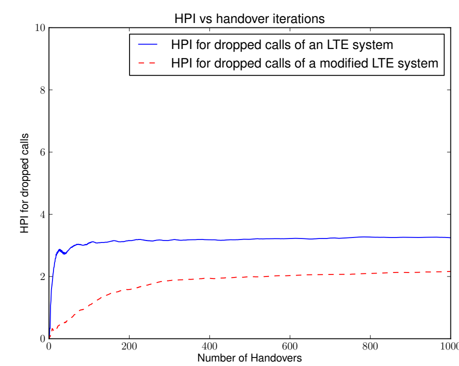

Figure 6.24: Case study 4: HPI for dropped calls

the number of handovers that take place.

Within Figures 6.9 to 6.12, the performance of the algorithm when the clusters were detected optimally was shown. Figures 6.9 to 6.24 show the performance of the algorithm when the number of clusters were not optimally detected. When the number of clusters is marginally incorrect (case study three), the handover ping pong HPI is 21.4 % more then when the number of clusters is correctly detected (case study one). When the number of clusters is extremely incorrect (case study four), the performance of the HPI for ping pong is 85 % more than when the number of clusters is detected optimally. The HPI for dropped calls using XSOM is 23.1 % more for case study three and 61.5 % more for case study four when compared to case study one. Using the HPIs it can be seen that when an incorrect number of clusters is used, the handover performance is an improvement over simpler algorithms. Any improvement over standard methods is a benefit to the network operator. However, when the correct number of clusters is detected and used, the algorithm performs well in adapting to its environment.

The number of handover ping pongs and failed handovers as a result of dropped calls are both lower for the optimized system that utilises the XSOM algorithm showing that the proposed algorithm is successful in optimizing the handover parameters. The locations of the handovers have been detected and the handover parameters have been optimized. The algorithm requires no prior information regarding the location of the windows or doors. This knowledge (or, at least, the equivalent radio environment knowledge) is gained via unsupervised learning.

6.3.2 X-means Cluster Detection

The result of inaccurate detection of clusters and its effect on XSOM was discussed in Section 5.3.2. The ability of X-means to autonomously estimate the correct clusters in the data, without any human input, is of prime importance and will be discussed

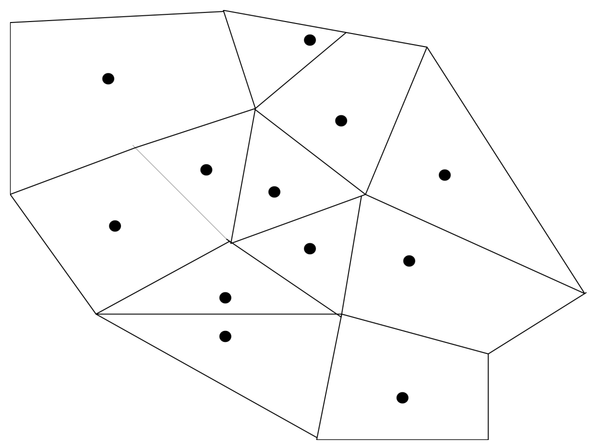

Figure 6.25: A general Voronoi diagram

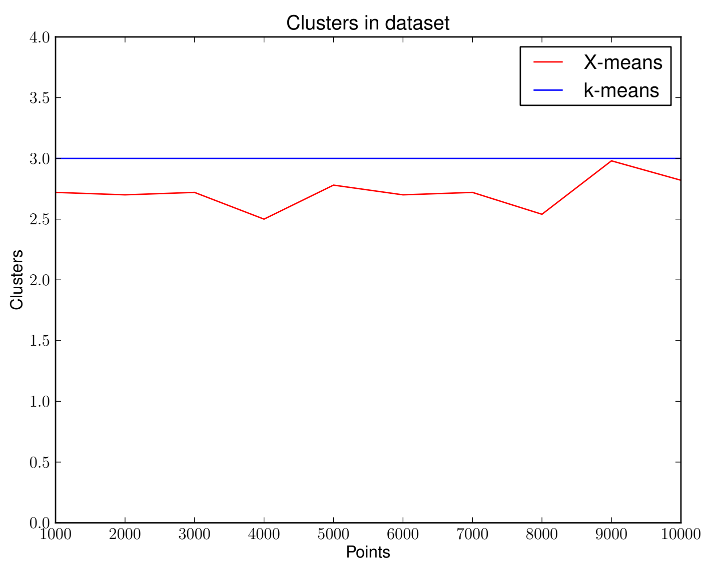

Figure 6.26: X-means calculating k with different numbers of data points

here. The more accurate the detection of clusters the better the XSOM algorithm can perform. The number of clusters detected is the number of cells within the Voronoi diagram (an example of a generic Voronoi diagram is shown in Figure 6.25). The number of clusters is made up of both the prohibition and permission zones within the indoor environment. Figure 6.26 shows the average number of clusters calculated using X-means with different numbers of data points (for 30 simulations providing an ensemble average) and compares this to the ideal number of clusters.

Figure 6.26 shows the average number of clusters calculated using X-means in a scenario that the number of clusters was 3. As can be seen, the X-means algorithm closely mimics the correct number of clusters which leads to successful operation of the XSOM algorithm. Case study three shows that when the clusters are detected slightly incorrectly, the performance degradation is minimal. Therefore, the X-means algorithm works well for use within a SOM algorithm (resulting in the XSOM algorithm).

6.4 Summary and Conclusions

In this chapter a novel kernel SOM algorithm has been proposed to improve the efficiency of handover within an indoor environment. The algorithm has been shown to effectively optimize both TTT and Hys values to reduce the number of handover too early and handover too late events. The handover parameters are optimized on the basis of radio environment location (related closely to physical location). The use of the kernel SOM allows the parameters being used in a handover-permissive zone to be different from those being used in a handover-prohibition zone. One of the main advantages of using this algorithm within SON is that it becomes more flexible with regards to the femtocell being able to adapt to its environment autonomically and improve handover efficiency in a fast and efficient manner.

The algorithm operates by monitoring the environment and waiting for a handover trigger to serve as an input to the algorithm. Once an input has been detected, the input is analysed and the algorithm plans and implements whether to increase, decrease or not change the handover parameters. The XSOM algorithm estimates the number of zones within the environment and classifies each zone as a prohibition or permission zone by detecting unnecessary handovers or dropped calls and altering the handover parameters accordingly. The algorithm quickly adapts to its environment and allows the femtocell to be more flexible to its environment by significantly reducing the handovers that occur.

Chapter 7

Conclusions and Future Work

7.1 Conclusion

LTE is expected to be a large and complex communications network. The increased size and complexity is a result of the improved data services required by users. The additional data services in indoor areas is provided by femtocells which result in an increase in handovers required to sustain user's calls. Within LTE, the handover signalling is optimised but the general handover process can still be improved. Detecting when handovers are not required is a complex task that when achieved can dramatically reduce the additional stress on the network.

This thesis has presented research into autonomous handover optimisation methods. By using the location of the user and neural networks, the number of handovers can successfully be reduced based on previous success or failure of handover in regions of the propagation environment. To this end, an effective software implementation of handover and neural network approaches have been implemented within NS3. Using this simulation platform, the merits of neural networks have been tested and evaluated.

Chapter 4 provides details of a proof of concept for improving handover performance using SOM based paradigms. Here, it was shown that by utilising the users location and a Kohonen SOM algorithm, the handover process can be optimised. The optimisation process was based on previous handover experience in a region of the propagation area of an LTE femtocell. By using the location of the user, false starts to handover can be minimised by prohibiting handover in areas that handover ping pong is likely to occur. This approach yields a potential reduction of handovers after a period of learning has taken place by removing unnecessary handovers. The algorithm leads to additional computational complexity in the femtocell but successfully optimises the number of handovers after a given period of time based on the location of the handover trigger. When implemented into a physical LTE femtocell, savings would be made on less power usage through less signalling messages and network resources which would benefit the network operator.

In Chapter 5 a novel SOM-based algorithm was presented. This novel algorithm was applied to the same simulation environment for handover optimisation as the algorithm in Chapter 4. The result of this chapter was that the XSOM algorithm provides an improvement to the performance of the Kohonen SOM algorithm and consequently the standard LTE approach. The learning rate of the XSOM algorithm was better than that of the Kohonen SOM. Also, the XSOM algorithm yields a potential reduction in the number of required handovers by prohibiting all unnecessary handovers. The improvement in the number of handovers requires more processing from the femtocell but leads to a more efficient use of the network resources. When the algorithm is in operation, any reduction in the level of handovers that occur is beneficial to the network operator. Prohibiting unnecessary handovers while not affecting the users perception of the network (through dropped calls and low QoS) results in a more efficient use of the network resources for the network operator.

In Chapter 6, the XSOM algorithm proposed in Chapter 5 was used for handover parameter optimisation. In Chapters 4 and 5 it was shown that handover can be prohibited in regions that it is not required. However, in situations where the handover regions are difficult to determine, this may not be an efficient method of optimising handovers. The handover parameter optimisation completed within this chapter was to alter the Hys and TTT within specific regions of the propagation environment of the femtocell. When handover ping pong was detected, the parameters would be increased to their next permitted values. When a dropped call was detected during the handover process, the parameters would be decreased to their next permitted values. Using this approach it was expected that the parameter values would be high in prohibition zones and low in permission zones to accommodate the difference in user behaviour within these areas. This was then tested and proven to provide improved handover performance by reducing both handover failures due to dropped calls and handover ping pong. This approach uses a parameter optimisation method in conjunction with the users location to optimise the parameter values. Optimally set handover parameters is beneficial for the user because the detrimental effect of badly set parameters can cause severe harm to the users QoS. This approach allows the network operator to optimally set the parameters with no human input and therefore no increase in OPEX.

Within this thesis, optimisation methods applicable to handover performance improvement have been proposed and tested. Handover is expensive for the network operator through consumption of radio channels and fixed links; through additional processing load in admission control, bearer setting and path switching; and the potential to degrade the QoS of ongoing connections. Optimising handover effectively improves the efficiency of the network but, if done incorrectly, the change to the system can be detrimental and affect the users use of the network. Reducing the number of handovers and the success of these handovers is a key aim of mobile network operators. This thesis has described approaches to fulfil this aim.

7.2 Summary of Contributions

The thesis has attained several achievements. The main contributions can be summarised as follows:

- The effective software implementation of handover and neural networks while adhering to LTE specifications within LTE.
- The implementation of a proof of concept that demonstrates the performance of SOM for handover optimisation.
- Creation of a novel algorithm implementing k-means into a kernel SOM.
- Creation of a novel algorithm that incorporates X-means into the standard kernel SOM algorithm (XSOM).
- Assessment of novel algorithms compared to standard LTE approaches.
- Effective parameter optimisation based on location while utilising neural networks.

The research contributions of this thesis have resulted in several publications that are listed in the List of Publications.

7.3 Future Work

Neural networks and their application to telecommunications management is a field that has many exciting possibilities. A mobile communications system with perfect handover performance has yet to be achieved due to the complexities of both the wireless communications environment and complex human behaviour. Providing simulation prototypes to demonstrate a performance improvement is only the first step in developing advanced systems. The simulations described within this thesis prove that SOMs can be used to effectively improve handover performance. Since this is the beginning of the work, there is much future work that can be undertaken based on the work presented within this thesis.

The work within this thesis involved using a SOM and progressively developing the algorithm to improve its performance in reducing the number of handovers that take place. The algorithm can be developed to further improve the algorithm's performance. Following the implementation of SOM based algorithms, there are other algorithms that could be tested. Reinforcement learning would be an interesting technique to improve the decision process of the algorithm. Reinforcement learning uses an intelligent process of exploration and exploitation to choose the best output from the given input. This would potentially yield a better performance than the current thresholding method being used. However, due to time restrictions this complex method has not been implemented or tested.

The algorithm developed for use within an indoor environment for handover performance improvement within LTE has been shown to be successful. In order to further test the implementation of this algorithm in a mobile communications environment, the simulation test bed should be developed to more closely mimic real world performance by including additional mobile communications issues. Currently most mobile communications issues have been abstracted off to allow development of the algorithm and not the implementation environment. The simulation environment could be changed to include aspects such as interference, additional base stations, base station load, throughput evaluation and multiple users. Additional case studies could also be developed within the simulation environment to further test the performance of the algorithms. Changing the case studies requires further development of the simulation test bed. The algorithm development was the main aim of the work completed within this thesis.

Now that simulations have been completed to demonstrate the improvement that the algorithm can be used to optimise the handover environment, practical implementation and testing would be the next step of the development. If an LTE femtocell could be purchased and altered to include the algorithms presented within this thesis then practical testing could be completed to confirm the viability of this approach. An LTE femtocell was not purchased for the work within this thesis because of their lack of availability during the planning stage of the PhD. These became more available at a later date and are available now since LTE is being deployed throughout the world.

Other future work would be to test the accuracy of the location detection ability of LTE femtocells. The work in this thesis makes assumptions about the error in the location detection ability of LTE femtocells using AoA and RSRP. The position estimation error has been shown within this thesis to have a minimal effect on the outcome of the algorithm. However, the lower the location error, the more likely that the algorithm will perform optimally in all deployed environments.

References

[1] S. Brightfield, "Femtocells - how to get the most out of your nework," 4G World Presentation, Chicago, USA, October 2011.

[2] 3GPP TS 32.500 V8.0.0, "Technical Specification Group Services and System Aspects; Telecommunications Management; Self Organizing Networks (SON); concepts and requirements (Release 8)," December 2008.

[3] White Paper, "The benefits of SON in LTE," Tech. Rep., 3G Americas, 2009.

[4] N. Sinclair, D. Harle, I.A. Glover, and R.C. Atkinson, "A neural networking approach for indoor to outdoor handover," in *Festival of Radio Science (FRS)*, Durham, UK, April 2012.

[5] N. Sinclair, D. Harle, I.A. Glover, and R.C. Atkinson, "A kernel methods approach to reducing handover occurrences within LTE," in *18th European Wireless Conference (EW)*, Poznan, Poland, April 2012, pp. 1 –8.

[6] N. Sinclair, D. Harle, I.A. Glover, J. Irvine, and R.C. Atkinson, "An advanced SOM algorithm applied to handover management within LTE," vol. 62, no. 5, pp. 1883–1894, 2013.

[7] N. Sinclair, D. Harle, I.A. Glover, J. Irvine, and R.C. Atkinson, "Parameter optimization for LTE handover using an advanced SOM algorithm," in *77th Vehicular Technology Conference (VTC Spring)*, Dresden, Germany, June 2013.

[8] 3GPP TS 25.367 V8.0.0, "Technical Specification Group Services and System Aspects; Universal Mobile Telecommunications System (UMTS); mobility procedures for Home Node B; overall description; Stage 2," 2009.

[9] 3GPP, "Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Architecture Description," TS 36.401, 3rd Generation Partnership Project (3GPP), December 2009, Release 9.

[10] F. Adachi, "BP-2: The Long-Term Evolution of 3G: Evolved UTRA and UTRAN," in *The Institute of Electronics, Information and Communciation Engineers (IEICE)*, Kitakyusyu-shi, Japan, March 2008.

[11] D.S. Shah, "A tutorial on LTE evolved UTRAN (eUTRAN) and LTE Self Organizing Networks," M.S. thesis, The University of Texas, December 2010.

[12] 3GPP, "Universal Terrestrial Radio Access Network (UTRAN); Overall Description," TS 25.401, 3rd Generation Partnership Project (3GPP), December 2009, Release 9.

[13] N. C. Lobley, "GSM to UMTS - architecture evolution to support multimedia," *BT Technology Journal*, vol. 19, no. 1, pp. 38–47, Jan. 2001.

[14] 3GPP, "Radio Access Network: Radio Transmission and Reception," TS 45.005, 3rd Generation Partnership Project (3GPP), December 2009, Release 9.

[15] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA) ; S1 Application Protocol (S1AP)," TS 36.413, 3rd Generation Partnership Project (3GPP), December 2009, Release 9.

[16] P. Lescuyer and T. Lucidarme, *Evolved Packet System (EPS): The LTE and SAE Evolution of 3G UMTS*, Wiley Publishing, 2008.

[17] 3GPP, "Evolved Universal Terrestrial Radio Access Network (E-UTRAN); X2 Application Protocol (X2AP)," TS 36.423, 3rd Generation Partnership Project (3GPP), December 2009, Release 9.

[18] T. Jansen, I. Balan, J. Turk, I. Moerman, and T. Kurner, "Handover parameter optimization in LTE Self-Organizing Networks," in *72nd Vehicular Technology Conference (VTC Fall)*, Ottawa, Canada, September 2010, pp. 1 –5.

[19] W. Shaohong, Z. Xin, Z. Ruiming, Y. Zhiwei, F. Yinglong, and Y. Dacheng, "Handover study concerning mobility in the two-hierarchy network," in *69th Vehicular Technology Conference (VTC Spring)*, Barcelona, Spain, 2009, pp. 1–5.

[20] Z. Ghadialy, "Handover principles and concepts," Tech. Rep., February 2005.

[21] L. Wang, Y. Zhang, and Z. Wei, "Mobility management schemes at radio network layer for LTE femtocells," in *69th Vehicular Technology Conference (VTC Spring)*, Barcelona, Spain, 2009, pp. 1–5.

[22] 3GPP TS 36.331 V9.3, "Technical Specification Group Radio Access Network; evolved Universal Terrestrial Radio Access (e-UTRA); Radio Resource Control (RRC); protocol specification (Release 9)," June 2010.

[23] 3GPP TS 36.214 V8.5.0, "Technical Specification Group Radio Access Network; evolved Universal Terrestrial Radio Access (e-UTRA); physical layer - measurements (Release 8)," December 2008.

[24] ETSI TS 132.521 V9.0.0, "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); telecommunications

management; Self Organizing Networks (SON); policy Network Resource Model (NRM); Integration Reference Point (IRP); Requirements (3GPP TS 32.521 version 9.0.0 Release 9)," April 2010.

[25] 3GPP, "Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Self-Configuring and Self-Optimizing Network (SON) Use Cases and Solutions," TR 36.902, 3rd Generation Partnership Project (3GPP), December 2009, Release 9.

[26] H. Claussen, L.T.W. Ho, and L.G. Samuel, "Self-optimization of coverage for femtocell deployments," in *Wireless Telecommunications Symposium (WTS)*, Pomona, California, USA, April 2008.

[27] J. Zhang and G. de la Roche, *Femtocells: Technologies and Deployment*, Wiley, 2011.

[28] S. Saunders, S. Carlaw, A. Giustina, R.R. Bhat, V.S. Rao, and R. Siegberg, *Femtocells: Opportunities and Challenges for Business and Technology*, Telecoms Explained. Wiley, 2009.

[29] Informa Telecoms and Media, "Small cell market status," Tech. Rep., Small Cell Forum, June 2012.

[30] S. Wang, W. Guo, and T. O'Farrell, "Energy efficiency evaluation of SISO and MIMO between LTE-femtocells and 802.11n networks," in *75th Vehicular Technology Conference (VTC Spring)*, Yokohama, Japan, May 2012, pp. 1 –5.

[31] Vodafone, "Vodafone sure signal," http://www.itpro.co.uk/619615/vodafone-rebrands-femtocell-system-as-sure-signal, Accessed: July 2013.

[32] D. Duffy, "Small cells outnumber traditional mobile base stations," Small Cell Forum, October 2012.

[33] L.G.U. Garcia, K.I. Pedersen, and P.E. Mogensen, "Autonomous component carrier selection: Interference management in local area environments for LTE-Advanced," *Communications Magazine*, vol. 47, 2009.

[34] V. Chandrasekhar, J. Andrews, and A. Gatherer, "Femtocell networks: A survey," *IEEE Communications Magazine*, vol. 46, September 2008.

[35] P. Lee, T. Lee, J. Jeong, and J. Shin, "Interference management in LTE femtocell systems using fractional frequency reuse," in *12th International Conference on Advanced Communication Technology (ICACT)*, Gangwon-Do, Korea, February 2010, vol. 2, pp. 1047 –1051.

[36] C. Bouras, G. Kavourgias, V. Kokkinos, and A. Papazois, "Interference management in LTE femtocell systems using an adaptive frequency reuse scheme," in *Wireless Telecommunications Symposium (WTS)*, London, UK, April 2012, pp. 1 –7.

[37] M. Bennis, L. Giupponi, E.M. Diaz, M. Lalam, M. Maqbool, E.C. Strinati, A. De Domenico, and M. Latva-aho, "Interference management in self-organized femtocell networks: The BeFEMTO approach," in *2nd International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology (Wireless VITAE)*, Chennai, India, March 2011, pp. 1 –6.

[38] J. Alonso-Rubio, "Self-optimization for handover oscillation control in LTE," in *Proc. IEEE Network Operations and Management Symposium (NOMS)*, Osaka, Japan, April 2010, pp. 950 – 953.

[39] B. Jennings, S. van der Meer, S. Balasubramaniam, D. Botvich, M.O. Foghlu, W. Donnelly, and J. Strassner, "Towards autonomic management of communications networks," *IEEE Communications Magazine*, vol. 45, October 2007.

[40] D. Raymer, S. van der Meer, and J. Strassner, "From autonomic computing to autonomic networking: An architectural perspective," in *5th IEEE Workshop on Engineering of Autonomic and Autonomous Systems (EASE)*, Bari, Italy, April 2008.

[41] P. Horn, "Autonomic computing: IBM's perspective on the state of information technology," Technical report, IBM corporation, October 2001.

[42] Q. Mahmoud, *Cognitive Networks: Towards Self-Aware Networks*, Wiley-Interscience, 2007.

[43] S. Feng and E. Seidel, "Self-organizing networks (SON) in 3gpp Long Term Evolution," May 2008.

[44] T. Kurner, "Final report on self-organisation and its implications in wireless access networks," SOCRATES Report, 2010.

[45] L. Gabriel, M. Grech, F. Kontothanasi, A. Mukhopadhyay, M. Nicoloau, and A. Sharma, "Economic benefits of SON features in LTE networks," in *34th Sarnoff Symposium*, Princeton, USA, May 2011, pp. 1 –5.

[46] NEC Corporation, "Self Organizing Network: NEC's proposals for next-generation radio network management," White paper, NEC Corporation, February 2009.

[47] K. Kitagawa, T. Komine, T. Yamamoto, and S. Konishi, "A handover optimization algorithm with mobility robustness for LTE systems," in *22nd International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, Toronto, Canada, September 2011, pp. 1647 –1651.

[48] L. Ewe and H. Bakker, "Base station distributed handover optimization in LTE Self-Organizing Networks," in *22nd International Symposium on Personal Indoor*

and Mobile Radio Communications (PIMRC), Toronto, Canada, September 2011, pp. 243 –247.

[49] D. Duffy, "Small cells to make up almost 90% of all base stations by 2016," Small Cell Forum, February 2012.

[50] G. Yang, X. Wang, and X. Chen, "Handover control for LTE femtocell networks," in *International Conference on Electronics, Communications and Control (ICECC)*, Ningbo, China, September 2011, pp. 2670 –2673.

[51] Z. Becvar and P. Mach, "Adaptive hysteresis margin for handover in femtocell networks," in *6th International Conference on Wireless and Mobile Communications (ICWMC)*, Valencia, Spain, September 2010, pp. 256 –261.

[52] A.K. Jain, Jianchang Mao, and K. M. Mohiuddin, "Artificial neural networks: a tutorial," *Computer*, vol. 29, no. 3, pp. 31–44, 1996.

[53] T.M. Mitchell, *Machine Learning*, McGraw-Hill, Inc., New York, NY, USA, 1st edition, 1997.

[54] M.M.C. Nelson and W. T. Illingworth, *A Practical Guide to Neural Networks*, Addison-Wesley, 1994.

[55] S.F. Zornetzer, C. Lau, J.L. Davis, and T. McKenna, *An Introduction to Neural and Electronic Networks*, Academic Press Inc., Orlando, FL, USA, 2nd edition, 1994.

[56] R.D. Gardner and D.A. Harle, "Alarm correlation and network fault resolution using the kohonen self-organising map," in *Global Telecommunications Conference (GLOBECOM)*, Phoenix, USA, 1997, vol. 3, pp. 1398–1402 vol.3.

[57] S. Akoush and A. Sameh, "The use of bayesian learning of neural networks for mobile user position prediction," in *7th Intelligent Systems Design and Applications Conference (ISDA)*, Rio de Janeiro, Brazil, 2007, pp. 441–446.

[58] J. Espi, R. Atkinson, D. Harle, and I Andonovic, "An optimum network selection solution for multihomed hosts using hopfield networks," in *9th International Conference on Networks (ICN)*, Menuires, France, 2010, pp. 249–254.

[59] S. Haykin, *Neural Networks and Learning Machines*, chapter 9: Self-Organizing Maps, pp. 453 – 502, Pearson, third edition, 2009.

[60] sparknotes, "Structure and function of animals," http://www.sparknotes.com/testprep/books/sat2/biology/chapter9section1.rhtml, 2011, Accessed: July 2013.

[61] B.M. Wilamowski, "Neural network architectures and learning algorithms," *IEEE Industrial Electronics Magazine*, vol. 3, no. 4, pp. 56 –63, December 2009.

[62] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, vol. 43, pp. 59–69, 1982.

[63] R. Thanawala, M. El-Sayed, T. Morawski, A. Mukhopadhyay, J. Zhao, and C. Urrutia-Valdes, "The mobile data explosion and new approaches to network planning and monetization," in *14th Telecommunications Network Strategy and Planning Symposium (NETWORKS)*, Warsaw, Poland, September 2010, pp. 1 –4.

[64] D. MacDonald and C. Fyfe, "The kernel Self-Organising Map," in *4th International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies (KES)*, Brighton, UK, 2000, vol. 1, pp. 317 –320 vol.1.

[65] M. Martin-Merino and A. Munoz, "Extending the som algorithm to non-euclidean distances via the kernel trick.," in *11th International Conference on*

Neural Information Processing (ICONIP), Calcutta, India, 2004, vol. 3316 of *Lecture Notes in Computer Science*, pp. 150–157, Springer.

[66] D. Pelleg and A. Moore, "X-means: Extending k-means with efficient estimation of the number of clusters," in *17th International Conference on Machine Learning (ICML)*, California, USA, 2000, pp. 727–734, Morgan Kaufmann.

[67] W. Min and Y. Siqing, "Improved k-means clustering based on genetic algorithm," in *International Conference on Computer Application and System Modeling (ICCASM)*, Taiyuan, China, October 2010, vol. 6, pp. V6–636 –V6–639.

[68] G. Hamerly and C. Elkan, "Learning the k in k-means," in *17th Neural Information Processing Systems (NIPS)*, Vancouver, Canada, 2003, p. 2003, MIT Press.

[69] L.Y. Tseng and S.B. Yang, "Genetic algorithms for clustering, feature selection and classification," in *International Conference on Neural Networks (ICNN)*, Houston, USA, June 1997, vol. 3, pp. 1612 –1616 vol.3.

[70] B. Scholkopf, "The kernel trick for distances," in *TR MSR 2000-51, Microsoft Research*, 1993, pp. 5–3.

[71] I. Balan, T. Jansen, B. Sas, I. Moerman, and T. Kurner, "Enhanced weighted performance based handover optimization in LTE," in *Future Network Mobile Summit (FutureNetw)*, Warsaw, Poland, 2011, pp. 1–8.

[72] M. Carvalho and P. Vieira, "An enhanced handover oscillation control algorithm in LTE self-optimizing networks," in *Wireless Personal Multimedia Communications (WPMC)*, Brest, France, 2011, pp. 1–5.

[73] H. Zhang, X. Wen, B. Wang, W. Zheng, and Z Lu, "A novel self-optimizing handover mechanism for multi-service provisioning in LTE-advanced," in *International Conference on Research Challenges in Computer Science (ICRCCS)*, Wuhan, China, 2009, pp. 221–224.

[74] A.M. Law and D.M. Kelton, *Simulation Modeling and Analysis*, McGraw-Hill Higher Education, 3rd edition, 1999.

[75] R.Y. Rubinstein and D.P. Kroese, *Simulation and the Monte Carlo Method*, Wiley Series in Probability and Statistics. Wiley, 2008.

[76] J. Banks, *Discrete-Event System Simulation*, Prentice Hall, 2010.

[77] G.A. Wainer, *Discrete-Event Modeling and Simulation: A Practitioner's Approach*, Computational Analysis, Synthesis, and Design of Dynamic Systems. Taylor & Francis, 2010.

[78] "ns-3," http://www.nsnam.org/, February 2013, Accessed: July 2013.

[79] "ns-3 tutorial," Tech. Rep., December 2012.

[80] "ns-3 manual," Tech. Rep., December 2012.

[81] "The network simulator - ns-2," http://nsnam.isi.edu/nsnam/index.php/ User-Information, February 2013, Accessed: July 2013.

[82] "ns-3 documentation," http://www.nsnam.org/docs/release/3.9/doxygen/ index.html, February 2011, Accessed: July 2013.

[83] "ns-3::propagationlossmodel class reference," http://www.nsnam.org/docs/ release/3.9/doxygen/index.html, February 2011, Accessed: July 2013.

[84] "ns-3::mobilitymodel class reference," http://www.nsnam.org/docs/release/3.9/ doxygen/index.html, February 2011, Accessed: July 2013.

[85] "ns-3 testing and validation," Tech. Rep., January 2010.

[86] S. Haykin, *Neural Networks and Learning Machines*, chapter 9: Self-Organizing Maps, p. 474, Pearson, third edition, 2009.

Appendix A

System Modelling and Evaluation

A.1 Introduction

The rising complexity of communications systems increases the difficulty in evaluating the performance of new techniques. However, due to the high level of costs involved with system implementation and deployment, all potential techniques must be evaluated before being deployed within hardware prototypes or communications systems. Simulation modelling [74] is a useful tool for technique analysis and testing. Simulation modelling involves various abstractions of system behaviour and provides a compromise between cost, time and complexity. Practically, this means a computer can be used to execute the simulation runs in relatively moderate time scales while keeping the cost low. The appropriateness of the results is high because details can be incorporated as required with lots of details of user behaviour and few restrictions. Additionally, accurate conclusions can be obtained with a good simulation model. Computer simulations provide a platform to perform consistent repeatable results. This chapter concentrates on describing the simulation tools that have been used within system level simulations for the work in this thesis.

This appendix first describes the testing and validation procedure of pre-made models within NS3. After that, the testing completed on the neural network models within this thesis will be described. A SOM (described in Chapter 4) is then used as an illustrative example to explain the testing procedure.

A.2 Simulation Types

There are multiple different types of simulation models that can be created. To accomplish the requirements of the simulation models there are multiple types of simulation: Monte Carlo simulation and Discrete Event Simulations will be explained. Emulation is similar to simulation and will also be discussed. Each different simulation type has its own strengths, weaknesses and purpose. The main types of simulations will now be described and explained.

A.2.1 Emulations

Emulations are a type of simulation model that replicate the functions of a computer system in another computer system. The emulated behaviour should closely resemble that of the real system. The concept of an emulation can be applied to many domains that retain the key concept of one system (hardware or software) pretending to be another system. In communications networking, one or more computers can be used to mimic the behaviour of an entire network. By utilising emulations, network attributes can be changed in a simpler manner than with a physical network while still behaving like a real network. The costs required for emulations tend to be higher than other styles of simulations since they usually involve more hardware. This method was not utilised in this thesis because the costs would be too high and the system would be complex to emulate in its entirety.

A.2.2 Monte Carlo Simulations

Monte Carlo methods [75] are a class of computational algorithms that are used to solve problems that are not analytically tractable. They are used to model scenarios that involve events that repeat over time. The Monte Carlo methods vary but generally involve: identifying possible inputs; generating inputs from the input domain; calculating the deterministic outputs; and finally, aggregating the results from each run of the simulator. Monte Carlo methods are mainly used in situations where the behaviour of the system does not change with time and is therefore not applicable to the work within this thesis.

A.2.3 Discrete Event Simulations

Discrete Event Simulators (DES) [76, 77] are a type of simulation technique that model systems based on a sequence of events using discrete state variables. When an event has been triggered in the simulation at a given time the state of the system will be changed. Since a DES is based on events, when no event occurs there is no change in the state of the system.

The operations of a DES are based on some principle components: clock; events; event handler; and event list. The simulation clock keeps track of the simulation time and is used to control the timing of events. The events are a series of structures generated by a simulation entity that occur at a specified time and are controlled by the event handler. The event handler is a call-back function that is called when the simulation clock is equal to an event activation time. The activation time can be both deterministic or stochastic in nature. The event list contains the list of scheduled events and their activation time. These elements are all combined to create the basis of a DES system. When a simulation is started, initial events are added to the event list. Then the event list is checked to ensure it is populated with events; if there are events in the list, the one with the earliest activation time is compared to the clock time and executed at the appropriate time. This process continues until there are no longer any events in the event list and/or the simulation is ended. It should be noted that when an event is being handled other events can be added, removed or modified within the scheduler. This process is depicted in Figure A.1.

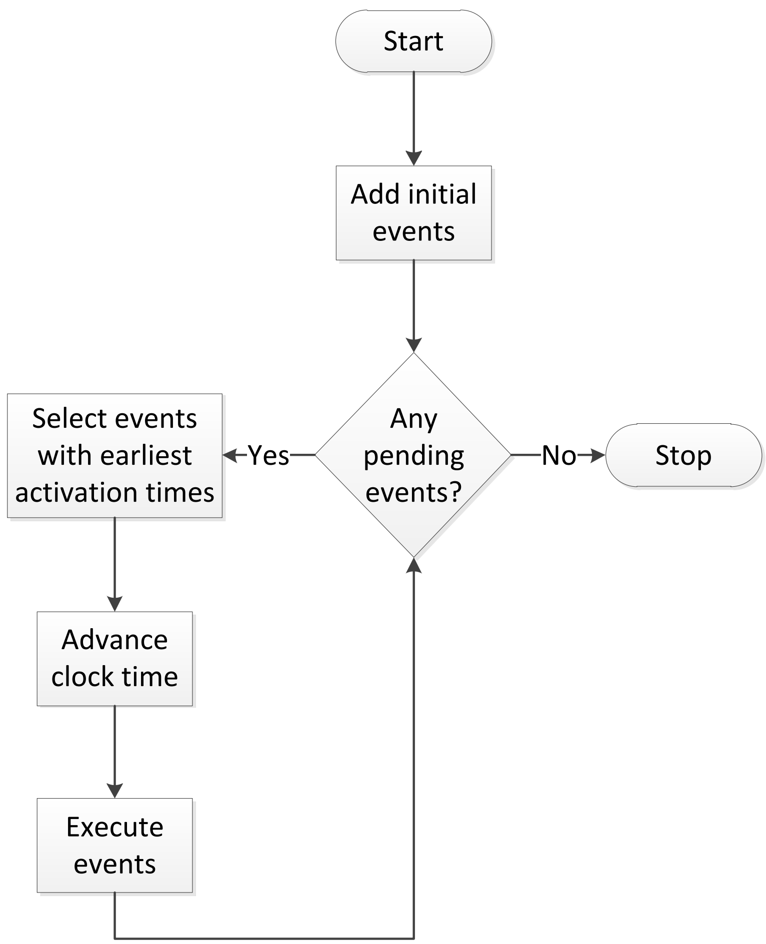

Figure A.1: DES execution flow

The fundamental nature of DES systems is controlled by Pseudo Random Number Generators (PRNG). Using such an approach there is a stochastic approach to the technique. Within a networking DES scenario events such as packet transmission, packet reception, call requests/pull down and mobility changes are part of the core operation. Random numbers are used within a networking scenario to generate aspects such as mobility patterns and traffic profiles.

A.3 Network Simulator 3

NS3 [78, 79, 80] is an open source, DES that was created to replace Network Simulator 2 (NS2) [81]. NS3 is not compatible with NS2 and was built from the beginning as a brand new simulator. The simulator and programs created are generally written in C++ but Python bindings are also available. This simulator has been created primarily for use within academic and research communities rather than for commercial use. The GNU GPLv2 license for research, development, and use applies directly to the use of NS3 and any models developed with the tool as its basis. Extensive Doxygen documentation for NS3 is available online [82].

NS3 is continuously under development and is constantly being extended to other areas of communication systems. A wide variety of protocols/models have been implemented including elements of MANET, TCP, IPv4, IPv6, WiFi and more recently LTE and WiMAX technologies [82]. Models, within NS3, have been written to analyse, study and develop specific elements of network protocols. Simulation modelling of communications systems is widespread because it enables a simplified view of complex interactions and mitigates issues associated with the availability and expense of real-world communications systems.

A.4 Using NS3

The source code for NS3 is organised, mainly, within a core directory and is written in C++ and Python. The organisation of the source code is described in Figure A.2; each module can have dependences on other models as shown in the figure. Any model written can use many aspects of the functionality that is already contained within the source directory.

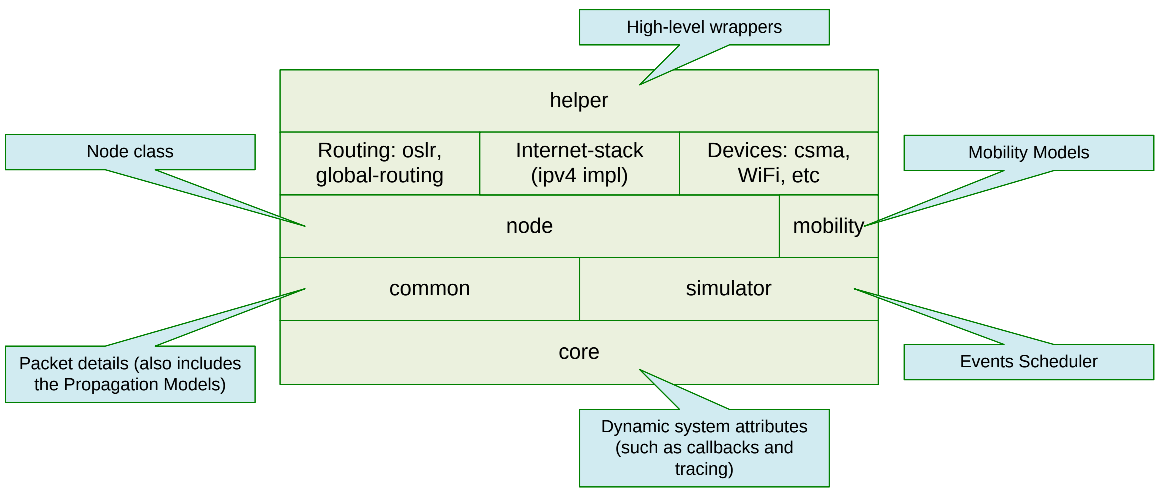

Figure A.2: NS3 code organisation

To use NS3, an understanding of the organisation of the source code must be obtained. The components held within the core of the simulator are essentially dynamic system attributes (callbacks, tracing, logging, etc) that are common to all models within the simulator. Packets and their associated attributes are fundamental in the simulation of any network. These components are used by all networking components. The simulator schedules all events and controls the flow of the network and its current state. These modules are the basis for all simulations completed within NS3 and can be used to represent behaviours of a range of networks.

The modules shown above the common and simulator modules within Figure A.2

(e.g. node, mobility, helper, etc) are specific to the network and device models being simulated. The mobility of a user, device type being used as well as all aspects of network topology communications channels, network interface cards and activity within the network are described by models within the diagram. Helpers are high-level wrappers that are designed to support many aspects of network simulations. By utilising all modules within the NS3 source directory, a communication network can be fully realised.

A.5 System Level Simulations

System level simulations have been used for all the analysis contained within this thesis. System level simulators are simulation programs that focus on network issues such as scheduling, mobility handling or interference management. These are useful for evaluating the performance of networking technologies (*i.e.* LTE) as well as algorithms implemented into such technologies.

Within NS3, a simulation environment has been developed that includes a femtocell, a macrocell and a UE. The femtocell and macrocell individually connect wirelessly to the UE. Simulating the radio links between the UE and the base stations in their entirety is an impractical method for completing system level simulations. Simulating the totality of the connections would require long simulation times and a high level of computational power. To efficiently perform system level simulations, most of the physical layer of the communications system is abstracted away. It is important to adopt a layer of abstraction appropriate to the problem being investigated. By creating a layer of abstraction, development effort can be saved, complexity and simulation time can be reduced and the modeller can clearly see the impact of specific changes being made. Using such an approach, a simplified model of a communications system can be created that captures the characteristics required for the simulation in question. By doing so, the complexity of the system is dramatically reduced while still providing useful insights into the system.

A.5.1 Implementation

The UE being connected to either the femtocell or the macrocell is not sufficient to undertake handover optimisation simulations. Handover must be implemented to permit a transfer of UE connectivity from the femtocell to the macrocell and from the macrocell to the femtocell. Within a basic LTE system, there is a handover process that must be adhered to. When a mobile user moves around an environment, handover is triggered when the RSRP of a base station, other than the serving base station, becomes larger than the serving base station by a Hys value for a duration equal to or greater than the TTT. When this occurs, a measurement report is generated which may then result in handover execution and can be considered as the handover trigger, as explained in Section 2.3.1.

In order to optimise the handover process, the UE requires the ability to move around the propagation environment of the femtocell, within the simulated environment. NS3 has propagation models and mobility models implemented for use within mobile communications simulations.

A.5.2 Radio Propagation Models

Within many simulations that involve wireless connections, a propagation loss model must be selected. Propagation loss models dictate the signal strength degradation of the signal from the point of origin: the base station. The decay of the signal strength from the base station has implications on many aspects of communications systems including range, handover and dropped calls. There are propagation models that have been implemented and tested within NS3 [83]:

- Cost231 Propagation Loss Model
- Fixed Rss Loss Model
- Friis Propagation Loss Model
- Jakes Propagation Loss Model
- Log Distance Propagation Loss Model
- Matrix Propagation Loss Model
- Nakagami Propagation Loss Model
- Random Propagation Delay Model
- Range Propagation Loss Model
- Three Log Distance Propagation Loss Model
- Two Ray Ground Propagation Loss Model

However, the propagation loss models implemented within NS3 are not generally suitable for in-building communications. As a result, a custom propagation model was created for use within the simulations environment used for the work in Chapters 4 to 6.

Single-Slope Propagation Loss Model

The single-slope propagation loss model used was based on an exponential decay and is shown in Figure A.3. The exponential decay is described by $b\exp(\text{dist})^{-1}$ where b is a magnification factor and dist is the distance from the base station.

This propagation loss model was created to approximate simple signal degradation over distance between the femtocell and the mobile terminal. The single-slope

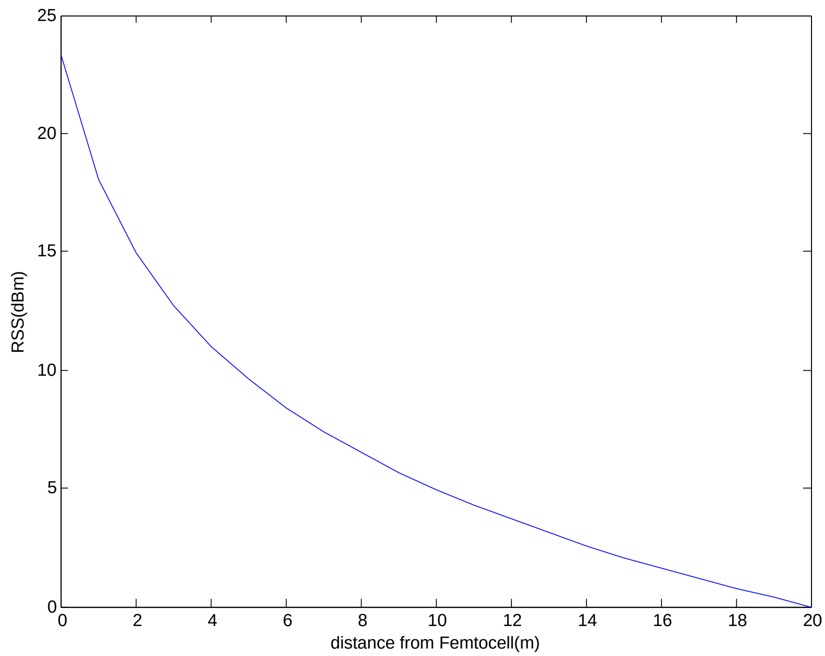

Figure A.3: Single-slope propagation model propagation model alters the RSRP of the femtocell as the mobile terminal moves around the simulation environment, based on its distance from the femtocell. The propagation model was tested to ensure correct operation both independently and as part of the NS3 simulator. The choice of propagation model does not effect the generality of the results and has, only, a secondary effect on the results as a result of the model working in an event-based manner rather than a temporal manner.

A.5.3 Mobility Models

User movement is often appropriate when evaluating the behaviour of system operations in a wireless environment. NS3 is designed to simulate environments that include mobile users. The chosen mobility model defines mobility characteristics of the user such as direction, speed and when to change direction. Some standard mobility models exist within NS3 [84], including:

- Constant Acceleration Mobility Model
- Constant Position Mobility Model
- Constant Velocity Mobility Model
- Gauss Markov Mobility Model
- Hierarchical Mobility Model
- Random Direction Mobility Model
- Random Walk Mobility Model
- Random Waypoint Mobility Model
- Steady State Random Waypoint Mobility Model

- Waypoint Mobility Model

These mobility models control the movement of users in different ways and allow for different patterns in movement. Both the base station and the mobile terminals involve the use of mobility models. Just as with the propagation model, the choice of mobility model does not effect the generality of the outcome. If the algorithms within this thesis work in a temporal manner, then the mobility model will have a greater effect on the result of the algorithm because the event triggers could be slower to occur based on the chosen mobility. The mobility models used within Chapters 4 to 6 will now be expanded upon.

Constant Position Mobility Model

In a situation where a networking element should be stationary, the mobility model still has to be defined within NS3. The constant position mobility model allows for a networking element to be stationary and never move. This mobility model is useful for base stations since they generally retain a constant position and do not frequently move around the environment.

Random Walk Mobility Model

A random walk mobility model emulates the random nature of movements of a mobile terminal and allows for unpredictable motion, as shown in Figure A.4. This mobility model permits the user to change direction randomly after a prescribed period of time or distance travelled. The movement changing metric (distance or time) and the bounds for the speed of the user can both be defined specific to each simulation. If the user hits one of the boundaries to the area being modelled, it will rebound on the boundary at a reflex angle with the same speed as before. The random walk mobility model is a memoryless model, therefore, the current velocity is independent of previous and future velocities and no feedback is involved.

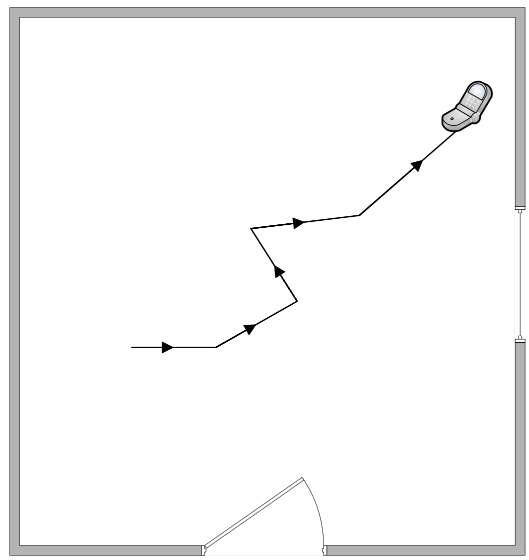

Figure A.4: Movements of a user governed by a random walk mobility model

This model was chosen because users in indoor areas are random in nature. When moving around people rarely move in a straight path or in a predictable manner. Therefore, a random model that frequently changes the users path was required. This model was modified within the context of this work to ensure the user mostly walked past a window and always through a door when these regions were entered.

Random Direction Mobility Model

A random direction mobility model emulates a subtly different type of motion compared to the random walk mobility model. The random direction mobility model allows for a random direction to be travelled whenever the user hits a boundary to the environment being modelled, as depicted in Figure A.5. The motion of the mobile terminal follows a random direction at a constant speed until it hits a boundary, the terminal then pauses for a random period of time, selects a new direction and speed then moves again. The bounds to the speed of the user and the pause time can be defined for each individual simulation.

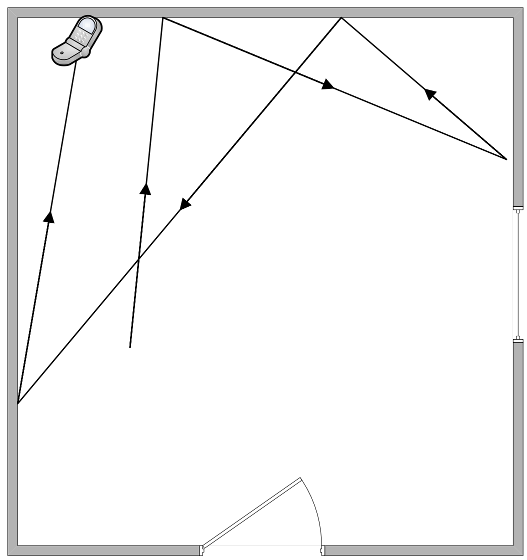

Figure A.5: Movements of a user governed by a random direction mobility model

The random direction mobility model was chosen, again, because it was random in nature. The path of a user could be similar to this model in an area with few obstructions. This model was modified within the context of this work to ensure the user mostly walked past a window and always through a door when these regions were entered.

Simulation Environment

Now that the environment being modelled has all the components required for standard handover management within LTE, some advanced methods can be considered. Neural networks will be implemented into handover management using an autonomic control loop (explained in Section 3.2).

Within this environment, the model allows movement of a user and the location of the user when a handover is required (Monitor stage). The situation can be analysed using a neural networking algorithm (Analyse stage) that can plan what to do about the handover (Plan stage) and execute the decision in a way that adheres to the requirements of handover in LTE (Execute stage). The results can then be recorded and stored in order to evaluate the effectiveness of the algorithm.

The handover management using neural networks is integrated into NS3 and evaluated in comparison to a standard LTE approach. In order to assess the performance of the algorithm, HPIs are required (explained in Section 3.3.4). Appropriate HPIs are ping-pong handover ratio ($\mathrm{HPI}_{\mathrm{pp}}$) and handover failure ratio ($\mathrm{HPI}_{\mathrm{drop}}$).

A.6 Software Testing

When creating any simulation program, it is important to thoroughly test the software to ensure error-free operation; this is not an easy task. Software testing can be defined as the process of executing a program with the aim of finding errors. The term testing is ambiguous and general aims are to ensure that the software is correct and robust. Tests should be created with the aim of testing specific elements of the software to identify when it does not operate effectively. Testing software to ensure that it is fully verified and validated is a very long and extensive task. Software testing is completed to ensure that the results seem reasonable and error-free.

To ensure that a model has been created correctly, both validation and verification have to be completed. The process of ensuring that a particular model is implemented according to its specification is called verification. Validation is ensuring that a model operates according to its intended use. Both validation and verification are required of any model and those models already held within NS3 have previously been both validated and verified. Three categories of tests are used to ensure successful model implementation: build verification tests; unit tests and system tests

A.6.1 NS3

There is a testing framework within NS3 that is used to govern the operations of all the models held within it. Testing is completed out with the main organisation structure of code within NS3 but is completed on all aspects of the NS3 simulator. Due to the popularity of the simulator, all models used have to be extensively tested before being included within the simulator [85, 80]. Automated build robots that perform robustness testing by running the test framework on different systems with a range of configurations. Build robots allow for NS3 to be rebuilt and tested each time something has changed. Users and developers do not generally interact with build robots other than to read any test results that are produced. The build robots use a file called *test.py* to execute tests and examples. This file is responsible for running all tests and collating the data into a report provided to users/developers and can be referred to at any point in time. The user can specify what tests and how many test should be run.

The tests completed can loosely fall within 3 categories: build verification tests; unit tests and system tests. Build verification tests are built along with the distribution and are used to remove most errors with the models. Unit tests test the models in isolation and are more rigorous than build verification tests. The tests are not built into NS3 and examine multiple aspects of the functionality of the model. System tests examine the models operations within NS3 and involve more than one model in the system.

As a result, all models included within deployed versions NS3 are rigorously tested before and during their implementation into the simulator. Researchers can use this simulator with confidence for all aspects of communications network simulations.

A.6.2 Neural Networking Models

NS3 currently has no neural networking models implemented within it. In order to implement advances in handover management, machine learning must be added to the simulator. The models were verified and validated before being added to the simulator.

Initially, the models were created separate from NS3 and tested independently. The models were then put through build verification tests and unit tests. By performing such tests, it was confirmed that the algorithms (from Chapters 4 to 6) worked in a manner that adhered to both expectations and previous examples of such algorithms. A SOM will be used as an example to demonstrate effective operation.

When creating a C++ implementation of a SOM, it was important to ensure correct operation. In order to verify that the algorithm was working as expected, a process similar to that of many textbooks was employed [86] and a uniform distribution of inputs was taken. Consequently, the weights within the network will spread from initially deployed locations that form another estimation of a uniform distribution of the simulation area (shown in Figure A.6) to locations that represent a grid (shown in Figure A.7) covering all locations in the area equally. Ideally the result would involve each weight being equidistant from each other while covering the entire simulation area. In reality, the weights rarely fit this description perfectly. An ideal fit is difficult to achieve because it requires no under or over fitting and a very good uniform distribution of inputs. By using a uniform distribution of inputs that are spread equally throughout the simulation area, an approximation of equidistant

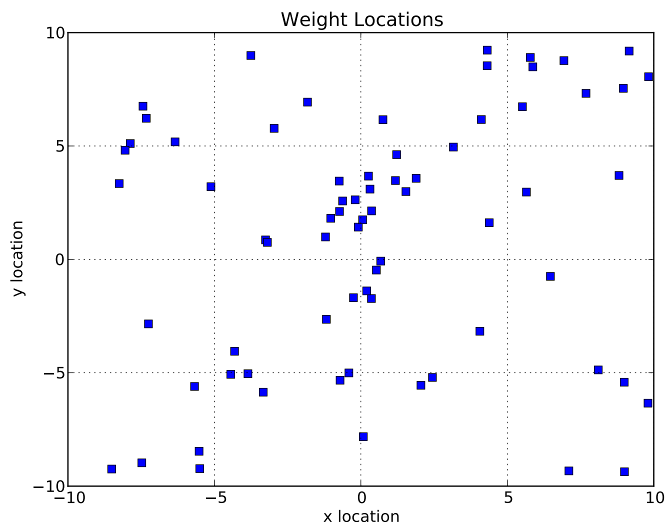

Figure A.6: Initial weight locations within a SOM

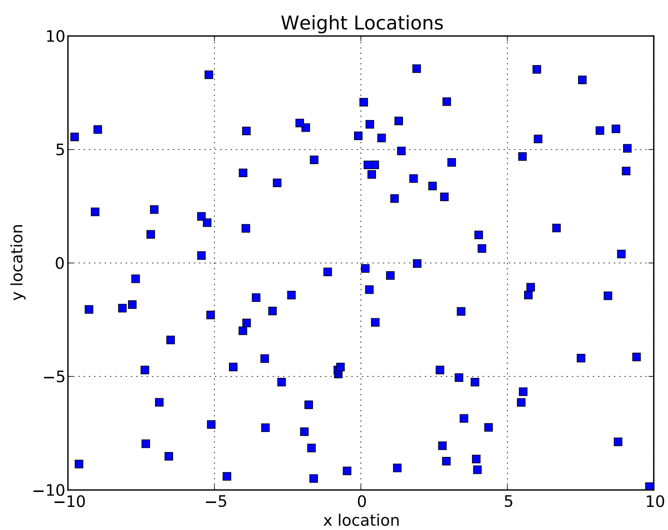

Figure A.7: Weight locations within a SOM after a time period with uniformly distributed inputs

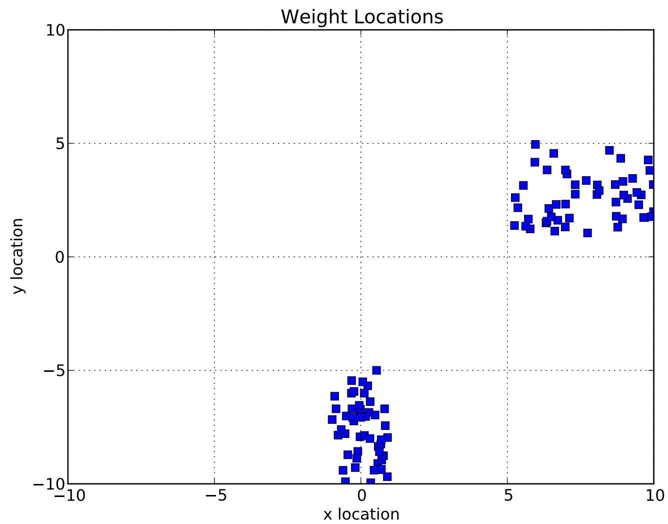

Figure A.8: Weight locations within a SOM after a time period with uniformly distributed inputs within specific regions

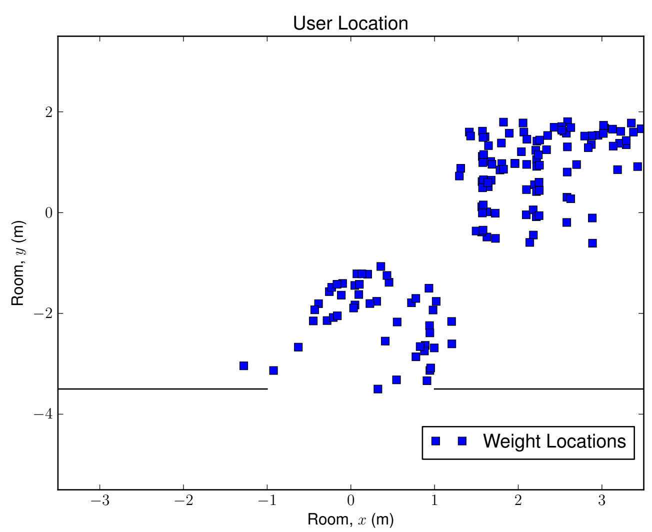

Figure A.9: Weight locations within a SOM after a time period with uniformly distributed inputs in specific regions using NS3 coverage can be obtained. At this point, the algorithm has been verified to work as expected based on related literature.

The algorithm was initially validated outside of NS3. The model was expected to allow the weights, that are initially uniformly distributed, to move to locations that relate to the input data. The moving of the weights refers to the weights being updated at each input and changes to the parameters that relate to position. The intended application requires the weights to migrate to areas within the simulation environment that inputs occur. As can be seen in Figure A.8, when the inputs are not uniformly distributed over the entire area but distributed over the specific areas, the weights migrate to these areas.

At this point in the validating and verification process, the implementation of the SOM is correct and can then be added into NS3. When implemented into NS3, the model has to undergo system tests to ensure its interoperability with the simulator. The model was previously verified as a stand alone module so the system tests are completed to validate its operation within the simulator. Using inputs to the algorithm that occur at specific locations, the weights migrate to regions that the inputs take place (shown in Figure A.9) resulting in a similar output to the result of the stand alone module, shown in Figure A.8. These Figures are similar because they both show weight locations but since they are the result of random simulations they will never be the same.

The validation and verification procedure applied has been completed to ensure correct operation. Once validation and verification is complete, simulations of scenarios can begin. Each of the algorithms, case studies and applications from Chapters 4 to 6 are extensively tested and correct operation confirmed.

A.7 Summary and Conclusion

This appendix has presented the simulation tools that have been used within the Chapters 4 to 6. All simulations have been completed within NS3 which is well known and a common development tool for network based simulations. The evaluation environment includes both models provided within NS3 and models created specifically for the work contained in this thesis. All models created have been tested and debugged. A SOM has been used as an example to demonstrate the testing stages involved with creating an algorithm and implementing it into NS3. The SOM is explained in more detail in Chapter 4 and developed upon within Chapter 5.