An Approach to Using Cognition in Wireless Networks

Lizdabel Morales Tirado

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Electrical Engineering

Dr. Jeffrey H. Reed, Chair
Dr. William H. Tranter
Dr. Luiz A. DaSilva
Dr. Allen B. MacKenzie
Dr. Narendran Ramakrishnan

December 18, 2009
Blacksburg, Virginia

Keywords: cognition, cognitive radio, cognitive networks, cognitive engine, radio resource management, machine learning, case-based reasoning, decision tree learning, coverage management, handover management, policy management

Abstract

An Approach to Using Cognition in Wireless Networks

Lizdabel Morales Tirado

Third Generation (3G) wireless networks have been well studied and optimized with traditional radio resource management techniques, but still there is room for improvement. Cognitive radio technology can bring significant network improvements by providing awareness to the surrounding radio environment, exploiting previous network knowledge and optimizing the use of resources using machine learning and artificial intelligence techniques. Cognitive radio can also co-exist with legacy equipment thus acting as a bridge among heterogeneous communication systems. In this work, an approach for applying cognition in wireless networks is presented. Also, two machine learning techniques are used to create a hybrid cognitive engine. Furthermore, the concept of cognitive radio resource management along with some of the network applications are discussed. To evaluate the proposed approach cognition is applied to three typical wireless network problems: improving coverage, handover management and determining recurring policy events. A cognitive engine, that uses case-based reasoning and a decision tree algorithm is developed. The engine learns the coverage of a cell solely from observations, predicts when a handover is necessary and determines policy patterns, solely from environment observations.

Acknowledgements

This dissertation would not have been possible without the help of many people. Thanks to everyone who has contributed either directly or indirectly in this journey.

I would like to express my gratitude to my advisor, Dr. Jeffrey H. Reed. His guidance, support and flexibility during this journey have been invaluable. To the members of my committee, thank you for your encouragement, support, insight and patience. To Dr. William B. Tranter, thank you for your words of advice and keeping me on track when I lost sight of the goal. To Dr. Luiz DaSilva, thank you for your friendship and for being such a great role model. To Dr. Allen B. MacKenzie, thank you for suggestions for strengthening the wireless communication concepts. To Dr. Naren Ramakrishnan, thank you for your recommendations on other learning techniques, that enhanced the engine's design. I would like to thank Dr. Manuel Pérez-Quiñones for his assistance during my graduate studies.

To my colleagues at Wireless@VT: Dr. Carl Diettrich, Joseph Gaeddert, Carlos Aguayo and An He. Thank you for all the fruitful meetings and discussions that have provided the backbone for this research. To Wireless@VT alumni: Dr. Kyouwoong Kim, Dr. Kyung Bae, Dr. James Neel, Dr. Rekha Menon, Dr. Ryan Thomas, and Dr. Youping Zhao for their collaborations in cognitive radio research. Also, I would like to thank Nancy Goad, Shelby Smith and Cynthia Hopkins for all their support during my stay at Virginia Tech.

I would like to thank the University of Puerto Rico, Mayagüez (UPRM) Campus for their continued support. To Dr. Ramón Vázquez, Dean of Engineering at UPRM for giving me

this opportunity and believing in me. To fellow faculty and to the department staff for their continued support during the length of my studies.

I am extremely grateful to my in-laws Ana María and Juane for their love, support, and time. Their lengthy trips to Blacksburg and countless hours of babysitting contributed to this dissertation becoming a reality. I would like to thank my nieces Myriam Mercader and Maria Ester Mercader, who voluntarily moved to Blacksburg during a summer so that I could continue working on my research. To my brother José and my sister Nancy, thank you for all your kind words and love. To my sisters-in-law Carla and Carmen, and my cousin Doris, thank you for listening and encouraging me.

To my mother, whose love, support and encouragement have been a driving force in me. To her many talks over the phone when my spirits were low, and for the many times she took care of my family while I was working. Thank you, Mom. I am extremely grateful.

I would like to thank my children, they are my inspiration and a source of love and joy. A discouraging day was instantly cured with their hugs and smiles.

Lastly, but not least, I would like to thank my friend and colleague Juan Emilio. His encouragement, support and patience have been essential towards the completion of this dream.

Funding Acknowledgements

This work was partially supported by a John Lee Pratt Graduate Fellowship during my years at Virginia Polytechnic Institute and State University. Also, by a Graduate Fellowship from the Virginia Space Grant Consortium, and by research assistantships by the following sponsors: Electronics and Telecommunications Research Institute (ETRI), Army Research Office (ARO), Tektronix, Inc., Wireless@VT Affiliates, and the Communications Network Services (CNS) office. I also would like to acknowledge the University of Puerto Rico Mayagüez Campus for their continuous support during the length of my studies.

Dedication

In memoriam of José H. Morales Tirado,
my beloved brother.

To Angélica María, Juan Emilio and Juan José,
my inspiration.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

With the rapid proliferation of wireless applications in recent years, there has been a greater demand for the electromagnetic spectrum. Although the “feeling” is that there is no available spectrum, the truth is that most of the spectrum remains under-utilized. Recent reports show that the actual utilization can be as low as 1% in some areas, thus the recent interest in techniques and technologies that allow for sharing of the spectrum [6–10]. This interest has led to research in the efficient use of the electromagnetic spectrum, the focus primarily on dynamic spectrum access (DSA) techniques that allow use of the spectrum more efficiently and more effectively. Currently, spectrum rights are assigned very similarly to real-estate rights. Primary users have “property” rights to their assigned spectrum, thus no sharing is allowed. New spectrum sharing policies and technologies are being researched to use the spectrum more effectively. Agencies such as the Federal Communications Commission (FCC) have had the task of investigating technologies that improve utilization of the spectrum; one such technology is the cognitive radio (CR).

The term cognitive radio was originally coined by Mitola back in 1999 [11], later it was the subject of his dissertation work in 2000 [12]. Since then, it has received significant attention from the research community, regulatory agencies and industry. Cognitive radio is a new technology that promises a paradigm in wireless communications but it is still in the very early stages of development. Cognitive radio is expected to make significant improvements not only in spectrum accessing techniques, but also in quality of service (QoS) optimizations, radio resource management (RRM), emergency communications, communications bridging, femtocell deployment and integration, broadband wireless networking, among others. While Mitola's vision of a "full-blown" cognitive radio will take many years to come, cognitive radios with a simpler level of cognition can still provide ample improvements in current networks [13, 14].

The current research efforts in DSA techniques have inspired us to apply cognition to other network resource problems. In this work, the path towards the development of a cognitive engine is discussed. An approach to apply cognition to various wireless network problems is presented. We discuss the concept of cognitive radio resource management and describe possible applications to current and future wireless networks. We apply the proposed approach and develop a hybrid cognitive engine based on Case-based Reasoning (CBR) and Decision Tree (DT) learning. The generic cognitive engine can be applied to any existing and future wireless network, but in order to validate the engine we have decided to provide a definite context for the engine. We selected 3G wireless networks for various reasons: these networks have been well studied and optimized with traditional radio resource management methods, we are very familiar with them, and there exists a variety of simulation tools that can aid us quantifying the benefits of adding cognition to these networks.

Furthermore, with the addition of femtocells, also known as home base stations, 3G networks will face new challenges that were not considered during the initial standardization process. Femtocells can help operator achieve the goals outlined in 3G LTE[1] specifications such as: improving coverage and capacity, improving link reliability, reducing operator costs, and reducing subscriber turnover [15]. However, these improvements will be possible when optimal solutions to femtocells' technical challenges are found. Femtocells need to adapt

[1] LTE stands for Long Term Evolution, and is the name given to the 3GPP project that is focusing on improving the Universal Mobile Telecommunications System (UMTS) mobile phone standard to cope with future technology evolutions. LTE specifications have been approved by 3GPP. It is expected to be commercially available in 2010.

to their surrounding environment and allocate spectrum in the presence of interference. Femtocells also need to provide timing and synchronization with the network, in order to ensure handover between macro-cell and femtocell environments. Furthermore, femtocells need to provide acceptable QoS and reduce latency in the backhaul. Other issues such as performing handovers, providing service to nearby subscribers, and providing 911 location tracking also need to be considered. Cognitive radio can tackle these technical challenges in femtocells by providing cognitive capabilities such as awareness, intelligence and learning. Moreover, if we consider 4G wireless networks, cognitive radio will be an integral part of this evolution. These networks are dubbed "cognitive networks", thus cognitive capabilities are already presumed in their design. The intelligence and awareness capabilities of CR are necessary to support the amalgam of heterogeneous networks and ensure interoperability among them [16–23].

There are many challenges in the evolution of wireless networks. Cognitive radio can be a great tool addressing some of these challenges. In this work, we propose using cognition to address improving coverage in a 3G network. The design is flexible enough that will allow for other network problems such as: interference mitigation, link reliability improvement, handover prediction to be solved. In this dissertation, we simulate the engine's radio environment and validate the engine design by applying it to real-life problems: learning the coverage of the cell by exploiting the network history, managing handover, and determining policy patterns.

1.1 Applying Cognition to Wireless Networks

It is clear that cognitive radio technology will be of great benefit to current and future networks. Literature published in recent years shows that adding basic CR techniques for spectrum utilization improvement, radio resource management optimization and data mining of user information can yield significant improvements in network performance. However, the end-user benefits have yet to be quantified. The costs versus benefits tradeoffs of cognitive radio techniques remain unknown.

Currently, operators of 3G networks are hesitant to embrace this new technology, as cognitive radio brings a radical change to current radio resource management approaches. On the other hand, efforts to define the evolution of 3G networks are well on their way, the goal is to further reduce operators costs and to improve service provisioning to the end-user [24]. In order to achieve this goal, operators and manufacturers are focusing in four key aspects: *increasing system coverage, increasing system capacity, increasing data rates, and reducing latency.*

Several solutions that address these issues have been proposed [3]. Researchers have suggested new architectures, multi-antenna solutions and evolved QoS and link layer approaches, among others [24]. From these solutions, only a few propose applying cognitive radio [25–31]. In this, work we explore adding cognition to wireless networks, we propose an approach to applying machine learning techniques and formulating the problems. We propose a generic cognitive engine architecture and give detailed descriptions of its components. We also suggest engine architectures for femtocell deployments and Beyond 3G (B3G) wireless networks. We apply the proposed approach and develop a cognitive engine that learns the coverage in a cell and performs handover management using the learned knowledge.

1.2 Problem Statement

Third generation wireless networks have been well studied and optimized with traditional radio resource management techniques, but still there is room for improvement. Cognitive radio technology can bring significant network improvements by providing awareness to the surrounding radio environment, exploiting previous network knowledge and optimizing the use of resources by applying machine learning and artificial intelligence techniques. Cognitive radio can also co-exist with legacy equipment thus acting as a bridge among heterogenous communication systems.

In this, work we address the design and implementation of a cognitive engine for wireless networks. First, we define cognitive radio in the context of 3G networks, we then identify the “knobs and meters” [2] for the wireless network. We implement the engine using a combination of case-based reasoning and decision tree learning. We validate the engine by applying it to real-life scenarios. The engine learns the coverage pattern of a cell purely by deriving rules and creating a decision tree using solely previous cases. The new learned knowledge is used to develop performance improving algorithms for handover management.

During this process we will also address broader cognitive engine development issues such as:

- *How much cognition is necessary?* - Mitola’s vision of cognitive radio is an agent that is autonomous and capable of understanding, planning, negotiating. His vision requires extensive machine learning techniques. Some researchers [13, 14], have shown that lower levels of cognition can bring significant improvements to the network. What level of cognition can provide substantial network improvements without the added complexity? What are the tradeoffs between cognition and complexity?

- *Where does this cognition should reside?* - Some suggests that cognition should be dispersed in all the levels of the stack, while others suggest cognitive implementations in the physical and data link layer only. How can we determine where cognition should reside? Does adding cognition in all layers affect the overall latency in the cognitive radio, and in the network?

- *What Artificial Intelligence (AI) and machine learning techniques are suitable for the engine’s design?* - At this point, ten years after the term “cognitive radio” was coined by Mitola, only a handful of AI and machine learning techniques have been investi-

[2] These terms have been used in the Software Defined Radio (SDR) community as described by Rieser [32].

gated. The amalgam of techniques offered in the machine learning field is very vast. There is a need to investigate these techniques and determine their suitability to cognitive engine design. Furthermore, some techniques may be able to address different network problems better than others. How can the engine employ different machine learning techniques, to solve the network problems while minimizing processing time and keeping complexity to a minimum?

- *Which engine architecture is best?* - Several engine architectures have been suggested, which architecture results in a better engine. The trend in wireless networks is for simpler architectures that reduce latency. Will the addition of cognition affect this trend negatively?

- *What kind of performance measurements should be used?* - Currently, we expect cognitive radio to bring significant improvements to wireless networks. But how can we measure these improvements, what metrics should be used?

- *How much previous information do we need?* - Cognitive radio will use previous information on the user, the network and the radio environment to predict future actions, therefore how much previous information is really needed. Latency in the cognitive engine increases as the amount of data to analyze also increases. How can we limit the amount of previous information used and still produced a good prediction model?

- *What are the tradeoffs between cognition and latency?* - Latency has a major influence on the user's experience. Conversational applications such as voice calls, video conferencing and VoIP require very low latency. The trend in wireless networks has been to further reduce latency. In GSM/EDGE networks the roundtrip time (RTT) is around 150 ms, in WCDMA/HSDPA the RTT has been reduced to 80 ms, the expected RTT for the LTE standard is around 5 ms [33]; with these strict latency requirements there is a need for cognitive algorithms that do not affect latency adversely.

- *What is the impact of policy on the cognitive engine's design?* - In order to implement

> spectrum maximization solutions and optimize the use of radio resource management major changes in regulatory policies must take place. As with any autonomous device, there are concerns regarding a cognitive radio's behavior. A CR may behave selfishly, resulting in interference for non-cognitive or legacy users, and other cognitive devices. To address this issue, regulatory policies must be developed to assure fair use of all the network's resources.

We attempt to address some of these issues in this document, and hope that this work will further advance the adoption of cognitive radio technology in current and future wireless networks.

1.3 Methodology

In this work, we discuss the path towards the development of a cognitive radio engine for radio resource management in 3G wireless networks. The main approach used in this work is system design by analysis, modeling and simulation. Also, this work builds upon experience acquired while working on three Wireless@VT projects: the CR test-bed with Tektronix, Artificial Intelligence Techniques survey with ARO and IEEE 802.22 WRAN research for ETRI. We explore adding cognition to wireless networks, we propose an approach to applying machine learning techniques and formulating the problems. We propose a generic cognitive engine architecture and give detailed descriptions of its components. We also suggest engine architectures for femtocell deployments and B3G wireless networks. We apply the proposed approach to three case studies. First, a hybrid cognitive engine is designed to learn the coverage of a cell. Then, the trained hybrid engine is used to manage handover in a wireless network. We evaluate the performance of the engine in terms of the error rate, Receiver Operating Characteristics (ROC) curves and computational complexity. As a third case study, we use the hybrid engine to determine policy event patterns.

1.4 Contributions

To date, the original contributions of this research include:

1. Describing a methodology for adding cognition in wireless networks. The emphasis of the approach is the application of cognition to network radio resource management tasks.

2. Proposing a hybrid cognitive engine that uses case-based reasoning and decision tree learning.

3. Providing an analytical framework that relates case-based reasoning and decision tree learning to the engine's learning objectives.

4. Design, development, and implementation via simulation of the proposed hybrid cognitive engine.

5. Developing three cognitive radio resource management algorithms: coverage learning, handover management and policy determination.

1.5 Outline

This document consists of nine chapters. The second chapter discusses the related work that has laid the foundation for the research presented in this document. It provides an overview of the cognitive radio and cognitive engine concepts. Also, provides a brief history on previous cognitive engine implementations. Chapter 3 discusses the proposed approach for applying machine learning techniques in the design of a cognitive engine. Chapter 4 describes the generic cognitive engine's architecture and its components. Chapter 5 discusses the concept of *cognitive* RRM and contrast this new concept with the *traditional* RRM approaches. In Chapter 6, the first case study: using cognition to improve the coverage of the cell is presented. Chapter 7, discusses the second case study: using cognition in handover management. Chapter 8 explores the concept of using cognition to determine policy event patterns. Chapter 9 provides concluding remarks, presents the future direction of this work and lists expected publications.

1.6 Notation

The notation used in this dissertation is included in Table 1.1.

Table 1.1: Notation

Symbol	Definition
p	Problem case.
c	Matched case.
a	Attribute from the set A.
A	Set of attributes.
x	Position on the x axis.
y	Position on the y axis.
θ	Angle of the mobile w.r.t. the base station.
r	Distance from the base station.
υ	Velocity in kmph.
$SINR$	Signal to interference and noise ratio.
PL	Path loss.
fc	Carrier frequency.
Δh_b	Antenna height measured above the rooftop level.
RSS	Received signal strength.
S	Set of objects.
C	Set of classes.
T	Set of tests.
O	Set of outcomes.
ρ	Positive example.
c	Class of positive examples.
η	Negative example.
$\hat{c}$	Class of negative examples.
m	Leaf nodes of the tree.
M	Set of all possible network conditions.
e	Event.
$E(S)$	Entropy of S.
$G(S,a)$	Information gain given attribute a.
ϵ	Error rate.
β	Probability of success.
γ	Probability of error.
ζ	Confidence.
δ	Confidence limit.
TP	Number of true positives.
TN	Number of true negatives.
FP	Number of false positives.
FN	Number of false negatives.

Chapter 2

Related Work

In this chapter, the related work that precedes this research is presented. The work described here serves as inspiration and guidance in this research. Cognitive radio is a multi-disciplinary field, advancements in many areas of research are needed in order to achieve a successful implementation. In this work, we focus on analyzing the impact of cognitive radio from a system-level perspective. We explore cognitive radio and its impact on radio resource management, focusing on how awareness of previous history and data mining of user information can provide performance improvements in some real life scenarios.

First, the concept of cognitive radio is defined. Also, its advantages and limitations are presented. We proceed by describing where cognitive radio will bring the most benefit. Then, we discuss the concept of a cognitive engine and give a brief history of the various cognitive engine (CE) implementations that have laid the foundation for this work. We continue with a discussion of 3G wireless networks, as it is in this context that we examine the application of cognitive radio. We list some of the potential applications for 3G cognitive networks, and also identify the *knobs* and *meters*.The chapter concludes with a brief summary.

2.1 Cognitive Radio

In this section, the concept of cognitive radio is discussed. Also, we provide a comparison between the different levels of cognitive radios and discuss when these radios will become available. We conclude the section by identifying the primary drivers for this technology and the motivation for this research.

A cognitive radio, as defined by Mitola [11], is a software radio that is aware of its environment and its capabilities, it can alter its physical layer behavior, and is capable of following complex adaptation strategies. Adding to this definition, a cognitive radio learns from previous experience and can deal with situations that were not planned at the radio's initial design time.

Another commonly used definition from S. Haykin [34] is: "An intelligent wireless communication system that is aware of its surrounding environment (i.e. outside world), and uses methodology of understanding-by-building to learn from the environment and adapt its internal states to corresponding changes in certain operating parameters (e.g. transmit power, carrier frequency, and modulation strategy) in real-time, with two primary objectives in mind:

- Highly reliable communications whenever and wherever needed;
- Efficient utilization of the radio spectrum."

The above definitions refer to what we know as "full cognitive radio", a device that is autonomous and makes its own decisions based on the radio environment observations, and can evolve from its original design as it learns from new experiences. Other agencies such as the FCC, Office of Communications, UK (OfCom), National Telecommunications and Information Administration (NTIA), Institute of Electrical and Electronics Engineers (IEEE) and the SDR forum have their own definitions of cognitive radio. These definitions may vary from a simple adaptive radio, to radios that include intelligent signal processing and use game

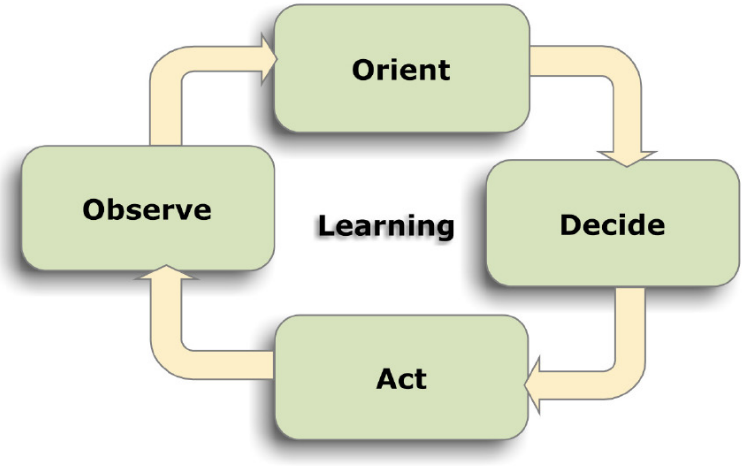


Figure 2.1: Cognition cycle

theoretic etiquettes to address the spectrum sharing problem. While various dissertations can be written solely on defining the term “cognitive radio”, in this work we define it as a software defined radio that is capable of :

- *sensing* its environment and drawing conclusions from the sensed information,
- *adapting* its physical layer parameters,
- *optimizing* over the use of the radio resources, and
- *performing* basic reasoning and learning by using artificial intelligence techniques.

Now, if we describe how a CR works, we can refer to figure 2.1 for a basic cycle adapted from the one described in [35]. The CR first senses the radio environment and draws conclusions from the sensed information. The CR proceeds to analyze the current state and the implications of the newly sensed information. If the new radio environment affects the CR’s current state it will proceed by reacting to new radio environment state by adapting its current configuration. While in the adaptation phase the CR will optimize the use of radio resources given three distinct inputs: the CR user requirements, the radio environment and the current spectrum policy. In order to achieve the adaptation the CR will reason using AI techniques and past experience in order to obtain an optimal solution. The CR reacts to the environment with this new adaptation, the CR receives feedback from the environment, the cycle begins again. In this cycle, the CR learns from previous experiences and applies what has learned to new challenges.

In his dissertation, Joseph Mitola also describes the levels of cognition that characterize various radio tasks [12]. These levels range from no cognition as in a "pre-programmed" software defined radio, to a cognitive radio that can autonomously propose, negotiate and adapt new protocols. While the latter is not attainable at the moment, researchers have found that radios with cognition levels limited to awareness of the environment and planning can yield significant network improvements [13, 14]. The higher levels of cognition require extensive machine learning not supported by current technology. In this work, when we use the term CR we refer to a radio with a cognition level that includes awareness, reasoning and basic machine learning.

Cognitive radio constructed with higher levels of cognition may seem ideal, but perhaps a lower level of cognition can adequately support the desired radio tasks. Higher levels of cognition require intensive AI techniques and machine learning, this implies larger memory requirements, greater processing power and thus, greater power consumption. If we consider that one of the major limiting factors for wireless equipment is battery life, a higher level of cognition inversely affects the life of the wireless device. Clearly, there is a trade-off between the level of cognition, the complexity and useful life of the cognitive radio.

Anil Shukla et al. in their "Cognitive Radio Technology" report for OfCom [1], compare four different levels of cognitive radios and the intelligence required for each one as seen in figure 2.2. At the top right corner lies a "full cognitive radio" or Mitola radio, if we consider the current technology trends we can expect these type of radios to become available by 2030. On the lower left corner of the diagram lie simple adaptive radios that are used today in WiFi, WLAN, PBR, MBITR and Bluetooth enabled networks. A CR that is achievable in

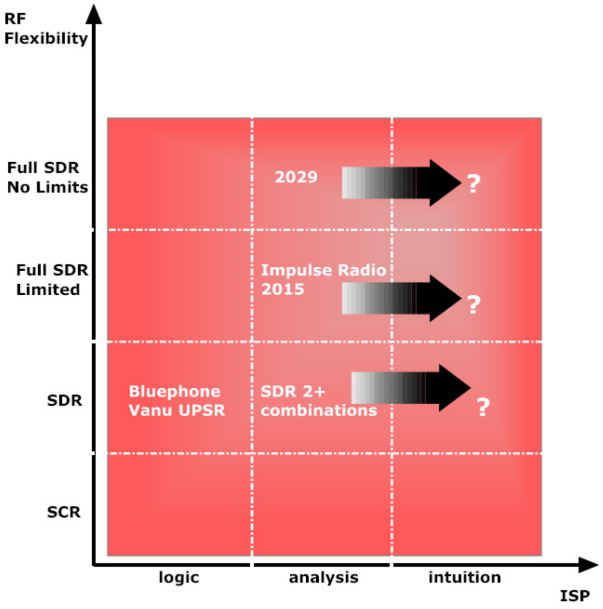


Figure 2.2: Matrix of CR technology available today, reprinted with permission from QinetiQ, Ltd. [1].

the next 5 to 10 years is one that can use intelligence to adapt its physical layer parameters using software radio techniques as mentioned in [1], and adding to this statement the CR should also optimize these parameters to improve the use of the radio resources.

If we can expect a basic CR to be available in the next 5 to 10 years, the next question that we need to address is, *what are the key benefits that cognitive radio provides?* First and probably the most significant of the benefits is spectrum flexibility. The CR's ability to be aware of the spectrum allows for its opportunistic use, hence improving spectrum utilization. This spectrum flexibility also allows for different approaches to spectrum sharing, thus creating new markets that were previously not feasible due to spectrum costs or availability [1,11,36].

Shukla et al., list optimal diversity as the key benefit of cognitive radio [1]. CR offers diversity not only in frequency but also in power, modulation, coding, space, time, and so on, but this flexibility is partly inherited from SDR. With this diversity CR will improve the availability and reliability of wireless services, thus increasing QoS from the user's perspective. Another benefit of CR is that it is a "future-proof" technology. CR's ability to adapt to new regulations, services and protocols that were not thought-of at deployment can save operators when upgrading systems or adding new services. Also, CR can minimize the burden to regulatory agencies in their regulation efforts. If the regulations and policies are included in a database they can be changed or updated quickly, CRs can then update their own local databases (i.e. radio environment map) with the new regulations.

Cognitive radio will provide many benefits to the end user, to the operator, to the manufacturer and to the regulator. Along with these benefits there are issues that cognitive radio raises. Probably the most significant issue is interference mainly due to the hidden node problem [1, 6, 7]. The hidden node problem arises when a CR is unable to detect all of the primary users in its sensing range. This situation occurs when the primary user is hidden from the CR's sensing range. As an example, let's discuss a possible scenario: there is a far-away non-cognitive transmitter, that is not in the sensing range of the CR. This non-cognitive transmitter is communicating with a non-cognitive receiver that is in proximity to the CR. The non-cognitive receiver may be at the edge of the cell. The CR will scan the spectrum for transmissions, however since the transmitter is very far from the CR, it's signal may be lower than the noise floor, and the CR may be unable to detect it. The CR may cause harmful interference to the transmitting primary user and affect its communications, without realizing it as it does not have any knowledge on the primary user. Several suggestions on how to solve this problem have been published in the literature; they include cooperative sensing, and adding a beacon signal to primary users or other non-cognitive devices operating in the same spectrum as the CRs. Other key issues for cognitive radio deployment include security and reliability. As with any autonomous device that can alter its initial programming, there are lots of concerns on how CR will behave after it has been deployed. Alternative etiquettes and protocols can be established in order to encourage cooperation and sharing of the resources among CRs in a network.

In summary, cognitive radio is a new technology that promises a new paradigm in wireless communications but it is still in the very early stages of development. Research and advancements in many areas are needed in order to have a successful CR implementation. The objective of this work is to further advance the cognitive radio field by researching cognitive engine design and applying the design to improving coverage in 3G wireless networks.

2.2 Cognitive Engine: Intelligence for the Radio

In the previous section we discussed the concept of cognitive radio, before we continue further into our discussion, it is necessary to describe what characterizes a radio as "cognitive". A radio has no cognition abilities; it is the cognitive engine (CE) that brings intelligence, awareness, reasoning and learning to the radio. Without the CE the radio is merely a programmable software radio at its best. Thus, we define a CE as an "intelligent" agent that manages the cognition tasks in a cognitive radio.

This agent can be thought of as an independent entity that oversees the cognitive operations of the radio. Given inputs from its user, regulatory agencies or the radio environment, the cognitive engine analyzes the situation, performs calculations and proceeds by responding or reacting to the stimulus. As an example, this response can be adapting parameters of the radio such as modulation scheme, transmission frequency, reception frequency, etc., given the user requirements and the current environment conditions.

Now, that we have defined what makes a radio cognitive we have to think about five inherent questions: *What is cognition?, How much cognition is necessary?, Where should this cognition reside?, What is the benefit of adding cognition?, What is the cost of adding cognition?*

First, what does cognition mean in the context of a radio. Cognition is defined as the mental process of knowing, including aspects such as awareness, perception, reasoning, and judgment, its Latin root is *cognoscere* which means *to learn.* When we characterize a radio as cognitive we imply awareness of its capabilities and of the environment. This awareness is achieved by means of sensing its surroundings, the use of radio environment maps (i.e.

REM), the use of policy databases, and by performing the needed calculations to interpret the data into information that can trigger actions in the radio.

In the context of machine learning, we imply the use of previous knowledge to create new knowledge when we refer to "learning". Learning can be achieved using AI and machine learning techniques that allow the radio to create knew knowledge from previous experiences. Learning also implies memory, and for a radio this means storing previous information or events that have triggered actions. For example, if we apply it to the coverage problem, the CR will deduct that a handover is necessary, if its moving to the cell boundary, and in previous instances the previous actions performed in order to maintain the call were to perform a handover.

Furthermore, there has been a lot of debate on the amount and location of *cognition.* A full Mitola CR requires extensive cognitive processing, in all the Open Systems Interconnection (OSI) layers of the stack, while some of the CRs proposed and tested by industry and academia require limited cognition. Generally, this cognition resides in the first two layers of the stack (Physical, Data Link). Recently, researchers at Virginia Tech proposed and defined the term cognitive networks extending *cognition* to the Network layer [37, 38]. In order to achieve a full CR implementation as proposed by Mitola and Haykin [11, 12, 34], *cognition* must reside in all layers of the OSI stack.

In previous sections, the benefits and costs of cognitive radio have been discussed. It is worth discussing the benefits and costs of adding *cognition.* By applying *cognition* to the radio environment, we will be able to solve problems that we were not capable of solving using traditional methods. We will use awareness and previous experience to predict future problems and solutions to these problems. Although, these capabilities do not come without a price. Adding cognition implies adding complexity to wireless networks. Wireless networks by nature are very complex and difficult to analyze due to their dynamic nature. Thus, we must consider the tradeoff between complexity and cognition.

2.3 Early Cognitive Engine Designs

In the next paragraphs we discuss the evolution of cognitive engine design. Also we describe various CE implementations and the level of cognition included in each design. Of particular interest are Wireless@VT's cognitive engine implementations as these implementations precede the work outlined in this document. The author built on her experience in these projects to create the proposed solution outlined in this document.

The CE as originally conceived by Mitola in [11], was modeled as a context and location based cognition cycle at the application layer. His research described the potential use of cognitive radio technology to enable spectrum rental applications, and to create secondary wireless access markets [12]. Since Mitola's dissertation significant research has been conducted in this area, not only by the military but also by commercial and educational institutions.

In 2002, the Defense Advanced Research Projects Agency (DARPA) recognized that the electromagnetic spectrum was highly under-utilized and funded the neXt Generation (XG) program [39,40]. DARPA's XG project objective was to create an adaptive radio that sensed and shared the use of the spectrum. DARPA termed this unused spectrum as spectrum holes. They called their approach opportunistic spectrum access. In their vision [40], the XG radio is implemented as a layer 2 process. The transceiver API may be enhanced to include certain XG components to provide an XG enhanced Transceiver API. There is no change in the network layer and above, and the legacy MAC layer may not be aware of the XG enhancements. The idea behind the XG process is to coordinate with each other to implement a dynamic spectrum sharing procedure among them, in a way that is designed to control interference to existing primary users. The XG process then creates a "state" in the physical layer for handling the packets consistent with the decisions made in the XG process. The XG radio did not posses cognitive learning abilities and adaptable operation capabilities like the ones proposed by Mitola.

In 2004, Virginia Tech's Center for Wireless Telecommunications (CWT) group realized the implementation of a biologically inspired cognitive engine based on genetic algorithms. The engine is capable of learning and intelligently evolving a radio's PHY and MAC layers when faced with unanticipated wireless and network situations [32]. In their approach CWT researchers created a "stand-alone" cognitive engine that could be used with legacy radios. The system focused on providing cognitive radio capabilities to the physical (PHY) and medium access control (MAC) data link layers. It is structured such that the CE is scalable with the capabilities of the host radio. The more flexible the radio is, the more powerful the cognitive control [41].

Also in 2004, Berkeley Wireless Research Center and the Technical University in Berlin proposed their cognitive approach for the usage of virtual unlicensed spectrum, CORVUS. CORVUS used techniques to avoid interference to primary users (PU) [42, 43]. Their design covered the physical and link layers of the ISO/OSI stack. They identified six main functions for the engine such as: spectrum sensing, channel estimation, data transmission, MAC, link management and group management. Their design also relied on two control channels that implemented the core functionality of the cognitive radio; the universal control channel (UCC) and the group control channel (GCC).

In 2005, Mobile Portable Radio Group (MPRG) and Tektronix collaborated in the development of Cognitive Radio Tektronix System (CoRTekS) [2]. The idea was to create a test-bed to validate cognitive radio as a proven technology for efficient spectrum utilization using Tektronix off-the-shelf components. CoRTekS was built on previous Virginia Tech and Tektronix SDR efforts and utilized Virginia Tech's open source Software Communications Architecture (SCA) implementation (OSSIE) and test equipment software components [44]. Initially, the focus of the cognition abilities of the radio comprised the identification of particular frequency bands suitable for transmission. The cognitive radio made adjustments to the center frequency, modulation and transmitted power according to the spectrum policy delineated by the user and the performance goal (i.e. QoS). The cognitive engine in this CR was implemented using artificial neural networks.

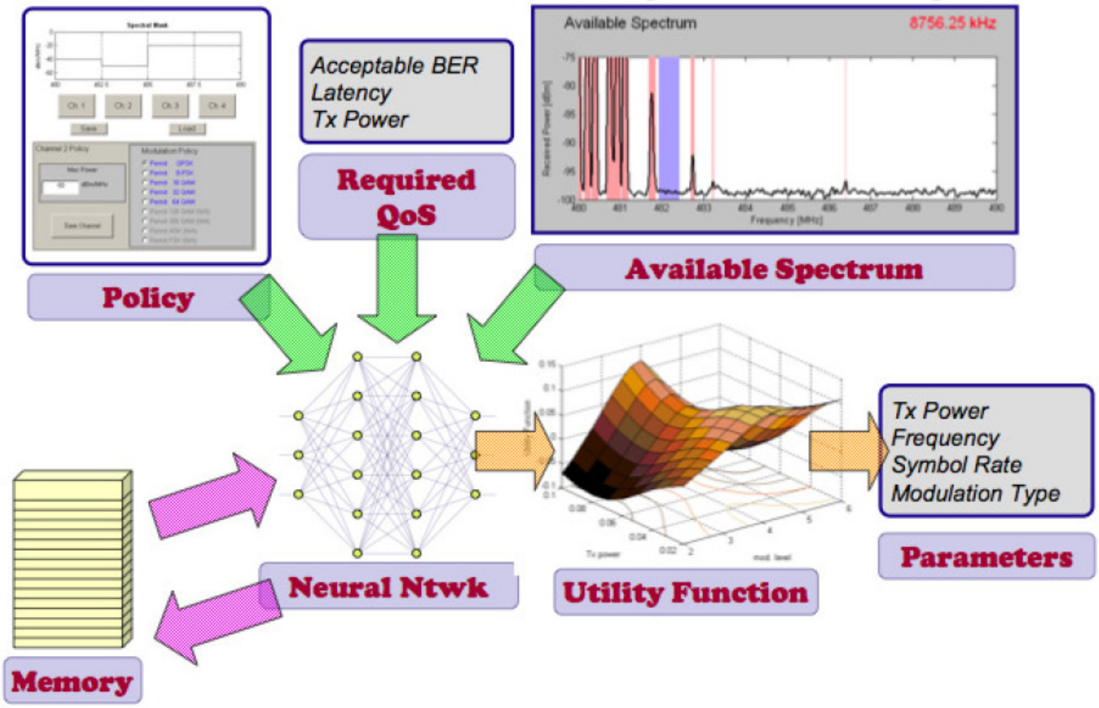


Figure 2.3: CoRTekS Block Diagram [2]

A block diagram of the test-bed is shown in Figure 2.3. The Arbitrary Waveform Generator (AWG430) was used to create a multi-mode transmitter capable of generating a variety of modulated signals. In scenarios where the operating center frequency was beyond the range of the signal generator, the operating frequency was set by a second Arbitrary Waveform Generator (AWG710B). The radio frequency (RF) interface contained several RF components such as filters and antennas in the operating frequency range. The Real Time Spectrum Analyzer (RSA3408A) performed the signal demodulation. The cognitive engine adapted the waveform parameters according to the results obtained after demodulating the signal, in order to achieve a desired performance goal. This test-bed was successfully demonstrated at the SDR Forum Conference in Orange County, California in November of 2005. More information on this implementation can be found in [2].

From 2005 until 2007, MPRG also collaborated with ARO and ETRI in projects that researched two important aspects of cognitive engine design:

- Artificial Intelligence Techniques for Cognitive Engine Design, and
- Cognitive Engine Algorithms for IEEE 802.22 Wireless Regional Area Networks (WRAN) Networks.

The first project provided the necessary background on artificial intelligence and machine learning theory to determine the techniques that were suitable for the various components of the engine. In the second one, the identified AI techniques were applied in a CE design and their performance evaluated. The initial cognitive engine consisted of a knowledge-based reasoner and a genetic algorithm multi-objective optimizer, Figure 2.4 depicts the engine's basic diagram. It was designed using a modular approach, thus the reasoner and optimizer modules could be used at any time during the algorithm adaptation procedure. One of the main advantages of this approach was that adaptation time decreased when the CE used knowledge to limit the search space of the GA optimizer. The results and conclusions drawn from these experiments are included in references [45–47]. The author's experience in these

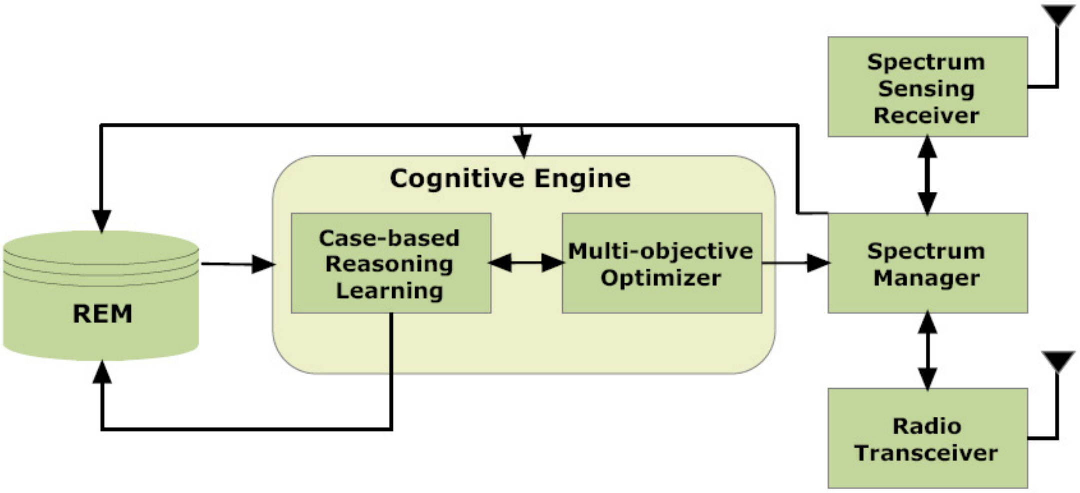

Spectrum Sensing Receiver
Cognitive Engine
Case-based Reasoning Learning
Multi-objective Optimizer
Spectrum Manager
REM
Radio Transceiver

Figure 2.4: MPRG's IEEE 802.22 Cognitive Engine

projects helped her gained insight and understanding in cognitive engine design issues, and encouraged her to continue research in this area.

In 2006, Erich Stuntebeck et al. proposed a generic architecture for an open source cognitive radio [48]. Their framework dubbed OSCR for open-source cognitive radio, facilitated the integration of a cognitive engine with one or more SCA based radios using Virginia Tech's SCA's implementation Open Source SCA Implementation-Embedded (OSSIE). The cognitive engine was implemented using the SOAR concept [49]. SOAR is a cognitive architecture based on the OPS5 production system. The authors followed with an implementation of the CE and its application in two cases: maximizing capacity in an Additive White Gaussian Noise (AWGN) channel and in a Non-AWGN channel. The authors found that the CE's adaptation time was in the order of tens of milliseconds for the AWGN channel case, and in the order of hundreds of milliseconds for the Non-AWGN channel case [50].

Also in 2006, collaborative efforts between Rutgers University, the University of Kansas and Carnegie Mellon University started a project to research architectural tradeoffs and protocol design approaches for cognitive networks at both local network and the global internetwork levels. The initiative is investigating architectural issues including naming, addressing, and routing, collaborative control and management protocols, and experimental system evaluation using measurement and management overlays and cognitive wireless implementations. This collaborative project focuses on the *cognitive stack* and it has two major thrusts: the first is to identify broad architecture and protocol design approaches for cognitive radio networks at both local network and the global internetwork level. The second thrust is to apply these architectural and protocol design results to prototype an open-source cognitive radio protocol (the CogNet stack) and use it for experimental evaluations on emerging cognitive radio platforms, more information on this effort is included in [51].

In the last year, several cognitive engine designs have been published in the literature. In 2007, Timothy Newman et. al, presented a genetic algorithm (GA) driven, cognitive radio decision engine that determines the optimal radio transmission parameters for single and multi-carrier systems. In their research, the authors also illustrated the trade-off between the convergence time of the GA and the size of the GA search space [52]. In the same year, Baldo and Zorzi propose the use of Fuzzy Decision theory for cognitive network access decision making [53]. In 2008, Baldo and Zorzi proposed a learning and adaptation scheme for IEEE 802.11 radios using neural networks [35].

It is evident that many research efforts are being devoted to designing cognitive engines for various cognitive radio applications. As seen in this brief history of cognitive engine design, the successful development of cognitive engines is a crucial aspect for the implementation of cognitive radio. More research efforts are needed in this area in order to develop robust and efficient engines capable of adequately managing the use of the electromagnetic spectrum and optimizing on the radio resources. To this date, only a handful of AI techniques have been proposed in the design of cognitive engines.

2.4 Cognitive Radio Applications

In this section, we discuss three broad applications for cognitive radio. We begin by exploring DSA techniques and provide a brief history on the subject. We proceed by exploring radio resource management and how cognition can bring improvements in this area. Next, we discuss how CR's knowledge of current and past information and awareness of the radio environment can help predict and adapt to future situations.

It is clear that cognitive radio technology will be of great benefit to current and future networks [54]. Literature published in recent years shows that adding basic CR techniques for spectrum utilization improvement, RRM optimization and data mining of user information can yield significant improvements in network performance. However, the end-user benefits have yet to be quantified. The costs versus benefits tradeoffs of cognitive radio techniques remain unknown.

In recent years, there have been numerous reports about the alarming scarcity of the electromagnetic spectrum. Research performed by various entities such as the FCC in the US, OfCom in the UK, and others, indicate that this assumption is far from reality; there is available spectrum since most of the spectrum allocated sits under-utilized [6–10, 36]. The current policies for allocating spectrum do not allow for sharing. Regulatory agencies grant licenses that offer exclusive access to the spectrum. When the licensees are not transmitting the spectrum remains idle, in other words under-utilized [36]. The under-utilization of the spectrum has encouraged researchers in engineering, economics and regulatory agencies to develop better spectrum management policies and techniques. Many ideas have been proposed and several models have been developed [55, 56]. Zhao and Sadler give a detailed description of these models in their recently published paper [55]. Research to validate the proposed models and determine the best solution to dynamically accessing the spectrum is still needed.

Although spectrum management is probably the most popular application for cognitive radio;

applying cognition to radio resource management methods can bring significant network performance improvements. Consider the particular case of 3G networks, these networks have been greatly optimized in terms of their radio resource management. The methods and techniques used are quite advanced; however these techniques are based on very network-specific software, or on proprietary algorithms developed for the telecommunication standard [57]. Generally, these techniques optimize over the Physical Layer (PHY) and Media Access Control (MAC) layers, and neglect the optimization of resources in the layers above. Thus, this current approach is not suitable for future heterogeneous networks, where a single base station will need to optimize the use of radio resources over a variety of networks.

RRM techniques in 3G networks can be divided into five groups: power control, handover control, admission control, load control and packet scheduling functionalities [58]. Power control is achieved by implementing open loop and close loop power control algorithms in the downlink and the uplink. It is necessary to perform power control since the capacity of a cell in Wideband-Code Division Multiple Access (WCDMA) is limited by interference. Also, power control is performed to avoid the near-far problem. Power control is the only RRM technique located in the UE, Node B and the radio network controller (RNC). The rest of the techniques are typically located in the RNC. Handover control is necessary in order to handle the mobility of UEs across cell boundaries and to sustain the desired QoS. Handovers can be intra-frequency, inter-frequency and intra-system (between WCDMA and Global System for Mobile Communications (GSM)). The rest of the techniques are used to guarantee QoS and to maximize the total throughput of the system. Usually, the system will select the combination bit rate, services and quality that will maximize throughput for the user and for the network.

Lastly, exploiting user and the network past information can aid in the decision making as we can use past experiences to predict or select future actions. Decision makers are predictors, they predict the best outcome given the experience and the past actions. Learning is defined as the act of acquiring new knowledge, in CR we exploit past experiences to create new knowledge, thus “learning”. In this work, we will focus on using past experience in terms of cases to predict the best outcome for our defined problem. As an example, using past cases that describe the coverage of the cell, the algorithm will induce rules and predict at a given location the probability that the call will be dropped. Chapter 6 describes the implementation in greater details.

2.4.1 Cognitive Radio Applications for Wireless Networks

Cognitive radio can be a great tool for existing 3G wireless network, it can certainly help operators and manufacturers achieve their evolution goals described above. What is uncertain at this point is the cost of applying CR to 3G networks, the adoption of CR as a secondary user and coexistence with the current (legacy) network architectures is probably the least disruptive approach. CR can improve system coverage by allowing communications between heterogeneous networks, also at the user level CR can exploit past information and current spectrum measurements to prevent coverage problems. CR can increase system capacity by managing the spectrum more efficiently and using non-contiguous bands of frequency. CR can also increase end-user data rates by providing broader frequency bands, either by underlaying techniques in the current spectrum or by using non-contiguous bands.

In summary, cognitive radio technology will enhance 3G wireless networks by:

- providing flexible spectrum management techniques,
- acting as a bridge between existing radio access technologies,
- enabling better quality of service to end-users,
- improving radio resource management methods with reasoning and learning,
- providing rapid service deployments, and
- optimizing networks using sensed data.

As mentioned above, spectrum management is probably the most popular application for cognitive radio; however applying cognition to radio resource management methods can bring significant network performance improvements. Also, this application of cognitive radio does not depend on approval of regulatory agencies, therefore it is a more attractive commercial application. Some of the possible RRM applications for our proposed cognitive engine include (but are not limited to):

- *Scheduling*: the CE can optimize packet scheduling based on 3G wireless networks service classes.
- *Managing handover*: the CE can anticipate when a handover is needed using past experienced.
- *Fault detection and prevention*: the CE can exploit previous data in the network's fault log and use them to tune the network.
- *Mode/service selection*: the CE changes from radio access network (i.e. GPRS, GSM, WCDMA) depending on the application requirements and the available connection.
- *Power amplifier optimization*: the CE reduces peak-to-average radio through power control, scheduling and handover.
- *Interference management in femtocells*: the CE can improve co-existence with the cellular infrastructure by providing scheduling and channel allocations.
- *Planning tool for layout*: the CE determines ways to incorporate past experience into cellular layout and tuning tools.

In the next section, the motivation for applying cognition to 3G wireless networks is presented.

Table 2.1: 3G Technologies
[59]

Systems	Service	Comments
CDMA2000 WCDMA ULTRA-TDD TD-SCDMA	Packet-data and voice services designed for high speed multimedia data and voice	• Defined by IMT-2000. • Europe uses UMTS/WCDMA and ULTRA-TDD. • America use UMTS/CDMA2000 and ULTRA TDD. • Asia use UMTS/CDMA2000, TD-SCDMA and ULTRA TDD. • Overlay approach for existing operators of 2/2.5G networks.

2.4.2 3G Wireless Networks

Third generation (3G) wireless networks are defined by the International Telecommunications Union in their specification International Mobile Telecommunications 2000 (IMT-2000) [3, 59]. The IMT-2000 specification comprises several radio and network access technologies that meet the overall goals of the specification such as: unifying technologies, providing high-speed data for the wireless markets, providing high speech quality similar to fixed networks, worldwide common frequency band and roaming capabilities, enabling multimedia applications services and terminals, and improving spectrum efficiency among others [59].

Table 2.1 presents a brief summary of the systems included in the 3G specification, the general services and the regions where these systems have been deployed. For the sake of generality, in this work when we refer to 3G wireless networks we refer to the Universal Mobile Telecommunications Service (UMTS) and more specifically, to the WCDMA air interface in the Frequency Division Duplex (FDD) mode.

Figure 2.5 shows the network architecture of the UMTS Release 99 system [3,59]. The radio access network for UMTS is known as Universal Terrestrial Radio Access Network (UTRAN).

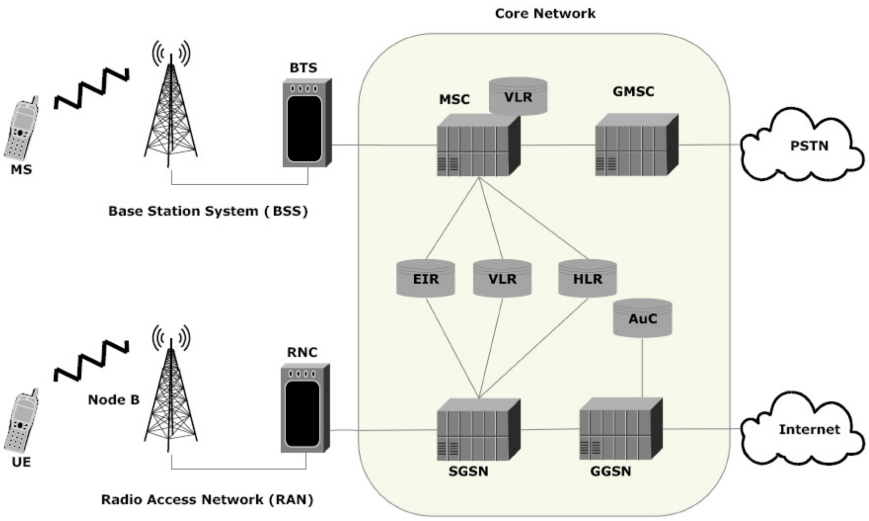


Figure 2.5: UMTS R99 Architecture [3]

The UTRAN is a WCDMA base station capable of handling the Frequency Division Duplex (FDD) and Time Division Duplex (TDD) modes. The core network is based on an evolution of the GSM network. The core network supports both UMTS and GSM. Going forward in 3G's long term evolution (LTE) the core network is expected to be an all IP-based network, as described in Release 8 and beyond. Furthermore, Release 4 focused on changes to the architecture and the core network, while Release 5 introduced a new call model thus changes to the user equipment, access network and core network were expected [3]. Release 6, integrated operation with wireless LAN networks and added HSUPA. Release 7 focused on decreasing latency, improvements to QoS and real-time applications such as VoIP. This specification also focused on HSPA (High Speed Packet Access Evolution). Release 9 is focused on specifications for the evolution architecture and interoperability between Wi-MAX and LTE/UMTS [3].

As noted earlier, UMTS most significant capability is the high data rate, up to 2 Mbps for fixed environment. Table 2.2 lists the UMTS environments and the minimum data rates for each of them. In order to distinguish among services and meet user's requirements UMTS defines four service classes [59]:

Table 2.2: UMTS Data Rates

Environment	Minimum Data Rate
Fixed Environment	2 Mbps
Pedestrian	384 kbps
Vehicular	144 kbps

- Conversational - this service class refers to voice and video conferencing. Voice requires low data rates, low delay and low jitter. Video conferencing requires higher data rates, low delay, low jitter and lower error rates.

- Interactive - These applications generally consists of request/response transactions. These applications have a lower tolerance for errors and a greater tolerance for delay. The data rate depends on the applications but usually is significant in only one direction at a given time.

- Streaming - These applications are characterized by one-way services, the data rates can vary. Streaming applications have very low tolerance for error but greater tolerance for delay and jitter.

- Background - These applications do not have a tolerance for errors, but have almost no delay constraints.

It is important to note that speech can still be a circuit-switched service. The user has access to dedicated resources throughout the call. Also, UMTS uses an Adaptive Multirate (AMR) speech coder. The AMR speech coder allows for the speech bit rate to change dynamically during a call. This can be done on a frame by frame basis as described in the specification, although in reality these rapid changes are unlikely. The PHY layer frames have a length of 10 ms. A speech frame is 20 ms, thus each speech frame requires two PHY frames.

Now, we shall focus on describing the UMTS air interface. It is relevant to describe the UMTS system since this is the environment that our proposed engine is being tested. Table 2.3 list the platform's specifications. In this work, we focus on the WCDMA FDD solution

Table 2.3: WCDMA Technical Specifications
[59, 60]

Parameter	Specification
Multiple Access Scheme	DS-CDMA
Duplexing Scheme	FDD and TDD
Switching Design	Circuit and packet
Nominal Bandwidth	5 MHz
Carrier Spacing	4.4 to 5 MHz
Frequency Reuse Factor	1
Chip Rate	3.84 Mcps
Channel Coding	Convolutional and turbo code
Spreading Code	OVSF
Data Modulation	QPSK, 16QAM
Spreading Factors	4-256
Frame Length	10 ms
Speech Frame Length	20 ms
Spectrum Efficiency	0.4
Power Control	Open/Close 1500Hz
Receiver	Rake
Inter-BS timing	Asynch./synch.
Handover	Intra-mode (Soft, Softer), Inter-mode and Inter-system (GSM)

for the UMTS system. The air interface consists of a pair of 5 MHz carriers, one for the uplink and one for the downlink. There is a 190 MHz separation between carriers in the US. The spacing between carriers can be anywhere from 4.4 MHz to 5 MHz in increments of 200 kHz. The purpose of this additional spacing is to avoid interference.

In WCDMA, the user data is spread to a greater bandwidth than the user rate by using a spreading code. The spreading code is a pseudorandom sequence of bits known as chips. All users transmit at the same time, on the same frequency. At the receiver each user's signal is separated from others by using a despreading code. The ratio of the spreading rate (number of chips per second) to the user data rate is known as the spreading factor. The higher the spreading factor the easier it is to despread the desired user's signal. The chip rate for WCDMA is 3.84 Mcps, this rate corresponds to a carrier bandwidth between 4.4 and 5 MHz.

In terms of the network architecture, the UMTS network is an evolution of the GSM/General Packet Radio Service (GPRS) network. There are many components in common such as the Mobile Switching Center (MSC), the Home Location Register (HLR), the Serving GPRS Support Node (SGSN) and the Gateway GPRS Support Node (GGSN). These network components can be upgraded in order to support both GSM and UMTS. However, the Radio Access Network (RAN) known as UTRAN, has been changed significantly, thus reuse of the Base Transceiver Station (BTS) and the Base Station Controller (BSC) is limited.

In this section, we provided a brief overview of the 3G wireless network, more detailed explanations can be found in Holma and Toskala's book [58], *WCDMA for UMTS*.

2.4.3 Femtocells: An emerging technology in 3G Networks

Femtocells, also known as home base stations, are cellular data access points typically installed by users to improve voice and data coverage in their homes. Femtocells are low power, fully-equipped base stations that can serve anywhere from 2 to 5 users. An UMTS femtocell

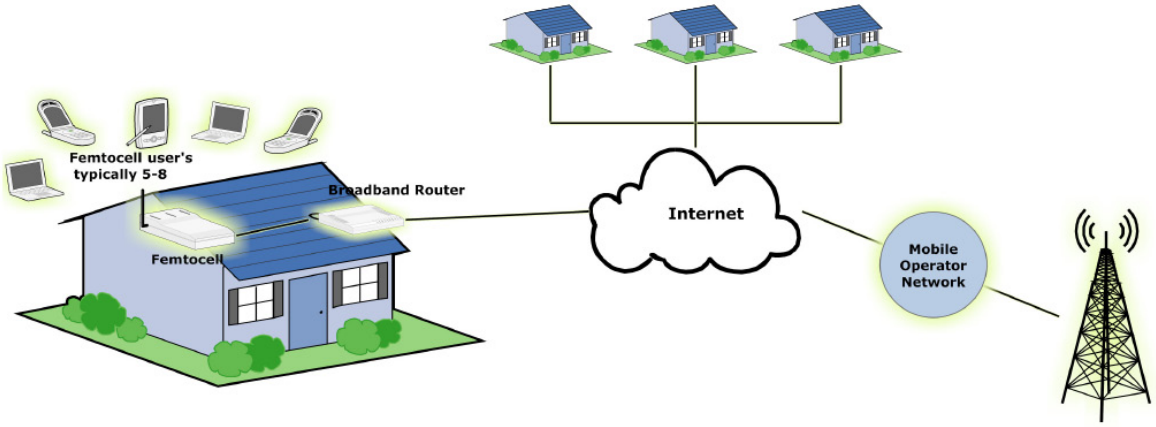


Figure 2.6: Femtocell concept diagram, reprinted with permission from the Femto Forum [4].

includes the following components: Node B, RNC and GSN. The femtocell connects to the backbone network by a broadband internet connection such as DSL or Cable. Femtocells increase system capacity minimizing the distance between the transmitter and the receiver and by increasing spatial reuse. Femtocell technology also offer some shortcomings: interference from macrocells and nearby femtocells limit the capacity of the system, and the increased strain on the backhaul may affect the system's overall throughput. Figure 2.6 shows a basic femtocell diagram.

Furthermore, according to the authors in [15], the deployment of femtocells can improve the performance of 3G networks by :

- providing better coverage,
- increasing capacity,
- increasing spectrum efficiency,
- improving link reliability,
- reduce operating costs,

- reduce subscriber turnover.

These key arguments in favor of the technology are also aligned with 3G operators LTE goals. However, there are key technical challenges that need to be addressed in order to successfully deploy the technology in current 3G markets. Femtocells need to adapt to their surrounding environment and allocate spectrum in the presence of interference. Femtocells also need to provide timing and synchronization with the network, in order to ensure handover between macrocell and femtocell environments. Furthermore, femtocells need to provide acceptable QoS and reduce latency in the backhaul. Other issues such as performing handovers, providing service to nearby subscribers, and providing 911 location tracking also need to be considered. We believe that cognitive radio can address these technical challenges in femtocells by providing cognitive capabilities such as awareness, intelligence and learning. The focus should be on simple cognitive engine architectures that can be contained in a single femtocell and exploited by all its components (Node B, RNC, GSN). Also, the algorithms used should be robust, provide fast solutions that do not increase processing time.

2.4.4 CR in the Evolution of Wireless Networks: 3G, LTE and B3G

Efforts to define the evolution of 3G networks are well on their way, the goal is to further reduce operators costs and to improve service provisioning to the end-user [24]. In order to achieve this goal operators and manufacturers are focusing in four key aspects:

- increasing system coverage,
- increasing system capacity,
- increasing data rates, and
- reducing latency.

Several solutions that address these issues have been proposed [3]. Researchers have suggested new architectures, multi-antenna solutions and evolved QoS and link layer approaches, among others [24]. From these solutions, only a handful propose applying cognitive radio [25–31].

In 2004, Aazhang et al. proposed two protocols to manage spectrum sharing by redistributing excess users to spectrum bands with excess user capacity. They simulated the protocols in a realistic cellular environment and both of these protocols showed gains in performance when base-stations shared spectrum resources versus just assigning spectrum to users on their licensed bands [25].

In 2006, Burnic et al. presented synchronization and channel estimation techniques for WCDMA systems based on cognitive radio. The focus was on the implementation of a sliding correlation-based fast iterative tap amplitude and delay estimation for RAKE receivers required for the downlink. The authors created a "proof-of-concept" demonstrator that performed the algorithm. It showed an improvement of approximately 3.7 times less in computational effort in comparison with the minimum requirements [26].

Furthermore in 2006, Shah et al. proposed an architecture of a sensing probe for wideband and distributed measurements method. They applied this architecture to the cellular band. Also spectrum measurements of one of the major cellular operators in the North Jersey area were conducted. The measurements showed that spectrum utilization peaked in the mornings and in the evening, the rest of the time the spectrum was underutilized; thus confirming that there is available spectrum to increase capacity in the cellular band [27].

In recent years, several papers that apply the cognitive radio concept to 3G networks have been published. Khajeh et al. extend the concept of cognitive radio to optimize the overall system power consumption for a WCDMA network, specifically for video transmission over the wireless medium. The authors reported 20% savings in the overall system power while maintaining the required quality of service [28].

Moreover, Harada showed a software defined cognitive radio prototype that combined WCDMA and IEEE 802.11a/b waveforms. The prototype senses the spectrum, chooses between the available waveforms, and then loads the waveforms to the devices. The cycle begins again after the time-to-sensing period completes. The prototype performed sensing and reconfiguration of these waveforms in the 400 MHz to 6 GHz band. The average reconfiguration time for each system was 1650 ms, and the actual sensing time was between 5-7 seconds [30,61,62]. In these paragraphs, we attempted to summarize some of the research that merges the cognitive radio concept and 3G wireless networks. Overall, little has been done to apply the CR concept to 3G networks.

Recently, research efforts have been focused on defining the architecture and capabilities of B3G (Fourth Generation (4G) or ITU IMT-Advanced) networks. In these next generation networks, wireless communication systems and the Internet will evolve into a new type of converged network. These networks will provide mobile users voice, data and multimedia streaming services *anytime, anywhere*. The design objectives include [16–23]:

- High spectral efficiency in terms of bits/Hz,
- high data rates (up to 100 Mbps in high mobility scenarios, 1 Gbps in low mobility scenarios),
- high network capacity (in terms of users per cell),
- seamless handover among heterogenous systems,
- global roaming and seamless connectivity,
- high QoS that allow for the next generation multimedia applications (mobile TV, life audio, teleconferencing, etc.),
- interoperability with existing wireless standards (2G, 3G, WLAN, etc.), and
- and all-IP packet switched network.

To achieve these objectives regulators, operators, manufacturers and researchers are considering many developing technologies including: orthogonal access schemes such as OFDM, smart antennas, mobile IPv6, software defined radio and cognitive radio. This converged network has been envisioned by some researchers [16–23], as a heterogeneous network that is reconfigurable, autonomous and has cognitive capabilities that allow the network to be aware of its environment, and is able to use previous knowledge to solve network problems. Others are looking into Self-Organizing Networks (SON)s as part of the solution for the long-term evolution of 3G wireless networks. SONs have been defined by the Next Generation Mobile Networks Alliance (NGMN) and Third Generation Partnership Project (3GPP) to have a standardized set of capabilities that include: self-configuration, self-operation and self-optimization [63–65]. SONs will provide operators reduced operating expenses (OPEX), while allowing for quick network deployment and technician-free configuration and optimization. The work presented in this dissertation can bring a significant contribution into the SON research area, by providing a low-computational complexity engine that can use history and previous knowledge to self-configure and self-optimize the network.

2.4.5 Knobs and Meters

In this section, we describe the *knobs* and *meters* of cognitive radio for a 3G wireless network. We define these terms in the context of cognitive radio as described by Rondeau in [66]. The knobs are the adjustable parameters that when set define the cognitive radio waveform. The meters are the metrics that we use to quantify the performance of the system. In 3G networks, the knobs are precisely defined by the standard and by the capabilities of the network components (Node B, RNC, etc.). The meters are also well defined, and depend on the reporting schemes in the protocol. However, how to adjust the *knobs* based on the *meters* is not specified in the standard, leaving opportunities for vendors to differentiate themselves based on their respective algorithms.

Table 2.4 lists the available knobs in the 3G system and Table 2.5 lists the possible meters

Table 2.4: Knobs

Knobs	Possible Values
DL Max. Transmitted Power	4W (Micro), 20W (Macro)
UL Max. Transmitted Power	0.2W (Micro), 1W (Macro)
Spreading Factor	UL: 4-256, DL:4-512
User Data Rate	480 kbps to 2.3 Mbps
Channel Coding	Conv. (1/2, 1/3) Turbo (1/3)
Frame Size	10ms - 60ms
Handover Type	Intra-mode (Soft, Softer), Inter-System (GSM)

Table 2.5: Meters

Meters
Received signal strength
Signal to Interference Ratio
Frame Error Rate
Bit Error Rate
User Priority
Time of day
Location
QoS Requirements
Traffic
Application

that measure performance in the system. The *knobs* and *meters* are critical in the cognitive radio design. The CR will optimize the allocation of resources and the performance of the system by adjusting the available knobs and evaluating the meters as part of its cognition cycle. As a result the CR will learn from these experiences and will identify combinations of *knobs* to desired values for the *meters*. If we consider B3G networks, additional *knobs* may include: Radio Access Technology (RAT) technology of choice, and other preferences such as: handover, service, cost, among others.

2.5 Summary

In this chapter, the concepts of cognitive radio and cognitive engine were introduced. Also, a brief history of previous cognitive engine implementations was presented. Moreover, future cognitive engine designs were suggested. The chapter also discussed cognitive radio applications for wireless networks, and its influence in their evolution. Finally, we listed the *knobs* and *meters* of the system. In the next section, an approach for applying cognition to wireless networks is presented. The proposed approach for combining case-based reasoning and decision tree learning is described.

Chapter 3

Machine Learning in Cognitive Engine Design

In this chapter, we propose a methodology for cognitive engine design. First, we discuss the impact of adding intelligence to communications systems. Then, we discuss key considerations in machine problem formulation. We proceed with a discussion of Case-based Reasoning and Decision Tree Learning, these are the techniques used in the implementation of the proposed cognitive engine. Next, we present an analytical framework that relates the learning mechanism to the learning objectives of the cognitive engine. We conclude the chapter with a brief summary.

3.1 Applying Machine Learning Techniques to Cognitive Engine Design

The field of Machine Learning (ML) has seen tremendous growth in the last three decades [67]. Meanwhile, wireless networks have also grown tremendously since their initial deployment in the 1980's. The expectation is that cognitive radio will merge advancements in these two fields, by applying machine learning techniques to current and future wireless networks. As discussed previously, cognitive radio can bring improvements in many areas such as: spectrum management, radio resource management, exploiting historic data, among others. Chapter 2 presented a brief history of specific machine learning implementations in the development of cognitive radio. Previous work at Wireless@VT surveyed six machine learning techniques and their applicability to cognitive radio [45]. Rather than discussing all the machine learning techniques that are suitable for cognitive engine design, first we would like to discuss the desired approach used for implementing these machine learning techniques in a successful and robust design. In the next section, we will present guidelines for the selection criteria of machine learning techniques.

In this research, we studied some of the published literature in the area [67–73] and found the approach suggested by Langley and Simon in [69] the most influential in our work. Also, Dietterich and Langley's most recent work in machine learning and cognitive networks provides an excellent survey of the machine learning techniques and the possible research challenges in cognitive networks [67].

Machine learning, as explained by Dietterich and Langley, tries to understand the computational mechanisms by which experience can lead to improved performance [67]. In other words, we say that "learning" has occurred if the system can perform an action that couldn't performed, or couldn't performed as well, before that experience. Langley and Simon [69] found that the application of learning methods followed a specific pattern as described in the following steps:

- Formulate the problem - this is the first step, in order to solve a "real-world" problem, we must formulate the problem in a way that can be handled or interpreted by a computational learning method.
- Determine the representation - in this step the attributes or features used to represent the problem are selected.

- Collect training data - this step can be performed automatically in real systems, or by simulation of accurate models. Another consideration is the accuracy and quality of the training data.

- Evaluate the learned knowledge - before knowledge can be used the quality of the knowledge must be verified.

- Fielding the knowledge base - once we acquired the knowledge, what is the best way to use it.

Furthermore, there are several agreements [67] in the machine learning discipline that help us narrow the approach:

- Experiments have shown that there is no mechanism that leads to better learning.

- Representational issues are integral to achieve the learning.

- Learning occurs in the context of a performance task.

Considering these general agreements, we must focus on *how to formulate the problem*, *how to represent the data*, and *how to define the desired performance task* in order to facilitate learning.

There are three general problem formulations for learning in the context of a performance task: classification and regression, acting and planning, and interpretation and understanding. Figure 3.1 summarizes how these formulations are related. If we focus on these three formulations we can get a clearer idea on the representation of the data and the desired performance task. The first formulation focuses on learning knowledge for the performance task of classification and regression. An example, of the first formulation in 3G wireless networks is classifying and predicting the coverage area of Node B, based on the number of dropped calls observed in the training data. There are two performance tasks in the example: one to classify the coverage area, the second to predict if the call will be dropped or not. In

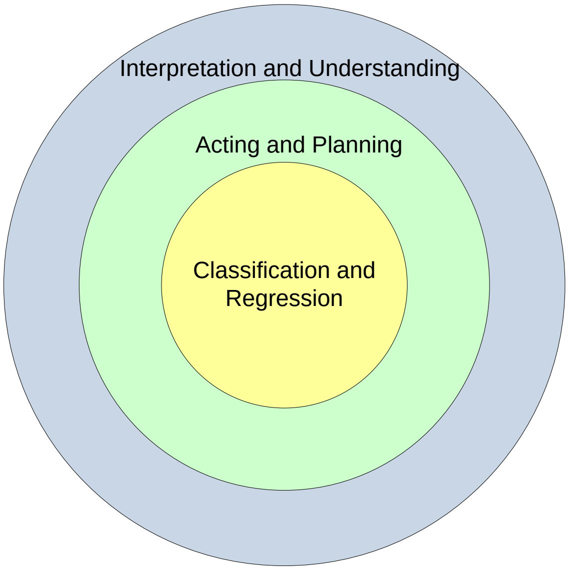

Interpretation and Understanding
Acting and Planning
Classification and Regression

Figure 3.1: Relationship of Machine Learning Problem Formulations

this case, the problem has been formulated as a classification and regression problem. The data is represented in cases, and the most important features (i.e. location of the UE, SINR, velocity) of these cases have been extracted. Classification and regression is the simplest formulation for a learning problem. Chapter 6 discusses this particular formulation of the coverage learning problem in greater detail.

The second formulation focuses on learning for selecting actions or plans for an agent [67]. The process involves making cognitive choices about future actions. This formulation usually employs a search through a space of possible outcomes, which can be constrained by the use of knowledge. An example of this type of formulation in 3G networks, is for the cognitive radio to plan for the lack in coverage, and decrease the probability of dropped calls in a specific area of Node B, by modifying its actions such as increasing power, handing over to another cell or to another system (i.e. GSM). This formulation can also be simplified to the first formulation if the actions and plans are determined in a reactive way, ignoring past information.

The third formulation addresses learning via interpreting and understanding the situation.

In this formulation the models and the data are contained in deeper structures, this process is often referred as *abduction*. One example where this formulation is used extensively is in natural language processing. The performance task in this case involves parsing sentences using a context-free grammar [67]. For a 3G cognitive radio, the performance task can be interpreting that the coverage gap in Node B is due to an obstruction, given the pattern of the dropped calls. This is something that a human can interpret easily by looking at the coverage map, the idea is for the cognitive radio to interpret this without the need of human intervention. The cognitive radio uses past experience, and the reported dropped calls from the User Equipment (UE)'s to figure out the coverage problem. This is Mitola's vision of a cognitive radio as discussed in the previous chapter.

In this work, we focus on the first formulation of learning for classification and regression. We employ case-based reasoning and decision tree learning to address the coverage problem in 3G wireless networks in Chapter 6. We extend this problem by providing an action for the acquired knowledge. We design an engine that predicts the need for handover based on the coverage map of the cell obtained from previous observations (past) and the user's mobility pattern (current), in Chapter 7.

In the next section, we discuss the machine learning techniques used in the implementation of the cognitive engine and criteria used to select these algorithms. We present a typical case for the case-based reasoner in the coverage problem.

3.2 Motivation for CBR and DT Algorithms in Wireless Networks

In previous research, we had the opportunity to survey a plethora of machine learning algorithms and determine their applicability to the development of a cognitive radio engine [45]. We concluded from our research that no single machine learning technique will yield a robust cognitive engine. It is rather the combination of algorithms according to the desired optimization task or application that produces a better cognitive engine [45]. Also, it is important to note as stated by Dietterich and Langley [67], that the formulation of the problem and the representation of the data, are keys in successful machine learning applications. In the next paragraphs, we will discuss the process involved in each technique, how we represent the data and discuss the merits and limitations of each technique.

3.2.1 Case-based Reasoning

Case-based reasoning (CBR) is an area of machine learning that focuses on using previous similar experiences, or in other words, cases to guide the problem solving process and to achieve a solution [74]. CBR is based on a memory-centered cognitive model in which previous experiences are recalled from memory and then adapted in order to solve the problem. In CBR, a solution to the problem is created by selecting the cases that are relevant to the problem, and then the best match is chosen and adapted to fit the current case or instance [5].

There are several types of CBR methods including: Exemplar-based reasoning, Instance-based reasoning, Memory-based reasoning, Case-based reasoning and Analogy-based reasoning among others [75]. The main differences between these methods are the types of problems that they solve, the algorithms used for indexing and retrieval, the type of cases used (i.e. concrete versus open, etc.), the use of general knowledge for adaptation or guidance in the CBR process, and whether the system interacts with the user or not. The CBR approach that we propose in this work, can be considered an example of Instance-based Reasoning (IBR), these systems are characterized by the lack of general knowledge, large number of instances and a simple representation of cases (e.g. feature vectors).

CBR offers several advantages [5] that makes this machine learning technique suitable for cognitive engine design. CBR systems are easy to implement if experience is rich, even if there is not enough general knowledge about the problem, and rules cannot be derived.

Another advantage of CBR, is that it provides a closer match to the actual human reasoning process. Furthermore, CBR can provide efficient reasoning by allowing the user to focus on the problem solving aspects that were important in previous cases; it can also help the user to avoid incorrect directions that were not successful in previous instances. CBR also allows for faster knowledge acquisition, given that the knowledge resides in a case memory. In some domains (i.e. medicine, law, engineering, etc.) there are existing case bases that can be used. In the case of 3G and B3G wireless networks, the operator can create a case memory by mining network logs. Also, maintenance efforts can be reduced as the CBR system will learn from the new cases and update the case library as new problems and solutions are tackled.

One of its limitations, is the system's reliance solely on previous cases. If previous cases have been solved incorrectly there is a likelihood of propagating those mistakes onto new cases. Also, the system may require a large case memory. Collecting a large case memory may be time consuming and in some instances difficult. As a solution, in 3G and B3G wireless networks the use of the Radio Environment Map (REM) can shorten the process of collecting the cases. Another solution is to integrate CBR with a knowledge-based reasoning system to improve the performance of the system and reduce the time it takes to build a strong case memory [76].

In this work, we investigate applying CBR to cognitive engine design since the technique performs very well in dynamically changing environments such as the 3G wireless network environment. This is most often true for wireless networks; models are very hard to develop due to the time variance of the system. Network operators rely frequently on field measurements. With the addition of cognition this process may be simplified as the user equipment can provide measurements on the radio's surrounding environment. Researchers at the University of London have employed CBR successfully to learn traffic patterns during periods of congestion and used them to control cooperating semi-smart antennas to optimize radio coverage, thus minimizing congestion [77, 78].

Figure 3.2 describes the general architecture of a case-based reasoner, while Table 3.1 de-

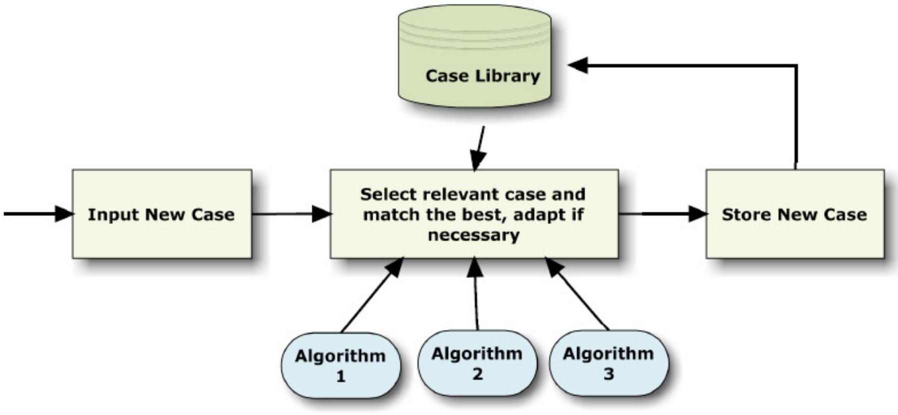


Figure 3.2: Case-based Reasoning System Architecture

Table 3.1: The Case-based Reasoning Process [5]

Step	Objective	Details
1	Organize case memory	Cases should be organized in a manageable structure.
2	Accept a new case and choose relevant ones	A new case is entered, retrieve relevant cases.
3	Select best match	The best match is selected, adaptation may be necessary.
4	Construct a solution for the new case	The system provides a solution to the current problem.
5	Test the new solution	The solution is tested, if the solution is optimal is implemented.
6	Add new knowledge to the case memory	The new solution can be added to the case memory for future problem solving.

scribes the steps involved in the case-based reasoning process. It is important to note that steps 2 and 3 are very complex. There are many case retrieval techniques that can aid in the decision process of selecting a case such as: fuzzy mathematical methods, nearest neighbor search, statistical weighting methods, preference heuristics and many others. As we will discuss in the next sections we have selected a DT search based algorithm to aid in the classification cases.

One of the key components in case-based learning is the case memory. The case memory is simply put a database of cases. Cases are detailed descriptions of activities, event, problems, etc. usually derived from real-world experience [79]. According Carroll and Rosson good cases have the following properties:

1. Cases are open-ended.
2. Cases unfold one at a time.
3. Cases involve specific actor and circumstances.
4. Cases can be used in many ways.

One important issue in cognitive engine design is how to define the “case” for the specific CR application, and how to measure the similarity between different cases for the case matching process. Typically a case includes the *problem*, which describes the state of the world at the time the case occurred, the *solution* which states the solution to the problem and the *outcome* which describes the state of the world after the case occurred [80]. The similarity of cases can be determined by using a similarity function. The similarity function measures the similarity between the problem case's attributes and the matched case attributes.

Table 3.2 shows a sample case for the coverage problem in 3G wireless networks. In the case we provide the position of the UE with respect to Node B, the velocity; and given knowledge on the channel and the environment, the Signal-to-Interference and Noise Ratio (SINR) observed by the UE is predicted. For the case retrieval step, there are many techniques

Table 3.2: Sample case for improving coverage in 3G Networks

Attribute	Variable
Position of UE in the x axis	x
Position of UE in the y axis	y
Angle of UE	θ
Velocity of UE in kmph	v
Signal to Interference and Noise Ratio	SINR

that can be used in the process case such as: fuzzy mathematical methods, nearest neighbor search, statistical weighting methods, decision tree searches, preference heuristics and many others. We have selected a decision tree algorithm to aid in case selection.

3.2.2 Decision Tree Learning

Decision Tree Learning (DTL), a technique used in data mining and machine learning, employs a decision tree to represent a classification system or a predictive system [81]. The resulting tree is a set of hierarchical rules that divide the data in groups, where a decision is made for each group. The hierarchy is called a tree, each segment is called a node and the initial segment is the root node of the tree. The terminal nodes are called leaves, they return a decision which is the predicted output given the input. Decision trees can be grouped in two general categories: classification trees (CT) and regression trees (RT). CT use discrete-valued functions, while RT use continuous-valued functions.

DTL has several advantages [81–83] that lead us to consider the technique for the cognitive engine design such as:

- *Simplicity*: DTs are easy to understand and interpret,
- *Robustness*: DTs manage large amounts of data quickly,
- *Reliability*: model can be validated using statistical methods.
- *Flexibility*: DTs can work successfully with discrete and continuous attributes.

- *Maturity*: DTs have been well studied in literature. Also, there are efficient algorithms (such as ID3 and C4.5) that reduce development time.

More formally, we can describe the procedure for generating decision trees as outlined by Hunt and described by Quinlan [82] in his seminal paper on decision trees.

Let S be a set of objects, each belonging to one of the classes $c_1, c_2, ..., c_k$, then the following steps are performed:

1. If all the objects in the set S belong to the same class, c_i, then the decision tree for S has a single leaf labeled c_i.
2. Otherwise, let T be some test with possible outcomes $o_1, o_2, ...o_w$. Then, each object in S has one outcome for T so the test partitions S into subsets $S_1, S_2, ..., S_n$, where each object with outcome o_i belongs to S_i. T becomes the root of the decision tree and for each outcome o_i a subsidiary decision tree is built by performing the same procedure recursively on S_i.

The attributes are considered adequate if there are no two objects in the training set that have the same value for every attribute and belong to different classes. Limitations of the decision tree algorithm arise when dealing with noisy data; in other words when attribute values contain errors, or there is lack of information for a proper classification of the objects in set S.

Generally, there are two approaches to deal with noisy data one is to apply a *stopping criterion*, the second is to let the tree grow without constraint and then eliminate unimportant portions of it, the latter approach is known as *pruning* [82]. Other limitations arise when the data is incomplete or when the classification of objects is not certain, say objects may belong to more than one class. In this work, we use a simplified version of the decision tree algorithm, where only two classes are used, c or $\hat{c}$.

ID3 Algorithm: Iterative Dichotomiser 3

The ID3 algorithm was developed by J.R. Quinlan in the 1980's. It was based on Occam's razor and it is iterative, therefore a heuristic. The idea was that if two decision trees produced the same solution for the same training set, the simplest tree was the best option [84]. The general process of tree induction in ID3 can be described as follows: the first step is to generate a tree from a random subset of the training data called a *window*. Then, if all the objects in the training set were classified properly, then the process terminates and the resulting tree is the one created from the *window*. If not, a selection of the misclassified objects is added to the *window*, and the process repeats [84]. The tree induction process in ID3 decreases the amount of time needed to induce the tree, thus a preferred solution for inducing trees even where the training sets are rather large (i.e. over 30,000 objects and over 50 attributes) [84].

Formally, we can describe the process of tree induction following Quinlan's approach in one of his seminal papers on ID3 [84]. The algorithm is carried out as described:

Let S be an arbitrary collection of objects. If S is empty or containing only objects belonging to the same class, the resulting tree is a leaf labeled with that class. Otherwise, let T be any test on an object with possible outcomes $o_1, o_2, ...o_w$. Each object in S will result in one of the outcomes for T, such that T produces the following partitions on S, $S_1, S_2, ..., S_w$, with S_i having the objects with outcome o_i.

The key in this algorithm is to choose test T, such that the resulting tree is simple. The way that ID3 chooses the test T is by determining the amount of information that each attribute contains, and testing on the attributes with the highest amount of information at the root of the tree.

Let S contain ρ objects from class c and η objects from the class $\hat{c}$. With the following assumptions:

- Any correct decision tree, will classify objects in the same proportion as they appear in S. That is, a random object from S will belong to class c with probability $\frac{\rho}{\rho+\eta}$, and to class $\hat{c}$ with a probability $\frac{\eta}{\eta+\rho}$.

- A DT classifies an object, by returning the class the object belongs to. Therefore, a DT can be viewed as source for the message 'c' or '$\hat{c}$' with the expected information needed to generate the message as:

$$I(\rho,\eta) = -\frac{\rho}{\rho+\eta}\log_2\frac{\rho}{\rho+\eta} - \frac{\eta}{\rho+\eta}log_2\frac{\eta}{\rho+\eta} \tag{3.1}$$

Then, if attribute a with values $a_1, a_2, ..., a_v$ is used for the root of the DT, it will partition S into $s_1, s_2, ..., s_v$ where S_l contains the objects in S that have the value a_l for attribute a. Let S_l contain ρ_l objects of class c and η_l of class $\hat{c}$. The expected information required for subtree S_l is $I(\rho_l,\eta_l)$. The expected information required for the tree with attribute a as root, is then obtained as the weighted average

$$E(S,a) = \sum_{l=1}^{v}\frac{\rho_l+\eta_l}{\rho+\eta}I(\rho_l,\eta_l) \tag{3.2}$$

where the weight for each branch is proportional to the objects in S that belong to S_i. The entropy function versus the probability for a two-class variable problem is shown in Figure 3.3.

The information gain, $G(a)$ is defined as:

$$G(S,a) = I(\rho,\eta) - E(S,a) \tag{3.3}$$

the information needed to generate the message minus the expected information for the tree with a as root. The idea is to branch on the attribute a_i that maximizes the information gain. The process is repeated recursively to form subtrees on the remaining sets. The pseudo code for the algorithm is included in Algorithm 1 [85]. Where N is the set of training examples and A is the set of attributes.

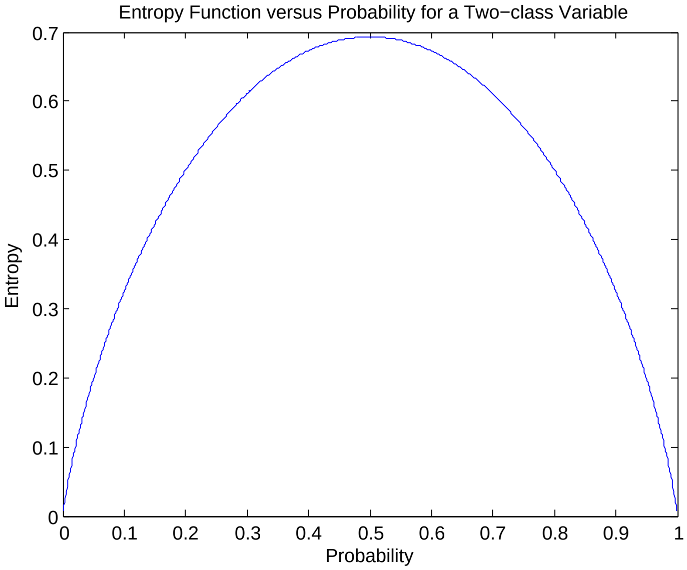


Figure 3.3: Entropy Function versus Probability for a Two-class Variable

Algorithm 1 ID3 Algorithm

```
Create a *Root* node for the tree
**if** all objects are in the same class $c$ **then**
  the tree is a leaf, return leaf labeled $c$
**end if**
**if** all objects are in the same class $\hat{c}$ **then**
  the tree is a leaf, return leaf labeled $\hat{c}$
**end if**
**if** attributes set is empty **then**
  Return single node tree *Root*, labeled with most common value of the target attribute in N
**else**
  Let $a_i$ be the attribute that best classifies $N$
  The decision tree attribute for *Root* is equal to $a_i$
**end if**
**for** each possible value $f_i$ of $a_i$, **do**
  Add a new tree branch below *Root*, corresponding to the test $a_i = f_i$
  Let $N(f_i)$ be the subset of objects that have the value $f_i$ for $A_i$
  **if** $N(f_i)$ is empty **then**
    below this branch add a leaf node labeled with the most common target value in $N$
  **else**
    below this branch add the subtree with objects $N(f_i)$ and attributes equal to $A - a_i$
  **end if**
**end for**
```

C4.5 Algorithm

The C4.5 algorithm, also developed by J.R. Quinlan in the 1990's was designed to overcome some of the limitations of the tree induction process and its predecessor, the ID3 algorithm. The main benefits of using C4.5 are: the algorithm's ability to handle discrete and continuous data, the ability to handle data with missing information, and also C4.5 allows the tree to grow without constraints and then prunes the irrelevant branches.

C4.5 builds a tree from a set of objects S, like ID3, it also applies the concept of information entropy. Information entropy is an estimate of the amount of information contained in set S, and the information gain is the change in entropy from one attribute to the next attribute in the context of the set S. The information gain is used to determine the relevance of an attribute when tested, thus if an attribute has a high information gain is desirable to test that attribute closer to the root of the tree to minimize computation and latency.

C4.5 employs the fact that each attribute a of the set S can be used to make a decision that splits the data into smaller subsets. Furthermore, C4.5 examines the normalized information gain, or in other words the difference in entropy that results from choosing an attribute a for splitting set S. Let a_k be the attribute in A with the highest normalized information gain, then this attribute is used to make the decision. The algorithm recurs on subsequent tests, pseudo code for the algorithm in included in Algorithm 2.

Algorithm 2 C4.5 Algorithm

```
if all objects in $S$ are in the same class $c$ then
    the tree is a leaf, return leaf labeled $c$
end if
for each attribute $a \in A$ do
    calculate $G(S, a)$ from splitting on that attribute
end for
Let $a_k$ be the attribute with the highest $G(S, a_k)$
Create decision node that splits on $a_k$
Recur on the branches obtained from splitting on $a_k$
```

Table 3.3: Sample coverage data with numeric attributes

Day	Outlook	Temperature	Wave Height	Surfing
1	Sunny	Hot	3	Yes
2	Sunny	Warm	4	Yes
3	Rainy	Cool	6	No
4	Sunny	Hot	4	Yes
5	Overcast	Cool	7	No
6	Rainy	Cold	4	No
7	Overcast	Cold	6	No
8	Overcast	Cool	4	Yes
9	Sunny	Warm	3	Yes
10	Overcast	Cool	7	No

3.2.3 Decision Tree Example

As an illustrative example, let us examine a small set of the training data included in 3.3. We want to predict the likelihood of going surfing on a future date, given previous weather conditions that include attributes such as: *Outlook, Temperature* and *Wave Height*. After classifying the data by testing on the attributes in the order that they appear, the resulting tree from that induction process is shown Figure 3.4. In this example, there are 5 examples from class c and 5 examples from class $\hat{c}$. Therefore, the entropy for this data sample is 1. From the set S of examples, there are 4 examples that are *Sunny*, 4 examples that are *Overcast* and 2 examples that are *Rainy*. The *Sunny* set has 4 examples from class c (Yes, gone surfing), and 0 examples from class $\hat{c}$ (No surfing). The *Overcast* set has 1 example from class c and 3 examples from class $\hat{c}$. The *Rainy* set has 2 examples from the class $\hat{c}$. Thus, the information gain due to sorting the original 10 examples by the attribute *Outlook* is as follows:

$$G(S, Outlook) = 1 - \frac{4}{10} Entropy(S_{sunny}) - \frac{4}{10} Entropy(S_{overcast}) - \frac{2}{10} Entropy(S_{rainy}) \quad (3.4)$$

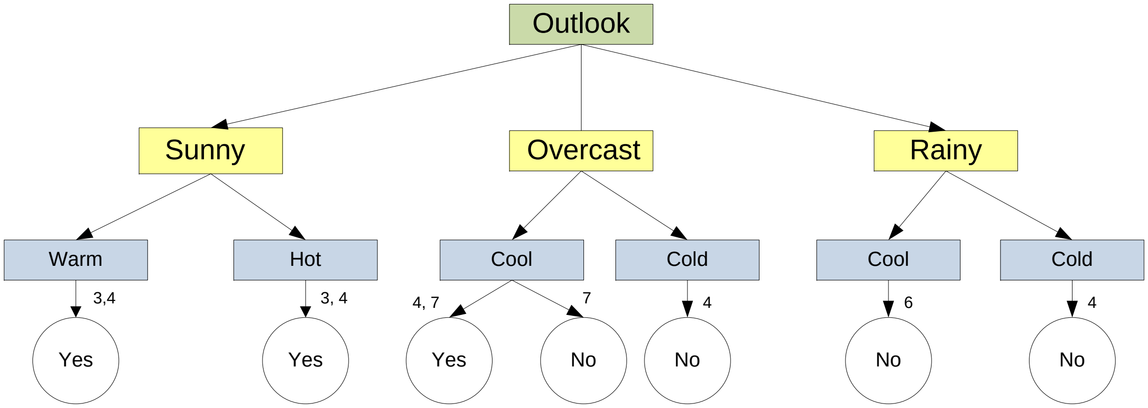


Figure 3.4: Sample Decision Tree

The entropy of each subset is determined as follows:

$$Entropy(S_{sunny}) = -1 * log_2(1) - 0 * log_2(0) = 0 \tag{3.5}$$

$$Entropy(S_{overcast}) = -\frac{1}{4} * log_2(\frac{1}{4}) - \frac{3}{4} * log_2\frac{3}{4} = 0.8113 \tag{3.6}$$

$$Entropy(S_{rainy}) = 0 * log_2(0) - 1 * log_2(1) = 0 \tag{3.7}$$

Thus, the resulting information gain from partitioning on the attribute *Outlook* is:

$$G(S, Outlook) = 1 - 0 - \frac{4}{10} * 0.8113 - 0 = 0.67548 \tag{3.8}$$

To follow the induction approach in the C4.5 algorithm, we must first evaluate all the attribute sets' entropies and place the attribute that maximizes the entropy at the root of the tree, and perform this procedure recursively on each branch as described earlier in Algorithm 2. In this example, perhaps testing on the attributes *Temperature* or *Wave Height* results in a simpler tree structure. In the next section, we will focus on how case-based reasoning can be improved with the addition of decision tree learning. Some of the literature published in this area is discussed. Arguments on favor to this approach are made.

3.2.4 Improving Cased-based Reasoning with Decision Trees

Case-based reasoning and decision tree learning are machine learning techniques that focus on analyzing existing data to develop some form of intelligent decision-support or decision making tool [86]. In this work, we propose a cognitive engine for wireless networks that employs both techniques.

Decision trees are great classifiers and predictors; while CBR systems focus on solving a problem, based on past experiences. CBR systems also employ indexing and retrieving algorithms in their problem solving process as discussed in previous sections. Decision trees can be used to improve the performance of CBR systems by aiding in the classification of cases and reducing the search space for relevant cases, by determining the features that are more important in the problem solving or case retrieving, and by speeding knowledge acquisition by the induction of generalized rules.

Now, let us review some of the literature in this area where improvements on CBR systems by applying DTs have been documented. In 1993, Cardie proposed using decision trees to improve case-based learning. The author employed decision trees to specify the features to be included in the k-nearest neighbor case retrieval. Results showed that the hybrid approach outperformed both single technique approaches (decision tree and case-based reasoning), and also outperformed two case-based reasoners that employed knowledge into the case retrieval algorithms [87]. Later in 1996, Richardson and Warren proposed using decision trees to determine the weight (i.e. importance) of features for the classification and retrieval of cases in a CBR system. The authors also assessed the tolerance of the technique to noise and incomplete data. Results showed that the addition of decision trees for rule induction in the CBR system improved the tolerance of the system to noisy and incomplete data, the noise or missing data had little or no impact due to the proper classification of the decision tree algorithm [86].

In 1997, Tsatsoulis et al. proposed using decision theory and CBR to aid chemists in the design of pharmaceuticals. The authors employed decision theory to help in the decision making process when uncertainty in the data was found, and when conflicting goals in the design process of a pharmaceutical arose [88]. More recently in 2005, Hüllermeier introduced a framework for experience-based decision making (EBDM) as an extension to case-based decision making (CBDM) [89]. The author proposed using satisficing[2] decision trees to establish direct relationships between decisions and the actions in problem solving. In 2008, Chang et al. proposed a hybrid approach using CBR and fuzzy DT to predict the stock price movement in the Taiwan Stock Exchange Corporation. Experimental results showed that their hybrid approach outperformed other approaches by predicting the stock movement with 91% accuracy.

CBR allows for problem solving from experience, while DT can make indexing and classification of this experience (e.g. cases) more efficient, thus significantly improving the performance of the case-based reasoner. After examining this brief sample of the literature, we feel that further research in this hybrid approach, and its application to cognitive radio engine design is worth pursuing.

3.3 Learning and Reasoning Model

In this section, we provide a detailed description of the learning and reasoning model used in the cognitive engine's design. Also, an analytical framework that relates case-based reasoning and decision tree learning with the engine's design is presented. It is important to note, that the environment where our intelligent agent is *learning* is the wireless network environment.

Using Mitchell's definition for *learning* [85] we have that:

[2] Satisfice = satisfy + sacrifice [89]

"A computer program is said to **learn** from experience Σ with respect to some class of tasks Υ and performance measure Φ, if its performance at tasks in Υ, as measured by Φ, improves with experience Σ."

As an example, let's examine the learning problem of handover management where the computer program is the cognitive engine:

- Task Υ: handover management.
- Performance measure Φ : handover probability, rate of false handover, rate of handover, and handover delay.
- Experience Σ: handover event observations in the wireless network.

Once the *learning* problem is formulated, we proceed to follow the methodology discussed in earlier sections by: determining the knowledge needed, determining the representation of the data, collecting the training data, evaluating the learned knowledge and finally fielding the learning problem in an application. Also, to perform these steps we must select a learning mechanism, in this work we will employ a combination of Decision Tree Learning and Case-based Reasoning.

A key element in our cognitive engine's design is the experience Σ. The engine's experience includes all of the previous observed conditions, as well as, the current network conditions. The current network conditions can be defined as a vector $\mathbf{a} = (a_1, a_2, ..., a_m)$, where a_i is the value of the i^{th} attribute. These attributes describe the current network conditions, such as: time, location information, geographical information, user preferences, network preferences, mobility information, power measurements, among others. Let M be the set of all possible attribute vectors.

In general, the design objective of the cognitive engine is to forecast the occurrence of an event at some future time, $e_{t+\Delta t}$, given the current observed conditions at time t, $\mathbf{a}_t \in M$. The cognitive engine will generate an estimate probability of the event occurring:

$$P(e_{t+\Delta t}|\mathbf{a}_t) \tag{3.9}$$

If this probability is reasonably close to 0 or to 1, the cognitive engine has successfully used previous experience to get information about future events, thus enhancing the decision process.

The event $e_{t+\Delta t}$ is modeled as a Bernoulli trial, where the probability of success is a function of $\mathbf{a}_t$. To accomplish this, we model the event $e_{t+\Delta t}$ as a mixture model as follows:

$$f_{e_{t+\Delta t}}(x) = \sum_{i=1}^{m} w_i(\mathbf{a}_t) f_{Y_i}(x), x \in \{0, 1\} \tag{3.10}$$

where $0 \leq w_i(\mathbf{a}_t) \leq 1$, $\sum_{i=1}^{m} w_i(\mathbf{a}_t) = 1$ and

$$Y_i(x) = p_i^x (1 - p_i)^{(1-x)} \tag{3.11}$$

Decision tree learning and case-based reasoning are used to estimate the parameters for the mixture model. Let m be the number of leaf nodes in the decision tree, such that the mixture model contains one component for each leaf node. Define $dt(\mathbf{a}_t)$ as a function that indicates the leaf node that $\mathbf{a}_t$ evaluates to, and:

$$w_i(\mathbf{a}_t) = \left\{ \begin{matrix} 1, & \texttt{if} \;\; dt(\mathbf{a}_t) = i \\ 0, & \texttt{o.w.} \end{matrix} \right\} \tag{3.12}$$

which implies that if $\mathbf{a}_t$ evaluates to leaf node i, then the i^{th} component of the mixture model will have a non-zero weight.

Each case in the case library represents the current network conditions at some time t and whether the event e occurred at time $t + \Delta t$. Let $H \subset M$ be a finite set which consists of the cases in the case library and define $H_i \subset H$ such that, for $h \in H$, if $dt(h) = i$ then

$h \in H_i$. This implies that all $h \in H_i$ are sample observations of the i^{th} component of the mixture model. Let $H_i^+ \subseteq H_i$ be the set of all $h \in H_i$ where the event e occurred at time $t + \Delta t$. We can estimate the parameter p_i for the i^{th} component in the mixture model with the maximum likelihood estimator (MLE) for the probability in a Bernoulli trial as follows:

$$p_i = \frac{|H_i^+|}{|H_i|} \tag{3.13}$$

Therefore, if $dt(\mathbf{a}_t) = i$ then

$$P(e_{t+\Delta t}|\mathbf{a}_t) = p_i \tag{3.14}$$

Therefore, the decision tree is used to estimate the components of the mixture model, and the case base library is used to estimate the parameter of each of the components.

3.4 Summary

In this chapter, we presented the motivation for using artificial intelligence and machine learning techniques in wireless networks. More specifically, the use of these techniques in the development and design of a cognitive radio engine. First, an approach to adding cognition to wireless systems was presented. Case-based Reasoning and Decision Tree Learning, the techniques used in the proposed cognitive engine implementation, were presented in greater detail. Furthermore, an analytical framework that relates the learning and reasoning techniques to the cognitive engine's design objectives was presented. This analytical framework is the major contribution of this chapter.

Chapter 4

Generic Cognitive Engine

4.1 Generic Cognitive Engine Model

In this section, we describe the cognitive engine model and its components. We discuss the cognitive engine and the interactions of its components in the context of 3G wireless networks, however these interactions can be translated to other networks (femtocells, B3G, mobile ad-hoc networks, etc.). The location of the components within the network may change as the intelligence and computational capabilities may be distributed differently.

4.1.1 Cognitive Process for the Engine

Figure 4.1 shows the basic diagram of the generic cognitive engine (CE). Figure 2.5 (see Chapter 2) shows of the UMTS network architecture according to Release 99 [3], this diagram will aid identifying the location of the engine components. The cloud in figure 4.1 represents the cognitive radio's surrounding environment, the 3G wireless network. Real-time data from the surrounding environment is gathered by the *Sensing* module. The *Sensing* module contains all the sensors and capabilities to sense the surrounding radio environment. Generally, the *Sensing* module resides in the user equipment (UE), although in some cases

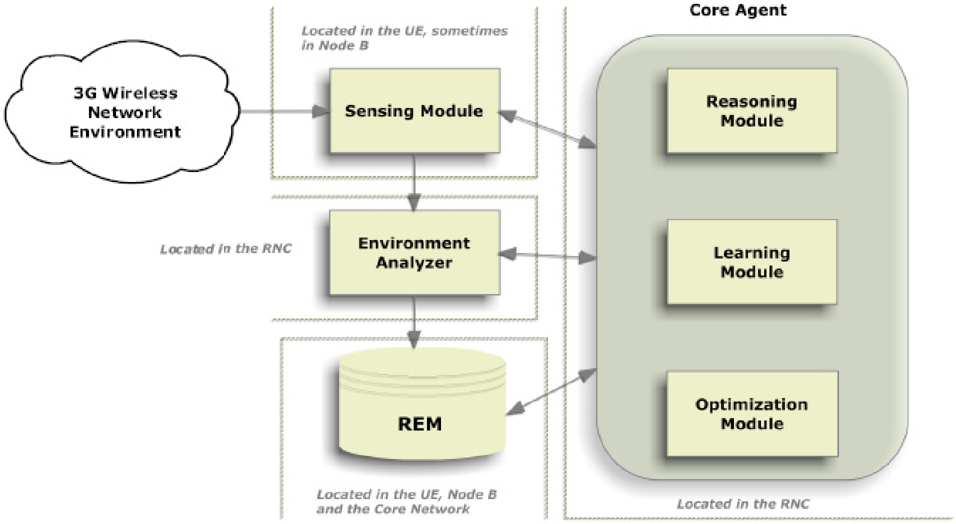


Figure 4.1: Generic Cognitive Engine

the radio access network (RAN) may have sensing capabilities as well. The *Sensing* module is instructed by the *Core Agent* when, how often and how much of the spectrum to sense. Since most of the intelligence in 3G networks resides in the radio network controller (RNC), it is apparent that the *Core Agent*, which is the intelligent agent that oversees all of the cognitive radio functions, it is also included in the RNC.

As seen in figure 4.1, the sensed data is passed from the *Sensing* module to the *Environment Analyzer* module, this module synthesizes and abstracts relevant environment information to the engine. This module generally also resides in the RNC, but in some cases it may be advantageous for the user equipment (UE) to synthesize and minimize the amount of transmitted data relevant to the current situation.

The newly sensed data is stored in the local *Radio Environment Map* (REM). The REM is an integrated database that consists of comprehensive multi-domain information for the cognitive radio, it includes geographical information, available services, spectral regulations, location and activities of nearby radio devices, policies and past experiences [90–93]. The

REM receives information from internal and external sources such as: the *Sensing* module, other REMs in the same network, REMs in external networks (i.e. 2G wireless network), policy databases, geographical information databases, the network management databases (HLR, VLR, AuC, EIR, etc.) and many others. The REM resides in various locations throughout the network, there is a local REM in the UE, another local REM in the RNC, and the global REM which is generally located in the core network where other entities such as the home location register (HLR), visitor location register (VLR), authentication center (AuC) and the equipment identity register (EIR) are located. Data derived from these databases is used in the REM, however, the information stored in the REM is more longitudinal, and may have been fused or modified for the environment analyzer. The idea is for the operator to be able to used stored information to identify patterns and specific user behavior that can help in the allocation of resources, and improve QoS. This in turn, is another way for operators to differentiate themselves in such a competitive market.

Once the sensed information is synthesized, the *Core Agent* proceeds to react to this information. The *Core Agent* may use the synthesized sensed data along with radio environment information stored in the REM, and the end-user requirements to create an abstraction of the current situation. This process is one of the most significant ones in the CE design process, since the machine learning technique and the approach used to derived a solution depends significantly on the type of information available and how this information is classified or organized. 3G networks are complex systems, the amount of variables that describe the system at a given time is very large, hence we must reduce the number of variables in order to reduce the amount of memory and the computation power needed. In order to achieve this, we may perform feature extraction or use classification algorithms that can eliminate redundant information.

The next step in the cognitive process is for the *Core Agent* to analyze the abstraction and determine the best reaction to the current situation. The *Core Agent* may employ the *Reasoning* module, the *Learning* module or the *Optimization* module. The *Core Agent* may select a single module at a given time or any combination of the three, generally these processes occur sequentially not in parallel. The idea is to allow the most flexibility to the *Core Agent* but at the same time minimize the convergence time to a solution. It is also important to note, that most of the learning is meant to be off-line learning. The *Core Agent* will employ data mining and machine learning techniques to acquire new knowledge about the system, it does not necessarily has to be online.

As an example, let's say the *Core Agent* determines that an action must be taken, and proceeds with the *Reasoning* module, the *Reasoning* module provides a solution to the *Core Agent.* The *Core Agent* proceeds to evaluate the solution, if the solution lies in the acceptance range, the *Core Agent* stores the solution for future instances, and reacts to the environment with the solution. If the solution does not lie in the acceptance range, the *Core Agent* may modify the solution provided by the *Reasoning* module by further optimizing it with the *Learning* or the *Optimization* modules. Once the *Core Agent* determines that an optimal solution is found the *Core Agent* stores the solution, as well as the process used to determine the solution. The *Core Agent* reacts to the environment with an action and then validates the solution by doing performance evaluation. The *Core Agent* may instruct the UE to sense the environment or send specific performance metrics, and the cognitive process repeats with this new sensed information. In the next section we will focus on describing each of the engine's components and suggest machine learning techniques for the implementation. We also specify the inputs and outputs of each module.

4.1.2 Cognitive Engine Components

In this section, we discuss the cognitive engine components in greater detail. We suggest possible implementations and suitable machine learning techniques. We also identify the inputs and outputs for each component and describe how these components can be implemented in our proposed cognitive engine.

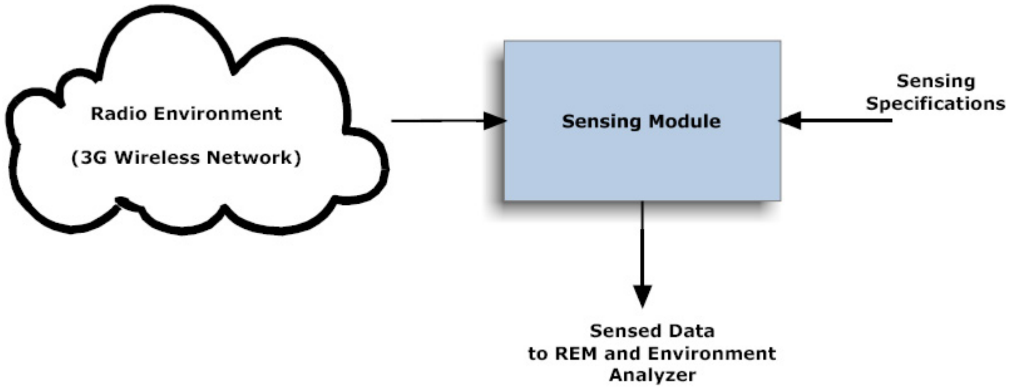


Figure 4.2: Sensing Module Input/Output

Sensing Module

Cognitive radio distinguishes itself from software defined radios and adaptive radios, by two main capabilities: awareness and learning. The CR enhances its awareness of the radio environment by sensing, analyzing, and then reacting with an appropriate adaptation that will optimize the CR's goal. In the next paragraphs, we will focus our attention in the *Sensing* module, which is an integral part to the awareness capability. Recalling from the CR's cognition cycle in Figure 2.1, the *Observation* process begins the CR cycle, sensing of the surrounding radio environment is one of the first steps in the *Observation* phase. Figure 4.2 shows the basic interactions of this module.

There are several signal processing techniques for spectrum sensing including: Matched Filter Detection, Energy Detection and Cyclostationary Feature Detection [7]. Researchers at Wireless@VT have focused on the last one, proposing detection of very low SNR signals using the cycle frequency domain profile (CDP) and Hidden Markov Models (HMM) for the preprocessing and classification of these signals [94–96].

In 3G wireless networks, there are two scenarios that require sensing of the spectrum:

1. The CR is a primary user and it is sensing the environment to determine the interference level at its location.

2. The CR is a secondary user and its accessing the 3G frequency band opportunistically, thus sensing is required to avoid interference to primary users.

Researchers have shown that although the cellular bands experience heavy traffic during peak hours (mornings, evenings) the rest of the time this spectrum sits underutilized. CR can enable efficient use of the spectrum in these bands and increase utilization [27]. In B3G wireless networks, with the convergence of wireless communications and IP-based networks awareness of the radio environment is a must. Many researches have suggested the term "cognitive networks" when referring to B3G networks. Sensing of the spectrum is presumed as heterogeneous platforms (i.e. 2G, 2.5G, 3G, 4G, Wi-LAN, Wi-Max, etc.) will be accessing the same spectrum and sharing cooperatively the radio resources.

In this work, when we refer to the CR's ability of "sensing", it is implied that the spectrum in the vicinity of the cognitive radio is scanned, and the existence of possible interfering signals is determined. Classification of the signals can be performed either by the *Sensing* module or by the *Environment Analyzer*. In some cases, some preprocessing and feature extraction of the signal is performed in the UE, but if the methods used for classification are intensive on computation and memory requirements, it is advisable to locate these processes in the RNC. Furthermore, when we refer to "awareness" in the CR context we refer to the radio's capability to adapt to its environment knowing its computing context, the user context and the physical context as described by Ilyas and Mahgoub [97]. In our implementation, since the focus is in the design of the cognitive engine rather than the sensing mechanisms, we assume that the *Sensing* module works as desired, with 100% accuracy in detecting and classifying signals.

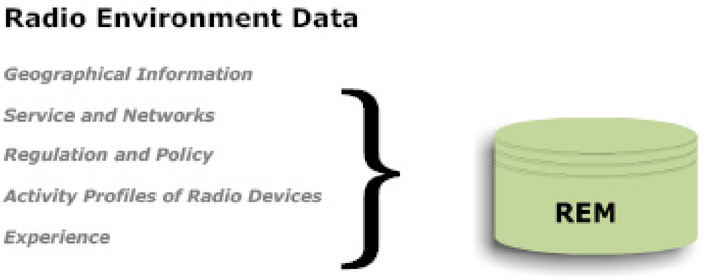


Figure 4.3: Radio Environment Map: Scenario Characterization

Radio Environment Map

The Radio Environment Map (REM) is an integrated spatiotemporal database that includes the cognitive radio's environment information such as: geographical features, regulation, policy, radio equipment capability profile, radio frequency emissions among others [98]. Figure 4.3 depicts how the REM supports the CR functionality by characterizing the radio scenario [91].

The REM can be also be viewed as an extension to the available resource map (ARM) proposed by Krenik [99, 100]. Furthermore, the REM can be employed to support cognitive functionality such as awareness, reasoning and planning. This network support can be realized by the global REM, generally located in the Core Network, and the local REMs which are located in Node B and the UE, respectively. The purpose is to reduce the memory footprint and the communication overhead. Figure 4.4 shows the dissemination of information throughout the network via the REMs.

Extensive investigations have been performed by Wireless@VT researchers on this topic, for further details the reader can refer to the published literature [90–93, 98]. In this work, although we do not implement a spatiotemporal database for the radio environment data, we assume that the cognitive engine has access to this data, locally and globally, and uses the available information along with the sensed information to be aware of its current radio

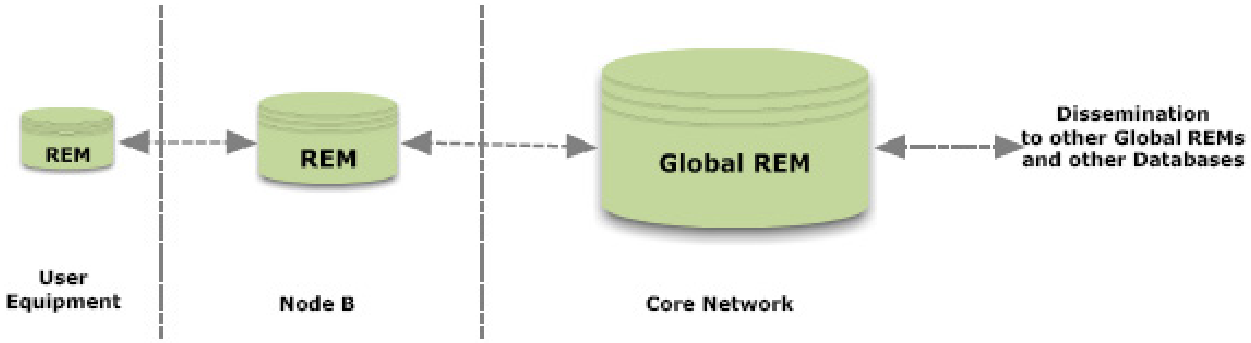


Figure 4.4: Radio Environment Map: Information Dissemination

environment. Also, the information located in the REM is used by the cognitive engine in the decision process. Furthermore, in our implementation, the REM information is readily available and there is no need for retrieval of this particular information, thus we are not considering additional time and memory requirements for the REM implementation.

Environment Analyzer

The *Environment Analyzer* synthesizes and abstracts the information gathered by the *Sensing* module and the REM. It is the module that fuses all the data, and classifies it into a problem of a certain type (handover, coverage, fault detection) or an optimization task for the *Core Agent*. Figure 4.5 shows the inputs and output of this module. The *Core Agent* is an intelligent agent that is flexible enough to solve different types of network problems, there is a need for preprocessing of the collected data from the available sources (user, network, radio environment, policy, etc.) for proper problem formulation.

Recalling the CR's cognition cycle in Figure 2.1, the *Environment Analyzer* is part of the *Orientation* phase of the cycle. The *Environment Analyzer* reports to the *Core Agent* the current "state of the world", as well as the current goals in a simplified version such that the *Core Agent* can make a decision on how to react to the current "state of the world". In

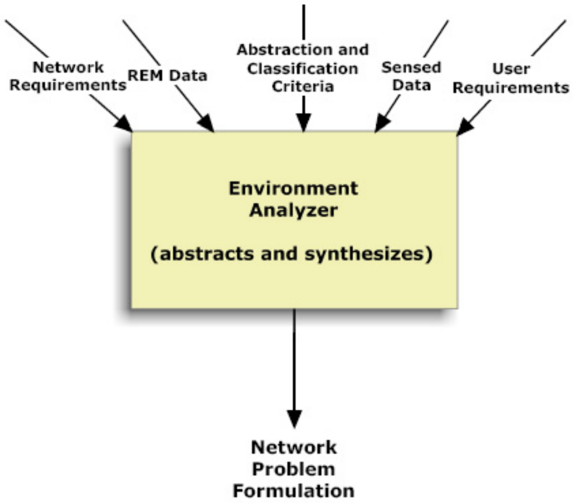

Network Requirements
REM Data
Abstraction and Classification Criteria
Sensed Data
User Requirements
Environment Analyzer
(abstracts and synthesizes)
Network Problem Formulation

Figure 4.5: Environment Analyzer Input/Output Diagram

terms of implementation we can consider the *Environment Analyzer* the data mining process for the REM. The *Environment Analyzer* performs data fusion from different sources (i.e. REM, Sensing Module, CE) to create new raw data that is represented as a new problem formulation.

As discussed in the previous chapter, the CE in 3G wireless networks can be used for several applications such as: scheduling, managing handover, fault detection and prevention, mode/service selection, power amplifier optimization, interference management in femto cells, and a planning tool for layout. The *Environment Analyzer* will consider the available user, network and environment data stored in REM, and the sensed information sent by the *Sensing Module* and formulate a problem for the specific cognitive radio application (i.e. managing handover, handling interference, etc.).

Core Agent

In the last sections we discussed the components that support the CR's awareness functionality, now we focus on the *Core Agent* which supports the cognitive abilities of *learning*

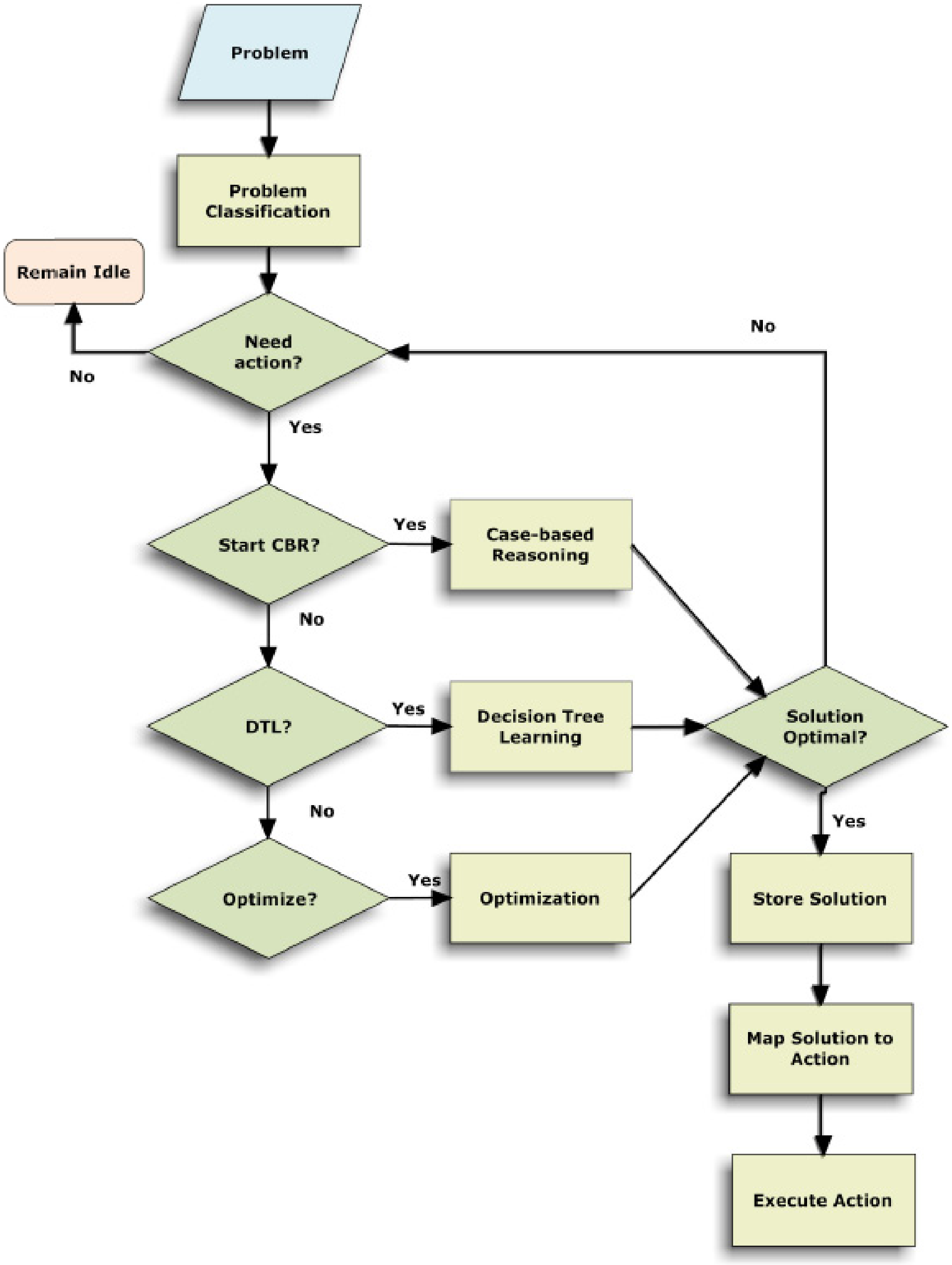


Figure 4.6: Core Agent Architecture

and *reasoning*. Figure 4.6 describes the *Core Agent*'s architecture. The *Core Agent* is an intelligent agent, that manages the cognition tasks of learning, reasoning and optimization, over the network resources in order to improve performance for the cognitive radio user and the network. In AI, an intelligent agent is defined as an entity that perceives and acts in an environment with the objective of determining the best actions that will maximize its performance goals [101]. Computationally speaking, an agent is said to be composed of an architecture and a program. In some cases, it is also desirable for the agent to be autonomous, the agent makes decision based on its own experience rather than on knowledge previously programmed.

If we consider the Intelligent Agent (IA) definition in the context of 3G wireless networks, we have an IA which is a computer program that uses machine learning techniques (i.e. case-based reasoning and decision tree learning) that resides in a powerful computing device (i.e. RNC) and that observes its environment (i.e. *Sensing, REM, Environment Analyzer* outputs) and reacts with the appropriate action that will maximize the CR's and the network's performance goals. In the next chapter, we will discussed the implementation details of our proposed engine.

4.1.3 Cognitive Engine Design for Femtocell Deployments

In a femtocell environment, the CE is integrated into the access point. Therefore the focus is in the use algorithms and artificial intelligence techniques that are computationally simple and reduce processing time. The components are the same as in the generic CE. However, in the case of femtocells the design has to be simplified significantly, as each femtocell deals with an average of five mobile users. In the femtocell CE, the sensing is conducted by the mobile and by the femtocell basestation. The interactions between components is the same, and the data flows in the same manner. It is important to note that the algorithms and processes for the femtocell CE should be computationally efficient. Femtocells do not have the same processing capabilities of macrocells basestations. The algorithms also should be very fast, since latency is a paramount issue in femtocell deployments.

4.2 Summary

In this chapter, the generic cognitive engine used in our simulations was presented. Furthermore, each of the components was described in detail, specifying the inputs and outputs of each component and possible approaches for implementation. The components are discussed in the context of the 3G wireless network. However, the implementation of these components in other environments (i.e. B3G, 2G, femtocells, etc.) may differ from what was described. Suggestions on how to implement these components in other environments were made.

Chapter 5

Cognitive Radio Resource Management

In this chapter, we discuss the concept of cognitive radio resource management (CRRM). We begin with a brief introduction and motivation to replace traditional RRM with cognitive RRM. We proceed by presenting our approach to CRRM and discuss some of the suggested implementations in the literature. Then, we present seven RRM algorithms and how by adding cognition the performance can be improved. Finally, we discuss three cases that are the main topics of the following chapters: using cognition to improve coverage, using cognition to perform handover management, and using cognition to determine policy event patterns.

5.1 Motivation

CRRM refers to the implementation of cognitive algorithms in the allocation of radio resources. The cognitive radio resource manager not only optimizes the use of the spectrum resources, but also can manage call admission, network and link capacity, network and cell load, packet scheduling, antennas, handover, and power, among many other resources. Recent trends in wireless communications are encouraging the use of cognition and intelligence in the design and management of wireless networks. First and foremost, is the migration from centralized, homogeneous but disjoint networks that do not allow for cooperation; to decentralized, heterogenous and cooperative networks. The idea is to converge current disjoint networks into a heterogenous network that optimally combines the different radio access technologies (RAT) under a global infrastructure known as B3G networks. Second, the demand for data applications, higher throughput, and user requirements for communication anytime, anywhere is contributing to the convergence to cognitive networks. Third, research efforts during the last decade in dynamic spectrum access techniques have lead us to apply cognition to other network resource problems, such as radio resource management. Furthermore, reconfigurability (cognition and adaptability) is foreseen to overcome some of the issues that arise from the management of heterogenous networks [16]. Furthermore, the cognitive approach to radio resource management exploits the network's awareness, previous knowledge, learning capabilities, and the ability to perform cross-layer optimization to achieve fair and efficient allocation of the radio resources. Thus, the cognitive radio concept is translated to a cognitive radio resource manager. In the next section, we discuss our approach in the design of a cognitive radio resource manager.

5.2 Our approach to CRRM

In this section we discuss our vision for the cognitive radio resource manager. In this work, the cognitive radio resource manager is defined as:

"An intelligent agent, that manages resource allocation tasks using learning, reasoning and optimization in order to improve overall performance for the network and for the individual link."

We follow the approach to adding cognition in wireless networks discussed in Chapter 3, for

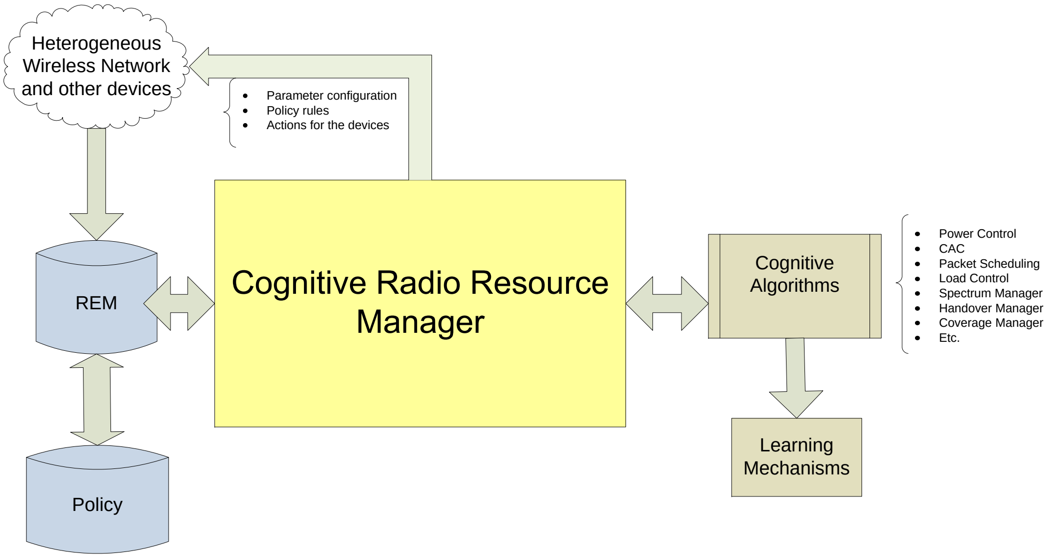


Figure 5.1: The Cognitive Radio Resource Manager

the design of the cognitive radio resource manager. However, we foresee the manager as able to tackle different network problems and able select various machine learning and artificial intelligence techniques in order to achieve optimal solutions and improve performance. Figure 5.1 depicts the concept.

The CRRM acts as the *Core Agent* in our generic cognitive engine. The CRRM architecture is distributed, with entities in each cell (Node B), a local version in each mobile equipment and a global entity located in the network's core. The CRRM is the decision maker, it acts and reacts to the stimulus (measurements, alerts, etc.) received from the network, from the individual links, and from a set of predefined objectives (i.e. user, network, operator, policy, etc.). Predefined objectives can be located in the global *REM* and local *REM* for the mobile equipment. The CRRM also has access to measurements and configuration information across the various layers in the network, and thus is able to optimize across the various layers.

Also, the CRRM is able to identify and synthesize the current network problem and select the type of cognitive algorithm to use. The CRRM can select among several algorithms including: Call Admission Control (CAC), packet scheduling, load control, power control, resource manager, handover control and coverage manager. The power control, handover control and coverage control can be distributed and included in cells (i.e. basestations, access points, femtocells, etc.). The power control and handover control algorithms are also distributed and included in the mobile equipment. The CRRM has access to a selection of machine learning methods, artificial intelligence techniques, optimization techniques, and heuristics. The CRRM is able to select the optimal technique given the problem formulation and representation. As an example, if the CRRM is encountering a handover problem in a cell, or a link, the CRRM can invoke the handover control algorithm that can solve the handover problem. In the next section, we focus on discussing the cognitive algorithms that are integral part of the CRRM.

5.3 CRRM Algorithms

In this section, we present seven applications for the Cognitive Radio Resource Manager, by no means this list comprises of all the radio resource management applications. We believe, that there are many aspects of the network management that can improve with the addition of cognition. The applications discussed here have been incorporated in traditional radio resource management, however we discuss how cognition can improve performance or what cognitive algorithms can be implemented.

5.3.1 Call Admission Control

CAC is the process of managing the arriving traffic (at the call, session or connection level) based on some established criteria [60]. Generally, CAC is applied as an algorithm that admits or rejects arriving users to optimize an objective function, while guaranteeing the QoS for arriving users, as well, as current users. There are several approaches to CAC including: admission based on a threshold, admission to guard channels, pricing-based admission, non-cooperative admission, among others. Two other approaches that exhibit cognition have been found in the literature: cooperative admission control and mobility-based admission control.

In collaborative CAC, neighboring basestations share information on the networks's condition such that resources can be allocated in advance, and as accurate as possible. In mobility-based admission control, the position and mobility patterns of the user can be employed to allocate the resources in advance to guarantee admission. Skehill et al. [102] presented a common radio resource management approach for call admission in heterogeneous networks. The approach selects the most appropriate wireless access system based on the service type, user preferences and the network load. A cognitive CAC algorithm interacts with other RRM algorithms (i.e. load control, handover control, power control, etc.) via the Cognitive Radio Resource Manager. The objective is to optimize admission goals while considering the network goals, and other radio resource management goals. Awareness of the network's admission control goals, as well as, neighboring basestations and/or neighboring systems' (i.e. in a heterogeneous network) admission goals can improve the control of admissions.

5.3.2 Packet Scheduler

Packet scheduling refers to the scheduling algorithm that manages bandwidth and monitors the importance of data packets; and depending upon the priority of the packet, gives it higher or lower priority or bandwidth levels. Packet scheduling is necessary for packet-based services in order to provide adequate QoS levels to the user (delay, delay jitter, amount of bandwidth, fairness), since many packets compete for a common outgoing link. In UMTS networks, there are four type of service classes as mentioned in Section 2.4.2. The conversational class (i.e. voice and teleconferencing) is transmitted without scheduling on a dedicated channel, the background and interactive classes have no guaranteed bit rates therefore are transmitted without a packet scheduler. The streaming class requires minimum bit rates and tolerates some delay therefore a packet scheduler is used.

As we move to B3G networks, where an all-IP architecture is envisioned, and traditional circuit-switched services such as voice and teleconferencing will be packet-based; there is a greater need for a packet scheduler and prioritization schemes. A cognitive packet scheduling algorithm can prioritize packets per service class, but also prioritization schemes based on cost or user preferences. An adaptive scheduling algorithm for cognitive radio was proposed by Li et al. [103], the algorithm exploited QoS and spectrum awareness in the cognitive radio to apply an adaptive algorithm that resulted in better QoS, higher system capacity and improved spectrum utilization, when compared to the traditional packet scheduler.

5.3.3 Load Control

Load control is generally perform to guarantee that the system is not overloaded and remains stable. If the system has been properly designed and the admission control and the packet scheduling algorithms work well, an overload in the system should be the exception. If an overload occurs then the load control algorithm returns the system back to the targeted load by performing several tasks, including: denial of downlink power-ups commands, reducing the target SINR used by the uplink fast power control, reducing the data throughput, performing handover to another cell or to another system and dropping low priority calls. The load control actions are performed at the cell level, and at the network level. A cognitive load control algorithm can use history to foresee overload situations such as during peak hours, or special events and start performing load control tasks to avoid overloads. The algorithm can interact with other RRM algorithms such as: CAC, packet scheduler, power control, etc. to minimize overloads. Intelligent load control algorithms have been suggested in the literature for wired networks [104, 105].

5.3.4 Power Control

Power control comprises of the algorithms used to manage and adjust the transmit power of basestations and mobiles [60]. Power control has several objectives including: reducing co-channel interference, maximizing cell capacity, minimizing the mobile's transmit power and hence power consumption, and managing data quality. There are two major effects that decrease the performance of wireless networks: the time varying nature of the wireless channel and interference. Power control is used to mitigate these effects by keeping SINR levels above the *minimum protection ratio* in order to provide an adequate communication link. There is a tradeoff in the power control problem that one should consider, *since increasing a link's power, will increase that link's SINR but in turn increase the interference to other links.* This power control could, under certain circumstances, become unstable.

Other constraints to the power control algorithm include:

- The algorithm should be a distributed one for the uplink, so that each link is responsible to use the limited resources cooperatively while maintaining QoS levels.
- The algorithm should be computationally simple. Limiting the overhead in the communications protocol.
- The convergence speed for the power-control algorithm should be faster than the changing speed of the fading channels.
- The algorithm should allow for heterogeneous QoS requirements.

In recent years, power control algorithms have been studied extensively. When a search with the key words *power control* and *cognitive radio* is performed on IEEE Xplorer over a hundred items are returned. Cognitive power control algorithms can help improve link quality, reliability and limit interference. Awareness and the ability to sense the environment can help the node assess its link's current condition, as well as, neighboring nodes's links and adjust its transmit power to optimize over its own link quality objective, neighbors and the network as a whole.

5.3.5 Spectrum Manager

The management of spectrum resources is probably the most influential aspect of RRM in the cognitive radio field. The search for efficient and effective spectrum management schemes has encouraged research in multi-disciplinary areas by academia, industry, manufacturers, government agencies, regulators and many others. Currently, spectrum rights are assigned very similarly to real-estate rights. Primary users have “property” rights to their assigned spectrum, thus no sharing is allowed. New spectrum sharing policies and technologies are being researched to exploit the unused spectrum. One of the solutions to the spectrum management problem is the cognitive radio. A radio that is able to use spectrum resources opportunistically and more efficiently, increasing the spectral efficiency of the systems in terms of bits/Hz. In our cognitive radio resource management approach, spectrum resource management is one seven applications. A cognitive spectrum management algorithm is aided by sensing and awareness in the mobile equipment (radio), and the knowledge and history stored in REM to optimized the use of the spectrum. The algorithm is distributed and located in the mobile, basestation and at the network’s core.

5.3.6 Handover Control

Handover control is the algorithm that transfers an active session (i.e. voice call or data session) from one cell to another as the user moves in the coverage area of the wireless network [106]. Handovers are performed in a cellular system if the mobile user has moved out of range from the basestation, or if the basestation serving the mobile user is full and needs to transfer the user to another nearby station, or if the current basestation is not able to provide a service requested by the mobile user. The handover control process is divided in three stages, these include: *initiation, new connection generation and data flow control* [107]. In recent years, handover control has received significant attention. It is also part of a broader concept known as *Mobility Management.* Mobility management comprises of two tasks: *Location Management* and *Handover Management.*

A cognitive handover algorithm can exploit previous knowledge and current information obtained from sensing the radio environment to predict when a handover is needed. Also, it can predict and allocate resources in advance. As an example, a cognitive handover algorithm can be helpful in a femtocell environment, when a femtocell user needs to handover from its femtocell to the macrocell, resources can be allocated in advance in order to minimize latency. An extensive survey of handover control is presented in Chapter 7. Also, we introduce a decision tree-based cognitive handover algorithm that uses previous information, mobility management, and prediction to perform handover control.

5.3.7 Coverage Manager

Although coverage management is not typically performed in traditional radio resource management functions, as we move to distributed RRM algorithms we can foresee performance improvements in coverage management if cells can manage and optimize their footprint. Generally, the coverage of a cell is determined during the initial design stages in the network system design process. RF coverage design along with network capacity is calculated based on the operator inputs such as: subscriber forecast, usage forecast per traffic type, coverage areas, type of coverage, available spectrum, QoS requirements, propagation characteristics, environment type, topography, etc. [59]. Once deployed, the actual coverage is determined using field measurements.

Assuming that reconfigurable antennas are available, a cognitive coverage algorithm can use current coverage information reported by the mobiles to adjust the cell's antenna characteristics in order to improve coverage. Also, a cognitive coverage algorithm can interact with other RRM algorithms such as power control, handover control, load control to maintain service requirements in a coverage gap. In Chapter 6, we propose a cognitive algorithm that determines the actual coverage of a cell based solely on reported observations from the mobiles. Such algorithm, can be very useful in self-organized networks, femtocell deployments, etc., where little human intervention is desired. The cell will adjust its coverage in order to maintain the user's QoS requirements. Recent literature shows some promising results in this area. Di Taranto et al. propose simple antenna pattern switching to minimize interference in cognitive networks [108]. Furthermore, Sayeed and Raghavan analyze the impact of reconfigurable antennas for dynamic spectrum access in cognitive networks [109].

5.4 Case Studies

In this section, we present three cases where the concept of CRRM has been applied. First, we present how to improve coverage knowledge in a network by adding cognition and learning capabilities to the network and the mobile equipment. Second, we present how to use cognition for handover management. Lastly, we present how cognition can be used to determine policy patterns. These cases are the main topics in Chapters 6, 7, and 8 respectively.

5.4.1 Case: Using Cognition to Improve Coverage

In this section, we present our first example on how to use awareness, previous knowledge, and machine learning to learn the coverage of the cell based solely on observations. Traditionally, the RF coverage of a cell can be determined by employing several well known methods. First, operators can get a rough idea of the RF coverage by performing a mathematical analysis. They may start doing a link budget and determining the radius of the cell. Then, they can use stochastic models of the channel and simulation to get an estimate of the coverage given their description of the site such as: the type of environment (i.e. urban, suburban, rural, indoor, etc.), and the characteristics of their system (i.e. maximum transmitted power for the DL, UL, etc.). Also, they may employ advanced simulators that include topographical

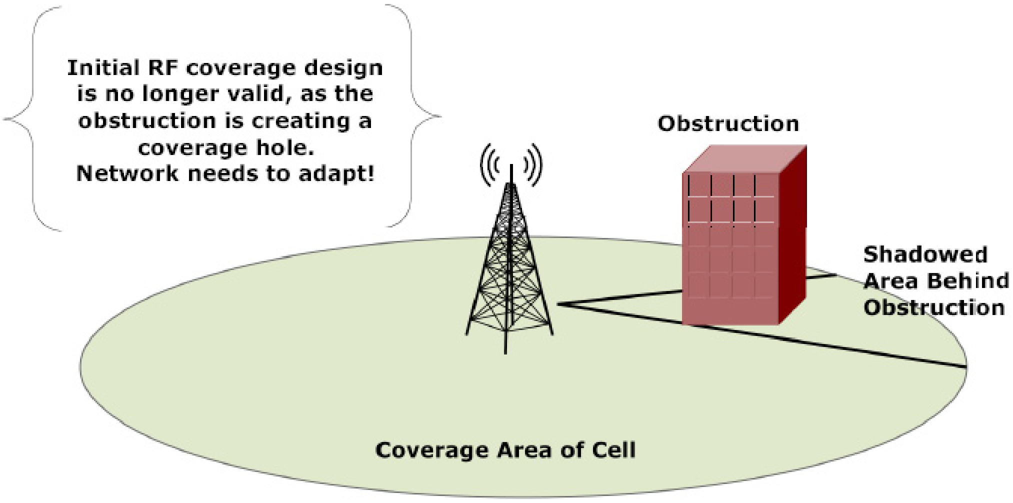

Initial RF coverage design is no longer valid, as the obstruction is creating a coverage hole. Network needs to adapt!
Obstruction
Shadowed Area Behind Obstruction
Coverage Area of Cell

Figure 5.2: Typical Coverage Problem in 3G Networks

information to get even more accurate predictions of the cell's coverage. Finally, they send their RF engineers to do field measurements and drive-by tests. After, their system is deployed mobile customers may report coverage gaps by calling the operator's customer service.

If we examine a cognitive network, we add cognition to the current mobile equipment, by this we mean the ability to sense the environment, the ability to know where it is (i.e. GPS) and knowledge of what has happened before (i.e. memory). We exchange the current network for a cognitive network, the system will be able to acquire new knowledge based on previous experiences and determine the coverage of the cell as it is changing. To maintain processing costs at a minimum, this process can be done offline everyday or as dictated by the operator's QoS requirements. In this case, cognition can help improve the network's coverage and assist in RF planning as discussed in Chapter 2. Chapter 6 discusses this example in detail.

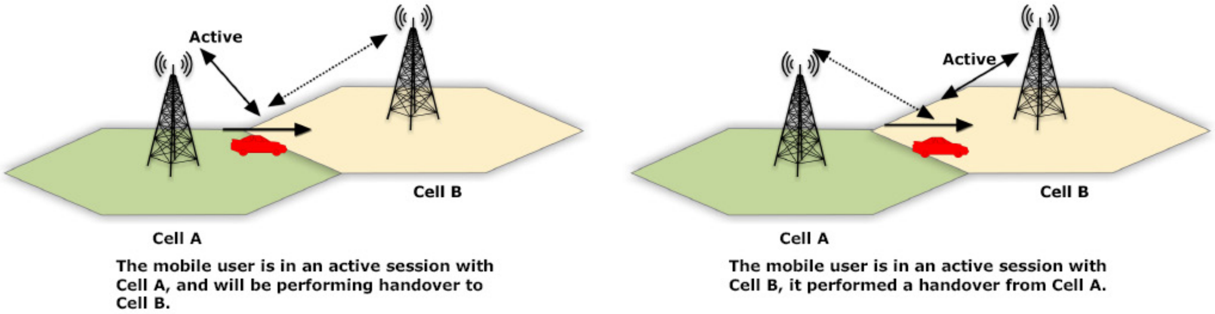

Active
Cell A
Cell B
The mobile user is in an active session with Cell A, and will be performing handover to Cell B.
Active
Cell A
Cell B
The mobile user is in an active session with Cell B, it performed a handover from Cell A.

Figure 5.3: The Handover Concept

5.4.2 Case: Using Cognition to Manage Handover

In this section, we extend our original example to use the learned knowledge for the engine and define a set of actions. The performance task involved is to predict the need for handover using previous knowledge and the mobility pattern of the mobile user. Figure 5.3 shows a typical coverage problem for the end-user in a wireless network. Let's consider the following:

- The mobile user is moving from cell A to cell B.
- The mobile user moves at a velocity v in a specific direction.
- The call is maintained by cell A and given the direction of the user it will be handover to cell B.
- The call is dropped before an effective handover to cell B.
- The user needs to start a new call.

This scenario repeats every time the mobile user travels in that direction from cell A to cell B.

Today, with current technology the mobile user deals with these coverage holes. The mobile user may anticipate that the call is going to be dropped based on the previous experience, and soon and may:

- Stop moving and finish the call before reaching the coverage hole.
- Continue talking until the call is dropped and then start a new call.
- The mobile user can terminate the call and start a new call once she passes the coverage hole.

If we substitute the standard handset, for an enhanced cognitive handset and a cognitive base station we can expect the following:

- the CR will have the previous information on similar events,
- the CR will be able to predict that the call may be dropped given the time of day, the geographical location of the user, the velocity, the bearing of the user, etc.,
- the CR may increase power in order to complete the handover,
- the CR may handover to cell B before the quality of the call degrades,
- the CR may perform an inter-frequency handover,
- the CR may perform an inter-system handover (i.e. other RAT available).

In order to address the aforementioned problem, we develop a cognitive engine that uses a case-based reasoning approach and decision-tree searches for its learning and decision making. The engine exploits the user's previous information and prevents call drops, thus increasing coverage in the 3G wireless system. This in turn increases the quality of service to the end-user. The cognitive engine design will follow the proposed approach in Chapter 3. This type of engine can reside in the cognitive resource manager and be used as needed.

The handover can be horizontal or vertical, as we move to B3G wireless networks similar services will be provided by the various communication systems. This case is the main topic of Chapter 7.

5.4.3 Case: Using Cognition to Determine Policy Patterns

Another possible application for the hybrid cognitive engine is to determine policy patterns from observations. The learning problem can be formulated differently, such that the predicted variable is whether a policy event occurs or not. For example, let's examine a 3G network, where loading exceeds the maximum load allowed, this happens in a specific pattern, (i.e. workdays from 5-7PM). During this event, cell breathing occurs. Cell breathing is when the coverage of the cell decreases due to the noise floor of the cell increasing. The mobile nodes react by increasing the power in order to maintain a call, thus in turn, affecting the coverage of the cell.

If the network operator implements a policy event that during that length of time in order to limit this effect. The operator may set the transmission power to a lower level in order to avoid the cell breathing problem. However, this policy event reduces the cell's coverage footprint. Figure 5.4 depicts these events. The policy classification algorithm can extract the event from observations and then higher layers of the engine process can interpret the event and define some actions to deal with the event. In Chapter 8 of this work, we will explore this case study in more detail.

5.5 Summary

This chapter presented the concept of cognitive radio resource management and its application. Also, it introduced the three case studies addressed in this dissertation using the proposed approach and hybrid cognitive engine. The idea was to connect the concepts de-

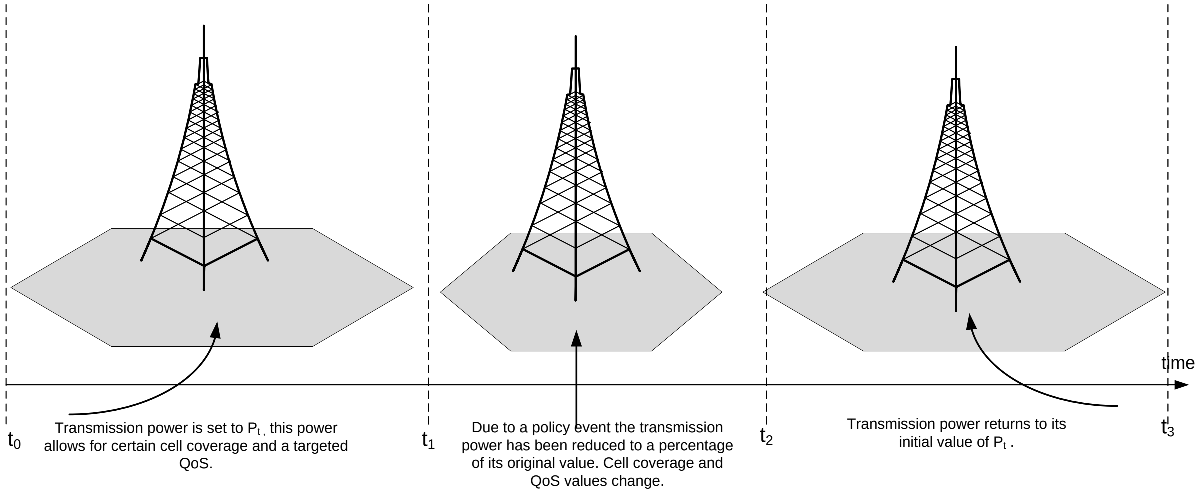

time
t0
t1
t2
t3
Transmission power is set to Pt , this power allows for certain cell coverage and a targeted QoS.
Due to a policy event the transmission power has been reduced to a percentage of its original value. Cell coverage and QoS values change.
Transmission power returns to its initial value of Pt .

Figure 5.4: Opportunistic use of primary channels

fined in Chapter 2, to the methodology presented in Chapter 3 and the applications discussed here in this chapter. In the next sections, we will focus on implementing cognitive solutions to these problems.

Chapter 6

Using Cognition to Improve Coverage

In this chapter, we present the first case study of the proposed cognitive engine. We begin with an overview of the system, a description of the network problem that the engine is trying to solve, and the models used in the simulation. We apply the engine to a 3G cognitive radio application: learning the coverage of the basestation, based on environment observations. We discuss obtained results and suggest ways that the acquired knowledge can be used to develop algorithms that can improve network performance. We also discuss how to measure the cognitive engine's performance.

6.1 Problem Background

One of the most important objectives in wireless network design is radio network planning, which includes coverage, capacity and network optimization. This process is generally done in the dimensioning stage of the design process. RF engineers approximate the number of sites, base stations, and their configurations based on the operator's requirements and the radio propagation characteristics of the area [58]. Capacity and coverage are closely related in WCDMA system, thus often are considered simultaneously in the dimensioning stage. As explained in the previous chapter, there are several methods to determine the coverage of a cell from mathematical analysis to field measurements for proper coverage estimation. In this chapter, we present how adding cognition can increase the network's and the mobile's knowledge on the actual coverage.

6.2 Problem Description: Using Cognition to Improve Coverage

In this section, we discuss how to apply cognition to improve the coverage of a cell. First, we describe the objectives of our proposed cognitive engine. Then, we compare and contrast the approach of improving coverage in a traditional network versus a cognitive network. We continue by describing the system's models, and discussing the simulation parameters. We present simulation results and evaluate our learning algorithm using several methods. Finally, we conclude the chapter with a summary.

6.2.1 Objective

The objective of the coverage learning engine (CLE) is to exploit the cognitive capabilities of the mobile and the network, to learn the coverage of the cell. We apply our approach to develop the cognitive engine, formulate the coverage problem as a *Classification and Regression* problem, and use a decision tree algorithm to predict whether the signal level is adequate to sustain a call. We then evaluate the engine's performance to measure improvements.

Recalling the cognition cycle in Figure 2.1 we can describe the cognitive process in terms of the coverage learning problem as follows:

- Observe - The cognitive mobile and the cognitive network observe the surrounding radio environment. In this phase, measurements such as: received signal strength, signal-to-interference ratio, etc. are taken. Also, location and mobility information is collected.

- Orient - In this phase, the cognitive engine (network, mobile or both) fuses the new information collected, with previous knowledge such as: geographical information, mobility patterns, handover history, user requirements, terminal capabilities, etc., and creates an interpretation of the current handover situation. In this step, the current state of the world is determined, and the problem is formulated in a way that the engine is able solve.
- Decide - In this phase, the cognitive engine evaluates the current network situation and makes a decision whether a call is going to be dropped or not. The engine classifies the observations and induces rules (new knowledge) regarding the actual observed coverage.
- Act - In this first example, there are no immediate actions for the engine. The engine derives patterns in the data collected. Later, these patterns and general domain knowledge are used to induce rules on the overall behavior of the system. The engine's goal is to learn the coverage based on observations. In the next chapter, we will see how adding a set of actions (handover options) improves the performance of the network.
- Learn - The system learns a coverage map of the cell, it classifies the regions based on the probability that a call is dropped. The amount of information regarding the coverage of the cell is increased.

6.2.2 System Overview

The overall system under investigation is composed of three modules: the network environment simulator, the Coverage Learning Engine (CLE), and a post processor used to interpret the results obtained by the cognitive engine. Figure 6.1 describes the interactions between these three modules. The network environment simulator, developed using MATLAB® [110],

Table 6.1: Case for CLE

Attributes	Type
x	numeric
y	numeric
R	numeric
SINR	numeric
Call Drop	boolean

was designed to generate cases for our cognitive engine to analyze. The input to the module are the simulation parameters and the output is a *.csv file with the cases. If data from a real system were available, the environment simulator can be replaced by a data mining process that extracts the necessary information to formulate the problem. The Coverage Learning Engine is based on Wireless@VT generic cognitive engine, however, decision tree learning is used for the reasoning and learning. The DTL algorithm used is C4.5 as described in Chapter 3. A post-processor was created in MATLAB® to interpret and present results graphically.

To design the engine we followed the proposed approach discussed in Chapter 3, the steps are outlined as follows:

- *Formulation* - the learning problem has been formulated as a *Classification and Regression* problem. The CLE uses previous observations in the form of cases to determine the expected SINR level at a given location.
- *Representation* - the data is represented in cases. Each case is constructed as in Table 6.1.
- *Collection of Data* - the cases have been generated using the Network Environment Simulator (described in Section 5.3 and a Random Waypoint (RWP) mobility generator.
- *Evaluation* - we train the CLE using a training set for each experiment. Then, the CLE is applied to a new data set. The CLE induces rules and a decision tree on the

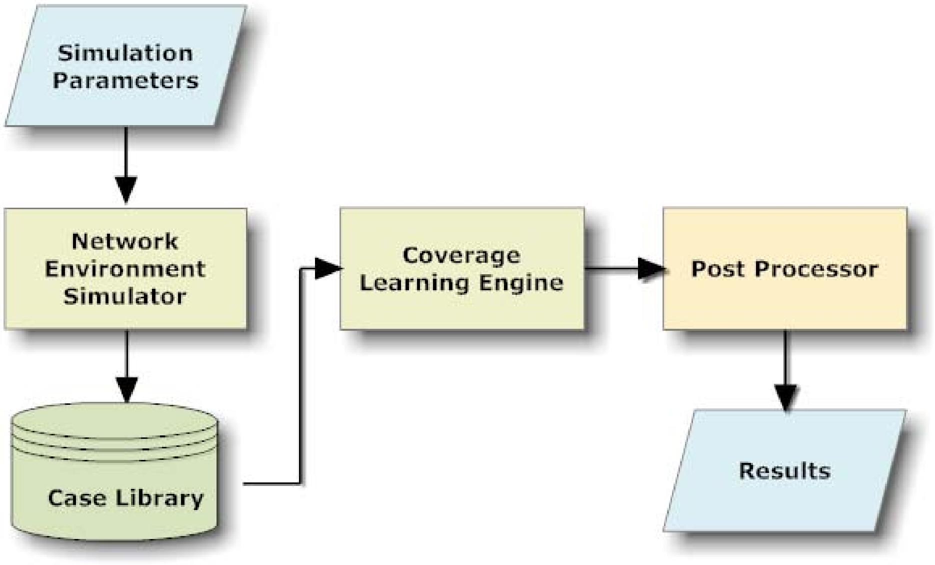


Figure 6.1: System Overview: Module Interactions

coverage of the cell, based solely on observations. The knowledge on the cell's coverage is increased.

- *Application* - the current application is to determine the exact coverage of the cell. However, this learned knowledge can be fielded to other applications such as Handover Management (i.e. Chapter 7), for power control algorithms, interference management, etc.

Traditional Network

In a traditional network, an operator can improve the network's coverage and increase capacity doing several things:

- deploying more cells,
- reducing the cell's footprint, and
- decreasing the distance between the transmitter and the receiver.

The last two also require the deployment of more cells in order to provide the same coverage, but in turn by reducing the cell's footprint and decreasing the distance between the transmitter and receiver, the coverage is reduced. Hence, the goals for increasing coverage and capacity are conflicting. There is a tradeoff between the deployment of more cells, increasing coverage and increasing capacity.

Cognitive Network

A cognitive network is aware of its own coverage and capacity, and ideally can reconfigure itself to compensate for coverage holes. One way is by collecting field measurements from the mobiles and creating an updated coverage map. Since the coverage in a cell can change dynamically due to the time variant nature of the wireless channel and susceptibility to various propagation phenomena (i.e. reflection, diffraction, scattering, etc.), the network can periodically update its coverage map. The cognitive network may react to the lack of coverage in several ways:

- the network may force a handover to another cell (horizontal) or another communication system (vertical),
- the network may do power control in order to maintain the user's QoS,
- the network may reconfigure its antennas' radiating pattern in order to cover the hole, and lastly,
- the network may alert the operator of the coverage problem.

The algorithm presented in this section has been design for a cognitive network, however legacy networks that do not allow cognition in all layers of the communication stack, can still benefit from the use of cognitive algorithms that use learning and previous experiences to solve new problems [54].

6.3 Network Environment Simulator

Although our generic cognitive engine is flexible enough that it can be used in any wireless networks, we needed to provide context in order to evaluate its performance. The 3G wireless network is the environment in which our engine, an intelligent agent, makes observations, analyzes the current situation, and then reacts with the appropriate action. We selected the 3G wireless networks since these networks have been largely studied, analyzed and optimized. Still, we think there is room for improvement, and even the simplest levels of cognition can tackle typical problems encountered in these networks.

Among the four platforms supported in the 3G wireless system, we have chosen the WCDMA FDD solution as our engine's environment. WCDMA has a bandwidth of 5 MHz, with a reuse ratio of 1. There are 4 services classes: conversational, interactive, streaming and background. Each of these services has a corresponding data rate and QoS requirements. Speech can still be a circuit switched service, thus the mobile user may have assigned radio resources in a call. There are three testing environments: indoor, pedestrian and vehicular. We have selected the vehicular environment which is characterized by macro-cells and high transmit power. It is important to note that the gains for the WCDMA fast power control are negligible for this test environment as the UE's speed is too high. In the next paragraphs, we describe the models used to characterized the 3G wireless network environment.

6.3.1 Propagation Model

The International Telecommunication Union (ITU) has recommended in their specification ITU-R M.1225 [111] several reference scenarios and propagation models for the IMT-2000 standard. Since, these models have been widely used by international standards organizations, we feel that it is best if we follow these recommendations in our simulation. In the vehicular environment, the propagation model is composed of the mean path loss, the slow variation around the mean due to shadowing and scattering, and the rapid variation in the signal due to multipath. In the next paragraphs we discuss the models used for the propagation phenomena in the WCDMA environment.

Path Loss

The path loss model for the vehicular environment in Equation 6.1 can be used in urban and suburban areas, outside the high rise core, where buildings have similar heights.

$$PL = 40\,(1\,-\,4\,\times 10^{-3}\,\Delta h_b)\log_{10} r\,-\,18\log_{10}\Delta h_b\,+21\,\log_{10} fc\,+\,80 \quad dB \tag{6.1}$$

where r is the distance between Node B and the UE, fc is the carrier frequency (2000 MHz is the default value), and Δh_b is the Node B antenna height, measure above rooftop level. The valid range for Δh_b is between 0 and 50 meters.

Log-normal Fading

The log-normal fading around the mean path loss can be represented as a zero mean Gaussian random variable and a standard deviation between 6 and 12 dB [112]. The ITU-R M.1225 recommended value for the standard deviation in suburban vehicular environments is 10 dB [111]. Thus, we have the following equation to characterize the probability density function of the log-normal shadowing effects:

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma}\exp^{-\frac{(x^2)}{2\sigma^2}} \tag{6.2}$$

where x is received signal level in decibels and σ is the standard deviation.

Table 6.2: Vehicular Tapped Delay Line Model Parameters: Channel A

Tap	Delay (ns)	Power (dB)
1	0	0
2	310	-1
3	710	-9
4	1,090	-10
5	1,730	-15
6	2,510	-20

Fast Fading

In WCDMA, multipath fading is simulated using an N-tapped delay line model. The model is characterized by the number of taps, the time delay relative to the first tap, the average power relative to the first tap, and the Doppler spectrum for each tap. There are two sets of parameters defined in the ITU-R M.1225 for each testing environment, one with relative low average delay spread and one with very high average delay spread. Table 6.2 shows the parameters for Channel A (average low delay spread model), a classic Doppler spectrum is recommended for this model.

The models described in ITU-R M.1225 have been widely used, because they were accepted by international standards organizations [113]. The main advantage that these models provide is uniformity, designers can evaluate all the radio transmission technologies for the IMT-2000 specification under the same propagation conditions. The disadvantages include: small number of taps, and that the models provide big differences in the power delay profile among the different test environment but provide little difference between suburban and metropolitan areas [113].

6.3.2 Spatial Model

In this section, we discuss the underlaying assumptions in the spatial model of the 3G environment simulator. Node B is placed at the center of the coverage area or cell, at a height Δh_b above the rooftop level, this value ranges from 0 to 50 meters, the default value is 25m. The maximum radius of the cell is set by the user, it is convenient to define the maximum radius of the cell (i.e. 3 km) greater than the expected coverage of the cell (i.e. 1 km), in order to obtain the actual cell footprint in the simulation. Node B is aware of the UE's location and velocity. Furthermore, the nodes or UEs are randomly distributed throughout the cell.

6.3.3 Mobility Model

The nodes move around the cell following a Random Waypoint (RW) mobility model. In this mobility model, UEs move along a zigzag path consisting of straight legs form one point to the next. Moreover, the UE moves directly towards the next waypoint at velocity v, once the UE reaches that waypoint the direction of the next waypoint is randomly selected from the uniform distribution over the plane A [114]. Figure 6.2 shows a sample mobility path for an UE.

Although this mobility model is the most popular model in the research community, it may not reflect real mobility of UEs in a 3G network. There are other mobility models that can be used in our simulation such as: the Reference Point Group Mobility (RPGM) model [115], the Freeway Mobility (FW) model, the Manhattan Mobility (MH) model and the Obstacle-based Mobility Model (OM) [116]. In Chapter 7, we present three different experiments with three typical environments and corresponding mobility models.

6.3.4 Node Model

The node is a 3G dual-band mobile, capable of accessing the WCDMA and the GSM platforms. The node has very limited cognitive radio functionality. It is equipped with a local REM that stores some of the UE's local radio environment information. The node is capable of sensing the spectrum and reporting results to the cognitive engine as indicated by the

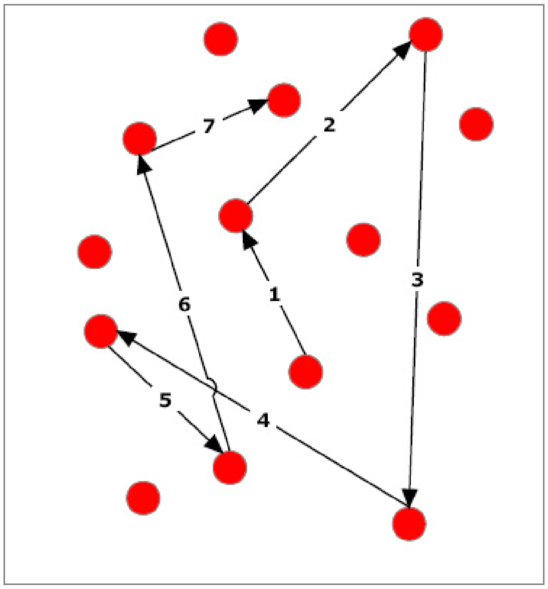


Figure 6.2: Random Waypoint Mobility Path

CE. The node acts as a slave, the cognitive engine that resides in the RNC is the master (intelligent agent). The idea is to keep cost, complexity and memory requirements at a minimum in the UE.

6.3.5 Observation Model

In the simulation, the network and the nodes are sensing the radio environment every second. New position information, as well as the received signal strength is measured every second. The total simulation time is 3,600 seconds (1 hour). There are 100 nodes assigned to each basestation. Thus, the total number of observations by the end of the simulation is 360,000.

6.4 Simulation

In this section, we discuss how the hybrid engine is implemented. We focus on determining the coverage of Node B in 3G wireless networks, thus the engine's modules can be simplified. The *Sensing Module* is simplified in the coverage problem implementation. The UE is the primary user of the spectrum, it shares the band with other users in the cell. Node B's capacity is limited by coverage not by interference. Thus, the coverage of the cell is determined by the propagation characteristics of the environment, including obstructions. The UE is able to determine the level of interference at its location and reports it to Node B. Node B relays the information to the *REM* that resides in the core network where the RNC and the *Core Agent* are located.

In the *REM*, we assume that only the coverage attributes are extracted from the database. The cases are generated by the 3G Wireless Environment Simulator as described in the previous section. Each case is characterized by the attributes listed in Table 6.3. For this problem, the *Environment Analyzer* is not needed, as there is only one type of problem to solve: learning the coverage of Node B. The *Core Agent* is composed of a case-based reasoner that includes the case library, an algorithm for indexing, matching and retrieval. The adaptation is a null adaptation since it is not necessary to modify the cases to determine the cell's coverage map. Some of the methods suggested for the adaptation module include: genetic algorithms, hill climb searches, local searches, and exhaustive search [46,47,117]. The *Core Agent* also includes the C4.5 decision tree learning algorithm. This DTL algorithm is used to classify the training cases and learn the coverage of Node B. In the next section we explain how the engine works.

6.4.1 How the hybrid engine learns the coverage

In order to validate the proposed hybrid engine, an engine that uses the C4.5 DTL algorithm was implemented. The first step was to populate the case library. In this step, we created

Table 6.3: REM Data and Type

Attributes	Type
x	numeric
y	numeric
R	numeric
SINR	numeric
Call Drop	boolean

the history and experience for the CE. If actual data sets were available, this step could be replaced by mining of actual data sets for relevant attributes. The case library is populated as follows: Node B is generated and placed in the center of the cell. Then, 100 UEs are assigned to Node B. There is a maximum transmitting power for the Node B and for the UE. In order to simulate mobility the Random Waypoint mobility model is used. The velocity of the UEs varies from 30 - 100 kmph. The chosen simulation environment is a suburban vehicular test environment where Node B has a high antenna height (15-50m) above the rooftop level, and the geometrical path loss rule is R^{-4}. We followed the International Telecommunications Union (ITU) recommendations described in ITU-R M.1225, and applied Channel A (low delay) parameters to the N-tap delay line channel impulse response [111]. A detailed list of the simulation parameters are included in table 6.4. For this case, 80,000 cases were generated.

Once the case library was populated, we proceeded to apply the decision tree algorithm to learn about our system. C4.5 was applied to the case library, the target value was the SINR. The algorithm classified the cases and derived rules regarding the value of SINR that resulted in a call being dropped. The algorithm uses the concepts of entropy and information gain as discussed in Chapter 3, to generate the tree. For the detailed description of the algorithm the reader can refer to Algorithm 2. Results are discussed in the next sections.

Table 6.4: Simulation Parameters

Parameter	Value
Antenna Height	25 m
Maximum Cell Radius	5 km
Environment	Vehicular Suburban
Channel Model	Channel A
Mobility Model	Random Waypoint
Service	Speech
UE velocity	40-100 kmph
Carrier Frequency	2 GHz

6.5 Results

The testing scenario included an obstruction as seen on Figure 6.3, the propagation behind the obstruction was modeled as a path loss of 100 dB in the shadow of the obstruction, then it gradually goes down to 0 dB at the leading edge of the obstruction. The blue shaded dots in the figure represent the cases where the E_b/N_o is lower than 5 dB and results in a call being dropped. Furthermore, let's say that at the initial stages of deployment the operator designed the coverage area with a 10% probability of a dropped call for the entire cell. Now, the operator can describe more accurately the areas and the probability of dropped calls. The knowledge on the coverage of Node B has increased. In order to classify the coverage, 13 rules where induced. The maximum depth size of the rules is 4. A detailed table is included in the Appendix.

If we recall the approach for applying a machine learning method to a real life problem discussed in Chapter 3, we have completed three of the five steps in the approach. We have formulated the problem, represented the data and collected the training cases. We have acquired new knowledge, thus implying learning. The steps remaining in the approach are: evaluating the learned knowledge and fielding the learned knowledge. In the next sections, we evaluate the engine'e performance and suggest applications of this knowledge to improve network performance in 3G networks.

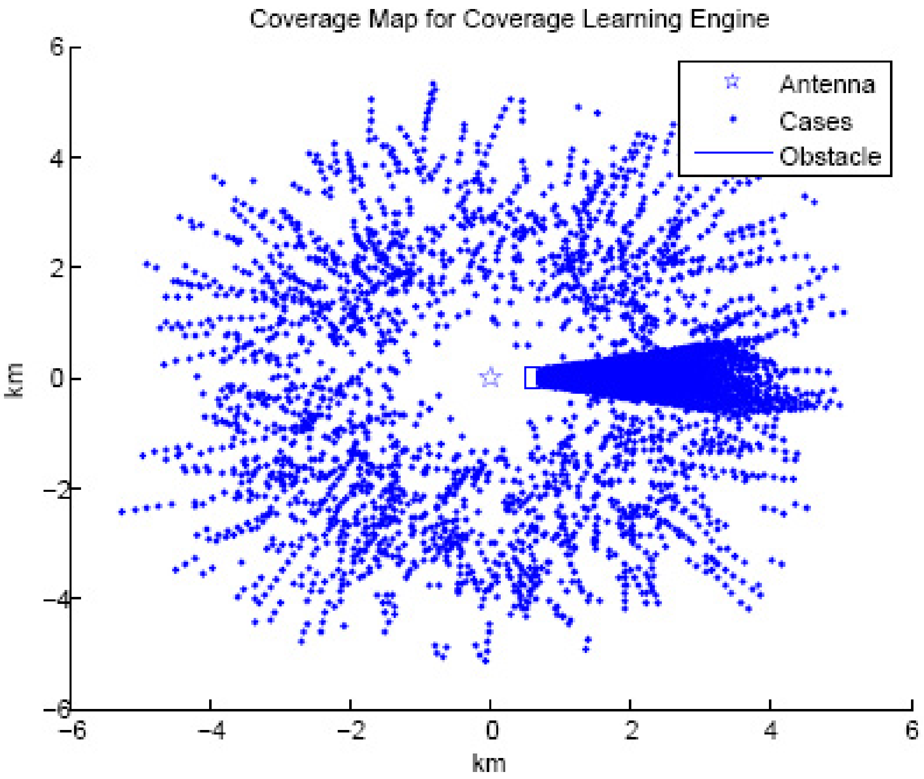


Figure 6.3: Scatter plot of the number of dropped call cases

6.5.1 Evaluating the engine's performance

In this section, we evaluate the engine's performance. We discuss two different methods to evaluate what has been learned: error rate and receiver operating characteristics (ROC) curves. Furthermore, we present a complexity analysis in terms of $O(n)$ for inducing and evaluating the tree.

Error Rate

Evaluating the learned knowledge is a very important step, the learning algorithm is not worth the computational complexity if the learned data is not useful in future cases. The coverage problem was model as a classification problem, thus we can evaluate its performance in terms of the *error rate*. The classifier predicts the class of each case: if it is a correct classification then it is a success, if not, it is an error. The *error rate*, is the ratio between the number of errors occurred and the whole set of cases as seen in Equation 6.3.

$$error\ rate = \frac{number\ of\ errors}{total\ number\ of\ cases} = \epsilon \tag{6.3}$$

Furthermore, in statistics the succession of independent events that are either success or fail is known as a Bernoulli process. If we refer to the set of instances at a particular node, the probability of error (misclassified case) at the node is denoted by γ, and the probability of success is denoted by β, where $\beta + \gamma = 1$. The question that we must answer now, *is how close to the true number of errors is that probability?* This is often expressed as a confidence limit δ on the confidence ζ, thus we have the following equation:

$$\Pr\left[\, \frac{\epsilon - \gamma}{\sqrt{\gamma(1-\gamma)/N}} \; > \; \zeta \,\right] \; = \; \delta \tag{6.4}$$

where N is the total number of instances, ϵ is the observed error rate and γ is the true error

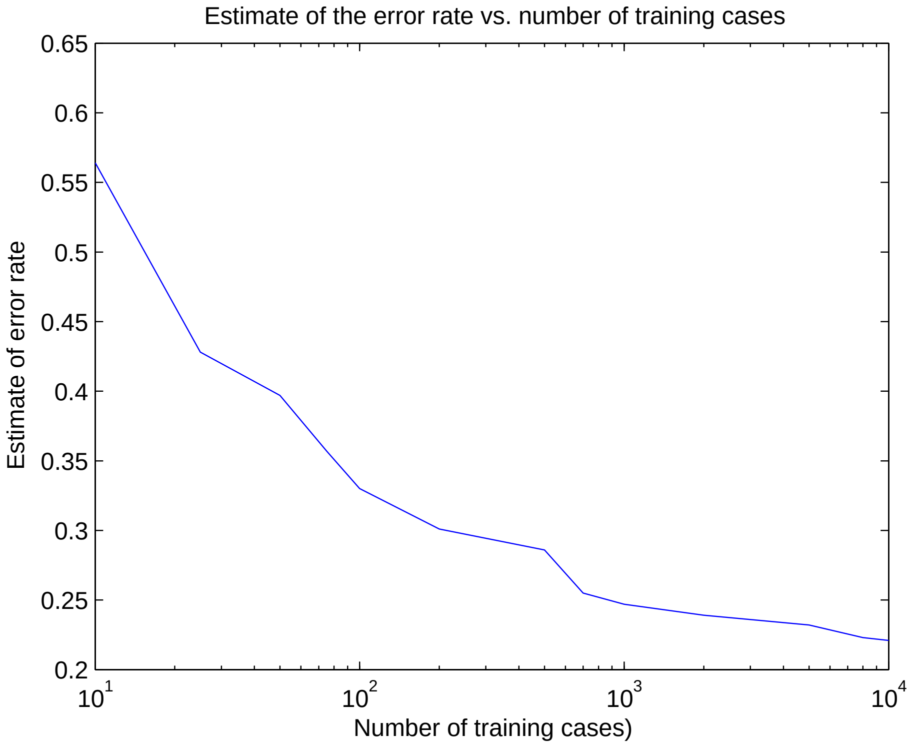


Figure 6.4: Estimate of the error rate vs. number of training cases

rate. Then if the upper confidence limit is used as an estimate for the error rate ϵ we obtain the following equation:

$$\hat{\epsilon} \;=\; \frac{\epsilon + \frac{\zeta^2}{2N} + \zeta\sqrt{\frac{\epsilon}{N} - \frac{\epsilon^2}{N} + \frac{\zeta^2}{4N^2}}}{1 + \frac{\zeta^2}{N}} \tag{6.5}$$

In the C4.5 algorithm, the default value of δ is 0.25, though this value can be changed. It is important to note, that in the C4.5 algorithm this probability of error is on the training data not on independent test data, it also uses the upper limit of the confidence interval, therefore it is considered a more pessimistic estimate of the error rate [83], nevertheless the error rate is still very useful. Figure 6.4 shows the estimated error rate for the algorithm with respect to the number of cases used in the training set, as we increase the number of cases in the training set, the estimate of the error rate is reduced. In order to obtained this graph each simulation was run 10 times, an average of the estimated error rate calculated.

ROC Curves

ROC curves are a graphical technique for evaluating data mining schemes. The acronyms stands for *receiver operating characteristics*, and it was used by electrical engineers in signal detection to characterize the tradeoff between hit rate and false alarm rate over a noisy channel [83]. ROC curves show the performance of a classifier without considering the class distribution or the error costs. The vertical axis shows the number of positives as a percentage of the total number of positives, while the horizontal axis shows the number of negatives as a percentage of the total number of negatives.

The true positive rate (TP) is given by the following equation:

$$TP\,rate = \frac{TP}{TP+FN} \times 100\% \tag{6.6}$$

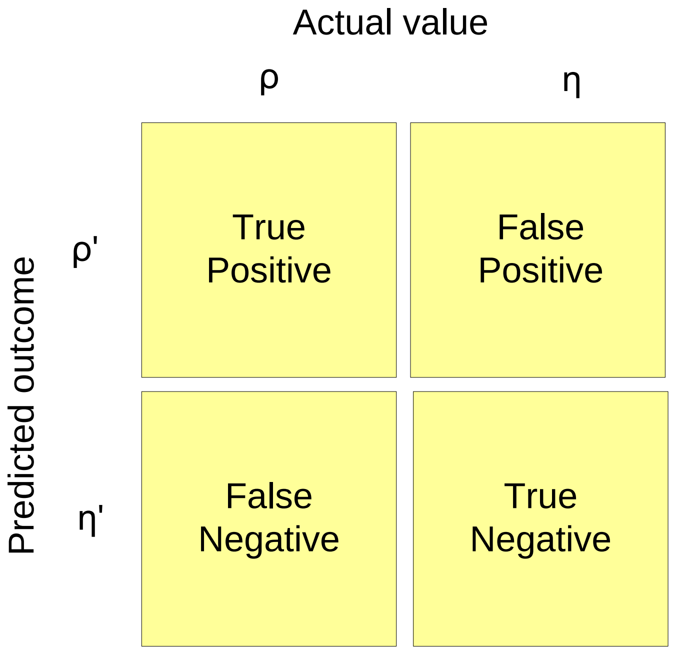
where TP is the number of true positives and FN is the number of false negatives.

The false positive rate (FP) is given by the following equation:

$$FP\,rate = \frac{FP}{FP+TN} \times 100\% \tag{6.7}$$

where FP is the number of false positives and TN is the number of true negatives.

In a binary classifier, in which outcomes are assigned two values either positive ρ or negative η. There are four possible outcomes as shown in Figure 6.5. If the outcome from a prediction is ρ and the actual value is also ρ, then the instance is labeled as a true positive (TP); however if the actual value is η then it is labeled as a false positive (FP). An instance is classified as a true negative when both the prediction outcome and the actual value are η, and false negative is when the prediction outcome is η while the actual value is ρ.



		Actual value	
		ρ	η
Predicted outcome	ρ'	True Positive	False Positive
	η'	False Negative	True Negative

Figure 6.5: Confusion matrix for a binary classifier

As an example, let's assume that we are classifying the area of the cell in two distinct classes, *dropped call* and *no dropped call*. A value of 1 is assigned to an instance where the call quality is good, and a value of 0 where the call quality results in a call dropped. The true positive rate is the amount of instances where the call quality was predicted to be good and were classified as *no dropped call*, over the total number of instances of *no dropped call*. The false positive rate is the amount of instances where the call quality was predicted to be good, but the actual classification was *dropped call*.

Figure 6.6 shows the ROC curve for the coverage learning engine. A good classifier is as close to the northwest corner of the chart as possible, and the area under the curve (AUC) should be close to 1. In this case, the algorithm performs well, however we see how the noise and randomness in the mobility model affect classification. The coverage algorithm induced 13 rules, with a maximum depth of 4. A list of the rules has been included in the Appendix. This decision tree is very simple, thus computational complexity of evaluating is very low. However, in certain areas where the call drops is neither close to 1 or close to 0, the classification is not as certain.

Computational Complexity

When we evaluate the performance of a learning algorithm another very important measurement is *computational complexity*. In computing, $O(n)$ is used to represent a quantity that grows linearly with n, $O(n^2)$ is used to represent a quantity that grows quadratically with n, and so forth. Following the approach described by Witten and Frank [83], let's define a set of training data that contains n instances and a attributes. The depth of the tree is assumed to be on the order of $\log n$, therefore it is expressed as $O(\log n)$. The computational cost of building a tree with a attributes is represented as $O(an \log a)$. Moreover, the computational complexity of pruning a tree by subtree lifting or replacing is added. The subtree replacement complexity is $O(n)$ and the complexity for subtree lifting is defined as $O(n \log n)$, thus the total complexity of reclassification is $O(n\log n^2)$. Then we have that the

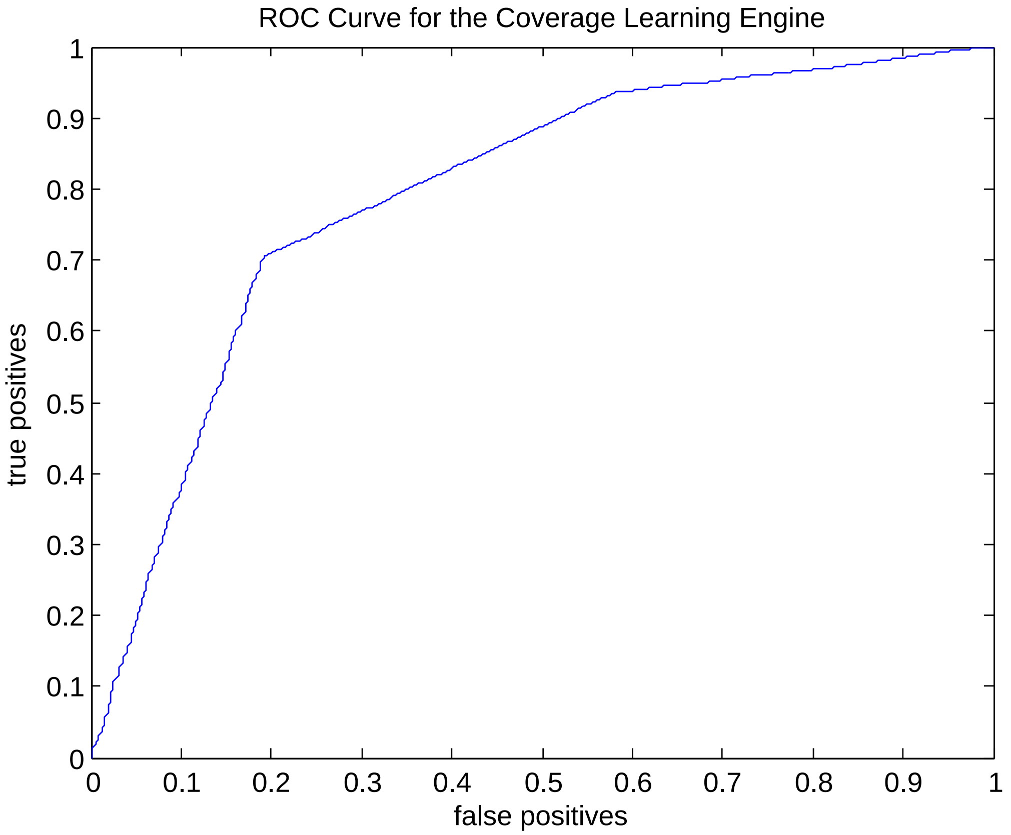


Figure 6.6: ROC Curve for the Coverage Learning Engine with RWP Mobility

total computational complexity for inducing a tree is:

$$O(a\, n \log n) + O(n\,(\log n)^2) \tag{6.8}$$

The computational complexity of evaluating the tree is defined as $O(\log n)$. As a result, once the tree is induced the computational complexity of evaluating the tree grows logarithmically with the amount of instances n.

6.5.2 How to use the learned coverage map

In previous sections, we discussed four of the five steps of our approach. In this section, we focus on the last step, what we refer as "closing the learning loop", or in other words applying the learned knowledge acquired by our engine to solve a problem that results in improved performance. The objective is to find a set of actions that use the learned knowledge to achieve the desired performance task. As an example, given the new knowledge that we have on the coverage map of the cell, we can develop a handover algorithm that prevents dropped calls in the vicinity of the obstruction, or a power control algorithm that increases the power while the UE is in the vicinity of the obstruction in order to maintain the call. Although we focus on the coverage knowledge obtained in the formulated problem, the approach can be applied to other problems in 3G networks as discussed earlier in Chapter 2.

Furthermore, the cognition cycle of a cognitive radio is considered an infinite loop. After, applying the knowledge and reacting to the environment with the appropriate action, the cognitive radio again observes the environment and determines if another action is necessary. This begins a new cycle, where learned knowledge (not trained knowledge) has been applied, then we must investigate how the environment has changed due to the engine's reaction, and the implications for the UE and for the network.

Some of the algorithms that can be developed in order to provide an appropriate set of actions to the engine are:

- *Handover Prediction Algorithm* - the coverage map can be used to predict when a handover is necessary before reaching the handover threshold. The algorithm can choose between the several handover options and select the one that maximizes the UE's objective function. This algorithm can be of great significance in femtocell environments. Femtocells will be self-deployed, requiring auto-configurability and adaptability to a changing environment. If equipped with a cognitive engine, and algorithms such as the proposed in this chapter, the femtocell will be able to learn the users' patterns in terms of their mobility, handover, power, and throughput requirements. The femtocell can apply that learned knowledge, predict when adaptations to its own configuration are necessary, and then reacting to this learned knowledge. As an example, the femtocell can learned handover patterns to the macrocell environment, predict when a handover will be needed, and then prepare and adjusts its parameters to guarantee a successful handover, while optimizing the use of the radio resources. This algorithm can also be helpful to enable opportunistic use of femtocells.

- *Power Control Algorithm* - in the vehicular environment the fast power control in WCDMA cannot compensate for the fast fading, thus the power control algorithm can predict when the SINR will drop below the desired levels to sustain a call and instruct the UE to increase power in order to maintain the call.

- *Interference Management Algorithm* - the cognitive engine can be applied to create an "interference map" similar to the coverage map. An algorithm that manages the interference experienced by the user can be developed.

- *Radio Network Planning Tool* - radio network planning requires extensive field testing measurements and knowledge expert for proper planning, the DTL algorithm can be a great tool to determine coverage from observed data without the need of human interaction.

Two other applications suggested not necessarily related to the coverage problem but to the cognitive engine design are:

- *Indexing and case classification* - The DTL algorithm can be used to perform indexing and classification instead of the k-Nearest Neighbor search algorithm or other case matching methods [87].

- *Feature Weighting* - The DTL algorithm can be used to determine the most important features for the problem formulation process of the cognitive engine and also to determine the weights of these features [86] as discussed earlier in Chapter 3.

6.6 Summary

In this chapter, we presented the implementation of the hybrid cognitive engine. We began with a system overview and discussed the assumptions and models used in simulations. We presented the results that showed how the cognitive engine classifies the observed cases to determine the coverage area for Node B. Then, we presented four methods to evaluate the performance of the algorithm: the estimate of the error rate and ROC curves. Also, we presented a complexity analysis and derived the computational complexity of inducing and evaluating the decision tree. We concluded with a discussion on how to apply the learned knowledge. Part of the material presented in this chapter was published in June 2009 at the IEEE International Conference in Communications [118].

Chapter 7

Using Cognition for Handover Management

In this chapter, cognition is applied to improve handover management in a wireless network. First, we provide a brief discussion on the background work in this area. We then discuss the concepts of handover management and handover prediction algorithms. We apply our approach to develop a cognitive engine that can improve the performance of the network by predicting handover based on the user's mobility and history. We compare and contrast how this process is performed in a traditional network and in a cognitive network. Finally, we discuss the metrics used to measure the improvements in performance and the results obtained from simulations.

7.1 Problem Background

Over the last two decades, the demand for capacity in wireless networks has increased dramatically. Generally, operators have solved this problem by deploying more cells and reducing the footprint of each cell. As a consequence mobile users have to change basestations

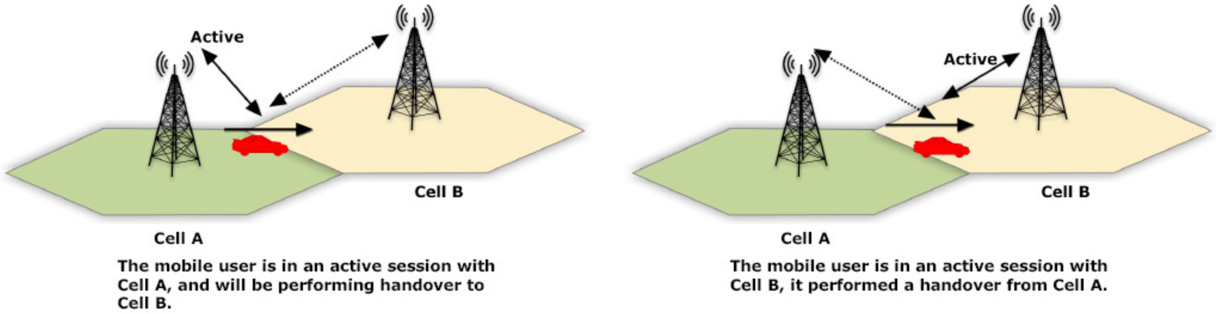

Active
Cell A
Cell B
The mobile user is in an active session with Cell A, and will be performing handover to Cell B.
Active
Cell A
Cell B
The mobile user is in an active session with Cell B, it performed a handover from Cell A.

Figure 7.1: The Handover Concept

with a greater frequency than before. Furthermore, as current wireless networks evolve, and converge into future heterogeneous wireless networks; cells size will be further reduced (i.e. femtocells) and the heterogeneity of the network will require the use of fast and efficient algorithms, not only within the same wireless network but within different communication systems (2G, 3G, B3G, WLAN, Wi-Max, etc.). The main objective is seamless communications, *anytime, anywhere*. Cognition can bring improved performance in the network by improving the handover management process, the idea is to exploit the awareness and the learning capabilities in the cognitive radio to predict when a handover is needed. Also, we will address how this approach differs from existing approaches, and how to measure the improvements when cognition is applied. We begin by discussing the handover concept and the related background information in the next section.

7.1.1 Handover Management

Handover is the mechanism that transfers an active session (i.e. voice call or data session) from one cell to another as the user moves in the coverage area of the wireless network [106]. Figure 7.1 illustrates the basics of handover. Handovers are performed in a cellular system if the mobile user has moved out of range from the basestation, or if the basestation serving

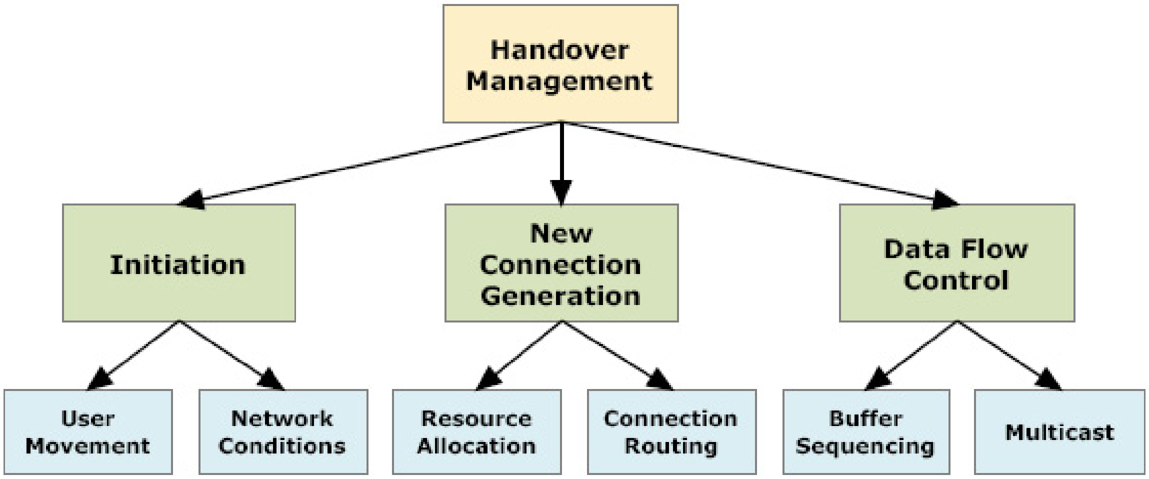


Figure 7.2: The Handover Management Process

the mobile user is full and needs to transfer the user to another nearby station, or if the current basestation is not able to provide a service requested by the mobile user.

Handover management oversees the stages needed to maintain the session active. The handover management process is divided in three stages, these include: *initiation, new connection generation and data flow control* [107]. Figure 7.2 depicts the process. In the first stage, *initiation*, the need for a handover is determined. This process can be started either by the mobile or the network. In the second stage, *new connection generation*, the network or the mobile must find the resources needed to perform the handover. In the third stage, the data flow from the old connection to the new connection is maintained, while considering handover QoS parameters such as: handover delay, handover blocking probability, etc. As we will see in the next section, handover schemes can be classified by the type of resources or by the type of control process used.

In recent years, handover management has received significant attention. It is also part of a broader concept known as *Mobility Management*. Mobility management comprises of two tasks: *Location Management* and *Handover Management*. In this work, we assume that the location information is provided by the mobile unit (i.e. GPS equipped) or in the case that GPS is not available location has been predicted using infrastructure-based location systems.

7.1.2 Handover Schemes

It is useful to discuss the various handover schemes, since proper algorithm selection depends on the type of wireless network, the resources available, and the type of control. Handovers are generally classified by the type of scheme used as follows [60]:

- *Hard Handover* - a hard handover requires the mobile unit to break the connection with the old basestation prior to making the connection to the new basestation. It is also known as "break-before-make". Femtocells currently use this scheme for handover.
- *Soft Handover* - a soft handover requires the mobile unit to establish the connection with the new basestation prior to breaking the connection with the old basestation. It is also known as "make-before-break". This scheme is often used in Code Division Multiple Access (CDMA) systems since basestations use the same frequency, and also because the mobile units are equipped with a rake receiver [60].
- *Horizontal Handover* - is the type of handover that is performed within the same access network. This type of handover can be a *Soft* or *Hard* handover. The type of mobility that is considered in this handover is termed *localized*.
- *Vertical Handover* - is the type of handover that is performed across heterogeneous access networks. As an example, the mobile user may switch from the 3G access network to the WLAN access network [119]. For this type of handover, the IEEE 802.21 is the emerging standard. It enables handover and interoperability between heterogeneous networks, including both 802 and non 802 networks [120].

Handovers can be further categorized by the handover control process [60, 120], as follows:

- *Network Controlled* (NCHO) - in this handover the network makes the handover decisions based on the measurements of the active links at the basestation. This scheme is generally used in first generation wireless networks, it requires information on all the link qualities to properly conduct the handover.

- *Mobile Assisted* (MAHO) - in this handover the mobile makes the measurements and the network makes the handover decision. This scheme is generally employed in second generation wireless networks such as GSM.

- *Mobile Controlled* (MCHO) - in this handover the mobile makes all the handover decisions. It collects the measurements on signal strength and interference in the surrounding area and completes the handover process. A cognitive radio's awareness of its surrounding environment and own capabilities makes this scheme feasible.

In the case of cognitive radio networks, the type of control scheme will depend on where cognition is located. In 3G networks, where cognition is located in the RNC, the handover scheme may be initiated by the UE, but all the control is performed by the RNC. In a femtocell deployment, if the femtocell is the only device equipped with cognition, thus the mobile unit is not a cognitive element, then the process may be handled by the femtocell entirely. In B3G networks, cognition is dispersed throughout the different layers and devices, the handover management is a multi-layer process handled by the cognitive elements and the cognitive network. In the next section, we discuss the handover criteria and/or inputs to the handover algorithm.

7.1.3 Handover Inputs

In this section, we discuss the handover criteria or inputs to the handover algorithm. Traditionally, handover algorithms have been designed based on the following inputs [121, 122]:

- *Received Signal Strength (RSS)* - is a measurement of the power in the received signal.

It is often used in traditional handover algorithms. One of the disadvantages of this criterion is the that does not consider interference. Handover algorithms that rely solely on RSS tend to perform an excessive number of handovers [106, 121].

- *Signal to Interference Ratio (SIR)* - is the ratio of the power of the received signal, to the total power of the interference signal [112]. This input considers interference, and its advantageous to use given the relationship with the Bit Error Rate (BER) and the link quality. However, in interference limited systems (i.e. CDMA, UMTS) a low BER and poor link quality does not necessarily imply a need for handover.

- *Distance* - is the distance of the mobile unit from the base station. This input is useful to determine the cell borders. It can be determined by the basestation given the signal strength measurements or calculated using the mobile's location information where Global Positioning System (GPS) is available. This input is not so accurate in indoor systems where precise location information is not available (i.e. femtocells).

- *Transmit Power* - the transmit power can be used as a handover input to limit the power requirements, reduce interference and extend battery life.

- *Traffic* - traffic information can be used as a handover input to avoid uneven traffic loads in adjacent cells, and to maintain QoS for mobile users.

- *History* - the mobile user's mobility pattern and preferences can be a helpful handover input, especially in handover prediction algorithms. In this work, we exploit the mobile user's history to predict when a handover is needed.

As networks evolve, new algorithms should also take into account other inputs as described in [122] such as:

- *Access Network Information* - as we move from centralized single technology networks to heterogeneous networks, the mobile user may select among different communication systems available, and also given the conditions of the network the mobile user may select the one with optimal conditions. As mentioned previously, IEEE 802.21 addresses heterogeneous network handover.

- *User Preferences* - may include QoS level requirements, access network, access points, costs, etc.
- *Service Type* - the user may assigned different handover priority given the service (voice, VoIP, teleconferencing, data, etc.).
- *Terminal Capabilities* - if its a multi-mode terminal or not, the access technologies it supports, battery life, power consumption, etc.
- *Cost* - there may be a cost advantage of using a particular system, for example a femtocell versus a macrocell. Or a WLAN access point versus the cellular network, thus the mobile user may prioritize given the cost of the service. The authors in [122] developed a cost function to determine the access network that minimized cost.

The DTL algorithm used in this work, can also prioritize the handover criteria in order to place the most relevant input towards the root of the tree and test on those features first for the prediction.

7.1.4 Handover Performance Metrics

When designing handover algorithms it is very important to measure the performance improvements that the algorithms brings to the network, as well as the cost involved in the handover. There are several performance metrics suggested in the literature [106, 123, 124] that help us evaluate handover algorithms, these include:

- *Call Blocking Probability* - this is the probability that a new call is blocked.

- *Handover Blocking Probability* - this is the probability that a handover attempt is blocked.
- *Handover Probability* - average number of handovers per session.
- *Call Dropping Probability* - the probability that a call terminates due to handover failure.
- *Probability of Unnecessary Handover* - the probability that a handover is initiated by the handover algorithm when the current link is adequate to maintain the ongoing session.
- *Rate of Handover* - the number of handovers per unit of time.
- *Duration of Handover Interruption* - the length of time during a handover that the terminal is in communication with neither base station. (This event occurs in hard handovers, or “break-before-make” type handovers. Where the communication link between the base station is terminated before associating with new the base station.)
- *Handover Delay* - the position at which the assignment probabilities of a mobile to both (current and target) basestations are equal [125].

The aforementioned handover performance metrics can help us evaluate the improved performance gained from implementing a cognitive approach in handover management.

7.2 Handover Management in Wireless Networks

In this section, we discuss the approaches to handover management in wireless networks. We begin by discussing handover in 3G networks. Then, we proceed by discussing the handover scenarios in femtocell deployments. We finalize this section by describing the handover management process in B3G wireless networks. This section will give us an insight on how cognition can be used to improve handover management in wireless networks.

7.2.1 Handover Management in 3G Networks

3G wireless networks support two types of handovers: soft handover and hard handover. However, with High Speed Downlink Packet Access (HSDPA) technology soft handovers are no longer supported [59]. The main difference between the two types of handovers are the system resources used for maintaining a session. As explained earlier, a soft handover requires communication with more than one cell at a given time. In 3G networks, there are two variants of soft handover: the UE may be in soft handover with two cells supported by the same RNC, or by two cells in the same Node B. A hard handover may occur in several situations: among two cells that use different carrier frequencies, or among cells that are connected to different RNCs. Another type of hard handover in 3G networks, is the *inter-system handover*. In this hard handover, the 3G mobile user uses the GSM network where 3G service is not available. In 3G networks, regardless of the type of handover the decision and resource handling of the handover is made by the RNC. Handovers in 3G are sometimes mobile-assisted (MAHO), where the UE triggers the handover decision by the reporting measurements. The handover procedures have been standardized by 3GPP, however the handover algorithms have not been standardized [59]. Handover algorithms are generally proprietary algorithms designed by equipment manufacturers. The handover criteria and handover decision thresholds are generally selected by the operators according to their network planning and overall network QoS goals, thus there is a lot of variety in the algorithms and thresholds used. Thus, there is room for network performance improvement by employing efficient handover algorithms that maximize network performance, reduce latency and minimize the use of resources.

7.2.2 Handover Management in Femtocell Deployments

In femtocell deployments only hard handovers are supported, regardless of the technology used. Therefore, all calls are switch to or from the femtocell zone to the macrocell network. Figure 7.3 illustrates handover in a femtocell deployment. In a femtocell deployment, there

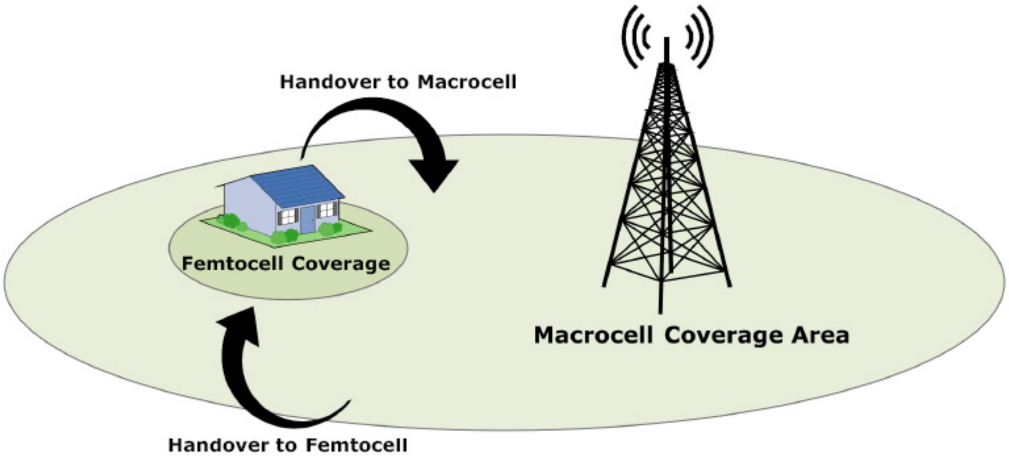


Figure 7.3: Femtocell Handover Scenarios

are two scenarios that indicate the need for a handover: the user moves from the femtocell zone to the macrocell environment, and the user moves from the macrocell environment to the femtocell zone. However, these scenarios increase if femtocells are able handover to both 2G or 3G networks in the macrocell environment, depending on the coverage available [126]. There are several issues in handover management that need to be addressed in femtocell deployments.

First, the type of access provided by femtocells. If femtocells provide *open access*, how will handovers be performed? Current handover algorithms work by broadcasting neighbor lists used by the mobile to identify candidate cells for handover. These algorithms do not scale to the expected large number of femtocells that lie in the same coverage area of the macrocell [15]. Thus, new handover algorithms need to be develop in order to guarantee proper access. The idea is to incorporate these changes in current networks, while minimizing the addition of new equipment and minimizing changes to current communication protocols.

Second, how can latency in a handover be minimized? Can we predict handover with enough accuracy such that the resources can be allocated ahead of time? This in turn, minimizes the latency; but may decrease the efficient use of resources. So there are tradeoffs to consider.

Another important aspect of handover is billing. Femtocell originated calls are covered in the monthly fee; if a call is in handover from the femtocell to the macrocell should billing differ? What happens if a call originated in the macrocell and is handover to the femtocell? What is approach to billing in these situations?

Handover management is key in the successful deployment of femtocells. The last thing a femtocell user wants is loose coverage as he enters the home. Therefore, the need for fast handover algorithms in femtocell deployments.

7.2.3 Handover Management in B3G Wireless Networks

B3G wireless networks can be described by two adjectives: *convergent* and *heterogeneous*. These networks are convergent, because at a high level they mark the intersection of data and circuit-switched services, the convergence point being the *Internet Protocol (IP)*. Mobile users no longer have to choose which wireless system is needed for their application [59]. Also, it is the convergence of multiple wireless devices (i.e. cellular phones, PDAs, laptops, etc.) into one single platform if the mobile users wishes to. Ultimately, whether to use one device or multiple devices is the user's choice, however, manufacturers and operators need to provide terminal equipment that is configurable to the user's needs. The single platform is envisioned to be an SDR module that the user can then configure according to his or her application needs.

In terms of handover management, there are two possible types of handovers in B3G networks: *horizontal handover* and *vertical handover*. The horizontal handover is an intra-system handover. This type of handover has been researched extensively in the literature, however there is still room for improvement. Currently, the vertical handover is the key challenge in these networks. The idea is to provide *seamless handover* even in inter-system handover scenarios in order to maintain QoS levels. If we further explore what the term *seamless handover* implies, we can describe this type of handover as fast and imperceptible by the mobile user. Today, in some scenarios with current technology a *seamless han-*

dover is not achievable [127]. The interruption of service is still perceptible by the user. Several solutions have been offered in the literature including: a common radio resource management approach for heterogeneous networks [102], a common based identifiers approach [127], a peer-to-peer (P2P) overlay network approach [128], an *a priori* database handover approach [129], also Layer 2 handover schemes [130–132], and several cognitive approaches [133–135], among others. An article that compares some of these approaches has been written by Stevens-Navarro et al. [136].

In summary, handover management in B3G networks is key to the successful integration of multiple communication systems. One of the main objectives is to offer *seamless communications* to the mobile user, even when performing vertical handovers. We believe that cognition can be useful in achieving this objective, however more research needs to be performed in order to develop fast and efficient algorithms that enable *seamless handover*, while minimizing changes to the infrastructure and equipment in current networks.

7.3 Handover Algorithms

In this section we provide an overview of traditional handover algorithms. We first describe algorithms based on the relative signal strength, the most common method. We then describe other traditional algorithms that trigger a handover based on the distance between mobile and basestation, the velocity of the mobile user, or the Signal-to-Interference Ratio.

7.3.1 Relative Signal Strength Algorithm

In this section, we discuss the RSS based handover algorithms. This type of handover scheme is very widely used and implemented, it was used in 2G networks and is currently being used in 3G systems with some modifications. The handover algorithm chooses the cell with the strongest relative signal at all times [106]. Figure 7.4 illustrates the basics of

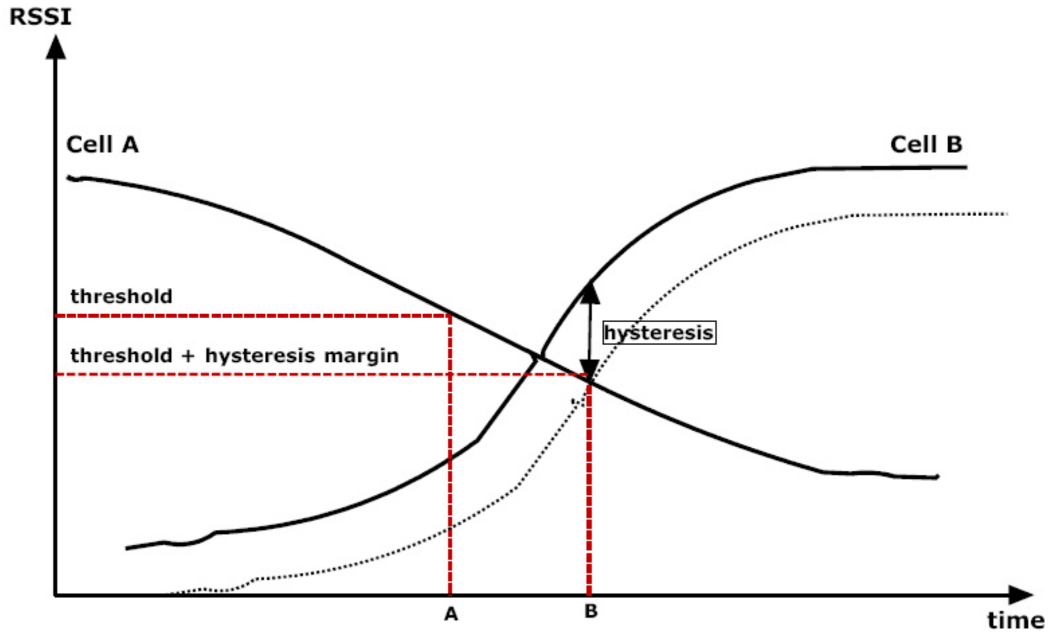


Figure 7.4: Handover based on Relative Signal Strength

this handover. The decision is based on the average of received signal measurement. The purpose of the averaging is to remove the rapid fluctuations on the signal strength due to the multi-path environment. One of the disadvantages of this method is the relatively high rate of handovers, even when the current basestation may be adequate. There are many variants of this method. In some a threshold, or a hysteresis margin, or both have been added to decrease the unnecessary handover rate [106,124,137]. Also as Zhang and Holtzman suggested an algorithm that combines absolute and relative signal measurements [138]. They are described as follows:

- *Relative Signal Strength with threshold* - In this algorithm, the network allows the user to handover only if the current basestation signal strength is less than a threshold, and the target basestation signal strength is the stronger one. This technique is generally not used because the selection of the threshold depends on prior knowledge of the crossover signal strength between basestations.

- *Relative Signal Strength with hysteresis* - In this algorithm, the network allows the user to handover only if the target basestation signal strength is sufficiently stronger (by a hysteresis margin) than the current basestation. This technique prevents the *ping-pong effect* caused by fluctuations in the received signal strength of both basestations.
- *Relative Signal Strength with threshold and hysteresis* - In this algorithm, both variants are incorporated. Therefore, the network allows the user to handover when the current basestation signal strength is below the designated threshold and the target basestation signal strength is stronger by the hysteresis margin. This method is used in 3G systems.
- *Combined Absolute and Relative Signal Strength* - In this algorithm, the handover decision is made if the following two conditions are satisfied [138]: the average signal strength of the serving basestation falls below an absolute threshold, and the average signal strength of the candidate basestation exceeds the average signal strength of the current basestation by a hysteresis margin.

The RSS algorithm and its variants are the most widely known handover algorithms, however as networks evolve into heterogeneous networks the need for flexible, fast and efficient algorithms increases. In the next section, we present other handover approaches in the literature.

7.3.2 Other Handover Algorithms

In this section, we describe other algorithms that are available in the literature [121, 139]. Some of these algorithms based their handover decision on the mobile user's distance from the basestation, the velocity of the mobile user, the Signal-to-Interference Ratio of received signals, the minimum transmit power and other criteria. While these algorithms are not as widely used as Relative Signal Strength Algorithms, their approach to handover triggering and handover decision may be useful in cognitive handover algorithms.

- *Distance-based Algorithm* - In this algorithm, the UE is assigned to the nearest cell. The handover decision is based on the RSS measurements and the relative distance of the mobile from the cell. The relative distance is calculated by comparing propagation delays. This algorithm allows handoff at the planned cell boundaries, however its efficiency over the RSS algorithm diminishes as cell sizes become smaller (i.e. microcells, picocells and femtocells). Some implementations of this algorithm can be found in [140–142].

- *Velocity-adaptive Algorithm* - In this algorithm, the length of the averaging window for the RSS adapts to the UE's velocity. This algorithm works well for fast moving users in overlay networks. Three velocity-adaptive algorithms are presented in [143, 144]. Recently in 2008, He et al. proposed a combined distance and velocity adaptive algorithm for heterogeneous networks [142]. The algorithm samples the received power from the basestation continuously, it also uses the basestation coverage radius and propagation parameters. The mobile terminal then estimates the distance and the leaving velocity from surrounding basestations. Simulation results showed significant improvements over the traditional handover decision algorithm. In addition, this algorithm can be solely implemented by the mobile.

- *Signal to Interference Ratio Algorithm* - In this algorithm, the handover decision is made if the Signal-to-Interference Ratio (SIR) from the current cell falls below a certain threshold. An advantage of using SIR as handoff criteria is that it is often the key to voice quality, system capacity and dropped call rate [139]. It is a metric for the link quality, however the SIR may oscillate due to the propagation effects and result in an increased handover ping-pong rate [139]. In references [145, 146], the authors combine the SIR criterion with a minimum transmit power algorithm.

- *Minimum Power Algorithm* - In this algorithm, the objective is to minimize the uplink transmit power by searching for a suitable combination of basestation and channel [145, 146]. This algorithm is useful when extending battery life, reducing power consumption, and reducing interference are the primary QoS requirements for the user. However, this algorithm results in an increased number of unnecessary handovers [121].

7.4 Handover Prediction Algorithms

In this section, we provide a brief overview of the handover prediction algorithms in the literature. Handover prediction algorithms have been studied extensively in the literature, however these techniques have not been well accepted in 3G or LTE systems due to their insufficient cost/performance ratio [147]. Therefore, the need to investigate handover prediction algorithms that are easy and economical to implement while improving the mobile user's QoS and the overall network performance.

The main idea in these algorithms, is to base the handover decision on the expected future value. The handover decision could be based on the signal strength, the mobile user's movement or distance from the current cell, handover history, and mobility patterns. There are three main types of predictive systems: history-based, location-based and hybrid, which combines both [148].

- *RSS and threshold Prediction Algorithm* - In this prediction algorithm, the UE periodically measures the RSS from the current basestation and compares it with a predefined threshold. If the expected future value of the RSS will lie below the threshold, then the UE triggers a handover. Some examples in the literature of the implementation of this algorithm are found in [124, 149]. In [124, 149] the authors developed a handover algorithm using Grey prediction theory to obtain the predicted RSS value. Results in [124] showed a smaller mean of handovers and smaller mean delay when compared to the traditional RSS handover algorithm without prediction.

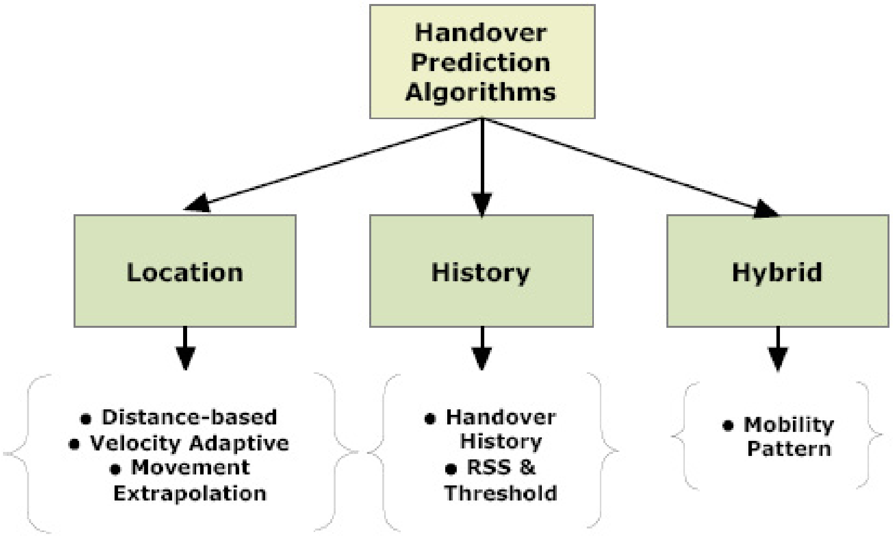


Figure 7.5: Handover Prediction Algorithms

- *Movement Extrapolation Prediction Algorithm* - In this prediction algorithm, the UE reports movement information including: location, direction, and velocity. The handover prediction algorithm calculates the expected next location of the UE and if the serving basestation for the area is a different from the current basestation, it initiates a handover. An enhanced version of this algorithm is presented in [150] where the movement trajectory of the UE is employed to make the target cell handover selection.

- *Handover History Data Prediction Algorithm* - In this handover prediction algorithm, a handover prediction is made based on previous handover history. The UE collects data on the cells visited, the dwell time in each cell, the candidate cells for handover, etc. Then the stored data is mined, the handover decision is predicted based on the history collected [119]. Many of the proposed algorithms today include the handover history data along with mobility patterns, thus algorithms that are solely-based on history are not as common. This type of algorithm may be useful where location services are not available (i.e. indoors where GPS is not accurate).

- *Distance-based Prediction Algorithm* - In this algorithm, a prediction is made based on the distance of the UE from the current basestation. If the UE is moving away from the serving cell, the algorithm calculates the time that it will be out of the coverage range and then decides on a handover and searches for the target cell. This handover tends to be fast but less precise since it uses only location information for the handover decision. An example of this algorithm is presented in [151] where a traveling distance handover prediction algorithm is applied to minimize unnecessary vertical handovers from cellular networks to Wireless Local Area Networks (WLAN)s.

- *Mobility Pattern Prediction Algorithm* - In this algorithm, a prediction is made using the mobility pattern of the UE. The UE collects movement data and handover history data. Then mobility patterns are extracted from the data, and the handover is predicted on a current mobility pattern using old mobility patterns in memory. Interest in this algorithm has been significant in recent years. It has been proposed for handover prediction in CDMA systems in [152–154]. Furthermore, in [119] it is use to improve handover in 3G-WLAN overlay networks. Also, it has been suggested for 3G LTE systems in [147]. Moreover, it has been suggested as part of the handover management scheme of next generation wireless networks in [155].

7.5 Problem Description: Using Cognition in Handover Management

In this section, we discuss how to apply cognition to the handover management process. First, we describe the objectives of our proposed cognitive engine. Then, we compare and contrast the approach to handover management in a traditional network versus a cognitive network. We continue by describing the system's model and the *knobs and meters* of the cognitive engine. With this implementation we execute the final step of the proposed approach by fielding the learned knowledge in an application.

7.5.1 Objective

The objective of the handover management engine is to exploit the cognitive capabilities of the mobile and the network to improve the overall performance. We apply our approach to develop the cognitive engine and formulate the handover problem as a *Classification and Regression* problem and use the decision tree algorithm to predict whether a handover is needed or not. We then evaluate the engine's performance to measure improvements.

Recalling the cognition cycle in Figure 2.1 we can describe the cognitive process in terms of handover management as follows:

- Observe - The cognitive mobile and the cognitive network observe the surrounding radio environment. In this phase, measurements such as: received signal strength, signal-to-interference ratio, etc. are taken. Also, location and mobility information is collected.

- Orient - In this phase, the cognitive engine (network, mobile or both) fuses the new information collected, with previous knowledge such as: geographical information, mobility patterns, handover history, user requirements, terminal capabilities, etc., and creates an interpretation of the current handover situation. In this step, the current state of the world is determined and the problem formulated in a way that the engine can solve.

- Decide - In this phase, the cognitive engine uses prediction to make a decision. In our work, we use a decision tree learning algorithm to predict whether a handover is needed or not, based on the expected future location and direction of the mobile user.

- Act - The handover process is initiated. Measurements are collected in order to determine performance improvements.

- Learn - The system learns to predict with accuracy when a handover is needed.

In summary, the cognitive engine should: predict when a handover is necessary with a certain accuracy probability, minimize the rate of handovers, minimize handover delay, and minimize the use of resources.

7.5.2 System Overview

The overall system under investigation is composed of three modules: the Network Environment Simulator (NES), the Handover Management Engine (HME), and a post processor used to interpret the results. Figure 7.6 describes the interactions between these three modules. The NES, developed using MATLAB, was designed to generate cases for our cognitive engine to analyze. A mobility simulator, called MobiSim, created by Mousavi et al. [156] was used to generate mobility traces for various mobility models. The HME was design using the approach described in Chapter 3. The DTL algorithm used is C4.5 as described in the previous chapter. A post-processor was created in MATLAB to interpret and present results graphically.

The engine was built upon the coverage learning engine discussed in Chapter 6. The steps are outlined as follows:

- *Formulation* - the learning problem has been formulated as a *Classification and Regression* problem. The engine uses previous observations in the form of cases, used to derive a decision tree that makes a prediction on the expected value of the received signal. If the expected value of the received signal is below a certain threshold the engine will make the decision to handover.

- *Representation* - the data is represented in cases. Each case contains is constructed as in Table 7.1.

- *Collection of Data* - Mobility traces are generated using MobiSim [156] for the various mobility models under investigation (Freeway, Manhattan and Random Waypoint). Then the cases are generated using the NES.

Table 7.1: REM Data and Type

Attributes	Description	Type
x	position in the x axis	numeric
y	position in the y axis	numeric
R	distance from Node B	numeric
v	velocity of the mobile	numeric
SINR	reported SINR level	numeric
HO	handover	boolean

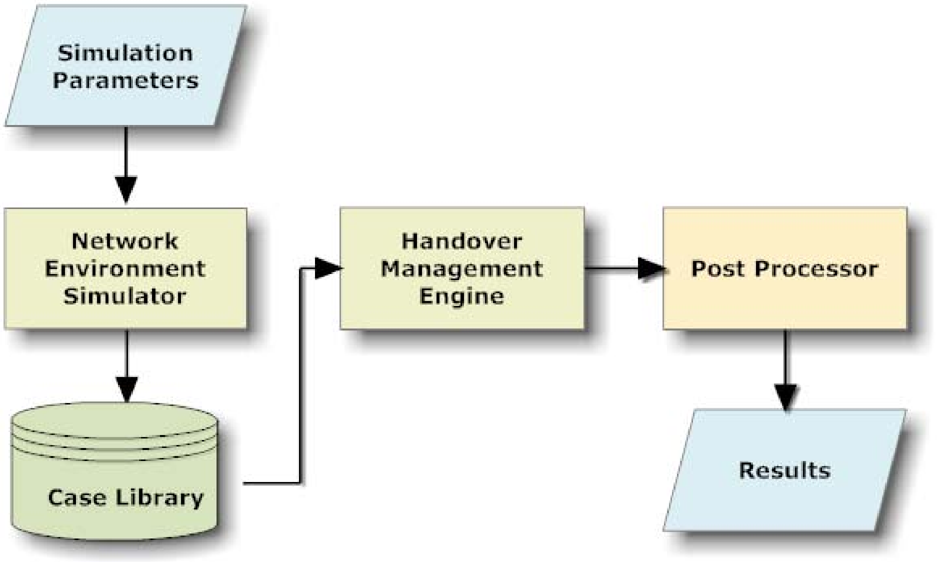


Figure 7.6: System Overview: Module Interactions

- *Evaluation* - We train the HME using a training set for each experiment. Then, HME is applied to a new data set. The HME makes a prediction on handover based on the training set and the rules derived for the training data. We analyze the performance of the prediction algorithm and compare it to the traditional handover algorithm.

- *Application* - In this case, the application of the learning method is use to predict and manage handovers in the networks.

Traditional Handover Algorithm

In a traditional 3G network, handover management is generally handled by the RNC. It may be assisted by the mobile (MAHO) or network controlled (NCHO). UMTS supports two types of handovers: soft and hard. However, in HSDPA soft handovers are no longer supported due to the amount of resources used and latency. Handover algorithms are generally proprietary to the equipment manufacturer, or have been designed by the operator to fulfill their QoS requirements. In this work, when we refer to the traditional handover algorithm we refer to a RSS averaging algorithm with threshold and hysteresis. The mobile reports the RSS measurements and the network controls the handover.

Algorithm 3 Traditional Handover Algorithm

```
Perform $RSS$ measurements
Average $RSS$ measurements
if current $RSS_C$ measurements greater than threshold $\tau$ then
    Compare $RSS$ measurements for available basestations
    Select best basestation:$N$
    if $RSS_N > RSS_C$ + hysteresis $h$ then
        Handover to basestation $N$
    else
        No Handover
    end if
else
    Perform $RSS$ measurements
end if
```

Cognitive Handover Algorithm

In our implementation of a cognitive handover algorithm, the mobile reports the current RSS levels. The HME uses the DTL algorithm to predict the RSS level of the mobile at time $t+\Delta t$. If we predict that the RSS level will be below the threshold τ, we initiate the handover process. If not, we continue to monitor the current state by performing RSS measurements again. The handover management process continues as the traditional one, the step that differentiates them is the prediction obtained by the DTL algorithm. We consider several

Table 7.2: Handover Management Engine Experiments

Experiment	Environment	Velocity	Mobility	Service
1	Pedestrian	0-5 kmph	RWP	Voice
2	Urban	0-40kmph	Manhattan	Voice
3	Suburban	40-100 kmph	Freeway	Voice

attributes of the mobile's mobility pattern. We use the initial location (x, y, R) and given the time elapsed, the mobile's velocity and direction we predict the mobile's location at $t + \Delta t$. We then examine the RSS level at time $t+\Delta t$ and if the RSS level lies below the threshold τ, we proceed with the handover management steps as in the traditional handover algorithm.

Algorithm 4 Cognitive Handover Algorithm

```
Perform $RSS_C$ measurements
Predict $RSS_C$ at $t + \Delta t$
**if** $RSS_C$ at $t + \Delta t >$ threshold $\tau$ **then**
    Compare $RSS$ measurements for available basestations
    Select best basestation:$N$
    **if** $RSS_N > RSS_C +$ hysteresis $h$ **then**
        Handover to basestation $N$
    **else**
        No Handover
    **end if**
**else**
    Perform $RSS$ measurements
**end if**
```

7.6 Simulation Results

In this section, we discuss the experiments that were used to test the HME. The experiments characterize typical wireless network environments for pedestrian, vehicular and urban models. Table 7.2 describes the experiments. In the next sections, we discuss each experiment in detail and present the results.

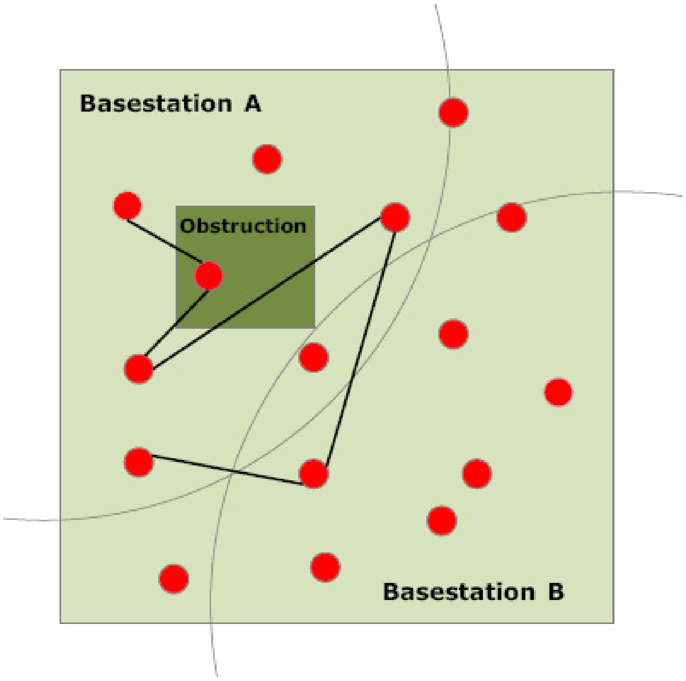


Figure 7.7: Experiment 1: Pedestrian Environment

7.6.1 Experiment 1: Pedestrian

In the first experiment, we model a pedestrian environment, where the users' mobility can be characterized using a RWP mobility model. The RWP is the "benchmark" model to evaluate the impact of mobility on network protocols due to its simplicity [157]. This model can characterize mobile users walking in a plaza, shopping center, or an open outdoor area. This model is implemented as follows: mobile users are randomly placed in the simulation area, then each mobile user travels towards its destination with a constant velocity chosen uniformly and randomly from $[0, V_{max}]$. V_{max} is the maximum velocity for all nodes. When a node reaches a destination it pauses for time T_{pause}. If $T_{pause} = 0$ then the mobility is continuous without pauses at each destination. After, this pause the node selects a random direction and the process repeats.

It is important to note that the RWP mobility model has some disadvantages such as: lack or regular movement modeling which provides no predictability. Also, it introduces sudden stops, is unable to achieve a steady-state and it is a memory-less movement behavior [158]. The speed of the users ranges typically from 0 to 5 kmph. We consider the service type to be voice, which is a circuit-switched service, thus requiring low latency. Furthermore, handover to another cell must be imperceptible by the user. To test the HME in a pedestrian environment, we have design the experiment represented in Figure 7.7. There are two basestations that cover the area. There is a coverage hole in each cell coverage area. The initial condition is for nodes to be assigned to the basestation that has the highest SINR. The idea is that the handover management engine will predict when the mobile user enters the coverage hole, and handovers to the other basestation.

Results

The results obtained from a scatter plot for the pedestrian experiment are shown in Figure 7.8. There are 360,000 observations corresponding to 1 hour of observation, and 100 nodes. The blue dots depict the cases observed by the engine, the shaded areas represent the coverage hole, with red being the worst coverage. As a node moves around the coverage area in a random waypoint pattern and is about to enter the coverage hole, the handover management algorithm will predict when the coverage hole will be entered and request a handover to the other basestation.

For this experiment, we found that in order to classify the coverage area, the algorithm induced 292 rules, with a maximum depth of 9 for some of the rules. The algorithm misclassified 1,113 cases out of 359,500, for a 0.3% classification error. A decision tree is created for each basestation. As expected due to the strong and random variations in the mobility of the nodes using RWP the resulting tree is relatively large, and cannot be easily drawn on a piece of paper. In this case, the idea is to generate the decision tree once during the day and update it when the processing requirements of the network are low. Thus, the latency to induce the tree can be handled offline, but still keep the coverage information current. When compared to the traditional handover algorithm, the results on the number of dropped calls,

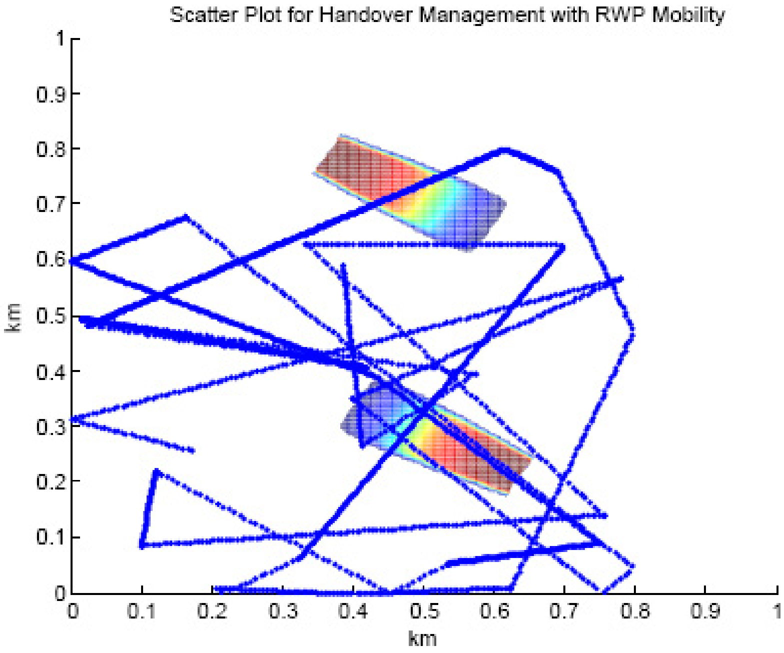


Figure 7.8: Scatter Plot for Handover Management in Pedestrian Experiment

Table 7.3: Traditional Versus Cognitive HO Algorithm - Pedestrian

Metric	Traditional Algorithm	Cognitive Algorithm
Number of Dropped Calls	192	81
Number of Handovers	1042	1155
Number of False Handovers	n/a	81

the number of handovers and the rate of false handovers for the cognitive algorithm are as follows:

Results showed that the amount of dropped calls was reduced by 57.8%, however the rate of handover increased by 10.8%. For the cognitive algorithm, the amount of false handovers was 81, representing a 7% of the total number of handovers.

Performance

The ROC Curve obtained for the handover management engine for the pedestrian experiment are shown in Figure 7.9. The ROC Curve validates our design and shows that the resulting classification obtained from the decision tree induced for this experiment offers very good performance, with an area under the curve close to 1.

7.6.2 Experiment 2: Urban

In the second experiment, we model an urban environment where the users' mobility can be characterized using the Manhattan mobility model. The Manhattan model uses a grid road topology [159] that mimics the street layout in urban cities. The map used in the Manhattan model is composed of horizontal and vertical streets. Each street has one lane in each direction, the mobile user moves along the streets in the grid. At each intersection the node can go straight, turn left or turn right. The model employs a probabilistic approach to make the turn at the intersection. Each node has 50% probability of remaining in the same

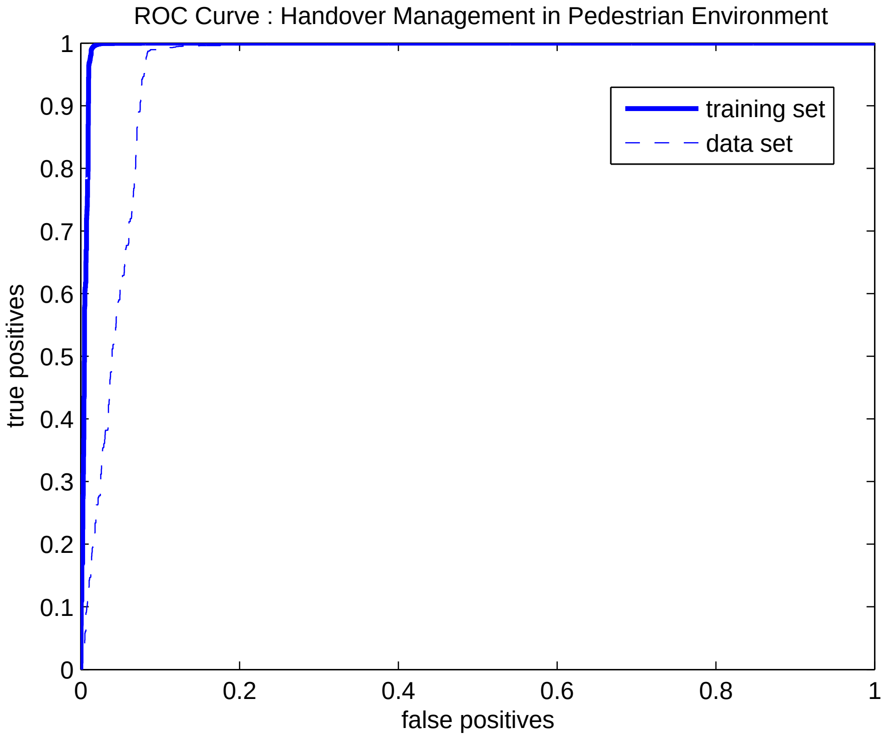


Figure 7.9: ROC Curve for Handover Management in Pedestrian Experiment

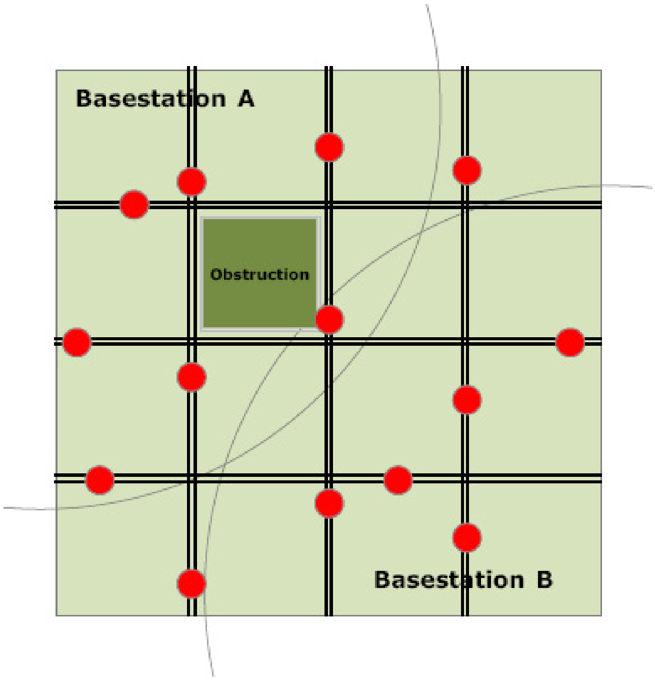


Figure 7.10: Experiment 2: Urban Environment

direction and 25% probability of turning either left or turning right. The Manhattan model differs from the RWP in several aspects [160]:

- Each mobile's movement is restricted to a grid-like pattern.
- The velocity of the mobile is temporally dependent on its previous velocity.
- If two nodes are on the same lane, the velocity of the following node cannot exceed the velocity of the preceding node.

The following speed rules describe the inter and intra node relationships:

$$|\vec{V_i}(t+\Delta t)| = |\vec{V_i}(t)| + \eta * |\vec{a_i}(\Delta t)| \tag{7.1}$$

$$\forall i, \forall j, \forall t, D_{i,j}(t) \leq SD \Rightarrow |\vec{V_i}(t)| \geq |\vec{V_j}(t)$$

Some of the shortcomings of this model are the lack of consideration on user travel decisions, and traffic safety conditions. Hence, this model is the middle ground between RWP and a car-following model [159]. This model is still considered a random mobility model.

The speed of the users ranges typically from 0 to 40 kmph. As with the previous experiment, the service type is voice, which is a circuit-switched service, thus requiring low latency (this is also needed for high-speed data services). Furthermore, handover to another cell must be imperceptible by the user. To test the HME in an urban environment, we have design the experiment as in Figure 7.10. There are two basestations that cover the area. An coverage hole is modeled in the middle of the area, the idea is that the engine will make a prediction based on the mobility of the user and handover from the basestation that has the coverage hole to the basestation with the stronger signal.

Results

The results obtained from a scatter plot for the urban experiment are shown in Figure 7.11. There are 360,000 observations corresponding to 1 hour of observation, and 100 nodes. The blue dots depict the cases observed by the engine, the shaded areas represent the coverage hole, with red being the worst coverage. The grid-like movement typical of the Manhattan mobility model is observed.

For this experiment, we found that in order to classify the coverage area, the algorithm induces 24 rules with a depth size of 6 for the first basestation, and 20 rules with a maximum depth size of 8 for the second basestation. The resulting trees are relatively small, thus it is feasible to induce the tree for this type of environment as conditions change. When compared to the traditional handover algorithm, the results on the number of dropped calls, the number of handovers and the rate of false handovers for the cognitive algorithm are as follows:

Furthermore, the amount of dropped calls is reduced by 99.5% in the cognitive algorithm,

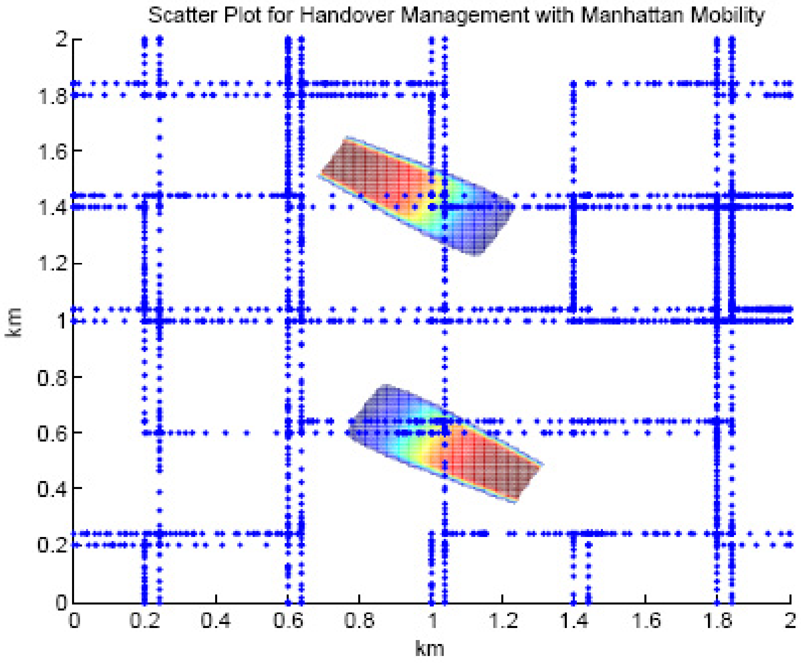


Figure 7.11: Scatter Plot for Handover Management in Urban Environment

Table 7.4: Traditional Versus Cognitive HO Algorithm - Urban

Metric	Traditional Algorithm	Cognitive Algorithm
Number of Dropped Calls	733	4
Number of Handovers	4,516	4,532
Number of False Handovers	n/a	13

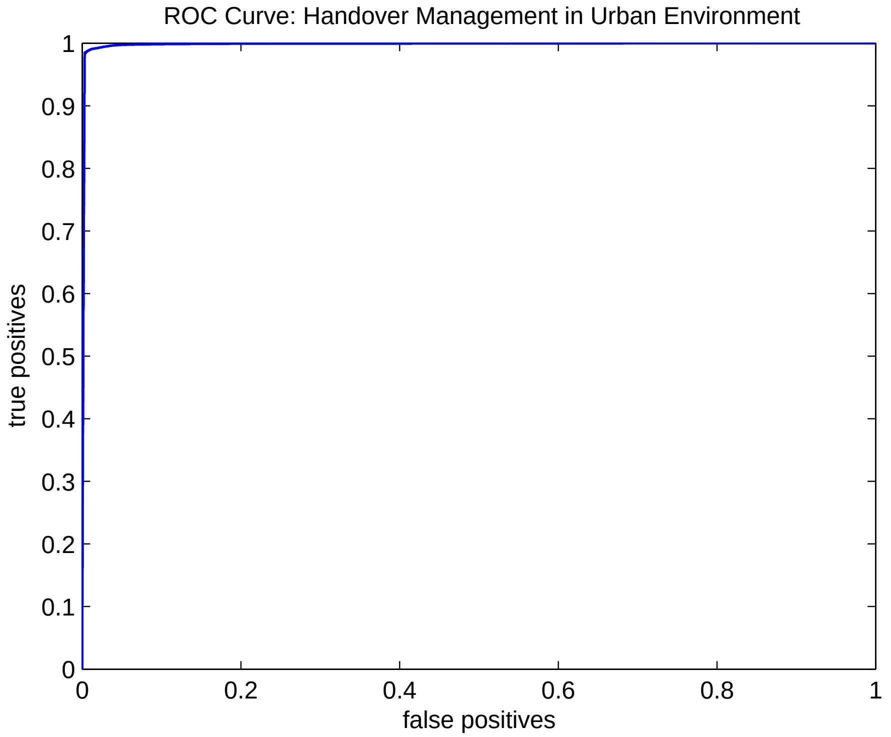


Figure 7.12: ROC Curve for Handover Management in Urban Environment

and the rate of handover is very similar it increased by 0.4%. The false handover rate is 0.3%. In this case, the amount of resources used is very similar in both cases, but the amount of dropped calls is significantly reduced by the cognitive algorithm. In the Urban experiment the tradeoffs between resources, complexity and performance improvements favor the use of the cognitive algorithm.

Performance

The ROC Curve obtained for the handover management engine for the pedestrian experiment are shown in Figure 7.12. The ROC Curve validates our design and shows that the resulting classification obtained from the decision tree induced for this experiment offers very good performance, with an area under the curve close to 1.

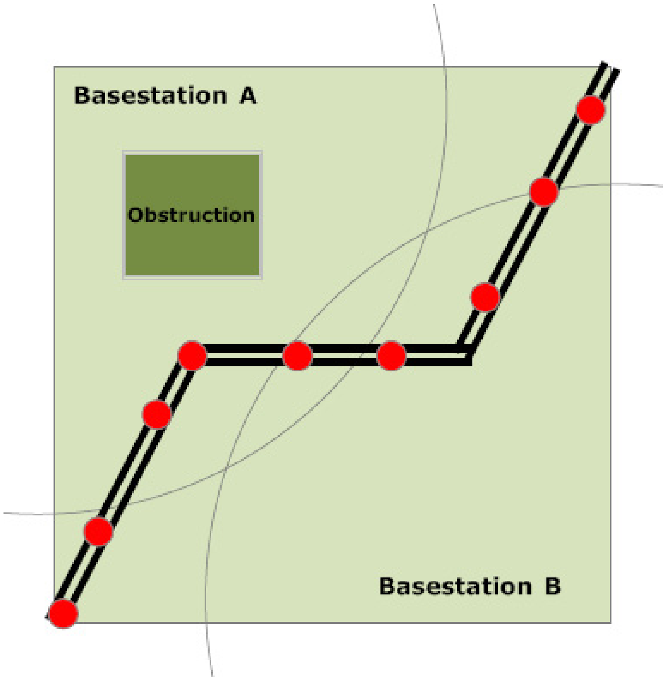


Figure 7.13: Experiment 1: Suburban Environment

7.6.3 Experiment 3: Suburban

In the third experiment, we model a suburban environment where the users' mobility can be characterized using a Freeway mobility model. The Freeway model mimics the mobility of users in a freeway. Nodes are restricted to move along a lane in one direction, as in the Manhattan model the same speed rules apply.

The speed of the users ranges typically from 40 to 100 kmph. As with other experiments, voice is the service type, which is a circuit-switched service, thus requiring low latency. Furthermore, handover to another cell must be imperceptible by the user. To test the HME in a suburban environment, we have design the experiment as in Figure 7.13. There are two basestations that cover the area. An coverage hole is modeled in the middle of the area, the idea is that the engine predicts with enough time the need for handover. In this case, where velocities are high and mobiles go in and out of the coverage area quickly, a fast and accurate prediction is very significant.

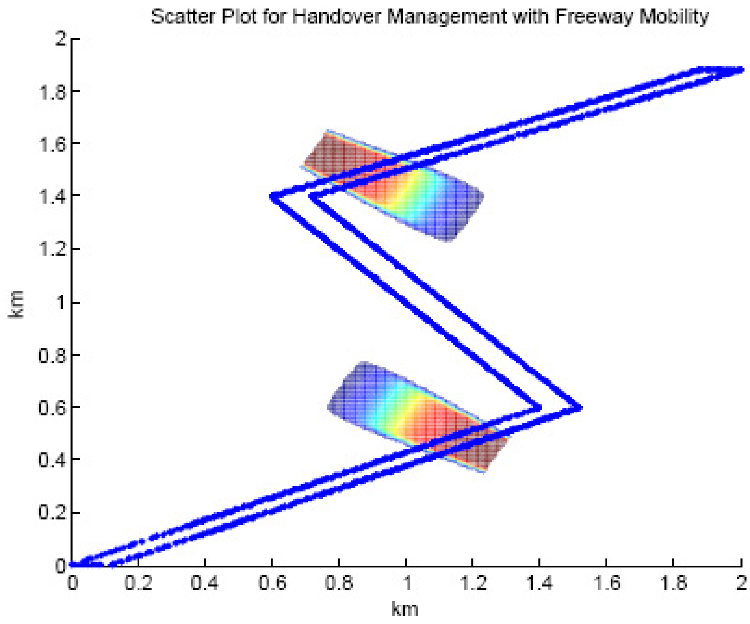


Figure 7.14: Scatter Plot for Handover Management in Suburban Environment

Results

The results obtained from a scatter plot for the suburban experiment are shown in Figure 7.14.

For this experiment, we found that in order to classify the coverage area, the algorithm induces 38 rules with a maximum depth size of 6 for one basestation, and 31 rules with a maximum depth of 6 for the second basestation. As with the urban experiment, the predictability in the mobility model allows for accurate predictions and a simple tree. Therefore, it is feasible to induce this type of tree several times during the day as network conditions change. When compared to the traditional handover algorithm, the results on the number of dropped calls, the number of handovers and the rate of false handovers for the cognitive algorithm are as follows:

Table 7.5: Traditional Versus Cognitive HO Algorithm - Suburban

Metric	Traditional Algorithm	Cognitive Algorithm
Number of Dropped Calls	1433	25
Number of Handovers	11,653	11,456
Number of False Handovers	n/a	131

In this experiment, the amount of dropped calls is reduced by 98.3% in the cognitive algorithm, and the rate of handover is reduced in the cognitive algorithm. In this case, the amount of resources used is very similar in both cases, but the amount of dropped calls is significantly reduced by the cognitive algorithm. For this experiment, the amount of handovers is reduced for the cognitive algorithm by 1.7%. The false handover rate is 1.1%, which is relatively low. In the Suburban experiment the tradeoffs between resources, complexity and performance improvements also favor the use of the cognitive algorithm.

Performance

The ROC Curve obtained for the handover management engine for the pedestrian experiment are shown in Figure 7.15. The ROC Curve validates our design and shows that the resulting classification obtained from the decision tree induced for this experiment offers very good performance, with an AUC close to 1.

7.7 Summary

In this chapter we presented another implementation of the hybrid cognitive engine. We began discussing the related background of handover management. We provided a system overview and discussed the assumptions and models used in simulations. We presented three different experiments that characterize typical wireless network environments. Results showed that our Handover Management Engine uses the learned knowledge from the Coverage Learning Engine presented in Chapter 6, and maps a handover management ac-

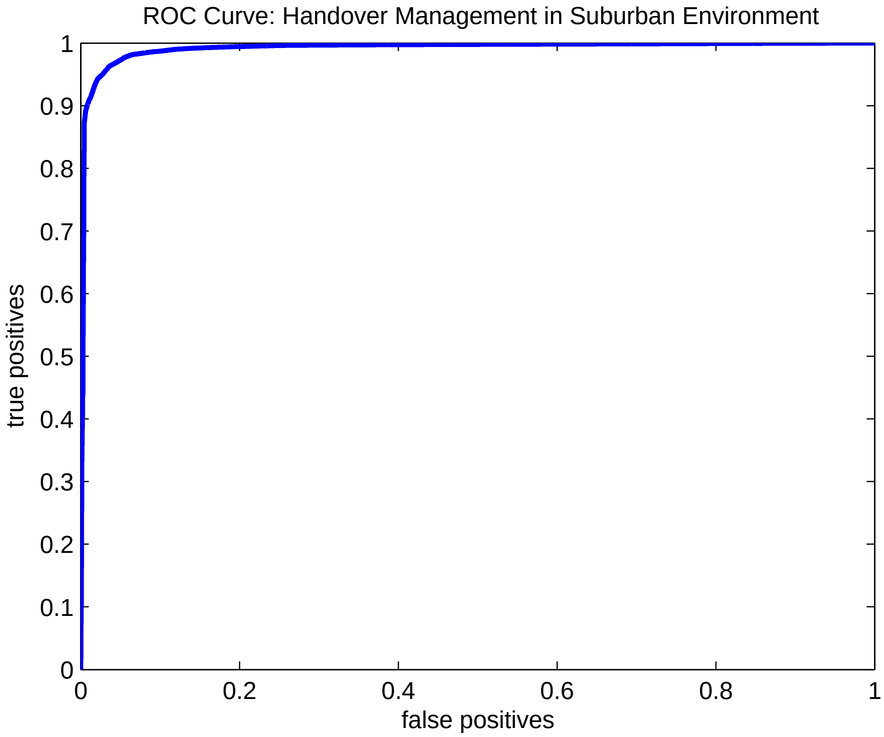


Figure 7.15: ROC Curve for Handover Management in Suburban Environment

tions that result in improved performance. We also presented how to evaluate the engine's performance.

Chapter 8

Determining Policy Events Using Cognition

In this chapter, we explore the impact of policy on our cognitive engine. First, we provide a brief discussion on the background work on policy-based management (PBM). Then, current approaches to applying policy in 3G networks, femtocell deployments and B3G networks are discussed. Furthermore, our cognitive engine is extended to learn policy events from environment observations.

8.1 Problem Background

The rapid proliferation of wireless applications has increased the demand for the electromagnetic spectrum. Current spectrum regulations limit its use, thus researchers in academia and industry have been investigating new approaches that maximize the use of this resource. Research has been conducted in many areas, including: DSA techniques, new technology such as: software radio, cognitive radio, and MIMO antennas, among others. However, in order to implement these spectrum maximization solutions a major paradigm shift in regulations, policies and market structures must take place. As with any autonomous device, there are concerns regarding a cognitive radio's behavior. A CR may behave selfishly, resulting in interference for non-cognitive or legacy users, and other cognitive devices. To address this issue, regulatory policies must be developed to assure fair use of all the network's resources. In the next section, we discuss the concept of policy-based management,and its applications in network management.

8.1.1 Policy-based Management

The concept of *policy* is used in many areas and fields from law to engineering. There is no consensus on a single definition for *policy*, mainly because of its variety of uses. However, in this work we will explore what *policy* means in the context of a network. In general, a *policy* is a set of guidelines that directs one's actions in order to achieve a set of goals or outcomes. The Internet Engineering Task Force (IETF) has defined *policy* as "a definite goal, course or method of action to guide and determine present and future decisions." In machine learning, *policy* can be defined in terms of an intelligent agent as: a control strategy for choosing actions to achieve the agent's performance goals [85].

Furthermore, one can also classify *policy* in terms of how it is used in the management of the network. According to Chadha and Kant [161] there are four basic applications of *policy* in networks:

- policy as rules,
- policy to grant/deny permissions,
- policy as constraints or parameters, and
- policy as configuration.

The first application is probably the most common one. A policy is a set of rules that dictates the behavior or actions for the network given the current conditions or state of the network. As an example, a policy rule for our cognitive 3G network could be:

"If Wi-Fi network available, use Wi-Fi for all data communications."

In this application the policy is prescriptive and procedural [161].

The second application is to grant or deny permissions, in this case the policies tend to be declarative rather than prescriptive. They specify the *what* not the *how.* As an example of this type of policy could be:

"If user is from service class A, do not connect to Wi-Fi."

In the third type of application, policy is used as constraints or parameters in the network. This type of policy application implements the network's target goals in a declarative form [161]. In this case, the policy also makes statements about relationships that must hold true among the network's elements. Furthermore, the network's target goals may come from different sources (i.e. operator's business goals, QoS requirements from the users, regulatory policy, etc.), thus the network operator must translate these higher goals into policy formulations that can be implemented in the network. An example could be:

"At most 20% of the bandwidth, is allocated to service class A."

The above policy statement ties two network elements such as the bandwidth, and the user service class to higher network goals such as targeted QoS, revenues, resource allocation, etc.

The fourth application is policy configuration. In this application, policy is used to configure the network elements, protocols and services. This type of policy is prescriptive and more precise as it sets parameters in the network configuration.

"Set maximum transmit power for users in service class A to -10 dBm."

The previous policy statement limits the transmission power of an user in a particular service class to -10 dBm.

In summary, the term *policy* can have different meanings in the context of network management. Analogous to applying machine learning to wireless networks, where formulation is key. The accurate application of policy relies on the proper use of policy given the network goals.

8.2 Problem Description: Determining Policy Events Using Cognition

In this section, we discuss how to apply cognition to determine policy event patterns. First, we describe the objectives of cognitive engine. We continue by describing the system's model and the *knobs and meters* of the cognitive engine. With this implementation we validate our approach, and show how different problems can be solved by mining the appropriate information and formulating the learning problem, as discussed in Chapter 3.

8.2.1 Objective

The objective of the policy management engine is to exploit the cognitive capabilities of the mobile and the network to determine policy patterns. We apply our approach to develop the cognitive engine and formulate the policy determination problem as a *Classification and Regression* problem and use the hybrid case-based reasoning and decision tree algorithm to predict whether a policy event has occurred or not. The policy event of interest is *transmission power control*. In this scenario, The target goal for the engine is to classify network events as policy events.

Recalling the cognition cycle in Figure 2.1 we can describe the cognitive process in terms of policy management as follows:

- Observe - The cognitive mobile and the cognitive network observe the surrounding radio environment over long periods of time. In this phase, measurements such as:

received signal strength, signal-to-interference ratio, cell load, etc. are taken. Also, location and mobility information for each mobile is collected.

- Orient - In this phase, the cognitive engine (network, mobile or both) fuses the new information collected, with previous knowledge such as: geographical information, mobility patterns, handover history, user requirements, terminal capabilities, etc., and creates an interpretation of the current network situation. In this step, the current state of the world is determined and the problem formulated in a way that the engine can solve. The objective for the engine has been clearly defined as policy determination.
- Decide - In this phase, the cognitive engine uses prediction to make a decision. In our work, we use a decision tree learning algorithm to predict the likelihood that a policy event will occur or not.
- Act - The engine classifies the events as being policy events or not. If it determines that the *transmission power control* will occur, then the nodes in excess to the new capacity (limited by the new maximum transmission power) will be handover to the other basestation.
- Learn - The system learns the network's policy patterns.

8.2.2 System Overview

The overall system under investigation has the same structure as the HME in Chapter 7. However, the objective of the engine is to determine policy patterns over long periods of time. The engine was built upon HME discussed in Chapter 7. The steps are outlined as follows:

- *Formulation* - the learning problem has been formulated as a *Classification and Regression* problem. The engine uses previous observations in the form of cases to derive a decision tree that makes a prediction on the likelihood that a policy event will occur.

Table 8.1: REM Data and Type

Attributes	Description	Type
τ	timestamp	numeric
x	position in the x axis	numeric
y	position in the y axis	numeric
R	distance from Node B	numeric
v	velocity of the mobile	numeric
SINR	reported SINR level	numeric
Policy	policy event	boolean

- *Representation* - the data is represented in cases. Each case contains is constructed as in Table 8.1.

- *Collection of Data* - Mobility traces are generated using MobiSim [156] for the various mobility models under investigation (Freeway, Manhattan and Random Waypoint). Then the cases are generated using the NES.

- *Evaluation* - We train the Policy Engine (PE) using a training data set for each experiment. A decision tree is generated for each training set. Then, the PE is applied to a new data set. The PE makes a prediction on whether a policy event occurred and the time period that it occurred.

- *Application* - In this case, the application of the learning method is use to predict policy events.

8.3 Policy Determination Experiment

In this section, we discuss the experiment used to simulate how the cognitive engine determines policy events. The experiment characterizes the 3G macrocell network environment with a Freeway mobility pattern. The engine performs network environment observations on a 2-lane highway, for a total period of 1 hour. There are 100 nodes, thus resulting in 360,000 cases.

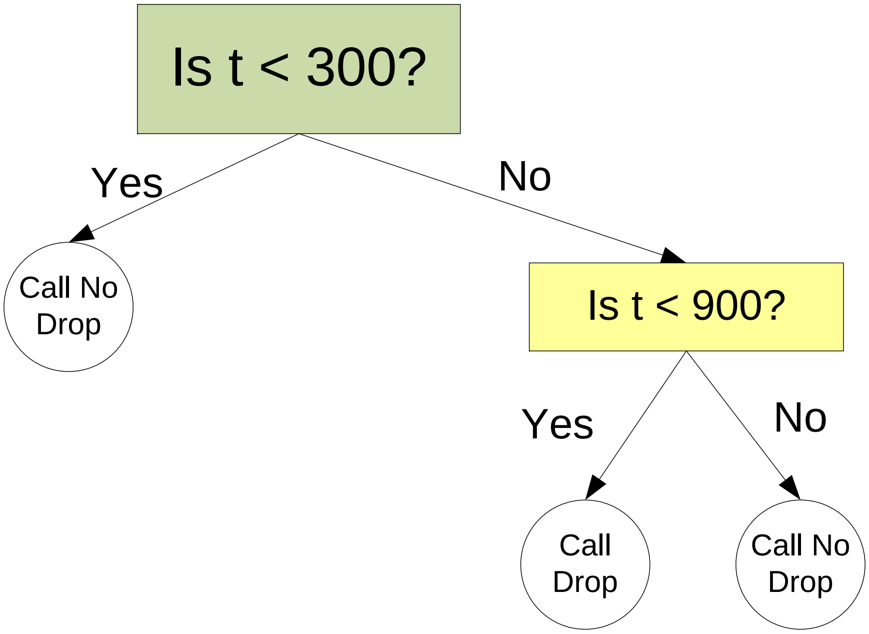


Figure 8.1: Experiment: Policy Events Determination

Furthermore, the nodes are moving along the lanes at random velocities that 40 kmph and 100 kmph. The metric used is analogous to the three bars on a 3G cellphone screen, availability of service. A policy event that constrains the transmitted power is added during an interval of time, Δt. The policy engine is induced with an initial simulation of 360,000 cases and tested on another set of 360,000 cases.

8.3.1 Simulation Results

The resulting decision tree is shown in Figure 8.1. In this case, the decision tree simplifies significantly as it determines that the attribute *timestamp*, τ, is uncorrelated from the rest of the attributes. It is the attribute with the highest entropy and splitting on that attributes results in the simplest tree.

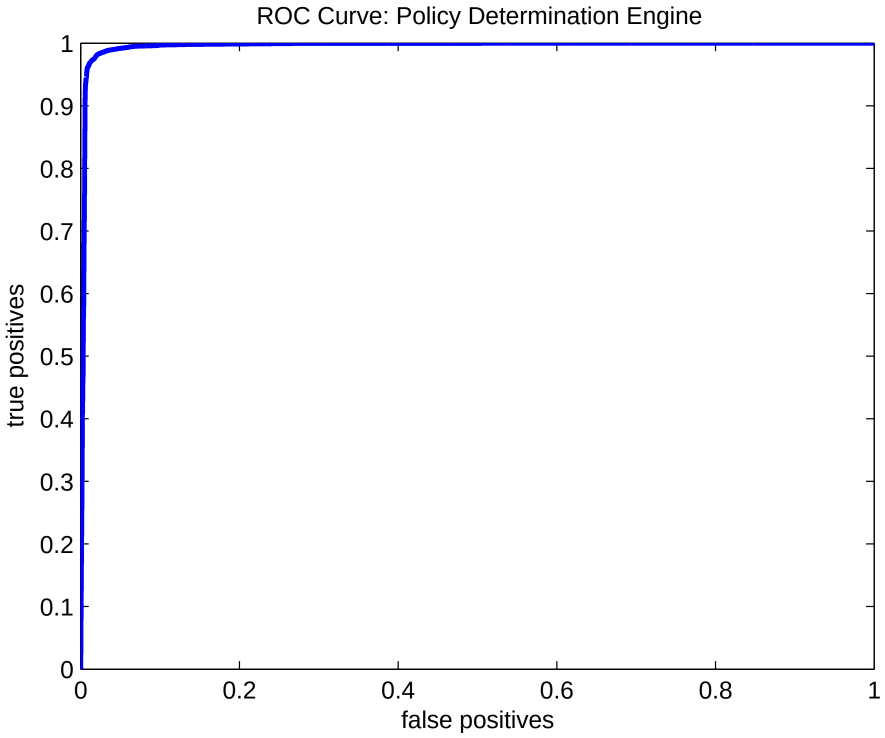


Figure 8.2: ROC Curve: Policy Determination Engine

The resulting ROC curve for the classifier obtained on the testing set is depicted in Figure 8.2. As seen from the figure, the policy engine is a good classifier for policy events, with an area under the curve very close to 1. The ideal classifier has an ROC curve with an area under the curve exactly one. The resulting decision tree algorithm is a good way to classify the event but it is not immune to noise or overfitting of the data.

This policy determination algorithm can be used in many network environment scenarios. As we move to autonomous devices and self-organizing networks, we can initialize these devices with "typical scenarios" knowledge, then these cognitive devices that are capable of *learning* via computation (i.e. using Machine Learning (ML) algorithms) can alter their initial pre-programmed configuration to improve performance as dictated by the observed network conditions.

As an example, a femtocell that has been deployed in a suburban setting, where other femtocells have been deployed, can determine from environment conditions that power transmission should be minimized at a certain time or in a certain direction (if equipped with a directional antenna) in order to limit interference to a nearby femtocell, or to the macrocell. Also, a cognitive device can generate policy rules from the observed conditions and disseminate the information with nearby nodes, either cognitive or non-cognitive devices. Also, this type of algorithm can be employed by policy regulators to validate that policy rules have been enforced.

8.4 Summary

This chapter presented how the decision tree learning CE can determine policy event patterns. We began discussing the concept of policy-based management and how it is currently applied to wireless networks. We proceeded to describe how the cognitive engine is used to determine policy events, or derive policy rules. We suggested several implementations of this approach. Finally, we evaluated the engine's performance via simulation.

Chapter 9

Conclusions and Future Work

This document proposed an approach to applying cognition in wireless networks. Also, discusses the concept of a cognitive radio resource manager. The proposed approach is used to design a hybrid cognitive engine that uses CBR and DTL for the reasoning and learning phases of the cognitive cycle. We applied the hybrid engine to three typical RRM problems: improving coverage, managing handover and determining policy events. This chapter summarizes this work, lists contributions and future research direction.

9.1 Conclusions

This document begins by providing the motivation for applying cognition to 3G and B3G wireless networks. Although, most of the focus of the cognitive radio research has been on the application of cognitive radio for management of the spectrum, we argue that cognitive radio can bring improvements in radio resource management by the addition of awareness, intelligence and the use of previous experience to solve current problems.

We proceeded with a brief discussion of the concepts of cognitive radio and cognitive engine. We contrasted some of the popular definitions and also provided the scope of these concepts in this proposal. We continued with a survey of previous cognitive engine implementations, the type of applications and the machine learning techniques used in these cognitive engines. We proceeded by discussing the application of cognition to 3G wireless networks. We listed possible applications, discussed the areas that can benefit from adding cognition and identified the *knobs* and *meters* for a cognitive radio implementation in 3G wireless networks.

In Chapter 3, we presented the proposed approach to adding cognition to wireless networks. First, we discussed the impact of adding intelligence to communications systems. We described our approach used to apply machine learning to "real-life" problems in order to achieve *learning.* We continued with a detailed discussion of the machine learning techniques and the data mining algorithms used in the hybrid cognitive engine. We examined case-based reasoning, its advantages, its disadvantages, and how to implement it in the cognitive engine design. As discussed, CBR is a better technique for *acting and planning* learning problems, while DTL works on problems that have been formulated as a *classification and regression* learning problems. CBR can be the decision maker and solver of general problems, while DTL can be used to solve very specific problems.

Chapter 4 presented the generic cognitive engine and its components. Inputs and outputs of these algorithms were described and discussed. Also, cognitive engine architectures for femtocell deployments and B3G wireless networks were suggested. Our generic cognitive engine is not limited to the 3G architecture but it was chosen to provide a definite context in order to evaluate performance.

Chapter 5 presented the concept of cognitive radio resource management. We first provided an introduction and the motivation for the concept of CRRM in wireless networks. We discussed possible CRRM applications and how adding cognition can improve performance. Finally, we introduced three RRM problems that are the main focus of Chapters 6, 7 and 8: applying cognition to improve the coverage in a cell and furthermore, using the acquired knowledge to perform handover management, and determining policy patterns from observations. This chapter merged the concepts of cognitive radio and cognitive engine to the radio resource management tasks in the network.

In Chapter 6, we presented the Coverage Learning Engine and its results. First, we provided an overview of the system, and a description of the coverage problem in Node B. Also, the assumptions and the models used in the simulation were discussed. We applied the engine to a 3G cognitive radio application: learning the coverage of Node B based on environment observations. The results obtained showed that the engine classified the drop call cases for Node B and deducted general rules on the coverage of the cell. The resulting tree was examined. Furthermore, we evaluated the performance of the engine in terms of the quality of the learned knowledge by examining the estimate of the error rate. We also presented ROC curve that depicts the performance of the classifier in terms of the tradeoff between hit rate and false alarm rate. Furthermore, we presented the cost of implementing the algorithm in terms of the computational complexity for inducing the decision tree. The computational complexity for evaluating the tree grows logarithmically with the amount of cases. The idea is to induce the tree with a representative amount of training data, such that the resulting tree is able to accurately predict or classify future events. An operator can choose to induce the decision tree offline, and update it once a day or as needed. In order to minimize computation in the system. This chapter concludes presenting how this new acquired knowledge on the network's coverage can be used to develop algorithms that improve the network's performance.

Chapter 7, presented an example of how to use the acquired coverage knowledge to manage handover and improve the performance of the network. By doing this we completed the steps of our proposed approach described in Chapter 3. First, we discussed the concepts of handover management and handover prediction algorithms. We applied our approach to develop a cognitive engine that improved the performance of the network by predicting handover based on the user's mobility pattern and history. We compared our cognitive handover management approach to a traditional handover approach. Finally, we discussed the metrics used to measure the improvements in performance and the results obtained from simulations.

In Chapter 8, another case study on how to apply cognition was presented. In this case, the learning problem was formulated such that the predicted value was whether a policy event occurred or not. The learning task for the cognitive engine was to determine if a *transmission power control* event occurred. With this case study, we also validate our approach of adding cognition to the wireless network.

9.2 Contributions

The contributions of our work are three-fold. First, we discussed the application of machine learning techniques and data mining methods in the design and development of a cognitive engine. There is a need to research AI and machine learning techniques that will allow for robust cognitive engine design, at the present time only a handful have been investigated. Furthermore, in order to successfully implement cognitive radio in 3G and B3G wireless networks, issues such as: latency, complexity, auto-configurability, and autonomy have to be addressed. In this work, we attempted to address some of these issues. The latency requirements in wireless networks will drive the cognitive engine design. We proposed a hybrid cognitive engine that uses case-based reasoning and decision tree learning. Case-based reasoning provides problem solving for acting and planning, while decision tree learning provides problem solving for classification and regression. We have divided the "learning" phase of the cognitive engine's cycle in two steps. Both of the techniques used in the engine's design exploit using past experiences to induce general rules and to gain new knowledge.

We selected these techniques as they provide key features for our engine's design. We found that CBR systems are easy to implement if experience is rich, they also provide a closer match to the actual human reasoning process, they provide efficient reasoning by focusing on the problem solving aspects that were important in previous cases, and they allow for faster knowledge acquisition. Also, maintenance efforts can be reduced as the CBR system will learn from the new cases and update the case library as new problems and solutions are tackled. DTL offers several advantages: the DTs produced are easy to understand and interpret, DTs manage large amounts of data quickly, DTs can work successfully with discrete and continuous attributes, and DTs have been well studied in literature. There are efficient algorithms such as C4.5 that reduce development time. Furthermore, if we consider the latency requirements in 3G wireless networks, we can expect that most of the training of the CBR system will be off-line learning in order to minimize the processing time, while the DTL can be trained using both offline and online learning. The DTL's algorithm abilities to classify large amounts of cases, and to prioritize among relevant features, minimizes the processing time of the cognitive engine, this in turn improves the latency of the engine.

Second, an analytical framework that relates the machine learning techniques (CBR and DTL) to the cognitive engine's learning tasks was presented. The cognitive engine learns when its performance at a certain network tasks improves, given its experience on the task. The design objective of the cognitive engine was to forecast the occurrence of an event at some future time, given the current observed conditions at a given time. The cognitive engine estimated the probability of the event occurring, given previous network conditions. If the probability was reasonably close to 0 or to 1, the cognitive engine had successfully used previous experience to get information about future events, thus enhancing the decision process. The predicted event was modeled as a Bernoulli trial, where the probability of success is a function of network conditions. A mixture model was used to accomplish this. Decision tree learning and case-based reasoning are used to estimate the parameters for the mixture model. Each case in the case library represents the network conditions at a given instant. The parameter for each distribution in the mixture model can be estimated using the maximum likelihood estimator (MLE) for the probability in a Bernoulli trial. Therefore, the decision tree is used to estimate the components of the mixture model, and the case base library is used to estimate the parameter of each of the components.

Third, we applied the hybrid cognitive engine to three network problems: learning the coverage of a cell, managing handover and determining policy patterns. In the first case study, the cognitive engine learned the coverage of the cell from observations; the objective was to provide a “proof-of-concept” on our engine design. Results showed that the engine learned the coverage and, furthermore, was able to induce general rules on the training cases. The engine increased the knowledge it had over the radio environment; now when future cases arise, the engine is able to predict if the SINR will be adequate to maintain a call. If the prediction is such that a call drop is expected, then the engine must map an action that will prevent the call drop from happening such as performing a handover. The performance of this engine was evaluated in terms of error rate, and ROC curves. We also derived the computational complexity of inducing the tree in terms of $O(n)$.

We expanded the initial engine design and completed our proposed approach by fielding the learned knowledge with an application. We developed a handover prediction algorithm using decision tree learning and tested the engine under several scenarios: pedestrian, urban and suburban (i.e. vehicular). We mapped actions in handover management to the learned knowledge. In other words, once we acquired the new knowledge on the radio environment we used the knowledge to improve the network's performance and maximize the user's goals. The handover prediction algorithm used the coverage map to predict when a handover was necessary before reaching the handover threshold. Results showed the for the pedestrian case, where the mobility was depicted by RWP model and small velocities (0 - 5 kmph), the number of dropped calls was reduced by 46%, however, the handover rate increased by 7.9%. The resulting classifier was a strong one with an area under the curve for the ROC curve very close to 1. For the urban case, the mobility pattern was characterized using the Manhattan mobility model and velocities between 0 and 40 kmph, the amount of dropped calls was reduced by 90.2%, the handover rate was very similar to the non-traditional handover algorithm. The rate of false handover for the urban case was 0.4%. For the last case, we examined mobility patterns in suburban environments. Where the mobility can be characterized by the Freeway model, and velocities ranging from 50 kmph to 100 kmph. Results showed that the amount of dropped calls was reduced by 75.9% and the rate of handover was slightly smaller than the traditional handover algorithm. The false handover rate was around 1.8%.

The handover prediction algorithm developed can bring significant improvements in many network scenarios. Femtocells will be self-deployed, requiring auto-configurability and adaptability to a changing environment. If equipped with a cognitive engine, and algorithms such as this one, the femtocell will be able to learn the users' patterns in terms of their mobility, handover, power, and throughput requirements. The femtocell can apply the learned knowledge, predict when adaptations to its own configuration are necessary, and then react to this learned knowledge. As an example the femtocell can learn handover patterns to the macrocell environment, predict when a handover will be needed, and then prepare and adjusts its parameters to guarantee a successful handover while optimizing the use of the radio resources. One of the limiting aspects of using this type of algorithm indoors is the need for location information. Currently, indoor location services are not very reliable or might not be cost-effective for a femtocell deployment. Therefore, other options for mobility prediction indoors have to be considered. The handover prediction algorithm can also be employed in B3G networks, where users will be able to handover to the various RATs available. User preferences can be included in order to maintain the highest QoS level while reducing costs. As an example a mobile user may default to free public Wi-Fi access rather than the more expensive 3G or 4G access.

The policy prediction algorithm developed classifies policy events accurately. The resulting decision tree is simple, and the algorithm is able to classify the regions precisely. This occurs because the attribute *timestamp* is independent from the other attributes, time passes and the policy event shall take place regardless of the network conditions. The ROC curve for the classification is very close to 1. There are some variations due to noise in the data, and overfitting by the algorithm. This type of algorithm is very useful in self-organizing networks where self-configuration and self-optimization are expected. The node can learn the network's policies at initiation, and self-configure as these change.

9.3 Future Work

The scope of this dissertation included application of machine learning techniques to cognitive engine design, the application of cognition to three CRRM algorithms: coverage management, handover management and determining policy patterns. Also, we focused on two machine learning methods: case-based reasoning and decision tree learning. Furthermore, for our simulations we considered the 3G wireless networks environment. Going forward we would like to extend this research in several areas including:

- Cognitive Engine Design - in terms of the cognitive engine's design the broader cognitive radio issues discussed in Chapter 1 should be further explored. There are still many unknowns and there is a need to further investigate: the amount of cognition needed in the engine, the location of this cognition in the system's architecture, the engine's architecture, the methodology used in the engine's design, the amount of previous information used, the transfer of learned knowledge, the amount of information used for training (as seen in Chapter 6), the tradeoffs between cognition, computational complexity and latency. These aspects have been investigated in the machine learning and artificial intelligence field, however research on the impact of these aspects in telecommunication systems design has just begun.

- Machine Learning and Data Mining Methods - the focus of this work has been on two learning techniques (CBR and DTL), recalling the agreements for applying learning methods: there is no algorithm that leads to better learning, but the representation of data and the definition of adequate performance tasks lead to successful implementations. The machine learning and data mining fields offer many learning alternatives that have not being explored. Research on other learning techniques that are computationally simple, easy to implement and easy to understand while still optimizing learned knowledge, will be continued.

- Multivariate Decision Trees - in this work we employed univariate decision trees, where each partition is made on a single attribute. Multivariate decision trees are able to generalize when dealing with attribute correlations, resulting in simpler tree structures and less overfitting of the data [162]. In wireless networks there exists correlation among the attributes that describe the network (i.e. cell loading and blocked probability, call blocked probability and QoS, etc.). Thus, creating tree structures that can exploit these correlations can yield to simplification of the tree structures, thus reducing the amount of time it takes to evaluate the tree.

- Algorithms - in this dissertation we focused on three CRRM algorithms: coverage management, handover management and policy management. In Chapter 5, other CRRM algorithms were discussed. In future work, the formulation of the learning problem can be such that other CRRM algorithms can be developed. Also, the context of these algorithms was the 3G network environment. The interaction of these algorithms in a multi-RAT environment have not yet been studied. Future work, will include the development of multi-RAT cognitive algorithms for radio resource management as described in Chapter 5.

- Networks - in this work, we discuss how to apply cognition in 3G, femtocell deployments and B3G networks. However, we tested our algorithms in the 3G wireless network environment. In our future work, we would like to develop algorithms and test them in other network environments such as: femtocell deployments, B3G networks, personal area networks, ad hoc networks, sensor networks, and others.

- Simulation Environment - we were able to develop realistic models by developing our own Network Environment Simulator, and employing a mobility simulator MobiSim created by Mousavi et al. [156]. However, the impact of mobility management and prediction based on expected mobility can be further explored. A more generic simulator that includes the amalgam of RATs available in B3G networks will be a great tool to research how the cognitive algorithms perform across communication systems. Therefore, in our future work we will focus on creating a more flexible simulation environment that encompasses the various RATs available in the next generation of networks.

- Testbeds and Hardware - the methodology used to perform the research outlined in this dissertation has been analysis, modeling and simulation. Future work, should include the implementation of these algorithms using commercially off-the-shelf equipment. The creation of cognitive test-beds is necessary to validate the algorithms in real-life scenarios.

The application of cognition to wireless networks is in the early stages of research and development. The author believes that developing cost-effective learning techniques that optimize: network goals, users goals, operators goals while maintaining complexity low, will be key in the success adoption of cognitive radio technology.

9.4 Expected Publications

The work presented in this dissertation will be published as follows:

1. L. Morales-Tirado, J. E. Surís-Pietri, and J. H. Reed, *"An Approach to Using Cognition in Wireless Networks,"* journal submission to IEEE Wireless Communications Magazine, special issue on Dynamic Spectrum Management in Wireless Networks, due January 15, 2010.

2. L. Morales-Tirado, J. E. Surís-Pietri, and J. H. Reed, *"Using Decision Tree Learning for Cognitive Radio Resource Management,"* journal submission to IEEE Transactions on Mobile Computing, due March 1, 2010.

3. L. Morales-Tirado, J. E. Surís-Pietri, and J. H. Reed, *"A Decision Tree Learning Algorithm for Handover Management in Wireless Networks,"* to be submitted to IEEE

and EURASIP 2nd Workshop on Cognitive Information Processing, due January 10, 2010.

9.5 Publications

The research presented here, as well as background research has resulted in the following publications:

1. L. Morales, J. E. Surís and J. H. Reed, *"A Hybrid Cognitive Engine for Improving Coverage in 3G Wireless Networks",* IEEE ICC Joint Workshop on Cognitive Wireless Networks and Systems, June 2009.

2. L. Morales, J. E. Surís, *"Cognitive Radio: A New Trend in Wireless Communications",* Colegio de Ingenieros y Agrimensores de Puerto Rico, Revista Dimensión, to be published in 2010.

3. An He, Kyung Kyoon Bae, Timothy R. Newman, Joseph Gaeddert, Kyouwoong Kim, Rekha Menon, Lizdabel Morales, Youping Zhao, Jeffrey H. Reed, and William H. Tranter, *A Survey of Artificial Intelligence for Cognitive Radios"*, accepted to IEEE Transactions on Vehicular Technology, to be published in 2010.

4. An He, Timothy R. Newman, Jeffrey H. Reed, Lizdabel Morales, Kyung K. Bae, Chan H. Park, *"Development of a Case-Based Reasoning Cognitive Engine for IEEE 802.22 WRAN Applications,"* accepted to ACM MC2R Special Issue on Cognitive Radio Technologies and Systems, to be published in 2010.

5. An He, Joseph Gaeddert, Timothy Newman, Lizdabel Morales, Jeffrey H. Reed, Kyung K. Bae, Chang-Hyung Park, *"Development of a Case-Based Reasoning Cognitive Engine for IEEE 802.22 WRAN Applications,"* ACM Mobile Communications and Computing Review, April 2009.

6. Y. Zhao, L. Morales, J. Gaeddert, K. K. Bae, J. S. Um; J. H. Reed, *"Applying Radio Environment Maps to Cognitive Wireless Regional Area Networks,"* IEEE Dynamic Spectrum Access Networks (DySpan), April 2007.

7. Y. Zhao, J. Gaeddert, L. Morales, K. K. Bae, J. S. Um, and J. H. Reed, *"Development of Radio Environment Map Enabled Case- and Knowledge-Based Learning Algorithms for IEEE 802.22 WRAN Cognitive Engines,"* International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CrownCom), August 2007.

9.6 Research Related Intellectual Property

1. Youping Zhao, Lizdabel Morales, Kyung Bae, Joseph Gaeddert, Jeffrey Reed and Jungsun Um, *"Case and Knowledge Based Learning (REM-CKL) Method for Cognitive Radios,"* VTIP:07-062.

2. Youping Zhao, Lizdabel Morales, Kyung Bae, Joseph Gaeddert, Jeffrey Reed and Jungsun Um, *"A Generic Cognitive Engine Design Based on Radio Environment Map (REM),"* VTIP:07-060.

Bibliography

[1] A. Shukla, A. Alptekin, J. Bradford, E. Burbidge, D. Chandler, M. Kennett, P. Levine, and S. Weiss, "Cognitive Radio Technology - A Study for Ofcom," tech. rep., QinetiQ Ltd., 2007.

[2] CoRTekS, "http://ossie.wireless.vt.edu/trac/wiki/corteks."

[3] 3G Partnership Project, "http://www.3gpp.org." Published Online.

[4] The Femto Forum, "http://www.femtoforum.org." Published Online.

[5] L. D. Xu, "Case-based reasoning," *IEEE Potentials*, pp. 10–13, December 1994 1994.

[6] Federal Communications Commission, "Spectrum Policy Task Force," tech. rep., FCC, November 2002.

[7] D. Cabric, S. M. Mishra, and R. W. Brodersen, "Implementation issues in spectrum sensing for cognitive radios," in *Proc. IEEE Asilomar Conf. on Signals, Systems and Computers*, 2004.

[8] M. A. McHenry and D. McCloskey, "Spectrum Occupancy Measurements: Chicago, Illinois, November 16-18, 2005," tech. rep., Shared Spectrum Company, 2005.

[9] T. Erpek, K. Steadman, and D. Jones, "Spectrum Occupancy Measurements: Dublin, Ireland, Collected On April 16-18, 2007," tech. rep., Shared Spectrum Company, 2007.

[10] T. Erpek, M. Lofquist, and K. Patton, “Spectrum Occupancy Measurements Loring Commerce Centre Limestone, Maine, September 18-20, 2007,” tech. rep., Shared Spectrum Company, 2007.

[11] J. Mitola, “Cognitive radio for flexible mobile multimedia communications,” *Mobile Networks and Applications*, vol. 6, pp. 435–441, September 2001.

[12] J. Mitola, *Cognitive Radio: An Integrated Agent Architecture for Software Defined Radio.* PhD thesis, Royal Institute of Technology (KTH), 2000.

[13] J. Neel, J. Reed, and R. Gilles, “Convergence of cognitive radio networks,” in *Proc. Wireless Comm. and Networking Conf. (WCNC 2004)*, vol. 4, pp. 2250–2255, 2004.

[14] J. Neel, *Analysis and Design of Cognitive Radio Networks and Distributed Radio Resource Management Algorithms.* PhD thesis, Virginia Polytechnic Institute and State University, 2007.

[15] V. Chandrasekhar, J. Andrews, and A. Gatherer, “Femtocell networks: a survey,” *IEEE Communications Magazine*, vol. 46, pp. 59–67, September 2008.

[16] P. Demestichas, G. Dimitrakopoulos, K. Tsagkaris, K. Demestichas, and J. Adamopoulou, “Reconfigurations selection in cognitive, beyond 3G, radio infrastructures,” in *Proc. 1st International Conference on Cognitive Radio Oriented Wireless Networks and Communications*, pp. 1–5, 8–10 June 2006.

[17] P. Demestichas, G. Dimitrakopoulos, J. Strassner, and D. Bourse, “Introducing reconfigurability and cognitive networks concepts in the wireless world,” *IEEE Vehicular Technology Magazine*, vol. 1, no. 2, pp. 32–39, 2006.

[18] P. Demestichas, G. Dimitrakopoulos, and A. Alexiou, “Reconfiguration techniques to enhance efficiency cognitive networks,” in *Proc. First European Conference on Antennas and Propagation EuCAP 2006*, pp. 1–6, 6–10 Nov. 2006.

[19] Z. Boufidis, N. Alonistioti, and L. Merakos, "RSS: The reconfiguration support subsystem for next-generation mobile networks," in *Proc. IEEE International Conference on Communications*, vol. 4, pp. 1825–1830, June 2006.

[20] J. Gozalvez, D. Martin-Sacristan, M. Lucas-Estan, J. Monserrat, J. Gonzalez-Delicado, D. Gozalvez, and M. Marhuenda, "SPHERE - A simulation platform for heterogeneous wireless systems," in *Proc. 3rd International Conference on Testbeds and Research Infrastructure for the Development of Networks and Communities TridentCom 2007*, pp. 1–10, 21–23 May 2007.

[21] K. Tsagkaris, P. Demestichas, and G. Dimitrakopoulos, "Policies for the management of services in CDMA-based segments of the B3G world," *IEEE Vehicular Technology Magazine*, vol. 2, pp. 21–28, Sept. 2007.

[22] K. Demestichas, E. Adamopoulou, M. Theologou, P. Demestichas, D. Boscovic, and D. Bourse, "Towards Cognitive B3G Networks: Autonomic Management of Access Points," in *Proc. 16th IST Mobile and Wireless Communications Summit*, pp. 1–5, 1–5 July 2007.

[23] G. Dimitrakopoulos, K. Tsagkaris, E. Adamopoulou, K. Demestichas, and P. Demestichas, "Management strategies for distributed cross-layer reconfigurations in the context of cognitive, B3G infrastructures," in *Proc. 16th IST Mobile and Wireless Communications Summit*, pp. 1–5, 1–5 July 2007.

[24] H. Ekstrom, A. Furuskar, J. Karlsson, M. Meyer, S. Parkvall, J. Torsner, and M. Wahlqvist, "Technical solutions for the 3G long-term evolution," *IEEE Communications Magazine*, vol. 44, pp. 38–45, March 2006.

[25] B. Aazhang, J. Lilleberg, and G. Middleton, "Spectrum sharing in a cellular system," in *Proc. IEEE Eighth International Symposium on Spread Spectrum Techniques and Applications*, pp. 355–359, 30 Aug.–2 Sept. 2004.

[26] A. Burnic, A. Viessmann, T. Scholand, C. Spiegel, A. Hessamian-Alinejad, A. Seebens, G. Bruck, and P. Jung, "Synchronization and Channel Estimation in Wireless CDMA Systems," in *Proc. IEEE Ninth International Symposium on Spread Spectrum Techniques and Applications*, pp. 481–487, Aug. 2006.

[27] N. Shah, T. Kamakaris, U. Tureli, and M. Buddhikot, "Wideband spectrum probe for distributed measurements in cellular band," in *TAPAS '06: Proceedings of the first international workshop on Technology and policy for accessing spectrum*, (New York, NY, USA), p. 13, ACM, 2006.

[28] A. Khajeh, S.-Y. Cheng, A. Eltawil, and F. Kurdahi, "Power management for cognitive radio platforms," in *Proc. IEEE Global Telecommunications Conference GLOBECOM '07*, pp. 4066–4070, 26–30 Nov. 2007.

[29] J. Nasreddine, O. Sallent, J. Perez-Romero, and R. Agusti, "Novel Inter-Cell Interaction Approach for WCDMA-based Cognitive Networks," in *Proc. IEEE International Conference on Communications ICC '07*, pp. 4573–4580, 24–28 June 2007.

[30] H. Harada, "Software defined cognitive radio prototype toward imt-advanced wireless communication systems," in *Proc. IEEE Radio and Wireless Symposium*, pp. 7–10, Jan. 2007.

[31] D. Das and V. Ramaswamy, "On the Reverse Link Capacity of a CDMA Network of Femto-cells," in *Proc. IEEE Sarnoff Symposium*, pp. 1–5, 28–30 April 2008.

[32] C. Rieser, *Biologically Inspired Cognitive Radio Engine Model Utilizing Distributed Genetic Algorithm for Secure and Robust Wireless Communications and Networking.* PhD thesis, Virginia Polytechnic Institute and State University, 2004.

[33] T. Blajic, D. Noguilie, and M. Druzijanic, "Latency improvements in 3G long term evolution," tech. rep., 2007.

[34] S. Haykin, “Cognitive radio: Brain-empowered wireless communication,” *IEEE J. on Selected Areas in Comm.*, vol. 23, pp. 201–220, February 2005.

[35] N. Baldo and M. Zorzi, “Learning and adaptation in cognitive radios using neural networks,” in *IEEE CCNC*, pp. 998–1003, January 2008.

[36] J. Peha, “Approaches to spectrum sharing,” *Communications Magazine, IEEE*, vol. 43, no. 2, pp. 10–12, 2005.

[37] R. Thomas, L. DaSilva, and A. MacKenzie, “Cognitive networks,” in *Proc. IEEE Int’l Symp. on New Frontiers in Dynamic Spectrum Access Networks (DySPAN 2005)*, pp. 352–360, 2005.

[38] R. Thomas, D. Friend, L. Dasilva, and A. Mackenzie, “Cognitive networks: adaptation and learning to achieve end-to-end performance objectives,” *Communications Magazine, IEEE*, vol. 44, no. 12, pp. 51–57, 2006.

[39] DARPA XG WG, “The XG Architectural Framework V1.0,” tech. rep., DARPA, 2003.

[40] DARPA XG WG, “The XG Vision RFC V1.0,” tech. rep., DARPA, 2003.

[41] T. W. Rondeau, B. Le, D. Maldonado, D. Scaperoth, and C. W. Bostian, “Cognitive radio formulation and implementation,” in *Proc. 1st International Conference on Cognitive Radio Oriented Wireless Networks and Communications*, pp. 1–10, 8–10 June 2006.

[42] R. W. Brodersen, A. Wolisz, D. Cabric, S. M. Mishra, and D. Willkomm, “Corvus: A cognitive radio approach for usage of virtual unlicensed spectrum,” tech. rep., July 2004.

[43] S. Mishra, D. Cabric, C. Chang, D. Willkomm, B. van Schewick, S. Wolisz, and B. Brodersen, “A real time cognitive radio testbed for physical and link layer experiments,” in *Proc. First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks DySPAN 2005*, pp. 562–567, 8–11 Nov. 2005.

[44] OSSIE, "http://ossie.wireless.vt.edu." Published Online.

[45] J. H. Reed, J. Gaeddert, K. Kim, R. Menon, L. Morales, Y. Zhao, and K. K. Bae, "Applying Artificial Intelligence to the Development of a Cognitive Radio Engine, report for the Army Research Office (ARO)," tech. rep., Virginia Polytechnic and State University, June 2006.

[46] J. H. Reed, C. Dietrich, J. Gaeddert, K. Kim, R. Menon, L. Morales, and Y. Zhao, "Development of a cognitive engine and analysis of wran cognitive radio algorithms," tech. rep., Virginia Polytechnic Institute and State University, December 2005.

[47] J. H. Reed, J. Gaeddert, K. Kim, R. Menon, L. Morales, Y. Zhao, K. K. Bae, and C. M. Spooner, "Development of a cognitive engine and analysis of wran cognitive radio algorithms," tech. rep., Virginia Polytechnic Institute and State University, December 2006.

[48] E. Stuntebeck, T. O'Shea, J. Hecker, and T. C. Clancy, "Architecture for an open-source cognitive radio," in *SDR Forum Technical Conference*, SDR Forum, November 2006.

[49] SOAR, "http://sitemaker.umich.edu/soar/home." Published Online.

[50] C. Clancy, J. Hecker, E. Stuntebeck, and T. O'Shea, "Applications of machine learning to cognitive radio networks," *Wireless Communications, IEEE*, vol. 14, pp. 47–52, August 2007.

[51] Z. Miljanic, I. Seskar, K. Le, and D. Raychaudhuri, "The WINLAB Network Centric Cognitive Radio Hardware Platform: WiNC2R," in *Proc. 2nd International Conference on Cognitive Radio Oriented Wireless Networks and Communications CrownCom 2007*, pp. 155–160, 1–3 Aug. 2007.

[52] T. R. Newman, B. Barker, A. Wyglinski, A. Agah, G. J. Minden, and J. B. Evans, *Cognitive engine implementation for wireless multi-carrier transceivers*, pp. 1–14. John Wiley & Sons, Ltd., May 2007.

[53] N. Baldo and M. Zorzi, "Fuzzy logic for cross-layer optimization in cognitive radio networks," in *4th Consumer Communications and Networking Conference, IEEE*, pp. 1128–1133, January 2007.

[54] J. M. Chapin and L. Doyle, "A Path Forwards for Cognitive Radio Research," in *Cognitive Radio Oriented Wireless Networks and Communications (CrownCom)*, August 2007.

[55] Q. Zhao and B. M. Sadler, ""a survey of dynamic spectrum access"," *IEEE Signal Processing Magazine*, vol. 24, no. 3, pp. 79–89, 2007.

[56] Q. Zhao and A. Swami, "A Survey of Dynamic Spectrum Access: Signal Processing and Networking Perspectives," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE*, vol. 4, pp. 1349–1352, April 2007.

[57] M. Petrova, P. Mahonen, and J. Riihijarvi, "Evolution of Radio Resource Management: A Case for Cognitive Resource Manager with VPI," in *Proc. IEEE International Conference on Communications ICC '07*, pp. 6471–6475, 24–28 June 2007.

[58] H. Holma and A. Toskala, *WCDMA for UMTS : HSPA Evolution and LTE.* John Wiley & Sons, Ltd., 2007.

[59] C. Smith, *3G Wireless Networks.* McGraw-Hill Company, 2007.

[60] Z. Han and K. J. R. Liu, *Resource Allocation for Wireless Networks.* Cambridge, 2008.

[61] H. Harada, H. Murakami, K. Ishizu, S. Filin, Y. Saito, H. N. Tran, G. Miyamoto, M. Hasegawa, Y. Murata, and S. Kato, "A software defined cognitive radio system: Cognitive wireless cloud," in *Proc. IEEE Global Telecommunications Conference GLOBECOM '07*, pp. 294–299, 26–30 Nov. 2007.

[62] H. Harada, “A software defined cognitive radio prototype,” in *Proc. IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications PIMRC 2007*, pp. 1–5, 3–7 Sept. 2007.

[63] N. G. M. N. Alliance, “Use cases related to self organizing network.” May 2007.

[64] I. Motorola, “”LTE operations and maintenance strategy using self-organizing networks to reduce OPEX”.” White Paper, 2009.

[65] N. Corporation, “”self organizing network”.” White Paper, February 2009.

[66] T. W. Rondeau, *Application of Articial Intelligence to Wireless Communications*. PhD thesis, Virginia Polytechnic Institute and State University, 2007.

[67] T. Dietterich and P. Langley, *Cognitive Networks*, ch. Machine Learning for Cognitive Networks: Technology Assessment and Research Challenges. John Wiley & Sons, Ltd., 2007.

[68] N. Shiratori, K. Takahashi, K. Sugawara, and T. Kinoshita, “Using artificial intelligence in communication system design,” *IEEE Software Magazine*, vol. 9, pp. 38–46, Jan. 1992.

[69] P. Langley and H. A. Simon, “Applications of machine learning and rule induction,” *Communications of the ACM*, vol. 38, pp. 54–64, November 1995.

[70] D. Allemang and B. Liver, “Abstractions: their purpose and application in telecommunications,” in *Proc. IEE Colloquium on (Digest No: 1997/094) AI for Network Management Systems*, pp. 8/1–810, 14 April 1997.

[71] M. Kaiser and Z. Cucej, “Artificial intelligence in modern telecommunication systems: Analysis and implementation,” in *4th Eurasip Conference on Video/Image Processing and Multimedia Communications*, July 2003.

[72] D. D. Clark, C. Partridge, J. C. Ramming, and J. T. Wroclawski, “A knowledge plane for the internet,” in *SIGCOMM ’03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pp. 3–10, ACM, August 2003.

[73] L. Kristiansen, “Mobility and Intelligence in Telecom: How and Where to Handle It?,” in *Intelligence in Communication Systems, IFIP International Conference, INTELLCOMM Proceedings*, vol. 3283, pp. 54–69, Springer, 2004.

[74] J. L. Kolodner, *Case-based Reasoning.* Morgan Kaufman Publishers, Inc., 1993.

[75] A. Aamodt and E. Plaza, “Case-based reasoning: Foundational issues, methodological variations, and system approaches,” *AI Communications*, vol. 7, pp. 39–59, 1994.

[76] Z.-W. Ni, S.-L. Yang, L.-S. Li, and R.-Y. Jia, “Integrated case-based reasoning,” in *Proceedings of the Second International Conference on Machine Learning and Cybernetics*, November 2003.

[77] S. Chantaraskul and L. Cuthbert, “Using case-based reasoning in traffic pattern recognition for best resource management in 3G networks,” in *Proceedings of the 7th International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, ACM, 2004.

[78] N. Yao and L. Cuthbert, “Resource management in 3G networks using case-based reasoning,” in *Proceedings of the 8th International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, ACM, 2005.

[79] J. M. Carroll and M. B. Rosson, “Cases as minimalist information,” in *Proceedings of the 38th Hawaii International Conference on System Sciences*, IEEE, 2005.

[80] I. Watson and F. Marir, “Case-based reasoning: A review,” tech. rep., University of Auckland, New Zealand, 1994.

[81] X. Yao and Y. Liu, *Search Methodologies*, ch. Machine Learning, pp. 341–374. New York Springer Science & Business Media, 2005.

[82] J. Quinlan, "Decision trees and decision-making," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 20, pp. 339–346, March–April 1990.

[83] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques.* Morgan Kaufman Publishers, Inc., 2005.

[84] J. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, pp. 81–106, 1986.

[85] T. M. Mitchell, *Machine Learning.* McGraw-Hill Company, 1997.

[86] M. Richardson and J. Warren, "Induced decision trees for case-based reasoning," in *Proc. Australian and New Zealand Conference on Intelligent Information Systems*, pp. 52–55, 18–20 Nov. 1996.

[87] C. Cardie, "Using decision trees to improve case-based learning," in *Proceedings of the Tenth International Conference on Machine Learning*, pp. 25–32, Morgan Kaufman Publishers, Inc., 1993.

[88] C. Tsatsoulis, Q. Cheng, and H.-Y. Wei, "Integrating case-based reasoning and decision theory," *IEEE Expert Magazine*, vol. 12, pp. 46–55, July–Aug. 1997.

[89] E. Hullermeier, "Experience-based decision making: A satisficing decision tree approach," *IEEE Transactions on Systems, Man and Cybernetics, Part A*, vol. 35, pp. 641–653, Sept. 2005.

[90] Y. Zhao, J. H. Reed, S. Mao, and K. K. Bae, "Overhead analysis for radio environment mapenabled cognitive radio networks," in *Proc. SDR '06.1st IEEE Workshop on Networking Technologies for Software Defined Radio Networks*, pp. 18–25, 25–25 Sept. 2006.

[91] Y. Zhao, L. Morales, J. Gaeddert, K. Bae, J.-S. Um, and J. Reed, “Applying radio environment maps to cognitive wireless regional area networks,” in *Proc. 2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks DySPAN 2007*, pp. 115–118, 17–20 April 2007.

[92] Y. Zhao, J. Gaeddert, L. Morales, K. Bae, J.-S. Um, and J. H. Reed, “Development of Radio Environment Map Enabled Case- and Knowledge-Based Learning Algorithms for IEEE 802.22 WRAN Cognitive Engines,” in *International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM 2007)*, 2007.

[93] Y. Zhao, D. Raymond, C. da Silva, J. H. Reed, and S. F. Midkiff, “Performance evaluation of radio environment map-enabled cognitive spectrum-sharing networks,” in *Proc. IEEE Military Communications Conference MILCOM 2007*, pp. 1–7, 29–31 Oct. 2007.

[94] C. da Silva, B. Choi, and K. Kim, “Distributed spectrum sensing for cognitive radio systems,” in *Proc. Information Theory and Applications Workshop*, pp. 120–123, Jan. 29 2007–Feb. 2 2007.

[95] K. Kim, I. Akbar, K. Bae, J. sun Urn, C. Spooner, and J. Reed, “Cyclostationary approaches to signal detection and classification in cognitive radio,” in *Proc. 2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks DySPAN 2007*, pp. 212–215, 17–20 April 2007.

[96] K. W. Kim, *Exploiting Cyclostationarity for Radio Environmental Awareness in Cognitive Radios.* PhD thesis, Virginia Polytechnic Institute and State University, 2008.

[97] M. Ilyas and I. Mahgoub, *Mobile Computing Handbook.* CRC Press, 2005.

[98] Y. Zhao, B. Le, and J. H. Reed, *Cognitive Radio Technology*, ch. Network Support: The Radio Environment Map. Elsevier Inc., 2006.

[99] W. Krenik and C. Panasik, “The potential for unlicensed wide area netwoks.” Texas Instruments White Paper, October 2004.

[100] W. Krenik and A. Batra, “Cognitive radio techniques for wide area networks,” in *42nd Design Automation Conference*, June 2005.

[101] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, ch. Intelligent Agents. Prentice Hall, 1995.

[102] R. Skehill, M. Barry, W. Kent, M. O??callaghan, N. Gawley, and S. Mcgrath, “The common RRM approach to admission control for converged heterogeneous wireless networks,” *IEEE Wireless Communications Magazine*, vol. 14, no. 2, pp. 48–56, 2007.

[103] J. Li, B. Xu, Z. Xu, S. Li, and Y. Liu, “Adaptive packet scheduling algorithm for cognitive radio system,” in *Proc. International Conference on Communication Technology ICCT ’06*, pp. 1–5, Nov. 2006.

[104] S. Albayrak, ed., *Intelligent agents for Telecommunication Applications*, Notes in Artificial Intelligence, Springer, August 1999.

[105] K. Prouskas, A. Patel, J. Pitt, and J. Barria, “A multi-agent system for intelligent network load control using a market-based approach,” in *Proc. Fourth International Conference on MultiAgent Systems*, pp. 231–238, 2000.

[106] G. Pollini, “Trends in handover design,” *IEEE Communications Magazine*, vol. 34, pp. 82–90, March 1996.

[107] B. A. Fette, A. Bensky, R. Aiello, P. Chandra, D. B. Miron, and D. M. Dobkin, *RF & Wireless Technologies.* Newnes, 2007.

[108] R. Di Taranto, K. Nishimori, P. Popovski, H. Yomo, Y. Takatori, R. Prasad, and S. Kubota, “Simple antenna pattern switching and interference-induced multi-hop

transmissions for cognitive radio networks," in *Proc. 2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks DySPAN 2007*, pp. 543–546, 17–20 April 2007.

[109] A. Sayeed and V. Raghavan, "On the impact of reconfigurable antenna arrays in cognitive radio," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2007*, vol. 4, pp. IV–1353–IV–1356, 15–20 April 2007.

[110] I. The MathWorks, "Matlab & simulink." Published Online.

[111] International Telecommunications Union (ITU), "Guidelines for Evaluation of Radio Transmission Technologies for IMT-2000," Tech. Rep. ITU-R M.1225, 1997.

[112] W. H. Tranter, K. S. Shanmugan, T. S. Rappaport, and K. L. Kosbar, *Principles of Communication Systems Simulation with Wireless Applications*. Prentice Hall, 2004.

[113] A. F. Molisch, *Wireless Communications*. John Wiley & Sons, Ltd., 2005.

[114] E. Hyytiä and J. Virtamo, "Random waypoint mobility model in cellular networks," *Wireless Networks*, vol. 13, no. 2, pp. 177–188, 2007.

[115] X. Hong, M. Gerla, G. Pei, and C.-C. Chiang, "A group mobility model for ad hoc wireless networks," in *MSWiM '99: Proceedings of the 2nd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, pp. 53–60, ACM, 1999.

[116] A. Jardosh, E. M. Belding-Royer, K. C. Almeroth, and S. Suri, "Towards realistic mobility models for mobile ad hoc networks," in *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*, pp. 217–229, ACM, 2003.

[117] J. H. Reed, "Development of a cognitive engine and analysis of wran cognitive radio algorithms," tech. rep., Virginia Polytechnic Institute and State University, January 2008.

[118] L. Morales-Tirado, J. E. Suris-Pietri, and J. H. Reed, “A Hybrid Cognitive Engine for Improving Coverage in 3G Wireless Networks,” in *To be presented in IEEE’s ICC 2009 Joint Workshop on Cognitive Wireless Networks and Systems*, 2009.

[119] H. Kwon, M.-J. Yang, A.-S. Park, and S. Venkatesan, “Handover prediction strategy for 3G-WLAN overlay networks,” in *Proc. IEEE Network Operations and Management Symposium NOMS 2008*, pp. 819–822, 7–11 April 2008.

[120] IEEE, “IEEE 802.21 Standard Webpage.” Published Online.

[121] N. D. Tripathi, J. H. Reed, and H. F. V. Landingham, *Radio Resource Management in Cellular Systems.* Springer, Inc., 2001.

[122] M. Mani and N. Crespi, “Wlc14-5: Handover criteria considerations in future convergent networks,” in *Proc. IEEE Global Telecommunications Conference GLOBECOM ’06*, pp. 1–5, Nov. 27 2006–Dec. 1 2006.

[123] M. Gudmundson, “Analysis of handover algorithms [microcellular radio],” in *Proc. st IEEE Vehicular Technology Conference Gateway to the Future Technology in Motion*, pp. 537–542, 19–22 May 1991.

[124] S. S. C. Rezaei and B. H. Khalaj, “Grey prediction based handoff algorithm,” in *World Academy of Science, Engineering and Technology*, vol. 2, pp. 58–61, January 2005.

[125] S. Ulukus and R. D. Yates, “Handover delay in cellular wireless systems,” in *IEEE International Conference on Communications*, 1998.

[126] D. Chambers, *A Femtocell Primer.* Lulu.com, 2008.

[127] A. Prasad, A. Zugenmaier, and P. Schoo, “Next generation communications and secure seamless handover,” in *Proc. Workshop of the 1st International Conference on Security and Privacy for Emerging Areas in Communication Networks*, pp. 267–274, 5–9 Sept. 2005.

[128] T. Hossfeld, S. Oechsner, K. Tutschku, F.-U. Andersen, and L. Caviglione, "Supporting vertical handover by using a pastry peer-to-peer overlay network," in *Proc. Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops PerCom Workshops 2006*, p. 5pp., 13–17 March 2006.

[129] M. Pischella, F. Lebeugle, and S. Jamaa, "UMTS to WLAN Handover based on A Priori Knowledge of the Networks," in *Proc. IEEE International Conference on Communications*, vol. 5, pp. 2009–2013, June 2006.

[130] N. Dailly, P. Martins, and P. Godlewski, "Performance evaluation of l2 handover mechanisms, for inter-radio access networks," in *Proc. VTC 2006-Spring Vehicular Technology Conference IEEE 63rd*, vol. 1, pp. 491–495, 7–10 May 2006.

[131] B. Liu, P. Martins, A. Samhat, and P. Bertin, "A layer 2 scheme for inter-RAT handover between UMTS and WiMAX in tight coupling architecture," in *Proc. IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications PIMRC 2008*, pp. 1–5, 15–18 Sept. 2008.

[132] B. Liu, P. Martins, A. Samhat, and P. Bertin, "A Cross-Layer Scheme for Inter-RAT Handover from WiMAX to UMTS," in *Proc. IEEE Global Telecommunications Conference IEEE GLOBECOM 2008*, pp. 1–6, Nov. 30 2008–Dec. 4 2008.

[133] C.-H. Lin, J.-S. Yang, and K.-C. Wu, "Mobile intelligent agent technologies to support voip seamless mobility," in *Proc. 19th International Conference on Advanced Information Networking and Applications AINA 2005*, vol. 2, pp. 177–180, 28–30 March 2005.

[134] L.-D. Chou, W.-C. Lai, Y.-C. Lin, C.-M. Huang, and C.-H. Lin, "Intelligent Agent over WLAN with Seamless Handover and Load Balancing," in *Proc. International Conference on Communication Technology ICCT '06*, pp. 1–7, 27–30 Nov. 2006.

[135] V. Stavroulaki, Y. Kritikou, and P. Demestichas, "Introducing cognition in the management of equipment in the future wireless world," in *Proc. IEEE 19th International*

Symposium on Personal, Indoor and Mobile Radio Communications PIMRC 2008, pp. 1–5, 15–18 Sept. 2008.

[136] E. Stevens-Navarro, U. Pineda-Rico, and J. Acosta-Elias, "Vertical Handover in beyond Third Generation (B3G) Wireless Networks," *International Journal of Future Generation Communication and Networking*, pp. 51–58, 2009.

[137] P. Marichamy, S. Chakrabarti, and S. Maskara, "Overview of handoff schemes in cellular mobile networks and their comparative performance evaluation," in *Proc. VTC 1999 - Fall Vehicular Technology Conference IEEE VTS 50th*, vol. 3, pp. 1486–1490 vol.3, 1999.

[138] N. Zhang and J. Holtzman, "Analysis of handoff algorithms using both absolute and relative measurements," in *Proc. IEEE 44th Vehicular Technology Conference*, pp. 82–86, 8–10 June 1994.

[139] A. Boukerche, *Handbook of algorithms for wireless networking and mobile computing.* CRC Press, 2006.

[140] K.-I. Itoh, S. Watanabe, J.-S. Shih, and T. Sato, "Performance of handoff algorithm based on distance and rssi measurements," *IEEE Transactions on Vehicular Technology*, vol. 51, pp. 1460–1468, Nov. 2002.

[141] H. Zhu and K. sup Kwak, "Adaptive handoff using distance information," in *Proc. VTC 2006-Spring Vehicular Technology Conference IEEE 63rd*, vol. 2, pp. 932–936, 7–10 May 2006.

[142] F. He, F. Wang, and D. Hu, "Distance and velocity assisted handoff decision algorithm in heterogeneous networks," in *Proc. Second International Conference on Future Generation Communication and Networking FGCN '08*, vol. 1, pp. 349–353, 13–15 Dec. 2008.

[143] M. Austin and G. Stuber, "Velocity adaptive handoff algorithms for microcellular systems," in *nd International Conference on Universal Personal Communications Personal Communications: Gateway to the 21st Century. Conference Record*, vol. 2, pp. 793–797 vol.2, 1993.

[144] M. Austin and G. Stuber, "Velocity adaptive handoff algorithms for microcellular systems," *IEEE Transactions on Vehicular Technology*, vol. 43, no. 3, pp. 549–561, 1994.

[145] C.-N. Chuah and R. Yates, "Evaluation of a minimum power handoff algorithm," in *Proc. Sixth IEEE International Symposium on Personal, Indoor and Mobile Radio Communications PIMRC'95. 'Wireless: Merging onto the Information Superhighway'*, vol. 2, pp. 814–818 vol.2, 1995.

[146] C. N. Chuah, R. Yates, and D. Goodman, "Integrated dynamic radio resource management," in *Proc. IEEE 45th Vehicular Technology Conference*, vol. 2, pp. 584–588 vol.2, 1995.

[147] T.-H. Kim, Q. Yang, J.-H. Lee, S.-G. Park, and Y.-S. Shin, "A Mobility Management Technique with Simple Handover Prediction for 3G LTE Systems," in *Proc. VTC-2007 Fall Vehicular Technology Conference 2007 IEEE 66th*, pp. 259–263, Sept. 30 2007–Oct. 3 2007.

[148] J. Martinez-Bauset, J. Gimenez-guzman, and V. Pla, "Optimal admission control in multimedia mobile networks with handover prediction," *IEEE Wireless Communications Magazine*, vol. 15, pp. 38–44, October 2008.

[149] C.-H. Lee and C.-J. Yu, "An intelligent handoff algorithm for wireless communication systems using grey prediction and fuzzy decision system," in *Proc. IEEE International Conference on Networking, Sensing and Control*, vol. 1, pp. 541–546 Vol.1, 2004.

[150] J. Kamal, M. Junaebur Rashid, T. Islam, and H. Sarwar, “An intelligent handoff decision algorithm,” in *Proc. International Conference on Computer and Communication Engineering ICCCE 2008*, pp. 37–41, 13–15 May 2008.

[151] X. Yan, N. Mani, and Y. Cekercioglu, “A Traveling Distance Prediction Based Method to Minimize Unnecessary Handovers from Cellular Networks to WLANs,” *IEEE Communications Letters*, vol. 12, no. 1, pp. 14–16, 2008.

[152] K. Al Agha, T. Al-Meshhadany, and L. Perato, “Resource allocation based on handoff prediction in WCDMA,” in *Proc. VTC 2002-Fall Vehicular Technology Conference 2002 IEEE 56th*, vol. 1, pp. 127–131 vol.1, 2002.

[153] L. Perato and K. Al Agha, “Handover prediction: user approach versus cell approach,” in *Proc. 4th International Workshop on Mobile and Wireless Communications Network*, pp. 492–496, 2002.

[154] L. Perato and K. Agha, “Handover prediction for cellular systems in a multi-services context,” in *Proc. 5th International Symposium on Wireless Personal Multimedia Communications*, vol. 1, pp. 67–71 vol.1, 2002.

[155] B.-J. Chang and J.-F. Chen, “Cross-Layer-Based Adaptive Vertical Handoff With Predictive RSS in Heterogeneous Wireless Networks,” *IEEE Transactions on Vehicular Technology*, vol. 57, pp. 3679–3692, Nov. 2008.

[156] S. M. Mousavi, H. R. Rabiee, M. Moshref, and A. Dabirmoghaddam, “Mobisim: A framework for simulation of mobility models in mobile ad-hoc networks,” in *Third IEEE international Conference on Wireless and Mobile Computing, Networking and Communications*, October 2007.

[157] P. Prabhakaran and R. Sankar, “Impact of realistic mobility models on wireless networks performance,” in *Proc. (WiMob’2006) Wireless and Mobile Computing, Networking and Communications IEEE International Conference on*, pp. 329–334, 19–21 June 2006.

[158] B. Pazand and C. McDonald, “A critique of mobility models for wireless network simulation,” in *Proc. 6th IEEE/ACIS International Conference on Computer and Information Science ICIS 2007*, pp. 141–146, 11–13 July 2007.

[159] M. Fiore, “Mobility models in inte-vehicle communications literature,” tech. rep., Politecnico di Torino, 2006.

[160] M. M. Javadi, “Mobisim a framework to manage mobility models,” tech. rep., Sharif University of Technology, 2006-2008.

[161] R. Chadha and L. Kant, *Policy-driven mobile ad hoc network management.* John Wiley & Sons, Ltd., 2008.

[162] M. Koziol and M. Wozniak, “Multivariate decision trees versus univariate ones,” in *Computer Recognition Systems* (M. Kurcynski and M. Wozniak, eds.), vol. 3, Springer, 2009.

Appendix

Appendix A

Sample Rules for CLE

Coverage Learning Engine Rules

Sample case: Reads 80000 cases (4 attributes).

- Rule 9: $x > 1.38826, \theta > -0.186259, \theta \leq 0.187019$, then class 0, with $p = 99.7\%$
- Rule 11: $x > 0.500936, \theta > -0.182438, \theta \leq 0.187019$, then class 0, with $p = 99.6\%$
- Rule 14: $R > 1.8129, \theta > -0.189561, \theta leq 0.189072$, then class 0, with $p = 99.6\%$
- Rule 3: $x > 0.500936, y > -0.19949, R \leq 0.726296, \theta \leq 0.196606$, then class 0, with $p = 99.1\%$
- Rule 17: $x > 0.500936, y \leq 0.19513, R \leq 0.716611, \theta > 0.196606$, then class 0, with $p = 96.7\%$
- Rule 22: $x \leq 2.36099, y > 3.98481, \theta <= 1.14328$, then class 0, with $p = 74.1\%$
- Rule 13: $R > 0.692618, R \leq 1.8129, \theta > 0.187019$, then class 1 with $p = 99.3\%$
- Rule 8: $x \leq 1.38826, R > 0.724814, R \leq 1.89371, \theta \leq -0.182438$, then class 1, with $p = 99.1\%$

Table A.1: Resulting Rules in CLE Case Study

Rule	Size	Error	Used	Wrong	%
9	3	0.3%	3031	3	0.1%
11	3	0.4%	1356	7	0.5%
14	3	0.4%	30	4	13.3%
3	4	0.9%	82	0	0.0%
17	4	3.3%	90	0	0.0%
22	3	25.9%	10	0	0.0%
13	3	0.7%	15144	74	0.5%
8	4	0.9%	15256	112	0.7%
20	3	2.8%	20298	854	4.2%
15	4	2.9%	106	11	10.4%
1	1	3.1%	15761	841	5.3%
23	3	3.3%	221	52	23.5%
2	2	3.8%	8516	478	5.6%

- Rule 20: $y > 0.19513, R \leq 4.18057, \theta > 0.196606$, then class 1, with $p = 97.2\%$

- Rule 15: $x > 0.500936, x \leq 2.26378, R > 0.692618, \theta > 0.189072$ then class 1, with $p = 97.1\%$

- Rule 1: $x \leq 0.500936$, then class 1, with $p = 96.9\%$

- Rule 23: $x \leq 4.03474, y > 0.19513, \theta > 0.196606$, then class 1, with $p = 96.7\%$

- Rule 2: $y \leq -0.19949, \theta \leq -0.189561$, then class 1, with $p = 96.2\%$

Evaluation on training data (80000 items):

Tested 80000 cases, found 2483 errors, resulting in 3.1% total classification error.

Appendix B

Acronyms

3G Third Generation

3GPP Third Generation Partnership Project

4G Fourth Generation

AI Artificial Intelligence

AMR Adaptive Multirate

ARO Army Research Office

AWGN Additive White Gaussian Noise

B3G Beyond 3G

BER Bit Error Rate

BSC Base Station Controller

BTS Base Transceiver Station

CAC Call Admission Control

CBR Case-based Reasoning

CDMA Code Division Multiple Access

CE cognitive engine

CNS Communications Network Services

CR cognitive radio

CRRM Cognitive Radio Resource Management

CoRTekS Cognitive Radio Tektronix System

CLE Coverage Learning Engine

DARPA Defense Advanced Research Projects Agency

DSA dynamic spectrum access

DT Decision Tree

DTL Decision Tree Learning

ETRI Electronics and Telecommunications Research Institute

FCC Federal Communications Commission

FDD Frequency Division Duplex

GGSN Gateway GPRS Support Node

GPRS General Packet Radio Service

GPS Global Positioning System

GSM Global System for Mobile Communications

HLR Home Location Register

HME Handover Management Engine

HSDPA High Speed Downlink Packet Access

IA Intelligent Agent

IBR Instance-based Reasoning

IEEE Institute of Electrical and Electronics Engineers

ITU International Telecommunication Union

LTE long term evolution

MAC Media Access Control

ML Machine Learning

MPRG Mobile Portable Radio Group

MSC Mobile Switching Center

NES Network Environment Simulator

NTIA National Telecommunications and Information Administration

OfCom Office of Communications, UK

OSI Open Systems Interconnection

OSSIE Open Source SCA Implementation-Embedded

PHY Physical Layer

PE Policy Engine

P2P peer-to-peer

QoS quality of service

RAN Radio Access Network

RAT Radio Access Technology

REM Radio Environment Map

ROC Receiver Operating Characteristics

RF radio frequency

RNC radio network controller

RRM radio resource management

RSS Received Signal Strength

RWP Random Waypoint

SCA Software Communications Architecture

SDR Software Defined Radio

SON Self-Organizing Networks

SGSN Serving GPRS Support Node

SIR Signal-to-Interference Ratio

SINR Signal-to-Interference and Noise Ratio

TDD Time Division Duplex

UE User Equipment

UMTS Universal Mobile Telecommunications System

UTRAN Universal Terrestrial Radio Access Network

WCDMA Wideband-Code Division Multiple Access

WLAN Wireless Local Area Networks

WRAN Wireless Regional Area Networks

XG neXt Generation