Aalto University
School of Electrical Engineering
Master's Programme in Communications Engineering

# Practical Implementation of Cognitive Radio Architecture for Radio Resource Management and Control

Anil Morab Vishwanath

Master's Thesis
Espoo, September 30, 2013

| | |
|---|---|
| Supervisor: | Professor Olav Tirkkonen |
| Advisor: | D.Sc. (Tech.) Kalle Ruttik |

Aalto University
School of Electrical Engineering
Master's Programme in Communications Engineering

ABSTRACT OF
MASTER'S THESIS

|  |  |  |  |
|---|---|---|---|
| **Author:** | Anil Morab Vishwanath |  |  |
| **Title:**<br>Practical Implementation of Cognitive Radio Architecture for Radio Resource Management and Control |  |  |  |
| **Date:** | September 30, 2013 | **Pages:** | xi + 77 |
| **Major:** | Department of Communications and Networking | **Code:** | S-72 |
| **Supervisor:** | Professor Olav Tirkkonen |  |  |
| **Advisor:** | D.Sc. (Tech.) Kalle Ruttik |  |  |

Cognitive radio is conceived to solve the problem of spectrum scarcity as it enables opportunistic use of radio spectrum, allowing unlicensed users to utilize licensed bands. The inherent flexibility offered by these systems, also pose a challenge for establishing equally flexible architecture in order to support them. This work includes design and implementation of one such robust Cognitive Radio Architecture (CRA) for radio resource management (RRM) and control. The established architecture provides a software platform for controlling physical layer resources and to investigate different resource management algorithms. Key network elements and interfaces are developed along the lines of Long term Evolution (LTE) systems. A Distributed channel allocation algorithm called Greedy distributed local search (GDLS) is implemented to demonstrate the self-organization capability of the system. The established CRA can communicate with external networks and offer services like paid-for premium video delivery to the latter. The feasibility of using scripting languages for developing such complex network architectures was assessed. The system was further evaluated for its functionality and performance in terms of speed, stability and scalability.

**Keywords:** Cognitive Radio, Cognitive Radio Architecture, Radio Resource Management, LTE, Self-organization, video delivery

**Language:** English

# Acknowledgements

This work is part of End-to-End Cognitive Radio Testbed (EECRT) project at Department of Communications and Networking, Aalto University. This thesis has been quite a journey and would have been impossible to accomplish without the support and encouragement of numerous people. At this moment, I would like to thank all of them who made it possible.

I am also extremely indebted to Prof. Olav Tirkkonen for accepting me into his research group. I am thankful for his patient guidance, valuable advice and his extensive discussions around my work. This work would not have been possible without his supervision, support and encouragement.

My sincere gratitude to D.Sc.(Tech) Kalle Ruttik, for providing me with an excellent atmosphere for doing research. Without your time, encouragement and effort, this thesis would not have been completed or written. One simply could not wish for a better or friendlier instructor.

I would like to thank my teammates Vishnu, Gerrardo, Yihenew and Liang for their kind cooperation and for those numerous discussions we had during this project. I sincerely thank my friends, Vishnu, Pradeep and Ashvin for motivating me to write this thesis. This thesis has taken its shape because of your help and suggestions.

Special regards to Monika, for being there beside me through the good times and bad. A journey is lot easier when you travel together. Lastly, but most importantly, I would like to thank my family: my grandmother, my younger brother and my parents for their sincere encouragement and inspiration. Words fail me to express my appreciation. I owe everything to them. Finally, thank you almighty for illuminating the darkest passages of life and leading me through it.

Thank you,
Anil Morab Vishwanath
Espoo, 15th Sep 2013

# Contents

# List of Figures

# List of Tables

# Abbreviations and Acronyms

| | |
|---|---|
| 3GPP | 3rd Generation Partnership Project |
| AMC | Adaptive Modulation and Coding |
| AS | Access Stratum |
| BS | Base Station |
| C-eNodeB | Cognitive-eNodeB |
| CN | Core network |
| CQI | Channel Quality Index |
| CR | Cognitive Radio |
| CRA | Cognitive Radio Architecture |
| CRN | Conitive Radio Network |
| CWC | Cellular worst Coupling |
| DSM | Dynamic Spectrum Manager |
| eNodeB | evolved-NodeB |
| EPC | Evolved Packet Core |
| EPS | Evolved Packet System |
| E-UTRAN | Evolved Universal Mobile Telecommunications System- Radio access network |
| FCC | Federal Communications Commission |
| FDD | Frequency division duplex |
| GDLS | Greedy distributed local search |
| GTP | General Packet Radio Service- Tunneling Protocol |
| HARQ | Hybrid Automatic Repeat request |
| HSS | Home Subscription Server |
| ICIC | Inter Cell Interference Coordination |
| IP | Internet Protocol |
| L-RRM | Local Radio resource Manager |
| LTE | Long term Evolution |
| MAC | Medium Access Control |
| MIMO | Multiple input Multiple output |
| MME | Mobility Management Entity |

| | |
|---|---|
| NAS | Non Access Stratum |
| N-RRM | Network Radio resource Manager |
| OFDM | Orthogonal Frequency Division Multiplexing |
| PAWS | Protocol to Access White Space databases |
| PCRF | Policy and Charging Resource Function |
| PDCP | Packet Data Convergence Protocol |
| P-GW | Packet Data Network Gateway |
| PU | Primary User |
| QoS | Quality of Service |
| RAN | Radio Access Network |
| RAT | Radio Access Technology |
| RB | Resource Block |
| RLC | Radio Link Control |
| RRC | Radio Resource Control |
| RRM | Radio Resource Management |
| RSS | Receive Signal Strength |
| RTT | Round Trip Time |
| SAE | System Architecture Evolution |
| SCTP | Stream Control Protocol |
| SDLS | Semi- Greedy Distributed Local search |
| SDR | Software Defined Radio |
| S-GW | Serving Gateway |
| SU | Secondary user |
| TDD | Time division duplex |
| TFT | Traffic Flow Template |
| UE | User Equipment |
| URI | Uniform resource identifier |
| USRP | Universal Software Radio Peripherals |
| WLAN | Wireless Local Area Network |

# Chapter 1

# Introduction

## 1.1 Overview

Increasing demands for capacity and high spectral efficiency by bandwidth-thirsty applications such as mobile gaming, video streaming etc are steering the technological evolution in wireless communications systems today. According to [6], mobile traffic has increased rapidly at about 130% annually over the last 5 years, generating a huge gap between the customer demand and the technologies enabling them. Today, relying on sophisticated modulation techniques alone is no longer an option to address this traffic explosion. Introducing new advanced technologies that provide improved capacity and efficient spectrum utilization hence is of paramount importance.

Long Term Evolution (LTE) is one solution proposed by the 3rd Generation Partnership Project (3GPP) to meet the bandwidth demands. In contrast to the traditional circuit-switched model of previous cellular systems, LTE supports only packet-switched services. It aims to provide seamless connectivity to its users with no disruption even during high mobility scenarios. Release 8 of LTE provides peak rates of 300 Mb/s, a radio-network delay of less than 5 ms and a new flat radio-network architecture designed to simplify operation. It also enables asymmetric spectrum utilization by providing the possibility for different uplink and downlink bandwidths. Even though LTE is the front runner in sufficing current demands, trend remains that wireless networks receive minimal spectrum efficiency gain by improving just the air interface technologies alone.

Currently, different services are allocated with different frequency bands and no violations from unlicensed users are allowed. However, a survey of spectrum utilization [7] made by the Federal Communications Commission (FCC) has indicated that a large portion of the actual licensed spectrum is used sporadically, resulting in spectral inefficiency. In this context, the capabilities of Cognitive radio (CR) offer the possibility to significantly enhance the performance of the wireless systems. Cognitive Radio, a term coined by Mitola [8], is a smart radio that is aware of its surroundings at all instances and adapts its behavior based on the knowledge acquired. In CR, users with no licenses, also called secondary users (SU), check the spectrum availability from time to time and choose the idle channels for communication. Once the primary user (PU) needs the channel, the SU switches to a different idle channel if available (to avoid interference with PU) or otherwise terminates the transmission altogether, thereby exploiting the underutilized spectrum opportunistically. It was also established by Mitola that an architecture was necessary for establishing a cooperating platform for such environment-aware nodes. Most of the research on cognitive radio architecture focus on providing basic cognitive frameworks [9], [10], [11], [12] designed to solve specific communication problems. Implementation on CR systems are limited to testbed developments [13], [14], [15]. No significant work has been done so far in establishing general control architectures for cognitive radio systems, a primary focus of this thesis. Implementation of CR assisted LTE networks was only done recently in [16].

## 1.2 Objective of the thesis

The extreme flexibility of cognitive radios has significant implications for the design of network algorithms and protocols. In particular, support for cross-layer algorithms which adapt to changes in physical link quality require an equally flexible architecture. The objective of this work is to design and implement a general architecture for Cognitive Radio Systems for the purpose of Radio Resource Management (RRM). The implemented Cognitive Radio Architecture (CRA) should serve as a platform for investigating different resource management algorithms. It should be aware of its environment and configure parameters based on the derived knowledge to produce RRM decisions and actions. This architecture should also be LTE compliant so that it has the capabilities to match the current standards for their performance. The feasibility of using scripting languages to implement such complex architectures is also to be tested. This not only makes the implementation faster, but also makes it easier to add more functionality with minimal degradation in performance. Interactions with external networks should be realized, for the latter to utilize the predefined services offered by the CRA. Resource Management algorithms that demonstrate the Cognitive and self organization capabilities of the established system should be implemented.

## 1.3 Thesis Organization

The organization of this thesis is as follows. Chapter 2 gives a basic introduction to LTE/LTE-A systems. The basic architecture of LTE and the network elements and interfaces it comprises of, is discussed. An overview of the Radio Resource Management entity and many protocols that formulate it is also provided. Chapter 3 gives a basic overview of cognitive radio networks and the key technologies that enable it. Self organization in Cognitive Radio networks and a few distributed channel allocation algorithms are also discussed here. A Proof of concept implementation of the Cognitive Radio architecture, defined in this work, is introduced in Chapter 4. The working of different network elements and interfaces is explained. Chapter 5 discusses the current capabilities of the system. The signaling models used for providing functionality such as radio access, interaction with Geo-location database, resource negotiation, self organization and providing service to external networks are explained. Chapter 6 provides the results obtained while evaluating the functionality and performance of the established architecture. Results obtained during examining the performance of GDLS protocol is indicated. The implemented CRA is also evaluated for its simplicity, stability, scalability and speed. Lastly, Chapter 7 concludes the thesis by presenting the main observations on practical implementation of the system. The future scope on extending the current architecture to include more components and services is also discussed.

# Chapter 2

# LTE System Architecture

This chapter gives an overview of the network architecture of a Long Term Evolution (LTE) system according to the Release 8 version. An overview of the functions provided by the core network (CN) and E-UTRAN is given. The protocol stack across the different interfaces, along with the functionality is briefly discussed.

## 2.1 Introduction

The recent increase in data usage by mobile terminals was the prime motivation for 3rd Generation Partnership Project (3GPP) to work on the Long-Term Evolution (LTE). With its highly flexible radio interface, LTE substantially improves end-user throughput, sector capacity and reduces user plane latency thereby significantly improving the user experience. It provides peak rates of 300 Mb/s, a radio-network delay of less than 5 ms [17]. Since IP protocol is being tipped as the favorite for carrying all types of traffic, LTE in contrast to the circuit-switched model of previous cellular systems, provides support for only packet switched services. It aims to provide seamless IP connectivity to its users, with no disruption in its service even during high user mobility. It supports both frequency-division duplex (FDD) and time-division duplexing (TDD). It relies heavily on physical layer technologies such as Orthogonal Frequency Division Multiplexing (OFDM) and Multiple-Input Multiple-Output (MIMO) systems to achieve its targets. LTE was also designed to minimize the system and User Equipment(UE) complexities, allow flexible spectrum deployment in existing or new frequency spectrum and to enable co-existence with other 3GPP Radio Access Technologies (RATs). The general network architecture of LTE is provided in the next section.

## 2.2 LTE Network architecture

Along with the enhancements in Radio Access technology, 3GPP also revisited the overall system architecture of both the Radio-Access Network (RAN) and the Core Network (CN). This work, also known as System Architecture Evolution (SAE), resulted in a flat RAN architecture, as well as a new core network architecture, called Evolved Packet Core (EPC). Together RAN and SAE formulate the Evolved Packet System (EPS). The RAN is responsible for all radio related functionality such as scheduling, radio-resource handling, retransmission, coding, multi-antenna schemes etc. The EPC on the other hand is responsible for functions not related to the radio interface but needed for providing a mobile broadband network [18]. This includes authentication, charging, setting up of end-to-end connections etc. EPS uses the concept of bearers to route IP traffic from the core network to the UE. A bearer is an IP packet flow with a predefined quality of service (QoS) between the core and the UE. The E-UTRAN and EPC together set up and release these bearers as required by applications. A single user can be provided with multiple bearers based on the type of application.

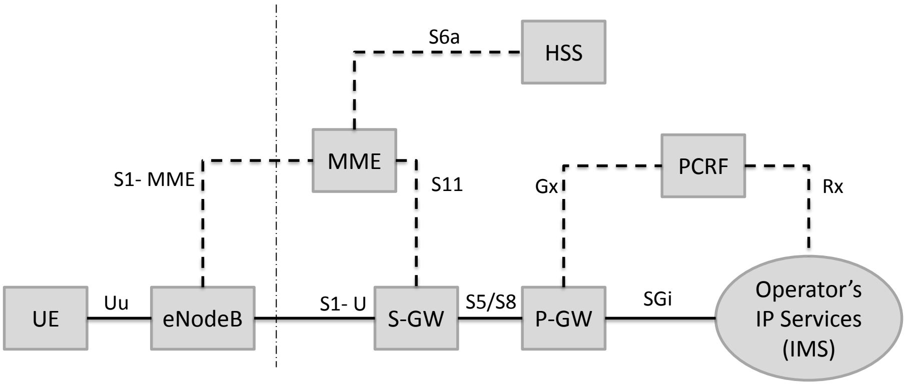

Figure 2.1: EPS network elements

Figure 2.1 shows the overall LTE network architecture, including the network elements and the standardized interfaces. While the EPC consists of many logical nodes to perform tasks such as authentication, bearer management etc, LTE RAN constitutes of essentially just one node, the evolved NodeB (eNodeB). Each of these network elements is interconnected by standardized interfaces that allow multi-vendor interoperability [19]. Figure 2.2 describes the functional split between the EPC and E-UTRAN in LTE.

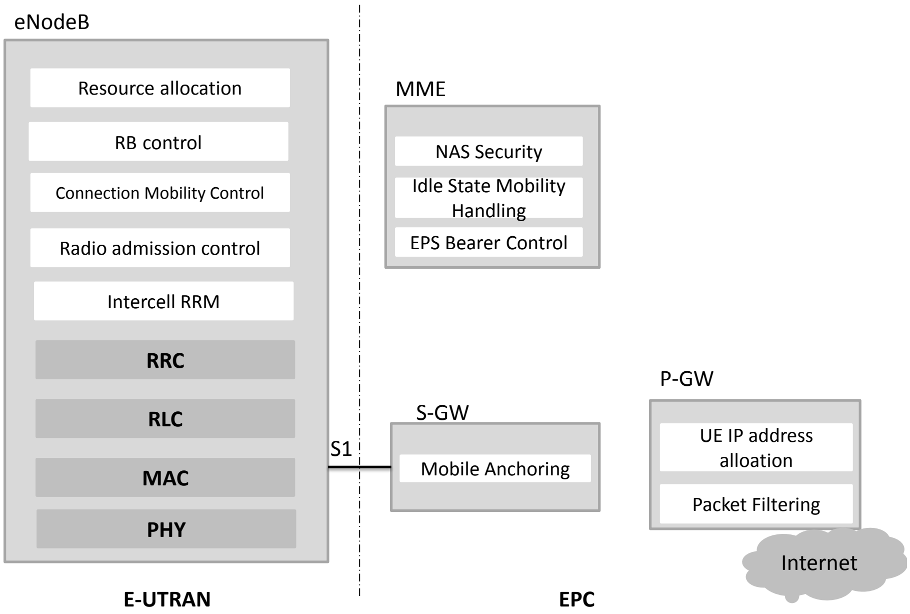

Figure 2.2: Functional split between E-UTRAN and EPC

The network elements that make up the EPC and E-UTRAN are described in detail in the following subsections.

### 2.2.1 Core network

The core network is responsible for the overall control of the UE and establishment of the associated bearers to it. The main logical elements of the core network are:

- **MME**: The Mobility Management Entity (MME) is the main control node that processes the signaling between the UE and the CN. It is responsible for Authentication and security, Mobility management, Idle mode UE tracking and paging procedure including retransmissions. It is also responsible for managing Subscription Profile and Service Connectivity of the UEs as well as bearer activation/deactivation process [20]. The protocols running between the UE and the CN are known as the Non Access Stratum (NAS) protocols.

- **P-GW**: The PDN Gateway (P-GW) connects the EPC to the internet. It is also the exit point for traffic from the UE. P-GW is responsible for allocating IP address to every UE attached to the system, as well as enforcing QoS and flow-based charging policies, according to rules set in PCRF. It is also responsible for the filtering of downlink user IP packets into different QoS-based bearers based on their Traffic Flow Templates (TFTs) [21]. It also acts as a mobility anchor between 3GPP and non-3GPP technologies.

- **S-GW**: Serving Gateway (S-GW) acts as the local mobility anchor for data bearers when UE moves between eNodeBs. All user IP packets are transferred to the access network through S-GW. It manages and stores UE contexts such as, IP bearer service parameters, routing information etc. It also retains the information about the bearers when the UE is in the idle state and temporarily buffers downlink data while the MME initiates paging of the UE to reestablish the bearers [22]. S-GW performs some administrative functions such as replication of the user traffic in case of lawful interception, collecting information for charging from visited network etc. It also serves as the mobility anchor for interworking with other 3GPP technologies.
- **HSS**: Home Subscription Server (HSS) is a database that contains subscription information about all permanent users. It stores the master copy of the subscriber profile, including information about feasibility of roaming into a particular visited network, the allowed PDN connections etc. For supporting mobility between non-3GPP technologies, HSS also stores the identities of those P-GWs that are in use. It also records the identity of the MME to which the user is currently attached or registered [20]. The HSS may also integrate the authentication center (AUC), which generates the vectors for authentication and security keys.
- **PCRF**: Policy and Charging Resource Function (PCRF) is the network element that is responsible for Policy and Charging Control (PCC) in LTE networks. It makes decisions on how to handle the services in terms of QoS, and provides QoS authorization information to the PCEF located in the P-GW, so that appropriate bearers and policing can be set up in accordance with the user subscription profile.

### 2.2.2 Access network

The LTE Radio Access network typically consists of a single entity, eNodeB which takes care of all radio related functionality of the system, thereby deeming the architecture to be flat. LTE E-UTRAN hence has only two network elements:

- **UE**: User Equipment (UE) is the device that the end user uses for communication. It provides user interface for communication applications and services offered to the users. UE is responsible for signaling the network for setting up, maintaining and removing the communication links when necessary. It also performs tasks instructed by the eNodeB, such as handovers and reporting the terminals location etc.

- **eNodeB**: The eNodeB acts as a layer 2 bridge between UE and the EPC, by being the terminating point for all the radio protocols towards the UE, and relaying data between the radio connection and the corresponding IP based connectivity towards the EPC [20]. To facilitate this functionality, eNodeB performs ciphering/deciphering of the user plane data, header compression/decompression of IP packets for avoiding significant IP overhead. It is also responsible for the Radio Resource Management (RRM) functions like radio bearer control, radio admission control, radio mobility control, scheduling and dynamic allocation of resources to UEs. In addition, the eNodeB also plays an important role in Mobility Management by taking decisions to handover UEs between cells based on the radio signal level measurements sent out by the UEs.

LTE integrates all radio controller functions into the eNodeB mostly because there is no need to support soft handovers unlike in previous technologies. This allows for tight interaction between different protocol layers thereby providing improved efficiency and reduced latency. The protocols that run between the eNodeBs and the UE is called as the Access Stratum (AS) protocols, the architecture of which can be seen in the next section.

## 2.3 Protocol architecture

This section describes the functions of the different protocol layers and their location in the LTE architecture.

### 2.3.1 User plane protocols

Figure 2.3 shows the user plane protocol stack of LTE RAN architecture. The user-plane runs the AS protocols (indicated in blue) between the eNodeB and the UE and is usually responsible for delivering the data to the UE by performing necessary functions such as header compression, segmentation and assembly etc.

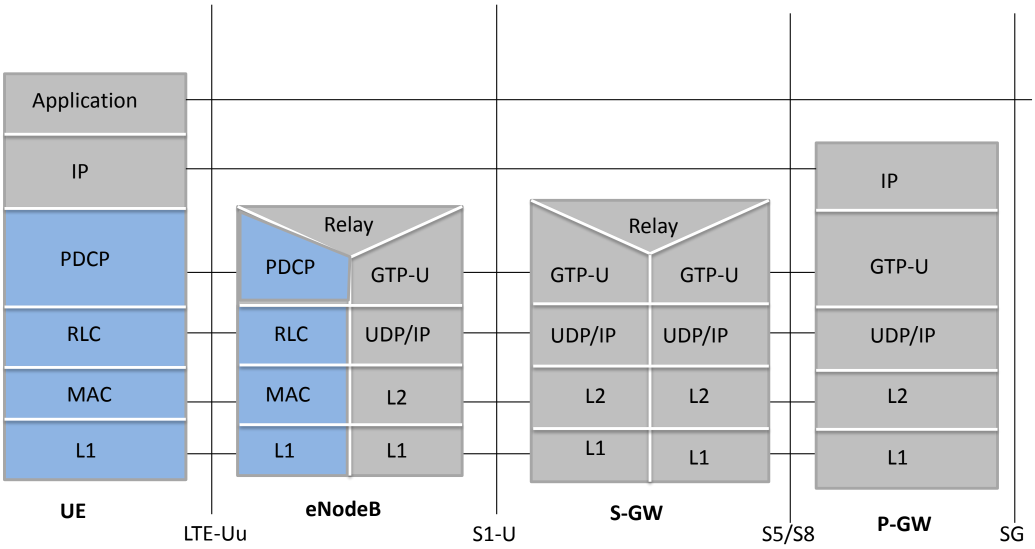

Figure 2.3: User plane protocol stack in LTE

The packet that has to be delivered to the UE is encapsulated in a GPRS Tunneling Protocol (GTP) packet and is tunneled between the P-GW and the eNodeB for transmission. The E-UTRAN user plane protocol stack consists of the following protocols, Packet Data Convergence Protocol (PDCP), Radio Link Control (RLC) and Medium Access Control (MAC) sublayers that are terminated in the eNodeB on the network side. A brief description of each of these protocols is given subsequently.

#### 2.3.1.1 PDCP

PDCP layer is responsible for compression and decompression of IP headers, which is necessary for reducing the number of bits transmitted over the radio interface. It uses Robust Header Compression(ROHC) algorithm for performing this task. It is also responsible for ciphering of both user plane and control plane data as well as integrity protection and in-sequence delivery of the transmitted data.

#### 2.3.1.2 RLC

Radio-Link Control (RLC) layer is used to format and transport traffic between the UE and the eNodeB. It is also responsible for segmentation/concatenation, retransmission handling, duplicate detection, and in-sequence delivery to higher layers. It provides three different reliability modes for transporting the user-plane traffic. The Acknowledged Mode (AM) is usually preferred for non real time traffic such as file downloads, the Unacknowledged Mode (UM) on the contrary is used for transmitting real time traffic such as voip. The Transparent Mode (TM) mode is used when prior knowledge about the PDU size is available at the eNodeB, a mechanism suitable for transmitting broadcast messages. In LTE, RLC supports variable PDU sizes in contrast to earlier mobile-communication technologies thereby providing a range of datarates from few kbps upto few Gbps. For high data rates, larger PDU sizes are transmitted resulting in a smaller relative overhead, while for low data rates, smaller PDUs are transmitted. By monitoring the sequence numbers of the incoming PDUs, RLC can identify the missing PDUs and send a request for retransmission if necessary. RLC also provides a single bit error-feedback mechanism called outer ARQ for handling the residual errors that are not corrected by the MAC's the Hybrid Automatic Repeat request (HARQ) mechanism.

#### 2.3.1.3 MAC

The main responsibilities of MAC layer is to handle multiplexing of logical channels to transport channels along with hybrid-ARQ retransmissions (HARQ) for error handling.

Table 2.1: Logical channels in LTE

| Channel | Type | UL /DL | Description |
|---|---|---|---|
| Broadcast Control Channel (BCCH) | Control | DL | Broadcasting System Control info |
| Paging Control Channel (PCCH) | Control | DL | Transferring Paging info |
| Common Control Channel (CCCH) | Control | UL | Transmitting system info (UE without RRC connection) |
| Multicast Control Channel (MCCH) | Control | DL | Transmitting MBMS Control info |
| Dedicated Control Channel (DCCH) | Control | DL | Transmitting Control info(UE with RRC connection) |
| Multicast Traffic Channel (MTCH) | Traffic | DL | Transmitting MBMS traffic |
| Dedicated Traffic Channel (DTCH) | Traffic | - | Bidirectional channel dedicated to single UE |

MAC provides services to the RLC in the form of logical channels. Based on the type of information carried, logical channels are divided into control channel: used for transmission of control and configuration information, or traffic channel: used for the user data. A brief description of each of these channels is provided in table 2.1. MAC uses services of physical layer in the form of transport channels. A transport channel is defined by how and with what characteristics the information is transmitted over the radio interface. Data on a transport channel is organized into transport blocks. In each Transmission Time Interval (TTI), at most one transport block of dynamic size is transmitted over the radio interface. A brief description of each of Transport channels is provided in table 2.2.

Table 2.2: Transport channels in LTE

| Channel | UL /DL | Description |
|---|---|---|
| Broadcast Channel (BCH) | DL | Transporting Master Information Block (MIB) |
| Paging Channel (PCH) | DL | Transporting paging information |
| Downlink Shared Channel (DL-SCH) | DL | HARQ, Dynamic link adaptation, support for UE DRX, dynamic resource allocation |
| Multicast Channel (MCH) | DL | SFN combining and static resource allocation |
| Uplink Shared Channel (UL-SCH) | UL | HARQ, Dynamic link adaptation, support for UE DRX, dynamic resource allocation |
| Random-Access Channel (RACH) | UL | Limited Control info, collision risk |

The MAC layer performs the mapping between the logical channels and transport channels, schedules the different UEs and their services in both UL and DL depending on their relative priorities, and selects the most appropriate transport format. The mapping of the logical channels to the transport channels is shown in figure 2.4. Scheduler in MAC layer is responsible for the assignment of uplink and downlink resources in terms of Resource Block (RB) pairs. Each RB corresponds to a time–frequency unit of 1 ms times 180 kHz. Scheduler in each 1 ms interval takes a scheduling decision and sends scheduling information to the selected UE. Coordination of scheduling decisions between eNodeBs is supported using signaling over the X2 interface. PDCP, RLC and MAC together for the LTE L2 protocol stack. More information on each of these L2 protocols is available in [1].

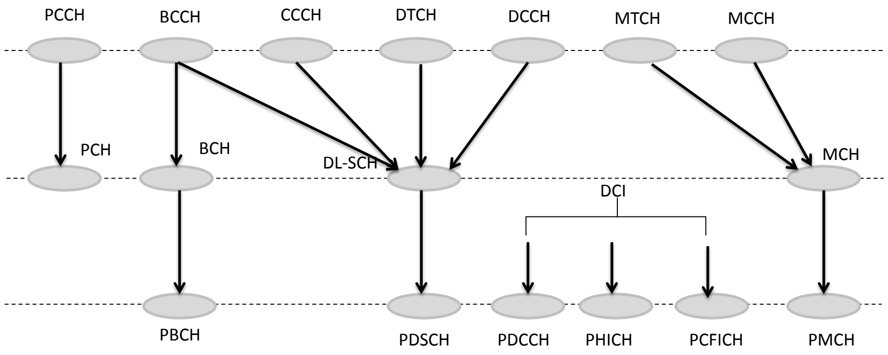

Figure 2.4: Mapping of logical channels to transport (Downlink)

### 2.3.2 Control plane protocols

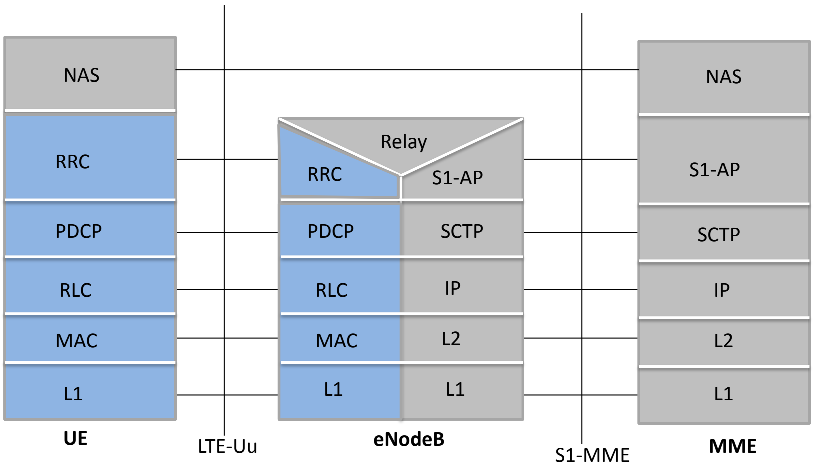

Figure 2.5: Control plane protocol stack in LTE

Figure 2.5 shows the control plane protocol stack of LTE RAN architecture. In the control-plane, the NAS protocols that run between the MME and the UE is used for control-purposes such as network attach, authentication, mobility management etc. Most AS protocols are common between the two planes apart from Radio Resource Control (RRC) protocol, a description of which is provided subsequently.

#### 2.3.2.1 RRC

The Radio Resource Control (RRC) is Layer 3 protocol in the AS protocol stack. It is the main controlling function responsible for establishing the radio bearers and configuring all the lower layers using RRC signaling. It is also responsible for making handover decisions based on the neighbor cell measurements sent by the UE. RRC is also used for, paging the UEs when in IDLE mode, broadcast system information, control periodicity of Channel Quality Information (CQI) reports sent by the UE etc. Also, RRC transfers UE context from the source eNodeB to the target eNodeB during handover and does integrity protection of RRC messages.

## 2.4 Network Interfaces in LTE

The connection between the E-UTRAN (eNodeB) to the EPC is by means of the S1 interface (to the S-GW by S1-u and to the MME by S1-MME). The eNodeBs are interconnected with each other by means of X2 interface. These interfaces are further split into control plane and user plane based on the type of traffic carried. The protocol structure for each of these is briefly discussed in the following subsections.

### 2.4.1 S1 interface

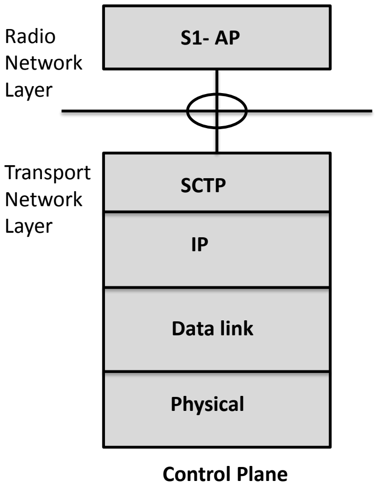

Figure 2.6: S1-MME Control plane protocol stack

The control plane protocol stack established in figure 2.5 is further shown in figure 2.6 with respect to S1 interface. This protocol stack is used for transferring signaling messages between eNodeB and MME regarding initiation, context management, bearer management, paging, mobility and load management [23]. S1 control plane is based on the well-known Stream Control Transmission Protocol (SCTP/IP) stack which inherits features from TCP for reliable delivery of signaling messages [22]. The mapping of S1-AP directly over SCTP results in a simpler protocol stack with no intermediate connection management protocols.

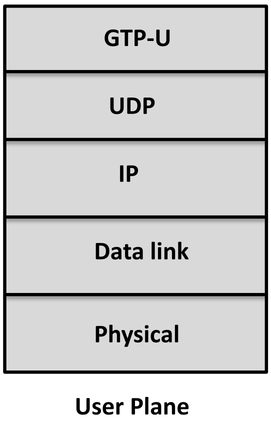

Figure 2.7: S1-U User plane protocol stack

The user plane protocol stack established in figure 2.3 is further shown in figure 2.7 with respect to S1 interface. The user plane protocol for S1 is based on the GTP/UDP/IP stack, an already established protocol stack for UMTS networks. One of the advantages of using GPRS Tunneling Protocol-User plane (GTP-U) is its inherent feature for identifying tunnels in order to facilitate intra-3GPP mobility [18]. Bearers can be identified by GTP tunnel endpoints and the corresponding IP address. This protocol stack is used to send downlink packets of the given bearer to the eNodeB IP address (received in S1-AP of eNodeB) associated to that particular bearer. It is also used by the eNodeB to send uplink packets of given bearer to the EPC IP address (received in S-AP of S-GW) associated to that particular bearer [23]. The choice of IP and data link protocols are left to the discretion of the vendors.

### 2.4.2 X2 interface

The main function of X2 interface is to support active-mode mobility and Inter-Cell Interference Coordination (ICIC). The control plane and data plane protocol stacks for X2 interface is similar to that of S1 interface, the only exception of X2-AP substituting for S1-AP. The use of the same protocol structure over both interfaces provides advantages such as simplifying the data forwarding operation [22]. Signaling involved for initiation, mobility, load and interference management over X2 interface is available in [24].

## 2.5 Physical layer

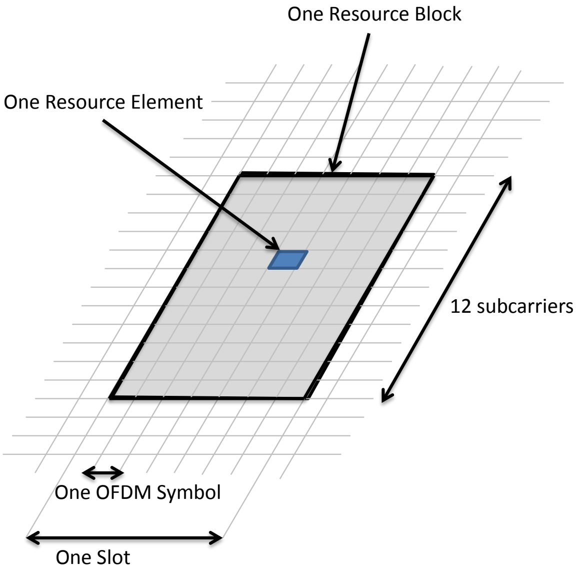

Figure 2.8: LTE Resource block

L1 or the physical layer is responsible for coding, modulation, mapping of the signal to the appropriate physical time–frequency resources etc. Adaptive Modulation and Coding (AMC) schemes are used to protect data against channel errors. Physical layer also provides indications to the upper layers regarding the link quality by processing the measurement reports from the UE. Multiple input multiple output (MIMO) configurations are supported at both eNodeB and the UE. Orthogonal Frequency Division Multiple Access (OFDMA) with a sub-carrier spacing of 15 kHz and Single Carrier Frequency Division Multiple Access (SC-FDMA) have been chosen as the transmission schemes for the DL and UL, respectively. Each radio frame in LTE is 10ms long. 1 frame further contains 10 sub-frames and one subframe contains two time slots, each slot capable of carrying 7 OFDM symbols. Each OFDM Symbol further contains 12 subcarriers in frequency domain. One such subcarrier is called a *resource element* and it is the smallest physical resource in LTE. 7 OFDM symbols (or one slot) in time domain along with 12 subcarriers (or one ODFM symbol ) in frequency domain together constitute as a *Resource Block*, the smallest frequency-time resource unit assigned to every UE. An LTE resource block is as shown in figure 2.8.

# Chapter 3

# Cognitive Radio overview

This chapter gives a basic overview of the Cognitive Radio and the technologies enabling it. The applications of CR along with the progress in standardization of CR technologies is briefly described. Dynamical channel allocation algorithms used for self organization are introduced in this chapter.

## 3.1 Introduction to Cognitive Radios

Electromagnetic spectrum is considered as a national resource and is managed by government agencies such as FCC (for United States), TRAI (for India), FICORA (for Finland) etc. In Europe, frequencies up to 6 GHz are used for various mobile technologies as shown in figure 3.1. The radio spectrum is usually allocated in chunks to different organizations for commercial purposes on a long term basis over wide geographical areas. Recent studies [7] have indicated that large portion of this spectrum is underutilized, resulting in large spectral inefficiency. This incompetence caused by allocating a primary user with a band of frequencies which he never uses, was termed as spectrum hole in [4]. Cognitive Radio, introduced in [25] makes efficient use of the spectrum by exploiting the existence of such spectrum holes. Secondary users (SU) check the spectrum availability from time to time and choose the idle channels for communication during the absence of primary users (PU). Coordination between SUs and PUs for efficient and fair spectrum sharing is usually done by a central network entity called the Spectrum Broker. The concept of spectrum hole is shown in figure 3.2(a).

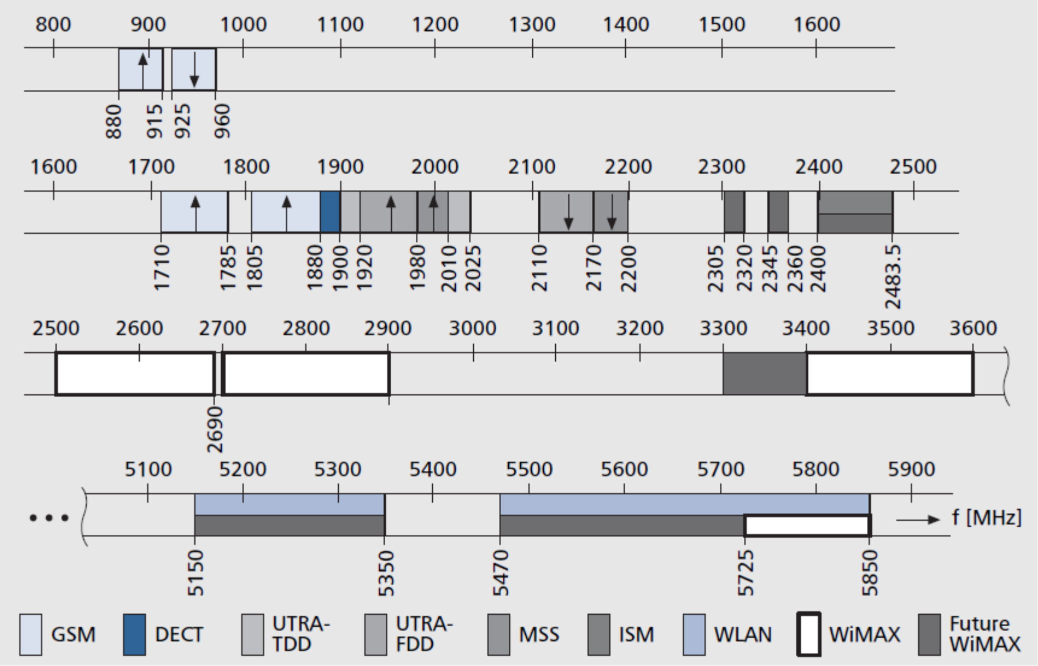

Figure 3.1: Mobile Spectrum in Europe, ref.[2]

## 3.2 Cognitive Radio characteristics

Haykin defines Cognitive Radio as an intelligent wireless communication system capable of analyzing its surrounding environment and adapting to the statistical variations by a *understanding-by-building* methodology, with efficient spectral utilization and reliable communication as its objective [4]. These systems are characterized by:

- **Cognitive capabilities:** Ability to sense and understand the environment.
- **Reconfigurability:** Ability to adapt the operational parameters according to the sensed information for improved performance.

CR looks towards software-defined radio (SDR) for reconfigurability. It is an extended version of SDR that additionally performs sensing and adaption based on its environment. The concept of SDR is discussed in the next chapter. The CR transceiver unit shown in 3.2(b) has a RF front-end (amplifier, mixer, A/D converter) which is capable of being tuned to any part of the spectrum and an equally flexible baseband processing unit. Next section

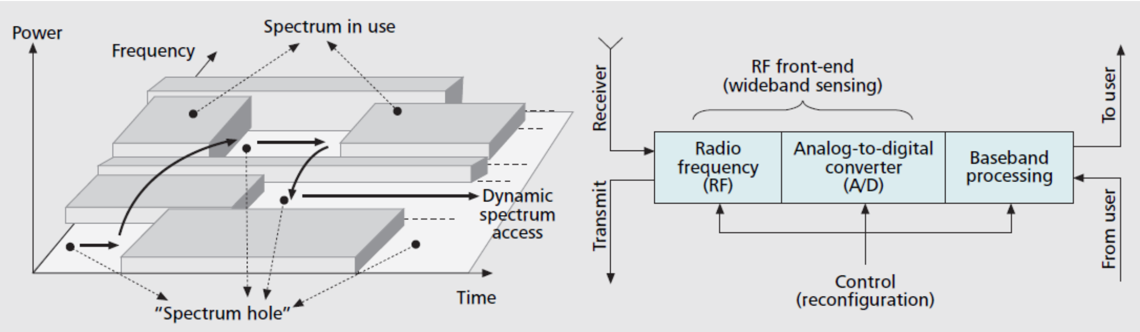

Figure 3.2: Cognitive Radio characteristics (a) Spectrum Hole concept, (b)Cognitive Radio transceiver, ref.[3]

describes different spectrum management functions that bring the cognitive capability to the system.

### 3.2.1 Cognitive Capability

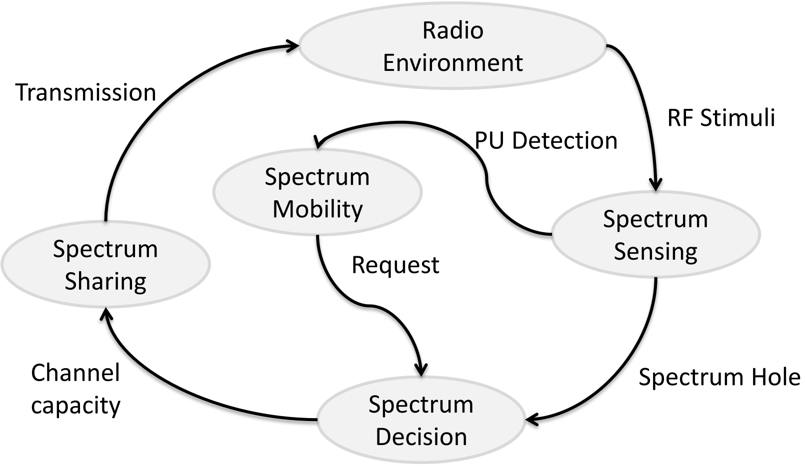

Figure 3.3: General Cognitive Cycle established in [4]

Different signal processing and machine learning algorithms have to be implemented for achieving cognitive capabilities. The basic cognitive cycle described by [4] is shown in figure 3.3. On receiving a RF stimuli due to the change in its operating environment, Cognitive radio starts scanning the entire radio spectrum for available spectrum holes. Based on its internal policies, it selects a suitable channel for communication purposes. Coordinated spectrum access is necessary since many CRs try to access the spectrum simultaneously resulting in collision. If the PU requires the channel, a spectrum handoff has to be initiated. Based on the above principles, CR cycle is classified into four basic spectrum management functions *spectrum sensing*, *spectrum decision*, *spectrum sharing*, and *spectrum mobility* each of which are explained further.

#### 3.2.1.1 Spectrum Sensing

Spectrum sensing is the most primitive functionality of any CR network. But detecting weak primary signals over a wide spectrum range in real-time can be quite a task. Different types of sensing techniques have been established over the years based on suitability and type of knowledge available at the CR node, a summary of which is available in [3]. A *matched filter detection* is employed when prior knowledge of the characteristics of the primary user is available. In this method, noise is assumed to be Gaussian and the source signal is deterministic and known to the receiver, making it is easier to match the source and received signals. If this prior information is not available, *Energy detection* is used wherein signal is assumed to be independent and identically distributed (iid), and the detection accumulates energy from all its signal samples. The performance of such techniques is vulnerable to the uncertainty in noise power. A more robust *Feature detection* technique can be employed to overcome this susceptibility. By exploiting the cyclostationarity of the signal, this technique extracts features in the Primary user signal. However, it is computationally complex and requires long observation intervals, as feature detection is performed by analyzing a spectral correlation function.

#### 3.2.1.2 Spectrum Decision

After sensing available free spectrum for its communication purpose, CR now has to choose the best possible spectrum hole. Even though this selection can be done at random, choosing spectrum hole based on the QoS requirements, channel characteristics and on the behavior of the PU results in a more efficient system [3]. Activities of other CR nodes may also have an impact while making this decision. Each available spectrum hole is allocated with a rank based on the above metrics (no of other CR nodes, channel characteristics etc.) and the spectrum with the highest rank is finally used for communication. The probability of primary user appearing in the same channel during a CR transmission also plays an important role in ranking the spectrum hole.

In case no single spectrum band is available to meet the given QoS requirement, multiple non-contiguous spectrum bands can be clubbed together for transmission thereby not only making the system immune to interference but also providing higher throughput.

#### 3.2.1.3 Spectrum Sharing

The shared nature of the wireless channel forces different CR nodes to cooperate between one another. This cooperation is increasingly difficult in CR networks, as CR nodes not only have to co-exist with one another but also have to establish cooperation with their primary users. Spectrum sharing can be classified based on several criteria. It can be classified into *centralized*, where spectrum allocation is controlled by a central entity or *decentralized*, where spectrum allocation is done distributively by applying local policies. Based on their information sharing mechanism, Spectrum sharing can also be classified into *cooperative*, where CR nodes form clusters to share interference information locally between one another or *non-cooperative*, where no such information is exchanged between the neighbors. Yet another type of classifying Spectrum sharing involves *Overlay sharing*, where CR and PU use only explicit spectrum that are not mutually used by one another or *Underlay sharing* where no such explicitness is defined and the transmission of one is considered as noise by the other.

#### 3.2.1.4 Spectrum Mobility

Whenever a primary user becomes active, it is compulsory for CR to switch to new operating spectrum band, as the priority always lies with the PU. This process is called the spectrum mobility. For undisrupted communication, it is necessary for this switch to have minimal overhead and also requires a new kind of handoff called the spectrum handoff. Different algorithms have to be implemented to take care of such handoffs. Every time a CR switches to a new channel, operation parameters have to be modified accordingly. It is advisable to hold-off the ongoing communications during this transition. The use of Spectrum mobility management (SMM) entities to ensure smooth transition and for minimum performance degradation during a spectrum handoff was initiated in [3].

### 3.2.2 Geo-location database

It is the responsibility of every Cognitive Node to perform the above spectrum management functions whenever it requires new frequencies for communication. Even though this is feasible, it sometimes becomes a tedious process to scan the entire spectra just to avail a suitable channel for a limited period of time, resulting not only in additional overhead but also in power draining the system. To eliminate this inefficiency, the concept of using Geo-location database for accessing TV white space was introduced in [26]. Instead of performing all the spectrum management functions by itself, a Cognitive Radio sends a query to a central database for the available frequencies it may use for a specific duration based on its current locations. The Geo-location database provides this response along with predefined rights and obligations attached in using the allocated spectrum. It now becomes the responsibility of the database to determine how the cognitive radio can coexist with the primary user.

One prerequisite for this method to work, is that the Cognitive Node should always be aware of its location and it has to update the Geo-location database of its position. Based on the location of the Cognitive node, the Geo-location database will prepare a list of the available channels, calculate the acceptable transmission power levels and the duration for which the channels are available. Every time the CR device changes its position by more than predefined distance, it has to obtain new set of parameters from the database. This results in lighter CR nodes, as the complexities involved in performing the spectrum management functionality is now moved to Geo-location databases [5]. In this work, we make use of one such Geo-location database provided by Fairspectrum Oy.

## 3.3 Cognitive Radio Applications

No technology can be self-sufficient in today's world. Cognitive Radio may not only have to co-exist with other heterogeneous networks in the future, but it can also be used to complement and support other wireless access technologies for improved performance. Wireless operators can now make use of the CR technology for helping them find white space as a supplement to their licensed spectrum to gain extra mileage in capacity. It can also be used to determine the threshold transmit power levels by applications so as to reduce co-channel interference. Performance of a CR-assisted LTE network is analyzed in [5].

This paper tries to increase the capacity of LTE system by using free TV spectrum and employs CR to find the spectrum holes. It concludes that LTE+CR systems had very less degradation and TV system coverage loss than just using the LTE systems. A CR-enabled LTE system could detect the primary users and quickly switch to other vacant channels thereby reducing

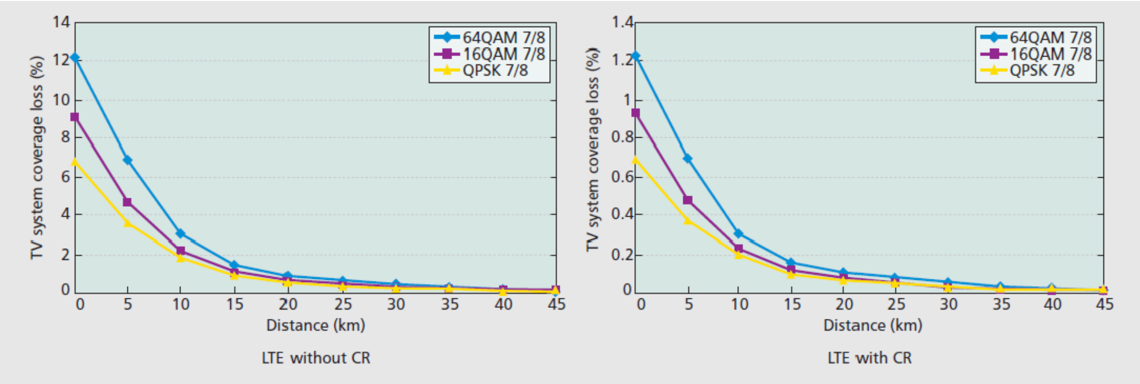

Figure 3.4: TV coverage loss, ref [5]

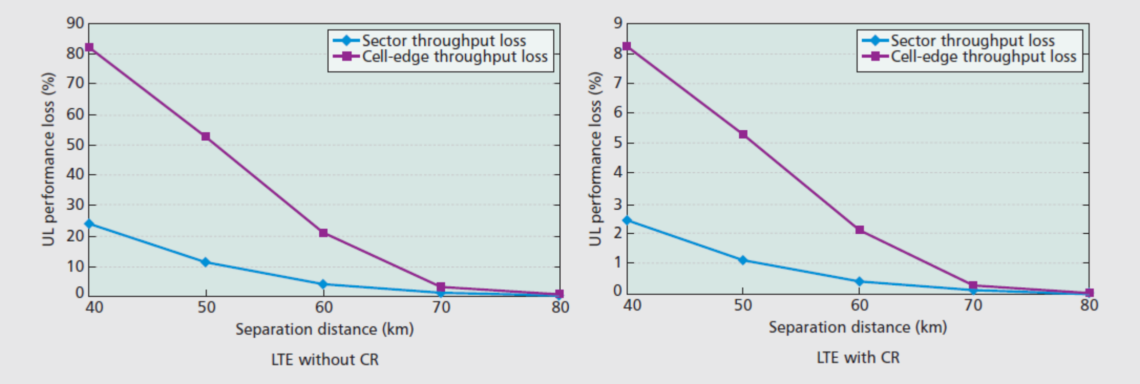

Figure 3.5: LTE UL performance loss, ref [5]

the interference between an LTE system and a TV system as indicated in figures 3.4 and 3.5.

## 3.4 Standards and Regulation

Several organizations are working on providing standards for Cognitive Radio Networks. The European Conference of Post and Telecommunications Administrations (CEPT) is providing the specifications for CR operations in 470–790 MHz range. The Institute of Electrical and Electronics Engineers (IEEE) provides standards such as 802.11af (an extension of 802.11 standards) to operate in TV white spaces. Wireless Coexistence Technical Advisory Group (TAG) has come up with IEEE 802.19 standards for developing coexistence between unlicensed wireless networks. IEEE 802.16h provides similar standards for coordinated and uncoordinated coexistence mechanisms for Wimax systems. Work is also in progress by IEEE 802 LAN/MAN Standards committee to develop IEEE 802.22 standard for cognitive wireless regional area networks (WRANs). IEEE 802.22 WRANs are designed to operate in the TV broadcast bands while ensuring that no interference is caused to the incumbent operation. IEEE Standardization Coordinating Committee 41 (SCC41), *Dynamic Spectrum Access Networks* deals with technological convergence of different areas like radio engineering, wireless networking, network management etc to contribute to the success of CR deployment.

The Internet Engineering Task Force (IETF) has started the standardization procedure for communication between CR devices and the Geo-location databases by providing *Protocol to Access White Space databases* (PAWS) protocol. According to PAWS, a Cognitive Radio must always access a Geo-location database to get the list of available channels for its location. PAWS explicitly provides the database identification mechanism (determining which database to connect to) and contents of the queries and responses (usually in JSON/XML). Field tests are already taking place in Finland as a part of Trial Environment for Cognitive Radio and Networks program from Tekes.

## 3.5 Self-organization in Cognitive Radio Architecture

Self-organization is an emerging trend in communication systems. It is defined as the emergence of system-wide adaptive structure and functionality from simple local interactions between individual entities [27]. In LTE/LTE-A systems, self organization and self healing have been emphasized in the eNodeB functionality. In Cognitive Radio Systems, where there is limited or no centralized control, the Cognitive Nodes are expected to sense the available channels by using distributed spectrum allocation schemes and organize themselves dynamically. Designing self-organizing networks embodies the following paradigms [27]:

- **Designing Local behavior rules that achieve Global properties:** In centralized solutions, a dedicated entity is responsible for establishing the global property. However, self-organization distributes the responsibility among the individual entities thereby contributing to a collective behavior. Hence, we must design local rules that automatically lead to the desired global property.

- **Exploiting Implicit Coordination:** Implicit coordination refers to gathering coordination information from the local environment instead of communicating explicitly by signaling messages thereby reducing the overhead inferred. This is very important in Cognitive Nodes as a node observes other nodes in its neighborhood and draws conclusions about the status of the network and reacts accordingly based on such observations.

- **Minimize Long-lived state information:** To achieve a higher level of self-organization, we should minimize the amount of long-lived state information. One approach is to employ discovery mechanisms, which nodes can use to obtain information about a certain network entity or service.

- **Designing protocols that adapt to changes:** This enforces nodes in the network to adapt suitably.

Many distributed channel allocation algorithms that help Cognitive Radio systems in achieving self-organization is available, a brief discussion of which is provided further.

### 3.5.1 Channel Allocation Algorithms

Dynamic resource allocation in CRNs has drawn a lot of attention recently [28]. The simplest frequency assignment problem, essentially a graph-coloring problem, considers allocating one resource to each cell with interference modeled by conflict graphs [29] [30]. A more complicated problem is when the number of resources are less than the number of cells resulting in a multi-coloring problem. Few works such as [31] and [32] have established a distributed solutions to the interference management problem. The focus of this work, lies is in understanding the characteristics of these channel allocation algorithms and implement them in order to achieve self-organization. Two classes of channel allocation algorithms are considered in this work, a generic greedy algorithm where the node does not give up its already chosen channel and a semi-greedy algorithm where the nodes give up the channel they have chosen under certain conditions. Each of these algorithms are discusses further.

#### 3.5.1.1 Generic Greedy algorithms

Many greedy algorithms for distributed channel assignment are available in the literature [33], [34]. These algorithms make choices without worrying about the overall system structure, without looking ahead. It makes a locally optimal choice at each step which may not lead to global optima further. The generic steps involved in Greedy algorithms is as follows: For all non colored nodes,

- Generate $P$, the list of possible colors.
- Apply appropriate **selection-rules** to choose a color from $P$, that best suits this node.
- Move to the next non colored node.

The selection rules depend on the respective protocols used for coloring. For example in First-Fit type of algorithms, node chooses the first available color from $P$ and continues to use the same color during its entire communication.

#### 3.5.1.2 Semi-Greedy Distributed Local Search (SDLS)

SDLS, established in [35], is a distributed algorithm which incorporates the principles of the zero-temperature Markov Chain Monte Carlo algorithm. According to this algorithm, each node periodically executes the following routine for selecting a suitable resource:

- Compute the number of neighbors $\mathrm{N}_{\mathrm{old}}$ using the same resource.
- Randomly select a resource that is not currently uniformly used.
- Calculate the number of neighbors $\mathrm{N}_{\mathrm{new}}$ using this newly picked resource
- If $\mathrm{N}_{\mathrm{new}} \leq \mathrm{N}_{\mathrm{old}}$, then start using the newly picked resource. Otherwise, do nothing

For SDLS to function, the knowledge of resource occupancy of neighbors should be known apriori. When an optimal state is found, no changes occur in the system even when other nodes are executing their routines. The last condition with its $\leq$ sign allows for transitions even when the number of conflicts do not change, thereby giving a good chance of stumbling upon the global optimum. This is because the non converged nodes can still pick resources from a subset of equal conflicts, deeming this algorithm to be *Semi-Greedy* [35].

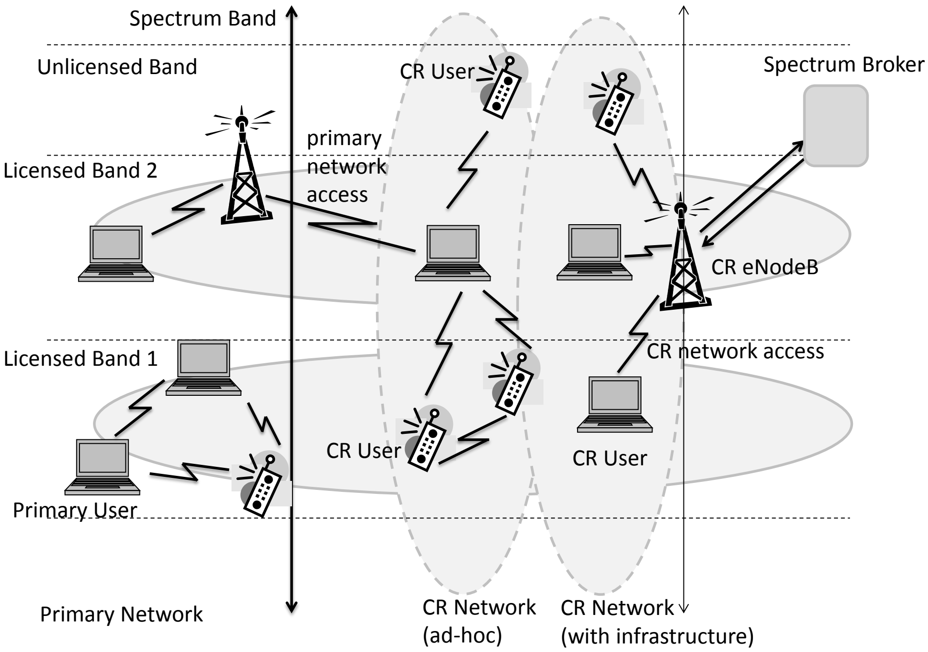

Figure 3.6: General Cognitive Radio architecture established in [3]

## 3.6 General Cognitive Radio architecture

A general architecture for Cognitive Radio networks as shown in figure 3.6 was established in [3]. The incumbent networks that have the license to operate are called the primary networks. They are usually controlled by a central base station. Cognitive networks on the other hand, do not have the license to use the spectrum and borrow it from the primary networks. The priority however always lies with the users of the primary network. CR networks can either be ad-hoc or infrastructure oriented. For many CR nodes to co-exist, they can make use of spectrum brokers for a fair distributing of the spectrum resources.

In this work, we implement a similar infrastructure oriented CRA, equipped with Cognitive-eNodeBs that provide single-hop connection to end users. The established CRA makes use of the TV spectrum for its communication purposes, hence becoming the primary networks in a sense. The Geo-location database provided by Fairspectrum acts as a Spectrum broker for allocating the radio spectrum when necessary. The established architecture is LTE compliant. This means that the current architecture can match the performance LTE/LTE-A systems in the future (Radio protocols have to be implemented) and also assist them for radio resource management when required. The Cognitive Radio architecture for radio resource management designed in this work, along with its network components and implementation is presented in the next chapter.

# Chapter 4

# Network architecture implementation

In this chapter, we define a working architecture for Cognitive Radio Systems. The design goals that were set before implementing this architecture is discussed. The overall network architecture, giving a description of different logical elements and interfaces is also presented. The main motivation for implementing this system in real time is not only to understand the practical difficulties involved with implementation, but also to help us in understanding the system better as it incorporates real time radio parameters and channel conditions. It is mostly difficult to model the limitations of hardware and tools in simulators and more often than not, we end up making assumptions that deviate from a real-time behavior.

## 4.1 Software Defined Radios

Cognitive Radio is contrived on the principle that adding dynamic intelligence to the receivers, with further assistance by external databases, can provide higher spectral efficiency than the existing systems. It is often associated with Software defined Radio (SDR), a term coined by Joseph Mitola III [8]. In a SDR, the components of radio that have been typically implemented in hardware (e.g. mixers, filters, amplifiers, modulators/demodulators etc.) are instead implemented by means of software on host computers [36] limiting the radio hardware only for RF transmission and reception purposes. This reconfigurability enables SDR to replace unalterable costly hardware functionality with compliant economical software. Given the pace at which wireless protocols are evolving these days, hardware can become completely obsolete due to their inability to confront to new standards. An SDR, however, could be reconfigured to support future standards and protocols.

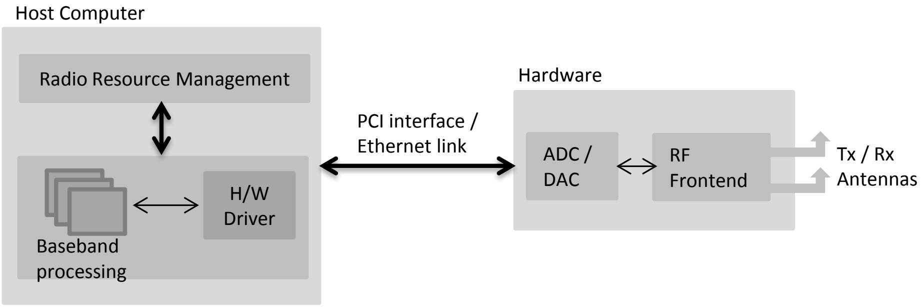

Figure 4.1: General architecture for Software defined Radios

A cognitive Radio system should generally be aware of its environment and adapt to it suitably [37]. In this work, we mainly concentrate on implementing a Radio Resource Management unit for supporting this adaptability. Figure 4.1 gives a brief overview of SDR implementation. A SDR system can be mainly characterized into two parts, a *Software entity* usually implemented on a Host computer and a *Hardware entity* for transmission/reception of the signal. In this work, the hardware entity is either WLAN or Universal Software Radio Peripherals (USRPs). USRP is a popular SDR platform manufactured by Ettus Research (now part of National Instruments) [38]. The USRP along with a host computer, creates a complete SDR system. It has three major components:

- **FPGA:** for high-speed digital up and down conversion from baseband to IF.
- **ADC/DAC:** for converting the signal from analog to digital domain and vice-versa.
- **Daughterboard:** which holds the RF transceivers used for translating the baseband signal into the carrier frequency.

When using USRP testbeds, the host computer consists of a RRM entity responsible for system level control of radio transmission characteristics. The baseband processing tasks related to the physical layer are also done on the host computer by software processes and the resulting data is then passed to the driver for managing the hardware entity. The host driver for USRP is called USRP Hardware Driver (UHD). The main function of UHD is to modify parameters such as RF center frequency, antenna selection, gain etc on the USRP. In our system, these parameters are usually set by the RRM entity present on the host computer, which is then passed to the UHD API by lower level C++. This data is later translated for the USRP FPGA by UHD. Hardware performs limited signal processing tasks necessary for transmission and reception of Radio signals. When using WLAN as the hardware entity, RRM unit on the host computer sets radio parameters such as center frequency, transmit power etc by making use of APIs provided by the WLAN driver. The objective of this work does not include developing a SDR testbed using USRP or WLAN, but rather focuses on implementing a working architecture for Radio Resource Management irrespective of the underlying hardware.

## 4.2 System Architecture

This section emphasizes on the design of the system and how the established architecture took its current shape. The different elements that make up the system are also described in detail.

### 4.2.1 System Design

The design of the system plays an important role while establishing any communication architecture. The performance of the system along with the functionality it can provide is usually reliant on the system design. A few design goals were kept at the focal point while designing this system architecture. The architecture was responsible for:

- Spectrum management functions: Spectrum Sensing, Spectrum Decision, Spectrum Sharing and Spectrum Mobility.
- Interacting with external networks and internet.
- Radio Resource Management and control.
- Configuring Cognitive nodes or end users connected to the system.
- Providing a platform to implement various RRM algorithms.
- Maintaining user profiles to help in Cognitive behavior.
- Providing a platform for communication applications.

With the above design goals in mind, the system architecture was designed as shown in figure 4.2:

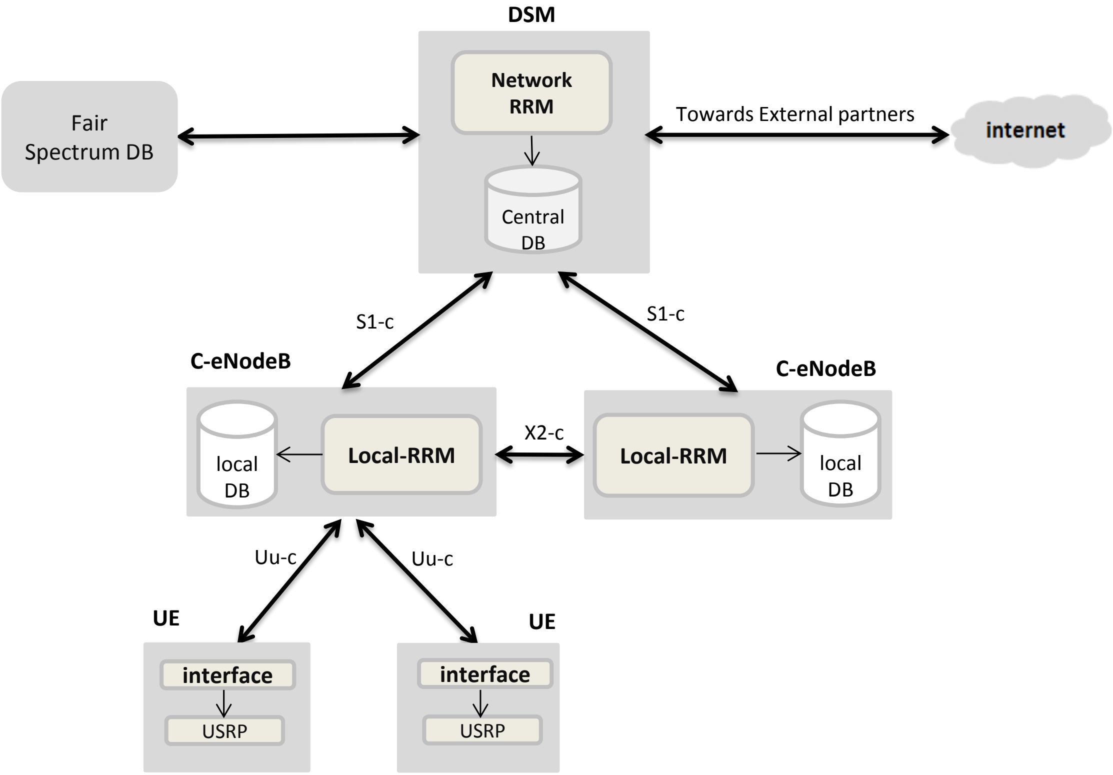

Figure 4.2: Cognitive Radio Architecture (CRA)

The Geolocation database provided by Fairspectrum Oy, is responsible for performing spectrum management functions. The job of the CR system is now reduced to communicating with this database for getting a list of available frequencies, duration of its availability and the permitted power levels. The *Dynamic Spectrum Manager (DSM)* is a central entity responsible for this function. It uses PAWS protocol (description in chapter 5) for this correspondence. DSM is supplemented with a *Central database*, for managing the parameters provided by the geolocation database and storing a local copy of its own. Due to its position in the hierarchy, DSM was also designed to act as an interface point for external networks that require access to the CRA. The *Cognitive eNodeB (C-eNodeB)* was designed for performing all the radio related functionality and management in our system. C-eNodeB, with its flat RAN is designed similar to the LTE E-UTRAN (but with limited functionality) for providing improved performance and reduced latency. Different RRM algorithms can be implemented and tested at C-eNodeBs. It is also made responsible for configuring the end users. To maintain user information, that is necessary for inducing a Cognitive behavior, C-eNodeB has a *Local database*, where it creates a new table for storing information from every user attached to the system. Lastly, the *User Equipment (UE)* provides a platform for executing the communication applications and services. The interfaces for communication between different network elements is also visible in figure 4.2. Although, these interfaces emulate functionality of its LTE counterpart, they are not LTE interfaces by themselves. A detailed description of these interfaces is provided in the later sections. In this work the terms DSM and N-RRM, C-eNodeB and L-RRM are used interchangeably.

## 4.2.2 Logical elements in CRA

Along with the logical nodes and connections, figure 4.2 also highlights the division of architecture into three main elements based on their functionality such as context awareness, spectrum management etc.

### 4.2.2.1 Dynamic Spectrum Manager (DSM)

Network Radio Resource Manager (N-RRM) along with the Central database acts as a Dynamic Spectrum Manager (DSM) in our system. As the main control element in our network, DSM is responsible for spectrum evaluation, spectrum assignment to C-eNodeBs and long term management of the radio spectrum.

- The N-RRM gets a list of available spectrum holes, their duration of availability and acceptable power levels from the Geolocation Database and stores these frequencies in a Central database.
- The N-RRM provides capabilities to write frequencies locally if necessary. Whenever there is a request for a new frequency from L-RRM, the first frequency available in the Central Database is given to the L-RRM and the number of NEs using the same frequency is marked.
- It also informs the new L-RRMs, the IP addresses of the L-RRMs already connected to it. This is very useful in establishing the X2-c interface.
- The N-RRM also commands the L-RRM over S1-c interface to take certain actions such as searching if a particular IP address is present

in its network or to get decision making context information such as measurement reports for a particular IP address and so on.

- It also acts as an interface for external networks to connect to our system.

N-RRM is implemented in Python and the central database in MySQL.

#### 4.2.2.2 Cognitive Radio eNodeB (C-eNodeB)

The main functionality of C-eNodeB is radio resource management of resources like frequency, transmit power, Resource Blocks, TDD mode, modulation and datarate. It is also responsible for configuration of the user equipment (UEs) connected to it. It consists of a local radio resource management unit called L-RRM and a local database for storing UE specific information. At any given instant of time, C-eNodeB may be serving multiple UEs, but each UE is connected to only one C-eNodeB.

- When a new C-eNodeB is started, it sends its IP address to the DSM over S1-c interface.

- Whenever a UE requests for a connection, The L-RRM provides the necessary configuration information to the UE over Uu-c. It gets the first available frequency from the central database and it updates the number of other C-eNodeBs using the same frequency. In our system, each frequency is divided into six RBs and each UE uses two RBs for communication. Three UEs can hence be allotted with the same frequency by a C-eNodeB. L-RRM also prepares a list of transmit power, Resource Blocks, TDD mode, modulation, datarate and sends all these parameters to the UEs for configuration and reconfiguration. When the UE terminates, it notifies the L-RRM of the same and all the resources allocated to it will be added back to the resource pool. For each UE, a new object/instance is created. So the L-RRM is capable of managing each and every parameter of the every UE under it individually.

- It receives constant feedback from the UEs regarding the channel quality (CQI), receive SNR and available throughput. It stores these parameters locally in a list and also in the local database. In the future, we have to make use of this feedback and implement resource management algorithms to decide on the configuration/reconfiguration parameters to be used. Currently, though the capability is ensured, the feedback parameters are not being made use of.

- The L-RRM can negotiate with L-RRM of other C-eNodeBs over the X2-c interface.
- On request, it provides decision making information for N-RRM based on the performance data collected from the UEs. It also performs certain actions for DSM such as download a video from the provided URL, search for IP address etc.

L-RRM is implemented in Python and the central database in MySQL.

#### 4.2.2.3 User Equipment (UE)

The UEs is the end user entity that is solely responsible for enforcement of configuration and reconfiguration decisions made by L-RRM and N-RRMs. It is a platform for communication applications which signals the C-eNodeB for setting up, maintaining and removing communication links.

- The UE when booted up asks for configuration information from L-RRM and uses this information until further notified by L-RRM.
- It measures the quality of its channel periodically and sends it as a feedback to the respective C-eNodeB.
- It performs tasks specified by C-eNodeB such as send report, receive a file etc.

UE has a wrapper written in Python for simplicity and management of the underlying hardware(USRP/WLAN). The system is context aware in a sense that, based on its context, C-eNodeB notifies UE to adapt few or all of its resources over Uu.

### 4.2.3 Logical Interfaces

Following are the interfaces defined for our Cognitive Radio architecture:

- **S1-c interface**: Connects DSM to the C-eNodeBs
- **X2-c interface**: Connection between C-eNodeBs
- **Uu-c interface**: Connects C-eNodeBs to UEs

To indicate the significance of these cognitive radio interfaces in comparison to the actual LTE interfaces and most importantly since they emulate a few important functionality of the LTE S1 and X2, cognitive radio interfaces continue to bear similar names as their LTE counterpart. The signaling messages used for initiation, interference management etc have been influenced from [23], [24]. The control plane and data plane protocol stacks are designed to be the same for all the Cognitive radio interfaces as shown in figure 4.3.

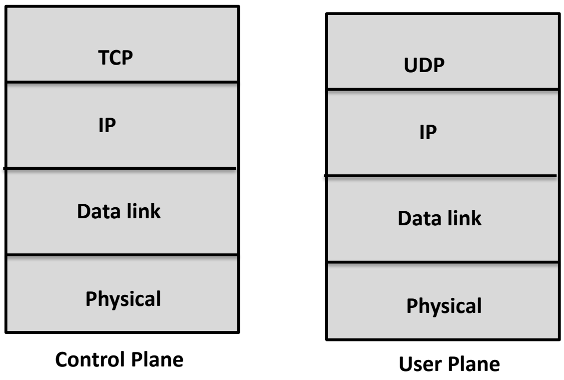

Figure 4.3: Protocol stacks for Cognitive Radio interfaces

S1-c interface takes care of the signaling required for informing C-eNodeB's IP address to DSM, requesting frequency from the central database, commanding C-eNodeBs to perform tasks such as search for an IP address, requesting Measurement reports etc. In our architecture, the sole purpose of X2-c is for ICIC. Signaling for resource allocation/reallocation, feedback report etc, are done over Uu-c. As seen from figure 4.3, the transport layer of User plane protocol stack relies on TCP for reliable transmission of user data, while the Control plane relies on UDP to deliver the control messages. This is because the system is designed in such a way that, a control message is retransmitted if the system does not receive an expected response within a period of time. However using UDP to deliver user data to the UE resulted in performance degradation in the system. We can notice that the interfaces defined for this CRA are all-IP interfaces. Now that the interfaces have been established, more functionality like Context transfer, Configuration transfer, Location reporting can be added to S1-c and functionality like mobility management, load management etc. can be added to X2-c by adding new signaling messages to the respective interfaces.

The basic principle of inter-cell interference coordination (ICIC) is to avoid high-power transmissions thereby reducing interference with cell-edge users of neighboring cells scheduled on the same resource. This kind of selective interference avoidance will not only benefit the QoS for the cell-edge-user but also enhances the overall system performance [1]. Since the

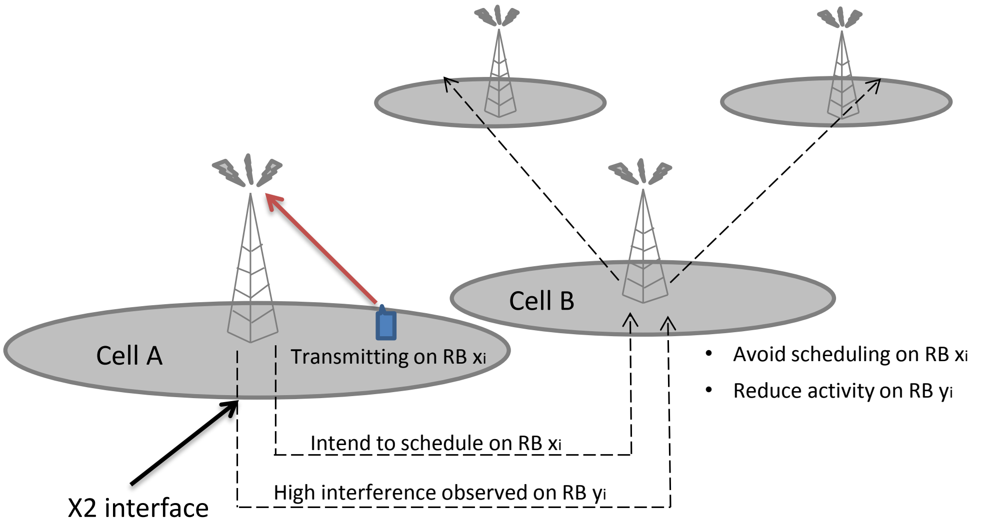

Figure 4.4: ICIC over X2-c

architecture is flat and resource allocation, scheduling etc are managed locally at the C-eNodeBs, it is advisable that the neighboring C-eNodeBs discuss the scheduling and other strategies over this X2-c interface as shown in figure 4.4. X2-c interfaces are not necessary between every C-eNodeBs in a network, they are required only between neighboring C-eNodeBs.

## 4.3 System Internals

This section gives a brief overview of the implementation details of the system and the preferred language and tools used.

### 4.3.1 Choice of tools

Choice of programming language definitely shadows its impact on practical design of any system. As there is no programming language that is suitable for all situations, we have to decide on it with utmost concern. Scripting tools are usually considered to consume lot of processing time and hence are not generally preferred for implementing high speed architectures. However they offer many other advantages such as faster implementation time, modularity etc. The objective of this work was to gather the impact on performance of the system when using scripting tools for development.

#### 4.3.1.1 Why Python?

Python is the most widely used language these days and is a popular choice for development due to its readability, elegant design, extremely portability and many other advantages. Its object oriented style that builds on classes provides easy abstraction, modularity and extensibility to include new attributes and behaviors. It has a perfect balance of high level and low level programming which suits the *Proof-of-concept* design we are adopting for our system. Its high level design results in reduced development time, readability and improved program maintenance. Adding new functionality in the future will be ornate and simple. Its closeness to low level programming results in reduced execution overhead as the code is compiled into interpreted bytecode. Python has automatic memory management and support for useful high level data structures and libraries. It's impressive set of standard library packages include scientific packages like SciPy, data plotting libraries like matplotlib, built-in database functionality, networking tools, XML support, regular expressions, unit testing, multi-threading etc.

Though the development speed is much faster in Python than in C, execution times are much slower. However, Python provides excellent language interoperability. It also provides compilers like Cython which combines advantages of both C and Python. It translates Python code into equivalent C code and then uses a C compiler to create a shared library which can be loaded as a Python module resulting in massive speed ups. Also, many core building blocks of Python are implemented in C. We can make use of such blocks to our advantage and obtain substantial performance gains.

Python with its interoperability can be written as a wrapper around the underlying PHY layer (written in C++ when using USRP). Therefore, we can make use of advantages of Python without having to code in complex C/C++ environment. It can be observed in the later sections that it is rarely the case where a problem needs solution only in C, Python in itself is usually self-sufficient.

#### 4.3.1.2 Why MySQL?

For a Cognitive Radio system, the availability of a database is necessary in order to manage large amounts of information regarding its surroundings. It has to gather useful information from huge data sets and use this information for adaptability and improved system performance. In our project, the central database at DSM manages a list of available frequencies, their duration and allowed power levels while the Local databases at each C-eNodeB, collect feedback information including receive power, channel quality etc from UEs attached to it.

MySQL with its scalability, platform flexibility and high performance is regarded as one of the most preferred choices of database for enterprises today. It can manage terabytes of information with negligible overhead due to its performance enhancing mechanisms such as cached memory, high speed load utilities etc. It is highly reliable and enjoys large online support. MySQL also provides connectors and drivers (ODBC, JDBC, etc.) that allow applications from Python, Java and many other languages to connect to and manage the databases. This flexibility along with robustness makes MySQL a natural choice for our databases.

### 4.3.2 Interfacing

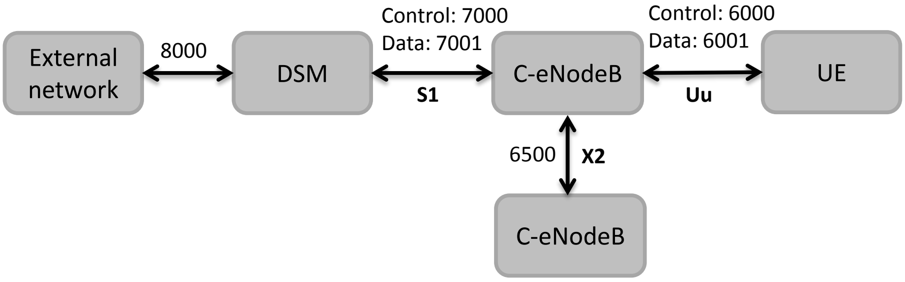

Figure 4.5: Signalling interface

The communication between different network elements in the CRA is achieved over UNIX ports. The interfacing diagram shown in figure 4.5 gives the port number used for this communication. bind() system call, provided by UNIX SOCKET interface, forces network elements to listen on pre-defined port number. Any element that wants to establish a connection with its counterpart should hence be aware of its IP address and port number beforehand. Whenever there is information to be read on that port, appropriate actions are taken based on the signaling model defined across the cognitive radio interface.

# Chapter 5

# System Capabilities

While the previous chapter gives an idea of how the network architecture is defined and implemented, this chapter gives a description of capabilities of the system. In this work, we have defined a few basic services for RRM. More functionality can be provided to the system simply by adding appropriate signaling between the already defined interfaces.

## 5.1 Attachment Procedure

To avail the services offered by the core network, any UE has to register itself to the network initially. This registration is known as network attachment and is as shown in figure 5.1. Once the UE is attached to the system, it is the duty of the C-eNodeBs to monitor the UE and provide requested services. The interaction between different NEs take place on predefined port numbers indicated in the figure 4.5.

The attachment procedure can be broken down into following steps:

- **Step 1:** The UE sends a ***new_client*** request to the C-eNodeB indicating its interest in joining the CRN.
- **Step 2:** C-eNodeB sends an acknowledgement ***got_req*** and starts preparing a set of radio parameters for configuring the UE.
- **Step 3:** Consequently, it sends a ***get_freq*** request to DSM for getting the first available frequency from the Central Database. DSM in turn responds with ***centre_freq*** message along with the allocated frequency.
- **Step 4:** The C-eNodeB sends a list of configuration parameters including Frequency, RB, Tx Power, Datarate, Modulation and TDD configuration mode along with the message ***parameters***. Currently,

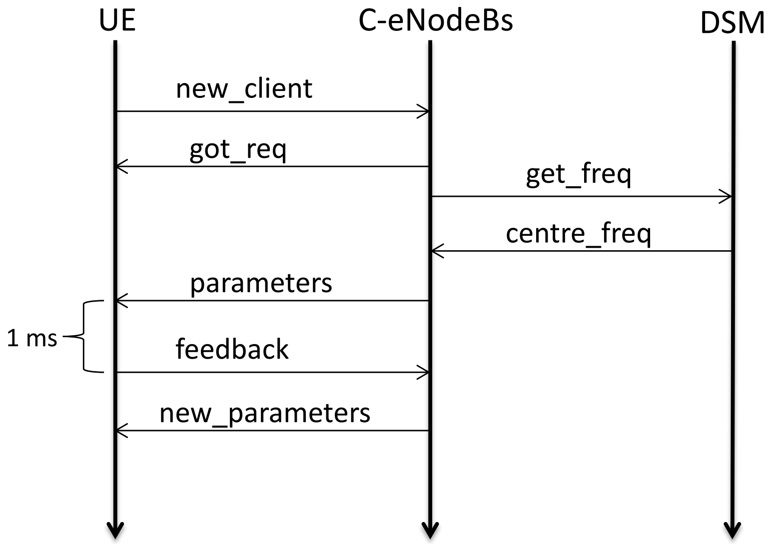

Figure 5.1: Signaling for the Attachment procedure

we do not have algorithms for deciding these parameters and hence we randomly choose a value from a set of prescribed values for LTE.

- **Step 5:** The UE makes use of these configuration parameters for its transmissions. It frequently monitors its environment and sends this information as ***feedback*** to respective C-eNodeB. Currently, when using WLAN interface, we measure the channel quality and receive SNRs at the UE and forward the same as feedback.

- **Step 6:** The C-eNodeBs writes this feedback into a local database. Based on the received feedback, it prepares a new list of resources and forwards it to the UE as reconfiguration parameters along with the message ***new_parameters***.

- **Step 7: Step 5** and **Step 6** are repeated as long as the UE is attached to the system. For detaching from the CRN, the UE has to send a ***terminate_client*** message after receiving which, the C-eNodeB releases all the resources held for that particular UE.

## 5.2 Interacting with Geo-location Database

One approach to manage spectrum sharing, makes use of Geo-location databases to report spectrum availability to CR nodes. To achieve interoperability among multiple nodes and databases, a standardized protocol called Protocol to Access White Space database (PAWS) is suggested. The established CRA communicates with the Geo-location database provided by Fairspectrum in order to obtain a list of available frequencies, the duration of its availability and the permitted power levels. The PAWS protocol includes the following steps [39]:

- Database Discovery: To discover the nearest Geo-location database based on Cognitive device's location.
- Initialization: To exchange the device capabilities.
- Device Registration: To Register Cognitive device with the database.
- Available Spectrum Query: To get the list of frequencies and related parameters.
- Device Validation: For authentication and security.

In this architecture, DSM has the Uniform resource identifier (URI) for Fairspectrum Database and hence no discovery is necessary. It establishes an HTTPS session with the database for further communication. Sending initialization messages are optional and are not currently used in this system, thereby resulting in a restricted signaling procedure. So the current signaling procedure between the DSM and the Geo-location database is as shown in figure 5.2 and can be described in the following steps:

- **Step 1:** The DSM registers itself to the database and sends an ***get-spectrum*** request to it.
- **Step 2:** The Database responds with an ***available_spectrum*** message along with a list of available frequencies, the duration of their availability and the permitted power levels. DOM parser provided by Python is then used to obtain this list from the body of the HTTP response. This information is further written into the Central database for local management and control.
- **Step 3:** DSM can also send a ***spectrum-usage*** notification message to the database if necessary. Even though this provision is provided by the PAWS protocol, it is not made use of in our system.

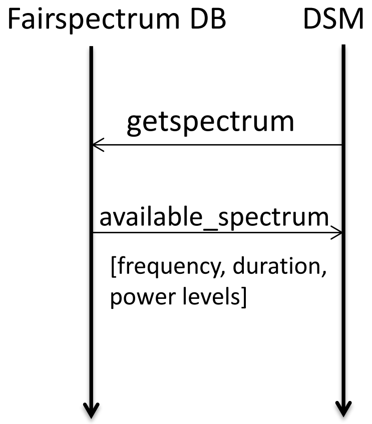

Figure 5.2: Signaling for interacting with Fairspectrum database (Geolocation DB)

## 5.3 Negotiating resources over X2

In a LTE system, X2 is a logical interface mainly used to support UE mobility. 3GPP standards also recommend using this interface for interference coordination for reducing inter-cell interference and improving throughput for cell edge users [24]. In our network architecture, the sole purpose of X2-c interface is for resource negotiation and ICIC. Whenever a new UE requests for a connection, a C-eNodeB before allocating the configuration parameters, checks with its interfering neighbor about its feasibility. Negotiations are done thereafter between the two C-eNodeBs according to the fig 5.3:

The steps involved for Resource negotiation over X2-c are:

- **Step 1:** Any C-eNodeB whose RSS exceeds a predefined threshold is considered a neighbor. When a UE executes the attach procedure (**Step 1 - Step 3**) requesting for radio resources, C-eNodeB will get the IP addresses of its interfering neighbors from the DSM and requests for an approval from all such neighbors before allocating the configuration parameters to the respective UE.

- **Step 2:** C-eNodeB sends its IP address along with the ***neighbor*** message introducing itself to its interfering neighbor.

- **Step 3:** It also sends the list of parameters it has deduced for the UE along with ***neighbor_param*** message for approval from the neighbor.

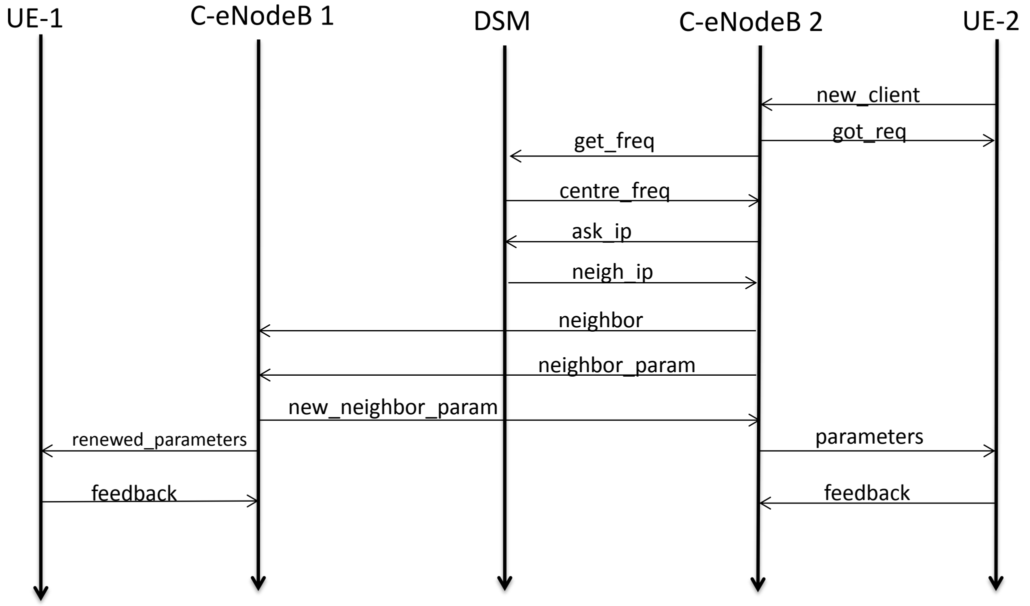

Figure 5.3: Signalling procedure for resource negotiation over X2-c

- **Step 4:** If the neighbor finds any of the parameters in the list to be unsuitable, it suggests new set of parameters for consideration along with ***new_neighbor_param*** message.
- **Step 5:** C-eNodeB on receiving these new parameters may consider the suggestion and allocate this new parameters to the UE, while waiting for its feedback.
- **Step 6:** Neighbor may also inform its own UE to modify a few parameters if necessary along with the ***renewed_parameters*** message and wait for its feedback.

Note that the decision to incorporate the suggestions provided by its neighbors lies with the C-eNodeBs itself. This is done to prevent the conflicts that may arise during negotiations. Even though this feature is made available, it is not compulsory for a C-eNodeB to send negotiation messages to its neighbors each time a new set of parameters have to be allocated to a UE. Avoiding resource negotiation feature reduces significant overhead to the system and hence is usually turned off.

## 5.4 Self-organization of Cognitive Radio Architecture

This functionality of distributed channel allocation and self-organization of CRA was implemented to demonstrate a real cognitive behavior of the established architecture. Here, based on environmental stimuli, CR adapts itself suitably. To be more precise, based on the available channels and their occupancy by its neighbors, the C-eNodeB selects channel that experiences least interference in the system. In this work, a distributed channel allocation algorithm called Greedy Distributed Local Search (GDLS) is implemented based on the steps involved in SDLS protocol and the concepts of generic greedy algorithms (introduced in chapter 3).

#### 5.4.0.1 Greedy Distributed Local Search (GDLS)

The steps involved in GDLS are similar to that of SDLS protocol. However, GDLS unlike SDLS is a greedy algorithm suitable for systems that can derive the global knowledge explicitly or implicitly. Instead of picking resource at random, GDLS applies local knowledge to carefully choose a resource such that the system experiences no/minimal interference. In some cases, GDLS enhances the overall system performance and also gives a chance to attain the global optima faster. It works as follows:

- Compute the number of neighbors $N_{old}$ using the same resource.
- Use global knowledge and choose resource such that the system experiences minimum interference.
- Compute the number of neighbors $N_{new}$ using this newly picked resource
- If $N_{new} \leq N_{old}$, then switch to use this newly picked resource. Otherwise, do nothing

In GDLS, if *No of channels* $\geq$ *No of nodes*, each node selfishly picks a resource such that no neighbor is present on its channel. Incase, *No of channels* $<$ *No of nodes*, each node greedily picks the channel of the least interfering neighbor based on the knowledge available at the system, thereby experiencing minimum interference. The system is considered to have achieved Global optima when a node ends up on channel with no/minimum interference. In GDLS, the node fails to give up the channel once this global optima is reached thereby making the algorithm *Greedy.* The flowchart of GDLS is shown in figure 5.4.

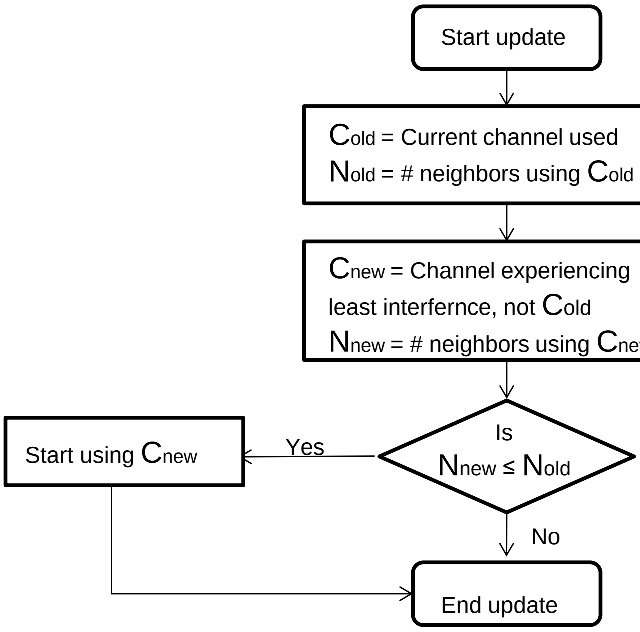

Figure 5.4: Flowchart of GDLS algorithm

#### 5.4.0.2 Implementation of GDLS using WLAN

This functionality is implemented by making use of the WLAN interface for Uu-c. One can observe how the self-organization paradigms (established in chapter 3) are incorporated in our system with the help of GDLS. Each C-eNodeBs gathers implicit information about other APs using its channel, by periodically scanning the WLAN interface. No long-lived state information is stored at the C-eNodeB. This information is either gathered from surrounding environment or by requesting the DSM for this information. Finally, GDLS adapts according to the changes in the system. Signals are included for informing the neighbors about the newly selected channel as $time_{feedback} >> RTT$. Even though this information can be implicitly derived before executing the routine, incorporating signaling will fasten the process of obtaining the knowledge about the network especially since the $time_{feedback}$ or TTI is about 5s when using the WLAN interface.

GDLS protocol also has the provision for providing a back-off mechanism. GDLS protocol when applied in its native version (without back-off) would result in symmetric conflict situations. For example, consider a case when *No of channels = No of nodes = 2*. If both the nodes are on the same channel, they sense that there is interference on the current channel and both the nodes execute their routine together to the switch to the new channel. Since they execute their routines together, both of them end up on the new channel again interfering with one another. This often ends up in a deadlock. This can be avoided if atleast one of the nodes is aware of the intentions of its neighbor. Asynchronous-GDLS (a-GDLS) precisely does this task. When in Asynchronous mode, GDLS backs-off for [0.5, 1 microseconds] before using the chosen channel and meanwhile employs signaling to inform the neighbor of this development. This forces the neighbor to stay on the current channel and thus achieve global optima. Practical implementation of the system makes us aware of such details that otherwise may have gone unnoticed. Performance comparison between a-GDLS and non backed-off GDLS, called Synchronous-GDLS (s-GDLS) can be seen in the next chapter. Even though it appears more logical to involve DSM in channel assignments (as it has a knowledge of channels used by C-eNodeBs), the idea is to make the system distributed and hence DSM is not made use of. A detailed description of the working of GDLS protocol and the corresponding setup involved is described in chapter 6. Next section describes the signaling procedure adopted for channel allocation.

### 5.4.1 Signaling

The signaling policy defined for distributed channel allocation is as shown in figure 5.5 and can be described as follows:

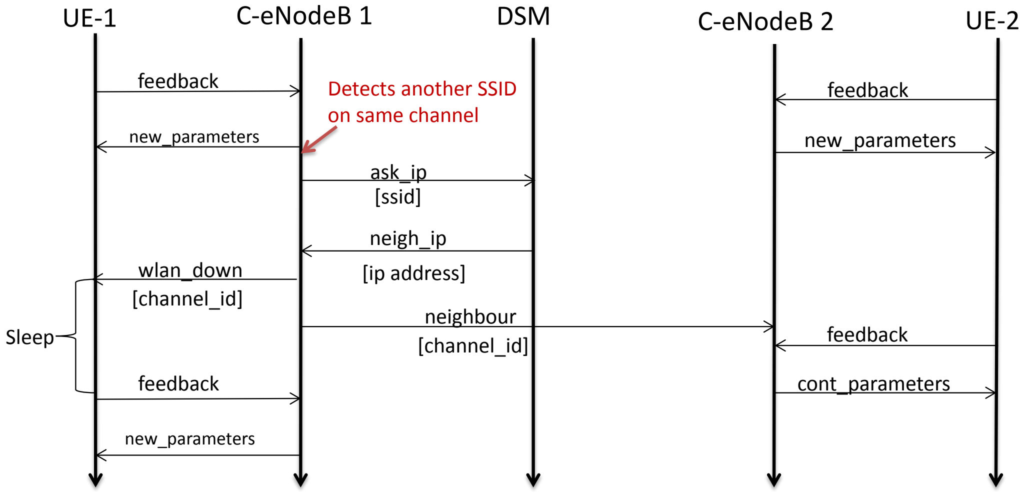

Figure 5.5: Signaling procedure for distributed channel allocation

- **Step 1:** All C-eNodeBs present within the system will have forwarded a list of IP address and WLAN SSID it uses, to the DSM.
- **Step 2:** Every 5 seconds, UE sends a feedback of SSID, Channel ID, Quality and RSS to its respective C-eNodeB which in turn writes them into the local database. Any other C-eNodeB using the same channel and whose RSS exceeds a predefined threshold is considered to be a neighbor. A threshold of -85dBm is selected for our system. To prevent frequent alternation between channels, a neighbor should be seen on the same channel as yours for atleast m consecutive feedbacks (m=3 in our system).
- **Step 3:** C-eNodeB requests the DSM for the IP address of the neighbor by sending a ***ask_ip*** message along with the SSID of the neighbor. The DSM unit checks for the IP address from its list and forwards the same to the C-eNodeB along with the message ***neigh_ip***.
- **Step 4:** C-eNodeB then applies the GDLS algorithm for selecting a better channel and before switching to it, sends a ***wlan_down*** message to the UE asking it to detach temporarily from the network. This is necessary because the UE and C-eNodeB should always be on the same channel at all instants of time when in *ad-hoc* mode. C-eNodeB then configures the WLAN driver to switch to the new channel for further communication. Meanwhile, UE detaches itself from the network temporarily and waits for about 5 seconds before reattaching to the system (As it takes 5 seconds for C-eNodeB to change to the new channel).
- **Step 5:** C-eNodeB also notifies all its neighbors about the newly selected channel with a ***neighbor*** message. Even though this information can be obtained implicitly, one must notice that it takes 5 seconds (feedback interval) for a neighbor to obtain this information implicitly, whereas explicitly signaling this information makes the C-eNodeBs aware of the changes in the system almost instantly.
- **Step 6:** The UE reattaches to the system by sending a ***feedback*** message.

This process is repeated until a global optima is obtained. On receiving a ***wlan_down*** message, UE moves from **RRM_connected** to **RRM_disconnected** as shown in figure 5.6. During **RRM_disconnected** state, UE will neither send any feedback nor will it be available for any service to its C-eNodeB.

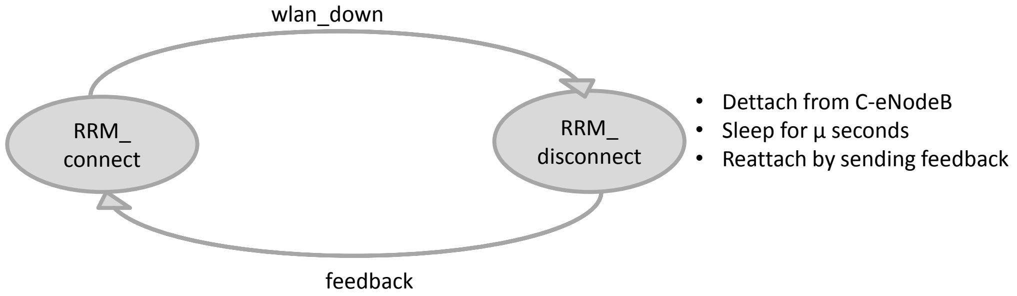

Figure 5.6: State machine for UE during distributed channel allocation

### 5.4.2 Measurement Collection system

C-eNodeB along with its local database behaves like a Measurement Collection System (MCS). Here, each UE attached to the system collects the SSID, Channel ID, Quality and Receive Signal Strength (RSS) of the surrounding APs over the WLAN interface and forwards the same to its respective C-eNodeB. The C-eNodeB writes this information into its Local DB thereby making the system aware of the interference levels seen at each of its UEs. This knowledge can be used for self-organization of the system aswell as to predict user behavior over a period of time. Three such C-eNodeBs when using WLAN can also help in indoor positioning of the system (triangulation methods).

## 5.5 Providing services to external networks

Inter connectivity with other heterogeneous networks or to the internet is extremely important for the established CRA. This interaction enables external partners to benefit from a set of predefined services rendered by this network. Video applications are the most popular and widely adopted services today thanks to large smart phone screens and high datarates provided by LTE. 60% of all data traffic is through video and this trend will definitely increase in the future. The emergence of paid-for premium video delivery model from sites like Netflix, and Hulu poses to be a challenge as well as an opportunity for mobile service providers. However, the cost of transporting and delivering such video content is much greater than the cost of traditional services such as voice or text short message service (SMS). In this work, we have implemented one such paid-for premium video transmission service by this CRA.

Any external network that wants video content to be delivered to a user can request the CRN to provide this service. The CRA checks the availability of the user in the system and based on its availability delivers the video according to its connection plan. The external network provides the URL of the video from video sharing websites such as YouTube. The C-eNodeB uses *youtube-downloader* program to download the video and transfer the content to the UE. Since the C-eNodeB has complete knowledge and history of the user profile, his channel conditions etc, appropriate charging model can be applied to suit the paid-for premium video delivery model. The signaling model adopted for this functionality is as shown in figure 5.7.

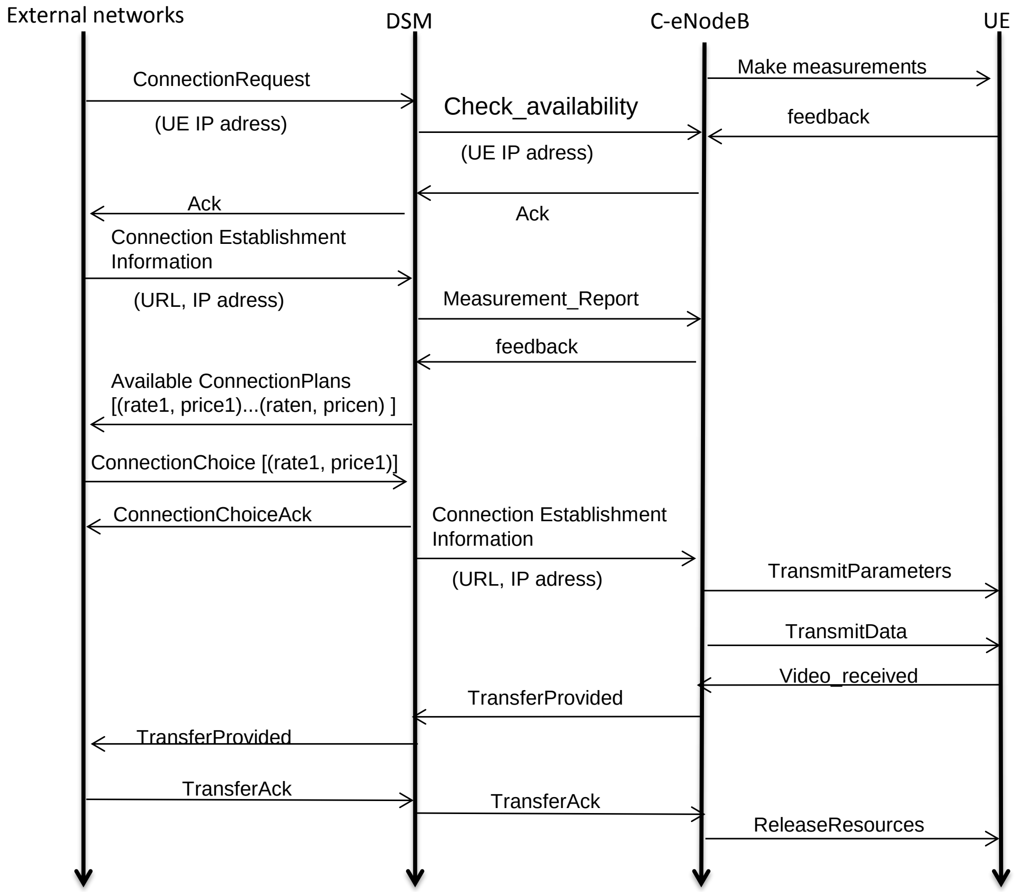

Figure 5.7: Signalling procedure for video transmission within CRN

The steps involved for video transmission within CRN is as follows:

- **Step 1:** External network sends a ***ConnectionRequest*** message to the DSM along with the IP address of the UE to which it wants the video to be delivered.

- **Step 2:** DSM in turn forwards this IP address to all the C-eNodeBs under it with a message ***Check_availabilty*** to verify whether or not that particular UE is attached to any of the C-eNodeBs in the network.
- **Step 3:** C-eNodeB will be aware of the IP addresses of all UEs attached under it. On receiving ***Check_availabilty***, C-eNodeBs check if the UE's IP address is present in its system and responds with a ***pos-ACK/neg-ACK*** based on UE's availability. If all C-eNodeBs respond with a ***neg-ACK***, DSM will intimate the external network about the unavailability of that UE within the CRN and hence its inability to provide the requested service. Otherwise, it proceeds with further steps.
- **Step 4:** C-eNodeB uses the *youtube-downloader* program to download any video on the internet with a distinct URL. On receiving ***pos-ACK***, the external system sends URL of the video to the DSM along with the message ***Connection Establishment information***.
- **Step 5:** Meanwhile, the DSM will request for measurement report from the C-eNodeB to gauge the history and profile of the requested user by sending a ***Measurement_Report*** message. The C-eNodeB forwards the connection history of that particular UE from its local database as ***feedback***. This report is required by the DSM for preparing a *Connection Plan* for charging the external systems for the service provided by it. Currently, no algorithms are present for preparing this connection plan.
- **Step 6:** DSM forwards a dummy connection plan in the form of a python list with the message ***Available ConnectionPlans***. The external network reciprocates with a suitable choice from the list along with the message ***ConnectionChoice***. This ***ConnectionChoice*** shall further be used for charging the external system.
- **Step 7:** DSM acknowledges ***ConnectionChoice*** message with ***ConnectionChoiceAck*** and forwards the ***Connection Establishment information*** to the C-eNodeB instructing the latter to deliver the content to the UE.
- **Step 8:** C-eNodeB downloads the video from the URL using *youtube-downloader* and sends a ***TransmitParameters*** message instructing the UE to prepare itself for receiving the content. It also sends a

***TransmitData*** message along with a 128-bit MD5 hash of the video file. C-eNodeB then strips the video file into binary data and forwards this content over the data channel. Since TCP (used for data channel) is a stream based protocol and it is very difficult to detect the end of file, an intrinsic ACK scheme has been implemented at C-eNodeB.

- **Step 9:** On receiving the entire video file, UE verifies the MD5 hash and notifies the C-eNodeB about the success or failure in receiving the video content along with the message ***Video_received***
- **Step 10:** If the message received at C-eNodeB indicates a failure in content delivery, the above process is further repeated. On the contrary, C-eNodeB sends a ***TransferProvided*** message to the DSM which further forwards it to the external network indicating the success in delivering the service.
- **Step 11:** External system on receiving ***TransferProvided***, acknowledges to the DSM with a ***TransferAck***. This is further forwarded to the C-eNodeB, which then releases all resources held during this service. C-eNodeB further notifies the UE to do the same with a ***ReleaseResources*** message.

# Chapter 6

# Results and Analysis

The results obtained in this work are of two types, one that evaluate the functionality described earlier and the other that evaluates the overall performance of the established Cognitive Radio architecture. This chapter presents and analyzes these results. The GDLS protocol established in chapter 5 is evaluated. The capability of the architecture to support external networks for providing services like video transmission is validated. Finally, the results characterizing the overall performance of the system in terms of simplicity, scalability, speed and stability is inspected.

## 6.1 Results Evaluating System Functionality

The results presented in this section validate the functionality of the CRA, illustrated by the means of two important demonstrations: **Self-organization of Cognitive Radio Architecture** makes use of attachment procedure of the UE to the CRA, interacting with geo-location database, negotiation over X2 interface and distributed resource allocation mechanism, to achieve its functionality. The other demonstration of **Providing paid-for premium video delivery service** is also validated in this section.

### 6.1.1 Self-organization of Cognitive Radio Architecture

Self-organization is defined as the process of establishing a global order through local interactions between a set of initially disordered elements, triggered by an agent within or outside the system. To validate the self-organizing capabilities of the system, we establish a setup wherein individual C-eNodeBs start on random channels and organize themselves in a distributed manner to achieve global optima with the help of GDLS algorithm (see chapter 5). Global optima is achieved when each C-eNodeB is on a different channel or if it uses the channel of its least interfering neighbor. The number of iterations required before this global optima is achieved is monitored for both asynchronous-GDLS and synchronous-GDLS algorithms. Results indicating the effect of different interference types on the system is also presented.

#### 6.1.1.1 Test Setup

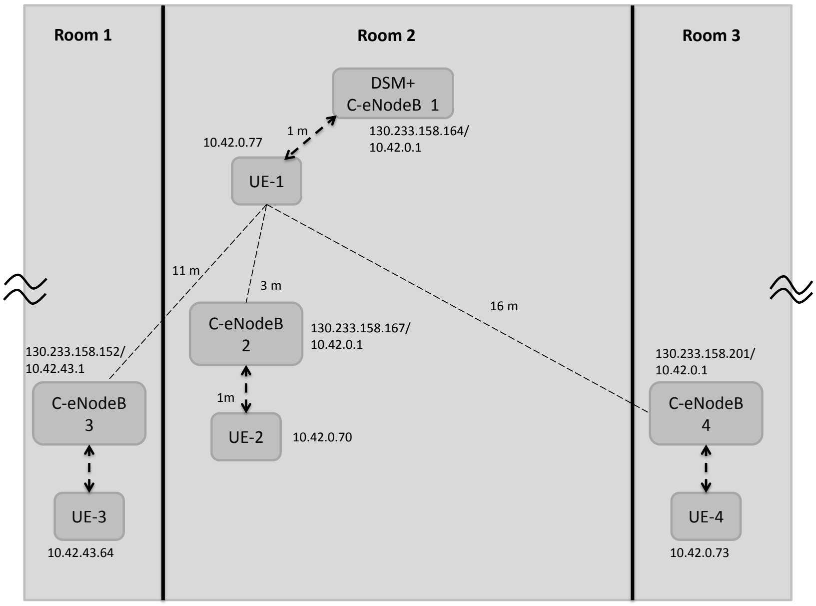

Figure 6.1: Test setup for Self-organized Resource allocation

We establish a network of Cognitive Radios (C-eNodeBs) that are communicating on random WLAN channels. Due to the presence of multiple C-eNodeBs on the same channel, they often cause interference to one another making the system disorganized. By sensing the environment (in this case interference levels), each C-eNodeB applies GDLS algorithm to obtain optimal channel, thereby resulting in a organized network. The setup consists of 4 UEs implemented on Linux laptops connected to 4 different C-eNodeBs running on Linux work-stations (1 UE per C-eNodeB for simplicity). Since the range of WLAN in dense indoor environments is typically about 20m, The C-eNodeBs are placed strategically in three different rooms separated by distances indicated in figure 6.1. DSM is implemented on one of these workstations. The C-eNodeBs and DSM which constitute the Core Network are connected through wired Ethernet interface, while C-eNodeB for connecting to its UE uses WLAN interface. IP addresses of all these NEs along with their SSIDs is given in table 6.1.

Table 6.1: IP address of different NEs

| Name of NE | Core IP | Radio IP | SSID |
|---|---|---|---|
| DSM | 130.233.158.164 | - | - |
| C-eNodeB 1 | 130.233.158.164 | 10.42.0.1 | eecrt |
| C-eNodeB 2 | 130.233.158.167 | 10.42.0.1 | eecrt_1 |
| C-eNodeB 3 | 130.233.158.152 | 10.42.43.1 | eecrt_2 |
| C-eNodeB 4 | 130.233.158.201 | 10.42.0.1 | eecrt_3 |
| UE 1 | - | 10.42.0.77 | eecrt |
| UE 2 | - | 10.42.0.70 | eecrt_1 |
| UE 3 | - | 10.42.43.64 | eecrt_2 |
| UE 4 | - | 10.42.0.73 | eecrt_3 |

#### 6.1.1.2 CINR as seen by different users

CINR at each UE can be calculated using the following formula,

$$CINR(dB) = 10 \log \frac{RSS_x}{N + \sum_{i=0}^{n} RSS_i} \tag{6.1}$$

In the above equation, $RSS_x$ is the received signal strength from C-eNodeB to which the UE is attached, $RSS_i$ is the signal strength of the interfering C-eNodeBs. Noise is usually restricted by the WLAN Receiver Sensitivity [40] and usually varies between -82 to -90 dBm depending on the receiver. All signal strengths on the right hand side are in mW and CINR is given in dB.

#### 6.1.1.3 Working of GDLS

The 802.11 specifications provides four distinct frequency ranges: 2.4 GHz, 3.6 GHz, 4.9 GHz, 5 GHz bands, with each band having a multitude of channels. C-eNodeB uses the 2.4 GHz band, which has 14 channels ranging from 2412 MHz to 2484 MHz out of which 13 are allowed in Europe. In this band, a proper deployment typically uses only the three non overlapping independent channels e.g Channels 1, 6, and 11 for North America and Europe [41]. With four C-eNodeBs and three non-overlapping channels in our network, we can contemplate the following two scenarios:

- **CASE 1:** Number of non overlapping Channels is greater than or equal to the number of C-eNodeBs, in which case each C-eNodeB uses a different channel
- **CASE 2:** Number of non overlapping Channels is lesser than the number of C-eNodeBs, in which case C-eNodeBs negotiate with each other for the best possible channel

The behavior of the CRA for the above two cases is monitored. The demonstration begins with **CASE 1** by starting just three C-eNodeBs and work towards achieving a global optima. Feedback is collected for a duration of one hour. Each C-eNodeB starts off by using one of available channels at random and applies GDLS to obtain a optimal channel (used by no other C-eNodeB). About 25 mins later, we begin a more interesting scenario in **CASE 2** by starting the fourth C-eNodeB. The addition of a new C-eNodeB into the system now creates a situation where atleast two C-eNodeBs will be sharing channel with one another. The new C-eNodeB measures the RSS from each neighbor on its WLAN interface and then starts on the channel of a neighbor from whom it measure minimum RSS. Since GDLS is greedy in its approach, each C-eNodeB will be lobbying for the best possible channel.

Figure 6.2 shows the channel numbers used by different C-eNodeBs during the run. From figure 6.2, one can notice that C-eNodeB 1 and C-eNodeB 3 start on Channel 1, while C-eNodeB 2 starts on channel number 11. Global optima is achieved during the initial few feedbacks (after the wait period is over). It is clearly visible, how starting the C-eNodeB 4 after 25 mins disturbs this optima. The CINR seen by each UE at different instants of time is shown in figures 6.3 and 6.4. The fluctuation of CINR observed at UE-1 (figure 6.3 (a)) is due to the impairment of WLAN receiver and is not because of the channel conditions. It is therefore exempt from any analysis further. During **CASE 1** a constant CINR is usually observed at all UEs after the completion of wait-period. However, **CASE 2** forces C-eNodeB 4 to share its channel with the incumbent C-eNodeBs thereby reducing the CINR levels of other C-eNodeBs. Each UE observes a CINR of 65-70 dB (based on its sensitivity) when it is not sharing its resource and this value gradually decreases otherwise. When UE-3 and UE-4 are on the same channel, the CINR observed at each of them is comparatively less than the CINR observed

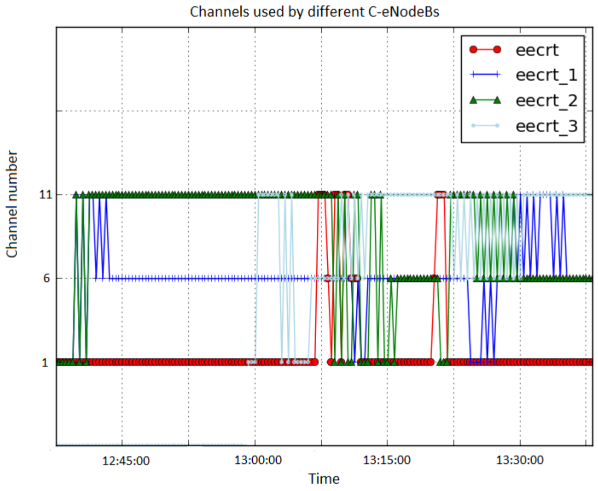

Figure 6.2: Channels used by different C-eNodeBs

when UE-2 and UE-4 are on the same channel. This can be attributed to the distance between the terminals.

#### 6.1.1.4 Performance evaluation of Asynchronous and Synchronous GDLS

When multiple C-eNodeBs start on the same channel, and start executing their routines together, there is a high probability that the system will never reach the global optima. Whenever a C-eNodeB senses neighbors on its channel, it attempts to switch to a new better channel with minimal interference. It is important to notify the neighbors of this switch, just to avoid them from using your current channel. This notification can be explicit (signaling the neighbors) or implicit (neighbors obtain this information when scanning the WLAN interface). In our system, since the WLAN interface cannot be scanned earlier than 5 seconds (time taken to scan the interface by the UE, send this as feedback information to the C-eNodeB and write this into the local dB), we adopt explicit signaling mechanism. This makes C-eNodeBs aware of the changes in the network much faster. When GDLS backs-off for a random duration before using the channel, and signals the neighbors to make them aware of its new channel, it avoids symmetric conflict situation assuming that the signals reaches the neighbor during the back-off interval i.e,

$$t_{back-off} \geq t_{rtt} \tag{6.2}$$

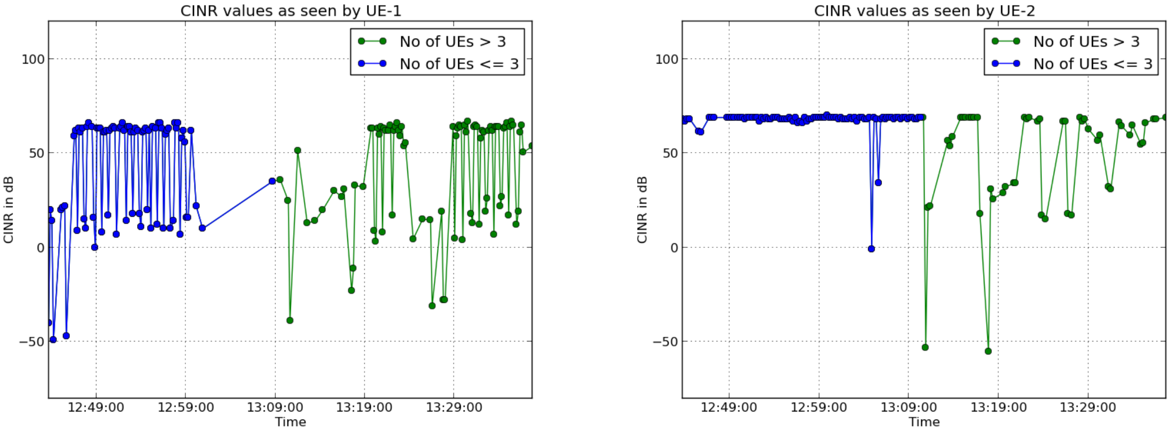

Figure 6.3: CINR as seen by UE-1 and UE-2 respectively

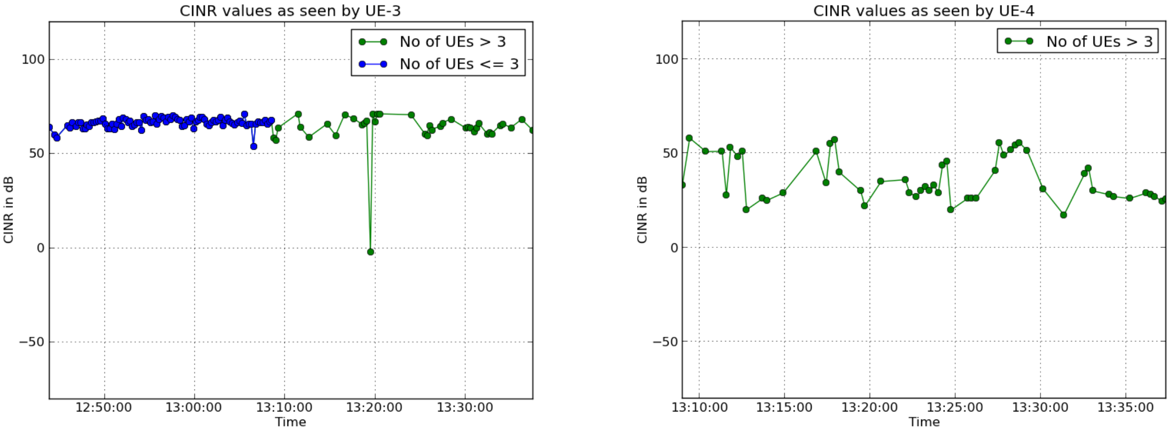

Figure 6.4: CINR as seen by UE-3 and UE-4 respectively

The performance of backed-off asynchronous-GDLS against non backed-off synchronous-GDLS is provided in this section:

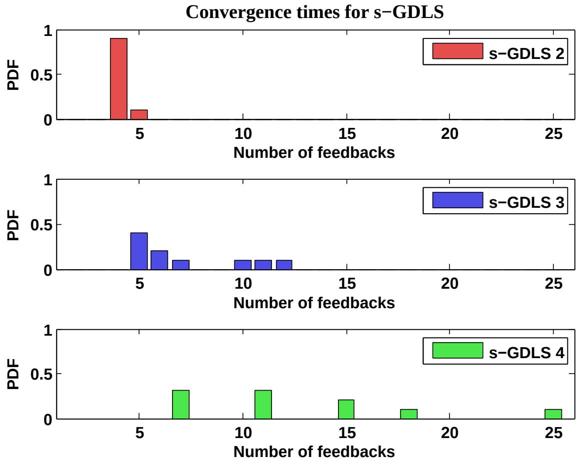

Figure 6.5: PDF of number of feedbacks for s-GDLS

We consider a system where the *No of C-eNodeBs = No of independent channels available*. Network is said to be converged and global optima is obtained when each C-eNodeB ends up on a different channel. The number of iterations required or the feedback cycles executed is then monitored for both a-GDLS and s-GDLS systems. A pdf of number of feedbacks required before achieving convergence is shown in figures 6.5 and 6.6. For s-GDLS with two C-eNodeBs in the network, global optima can be achieved for 4 feedbacks for about 90% of the time. The number of feedbacks however increase when more C-eNodeBs are added in the system. a-GDLS is on par with s-GDLS when considering two C-eNodeBs. However it outperforms s-GDLS when more C-eNodeBs are introduced. In this case, global optima can be achieved with a maximum of 6 feedbacks irrespective of the number of nodes in the system as illustrated in figure 6.6. Thus, asynchronous GDLS systems are much faster than the synchronous GDLS systems.

#### 6.1.1.4.1 Interference Couplings .

In this section we evaluate the performance of the CRA based on interference couplings experienced by the user, when a-GDLS and s-GDLS are employed. Again, we consider a case where *No of channels = No of C-eNodeBs*. Both the algorithms are run for a fixed period of time. The statistics of the C/I ratio experienced by users in the system is collected at the C-eNodeB. Metrics necessary for evaluating the system can be later gathered based on these

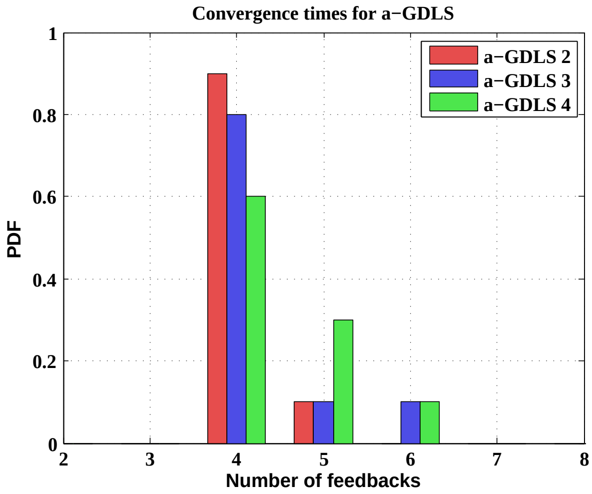

Figure 6.6: PDF of number of feedbacks for a-GDLS

statistics. Usually, to describe the efficiency of RRM, the following metrics are considered: The expected rate experienced by a user, Rate experienced by user under poor radio conditions (5% point of the cdf of C/I [35]). Therefore, we investigate the system performance based on the C/I statistics. We consider the following interference couplings scenarios:

- **Base Station to Base station coupling (BS-BS):** It indicates the interference power measured at C-eNodeB 2 from the surrounding C-eNodeBs [34].
- **Cellular Worst Coupling (CWC):** It indicates the worst interference experienced by UE 2 from the surrounding C-eNodeBs.

The selection of UE 2 and C-eNodeB 2 was based on the criteria that they lie at the center of established network and will experience maximum interference from all other network elements in the setup. SDLS uses the concept of thresholding interference levels while considering neighbors for BS-BS coupling [35] i.e, a conflict arises between two BSs when the interference coupling on a BS is larger than an absolute threshold value. The measurements here are done for cellular networks and neighboring BSs with receive powers larger than -64 dBm are considered to be conflicting BS. Threshold selection mechanisms are available in [42]. A direct comparison between GDLS and SDLS is not possible in this work because, unlike in [35] where threshold is calculated based on measurements from cellular networks, we consider measurements calculated for WLAN. Determining an optimal threshold value for classifying C-eNodeBs as conflicting neighbors hence becomes extremely difficult. However, a upper bound and a lower bound can be laid based on the measurements done at the C-eNodeB. An optimal threshold value $H$ will now lie between these bounds. An lower bound of -30 dBm and an upper bound of -90 dBm is chosen for our system. While $H=$ -30 dBm considers no conflicting neighbors as it leaves out all interference (The RSS seen from nearest neighbor C-eNodeB 1 is less than -30 dBm), $H=$ -90 dBm considers all neighbors as conflicting (Minimum RSS value seen at C-eNodeB 2 is around -90 dBm).

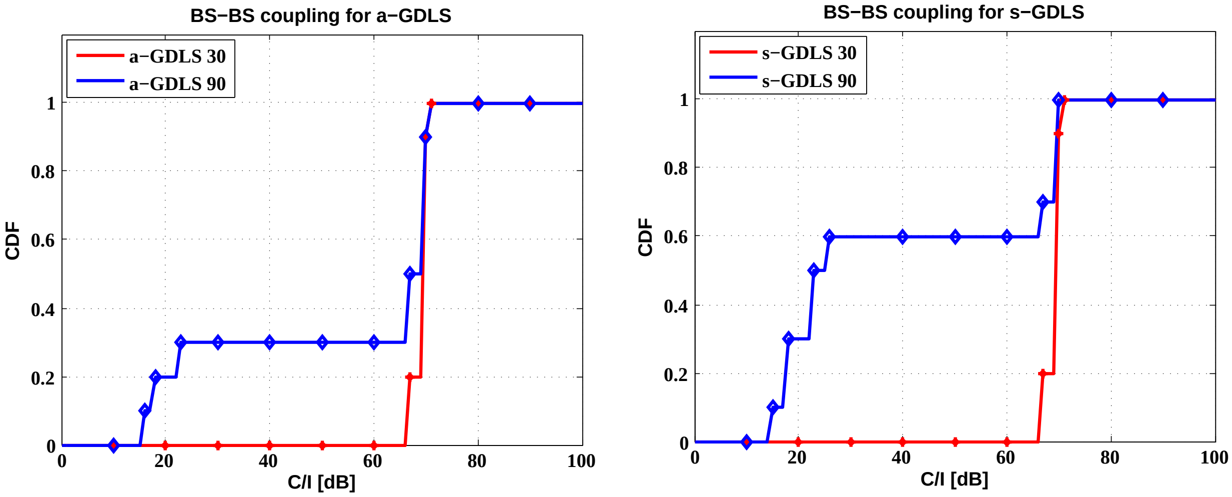

Figure 6.7: CDF of C/I for BS-BS coupling as seen at C-eNodeB 2

Both the algorithms are run for about 10 runs and the respective C/I values are noted. A probability density function on this data set yields the relative likelihood for each value of C/I. Once this data is normalized, cumulative sum is calculated in order to obtain cumulative distribution function (CDF). For each value of C/I, CDF indicates the probability of the result being equal to or less than C/I. C/I ratio at C-eNodeB 2 for BS-BS coupling with different thresholds can been in Figure 6.7. At $H=$ -30 dBm, when no interference is considered, a-GDLS and s-GDLS perform identically. However, at $H=$ -90 dBm a-GDLS clearly outperforms s-GDLS. For example, when C/I=60 dB the probability that C/I lies within -60 dBm is about 30% for a-GDLS while it is about 60% for s-GDLS. Since a-GDLS takes lesser time in converging than s-GDLS, high C/I values are visible even during earlier runs for a-GDLS.

Interference coupling for CWC and BS-BS are identical in our system, as the separation between C-eNodeBs and UEs is not more than 1m (refer

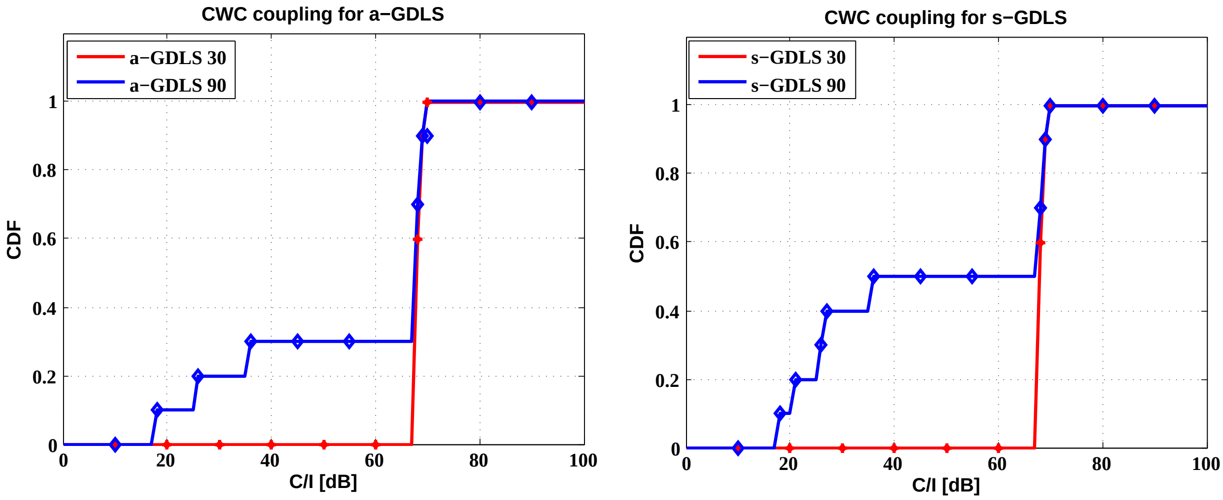

Figure 6.8: CDF of C/I for CWC coupling as seen at C-eNodeB 2

figure 6.1). Hence the system performance for CWC when running a-GDLS and s-GDLS algorithms is calculated exactly like in BS-BS coupling scenario. CDF of C/I ratio at UE-2 for CWC at $H=$ -30 dBm and $H=$ -90 dBm can been in figure 6.8. Again, at $H=$ -30 dBm, a-GDLS and s-GDLS perform identically due to the absence of interfering neighbors. At $H=$ -90 dBm a-GDLS clearly outperforms s-GDLS. The use of C/I statistics have no significance currently, apart from reestablishing the fact that a-GDLS performs better than s-GDLS. However, once the radio access for CRA is implemented with USRP boxes and the system becomes comparable to cellular networks, comparison between SDLS and GDLS algorithms will be interesting.

### 6.1.2 Providing Services to external networks

One service that the Cognitive Radio currently provides to external systems is delivering video content to a UE present in the network. This section describes the results obtained while furnishing this service. Any authorized external system can contact the DSM requesting for such a service and based on the signaling described in chapter 5, C-eNodeB downloads the video corresponding to the given URL and delivers it to specified UE over the data channel using WLAN interface. Since the data channels in this system are realized using TCP, the maximum receive throughput is,

$$T_{max} = \frac{R_{win}}{RTT} \tag{6.3}$$

where, $R_win$(TCP receive window size) is the amount of data that UE can accept without acknowledging the sender. It can be set upto 65,535 Bytes (64KB-1) for links with small round trip times (RTTs) like ours. However, for unfavorable links, setting large $R_win$ can lead to packet loss and excessive retransmissions thereby degrading the throughput of the system. RTT is the round-trip time for the path between UE and C-eNodeB, which is usually less than 1ms in our system (UE and C-eNodeB are less than 5m apart in distance). During one such illustration, transmitting a mp4 file of size 1.2 Gb required about 11 minutes with receive throughput as shown in figure 6.9:

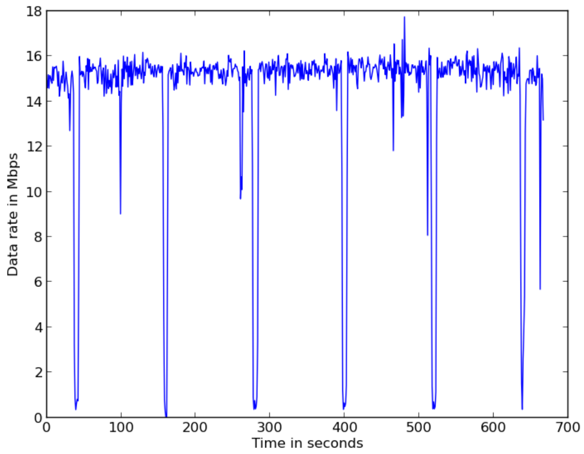

Figure 6.9: Receive throughput for video transmission over CRA

## 6.2 Results Evaluating System Performance

LTE network structure is flattened by the removal of the RNCs that were present in UMTS. If only this trend continues, more such decentralization can be expected in the next generation networks. With LTE being an *All-IP* network, future lies in implementing different LTE NEs on regular processors. Such processors should be capable of adhering to the LTE standards and should operate on large traffic at high speeds. The processors should be equipped with enough computation resources to satisfy demands on speed and throughput. The applicability and potential benefits of using IBM PowerEN processor (a multi-core, multi-threaded platform) for 4G eNodeBs was studied in [43]. In this section, we compare two processors Intel Pentium

Dual-core and Intel i7 processors with the specifications given in table, and verify the feasibility of using them for C-eNodeBs and UEs. Though the processors were chosen due to their availability rather than testing their feasibility to act as C-eNodeBs, they are a classic example for how the processing capabilities have increased over the last few years.

Table 6.2: Comparison of Intel Dual-core and i7 processors

| Characteristic | machine-1 | machine-2 |
|---|---|---|
| Processor | Dual-core | i7 - 3820 QM |
| No of cores | 2 | 4 |
| Cock rate | 1.6 GHz | 2.7 GHz |
| Release | 2006 | 2012 |
| RAM | 4 GB | 16 GB |

Network performance is defined as the ability of a network to provide the necessary functions to assist the user for communicating. It is characterized by parameters which are meaningful to the network and are used for the purpose of system design, configuration, operation and maintenance. In this work, we adopt a ***4S model*** as the metrics for measuring performance of the implemented Cognitive Radio System. The ***4S*** being:

- **Simplicity**
- **Speed**
- **Stability**
- **Scalablity**

The above metrics were considered due to the following reasons. Making the architecture simple adds a lot of mileage in designing a system. This makes implementation faster and easier to extend the system further. Speed is another important criteria for high performance architectures. The interaction between different network elements should be smooth and the interfaces have to be fast and efficient. Parallelism and threading of independent processes also play an important role in increasing the speed of the system. Since the system has to deal with high load and has to service each user with a predefined QoS, system has to be robust and stable. Practically, LTE eNodeBs can handle around 100 active users per cell even though this number depends on the memory, processing power, number of active bearers etc. The established CRA in comparison should be also be scalable to accommodate such loads. This section describes how the established CRA adheres to these metrics.

### 6.2.1 Simplicity

Simplicity of a design not only describes the ease of understanding the elementary design but also the amount of work required to add new functionality to the system. Using Python for implementation gives us this inherent advantage of simplicity. Besides, the modularity provided, along with the Object Oriented approach adopted, makes it easier to make additions to the system.

### 6.2.2 Speed

Speed is an important criteria in measuring the efficiency of the CRA. Since all the NEs are implemented using Python and Databases in MySQL, we correlate the speed of the system to the performance of MySQL and Python under various loads. Most time a C-eNodeB spends is in finding elements stored in its data structures or in performing intensive math operations. For example, searching if a UE is present in the list, searching for the last 100 CINR values as seen by a UE, Calculating CINR values based on feedback provided by UE etc. In high performance architectures, time remains a very critical factor and hence understanding the limits of the system and maximum loads it can process in required amount of time is very important.

#### 6.2.2.1 Evaluating Speed

Two functions, one for searching an element in a list and another for performing arithmetic (multiplication, ratios, averaging etc) are implemented and the performance of the two processors on these functions are shown in figure 6.10:

Understanding the importance of this graph is quite necessary for extending the system further. As expected, figure 6.10 shows that machine-2 (intel i7) performs better than machine-1 (intel dual core) on both the operations. For performing math operations on 200k elements in a list, machine-1 requires 20ms whereas machine-2 requires 10ms. The inference to be drawn here is that, any function which cannot afford atleast 10ms of processing time should not store elements of more than 200k in its list. For example, when DSM asks for the average CINR that a UE is experiencing over a day, and expects a response in say 1ms, the C-eNodeB should not have 86k elements in its list (assuming a feedback is provided every second) and then start processing. It should have maintained a different list of average hourly feedback. These kind of decisions can be taken based on the above graph.

Read performance(Number of records that can be read from the database)and

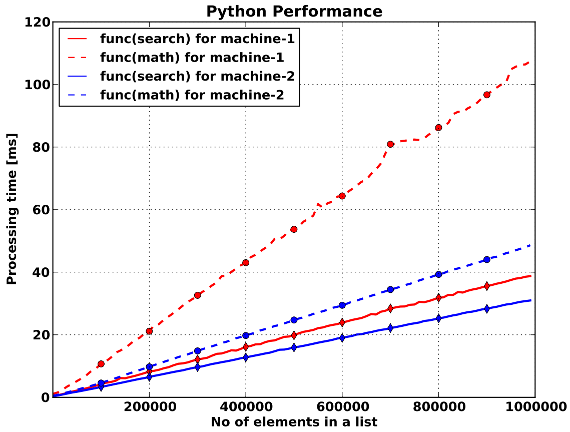

Figure 6.10: Python speed for search and math operations

Write performance (Number of records that can be written to the database) define the competence of any Database. We make use of the SSCursor class for fetching large data sets. Due to the parallelism offered by this class, huge difference in Read and Write performance is visible. Figures 6.11 and 6.12 describe the read and write performance of MySQL database on the two processors in consideration.

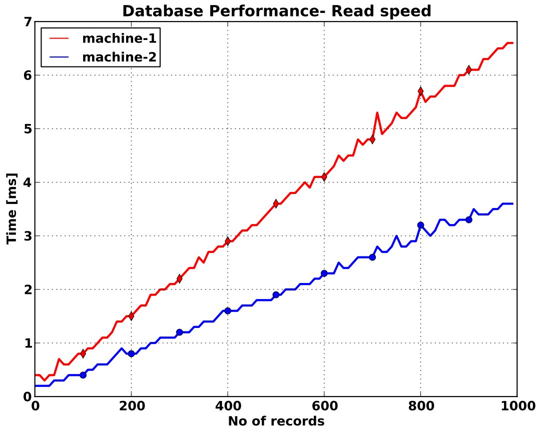

Figure 6.11: Database Read Performance

Figure 6.11 indicates that it takes about 3.5ms to read 1000 records on machine-2 while it takes almost twice the time on machine-1. Even

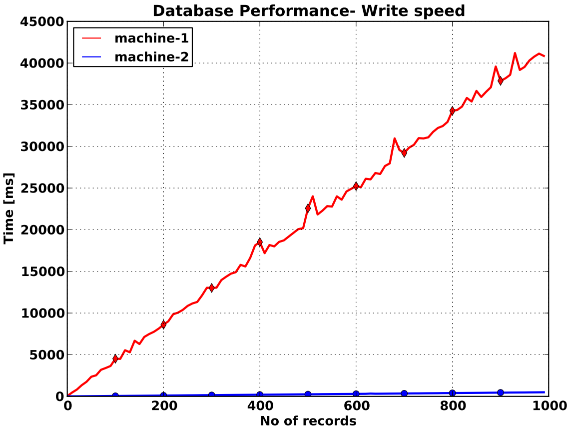

Figure 6.12: Database Write Performance

though the read performances are comparable, machine-2 clearly bamboozles machine-1 in terms of write performance. It takes about a minute to write 1000 records on machine-1 even when no other process was running alongside it. This high write times make them unfit for database writing. It is advisable that a copy of important records are also stored in the python list as read performance can become bottleneck for the system performance at times. At C-eNodeBs, it is advisable that read and writes should be written in separate threads and should not block the main process from performing its routine functions.

### 6.2.3 Stability

In LTE, channel state feedback reporting is used to provide the eNodeB with information about the downlink channel state and help them in optimizing the usage of frequency resources. These reports contain information about the scheduling and link adaptation related parameters the UE can support in the data reception. The channel state is estimated every 2 ms by the UE based on the downlink transmissions and feedbacks are sent to the eNodeB on PUCCH. This CQI measurement interval is said to be sufficient for capturing both the frequency selectivity and time variant behavior of the downlink channel [44]. Even though scheduling is performed in every 1ms interval (every transmission time interval/ TTl), the CQI reports need not be sent on every TTl by the UE and can be aperiodic in nature. However, the rate of CQI feedback definitely impacts the system performance. For systems with very less UE mobility such as ours, CQI reports should be sent atleast every 5 TTIs [45]. The stability of such a system can be measured by the consistency or the steadiness in which these feedbacks are provided within the specified duration.

#### 6.2.3.1 System Structure

The system consists of a C-eNodeB UE pair. For simplicity, we consider 1 UE per C-eNodeB and measure the capability of the system to provide channel state feedback every 1ms. The C-eNodeB sends a list of configuration parameters to the attached UE. The UE uses these parameters, makes an assessment about the operating conditions and sends a feedback to the system. The C-eNodeB based on the feedback received, reconfigures the UE with a new set of parameters. This process repeats continuously. The system is considered to be stable if the UE receives configuration/reconfiguration information every 1ms. Feedback interval is given by,

$$t_{feedback} = t_{processing} + (RTT/2) + t_{buffer} \tag{6.4}$$

assuming that it takes equal time to travel in both Uplink and Downlink directions. We consider, $(RTT/2)$ instead of just uplink travel time because the clocks at UE and C-eNodeB might not be synchronized (Even though Network Time Protocol (NTP) was used to synchronize the two clocks, millisecond precision was not achievable). $t_{buffer}$ is the time taken at the C-eNodeB to extract CQI from the received feedback report. Since we are considering just 1 UE per C-eNodeB, this time is negligibly small. Based on a constant $RTT$, the processing time $t_{processing}$ that can be allowed for RRM algorithms at the C-eNodeB can be deduced. Time at which the configuration information is sent and the time at which the feedback information reaches C-eNodeB (process level) is noted continuously at the C-eNodeB.

#### 6.2.3.2 Evaluating Stability

Figure 6.13(a) shows pdf of feedback rate measured at C-eNodeB. Clearly, machine-2 outperforms machine-1. Only 50% of the feedbacks are received on machine-1 within 1ms whereas 90% of the feedbacks are received before 1ms time on machine-2. For a better picture, figure 6.13 (b) indicates that more than 99% of feedbacks can be received on machine-2 within 3ms whereas it takes about 7ms on machine-1 to receive 99% of the feedbacks. This reestablishes the fact that python is quick enough for implementing high speed architectures.

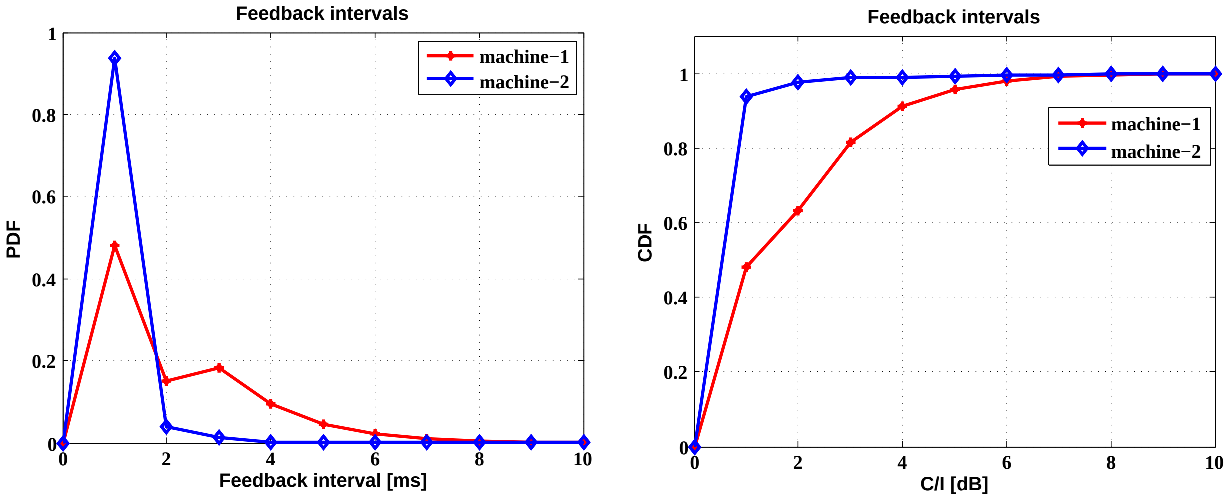

Figure 6.13: PDF and CDF of feedback rates at C-eNodeB

### 6.2.4 Scalability

Another important but often neglected metric of *4S* model is Scalability. The ability of the network to grow in terms of traffic load as well as the user load is critical. In our system, we define scalability as the number of UEs a C-eNodeB can accomodate without failing to provide necessary services.

#### 6.2.4.1 System Structure

The basic service provided by the C-eNodeB is configuring/reconfiguring UE with radio parameters for UL/DL communication. For a CR system where adapting according to environment is the prime focus, monitoring and updating the UE atleast every 10ms is necessary. The setup for evaluating the system in terms of scalabilty includes one C-eNodeB. The capability of the C-eNodeB to accommodate various UEs while providing service to each of them is monitored. Since, all the UEs will be sending data to the same port at C-eNode, the ability of the C-eNodeB to pick up the data from this port depends mostly on the buffer size available and speed of UNIX SELECT system calls. As the number of UEs increase, the time taken to serve each of them increases proportionally.

#### 6.2.4.2 Evaluating Scalability

Time taken by the C-eNodeB to service different UE loads on the two machines is shown in figure 6.14:

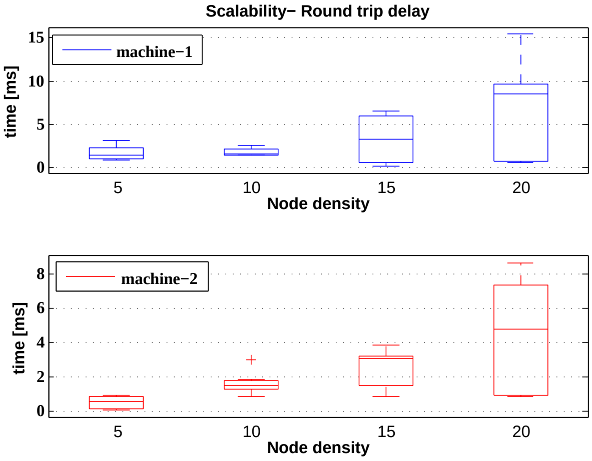

Figure 6.14: Scalability of CRA

This figure is very important in deciding the number of UEs that can be admitted by a C-eNodeB while providing satisfactory QoS. Again, machine-2 has a slight advantage over machine-1. While machine-2 takes a maximum of 4.5ms and an average of 2.5ms to provide service to 20 UEs (Note: Figure 6.14 shows RTT), machine-1 takes a maximum of 7.5ms and an average of 4.5ms to serve the same number of UEs. If we need to serve all UEs within 2ms, machine-1 can admit not more than 5 UEs while machine-2 can easily admit around 10 UEs.

# Chapter 7

# Conclusion and Future work

Cognitive radio is envisaged to solve wide number of wireless communication problems caused by spectrum scarcity and its inefficient usage, by exploiting the existing wireless spectrum opportunistically. They are usually characterized by *cognition capability*, *self-organization capability* and *reconfiguration capability*. In this work, a system level architecture is implemented for cognitive radio networks designed with the above characteristics in mind. Key functional modules (DSM, C-eNodeBs and UEs) and interfaces (S-c1, X2-c and Uu-c) are implemented along the lines of LTE, with limited but necessary functionality. The architecture has the ability to sense and gather information (for eg., signal strengths and interference) from its operating environment. It also interacts with geo-location databases to opportunistically make use of available spectra. The system can derive optimized parametrization based on the gathered information and reconfigure the operational parameters of the UEs according to the decisions made by C-eNodeBs. The architecture is flexible and extensible in the sense that the components in the architecture can be added, replaced or updated independently without impacting the other components and the integrity of the whole architecture. A software platform has been established to implement and test various Resource Management algorithms.

A dynamic channel allocation algorithm is implemented in order to demonstrate the self organization and resource negotiation capabilities of the implemented architecture. A setup where different UEs start on random WLAN channels and negotiate distributively in order to obtain global optima is established. During practical implementation of this algorithm, it was noted that asynchronous models perform much better than their synchronous counterparts as they avoid symmetric-conflict situations.

The established CRA is also capable of interacting with external networks and offer services if required. A paid-for premium video delivery service is implemented as an illustration where, C-eNodeB delivers video content to a cognitive terminal based on the instructions provided by an external systems to the DSM.

A system level performance of the network architecture is measured on two popular generic processors Intel Dual core and Intel i7 based on *4S* model defined in this work. Inferences were drawn regarding where to use threads, how to manage feedback data etc. The implemented architecture was found to be simple due to its choice of tools, fast due to the use of parallelism and threading, scalable as it could easily accommodate around 10 UEs while continuously serving each of them, stable as the system could satisfy the QoS requirements (TTI of 3ms in this case) for more than 99% of the time.

This network architecture can be extended in the future to include more functionality like mobility management, bearer management etc. Radio resource management algorithms have to be implemented for decision making and providing optimized radio parameters to UE. Charging policies can be placed in order to charge the external systems for service provided by CRA. USRPs can replace WLAN for radio access once the PHY channels are established and Radio interface protocols like RLC, RRC, MAC and PDCP are implemented using the already defined interfaces. This also makes room for comparing the performance of established CRA with LTE/LTE-A systems in the future.

# Bibliography

[1] E. Dahlman, S. Parkvall, and J. Skold, *4g: LTE/LTE-Advanced for Mobile Broadband*. Academic Press, 1 edition (March 21, 2011).

[2] F. Jondral, "Cognitive radio: A communications engineering view," *Wireless Communications, IEEE*, vol. 14, no. 4, pp. 28–33, 2007.

[3] I. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "A survey on spectrum management in cognitive radio networks," *Communications Magazine, IEEE*, vol. 46, no. 4, pp. 40–48, 2008.

[4] S. Haykin, "Cognitive radio: brain-empowered wireless communications," *Selected Areas in Communications, IEEE Journal on*, vol. 23, no. 2, pp. 201–220, 2005.

[5] J. Xiao, R. Hu, Y. Qian, L. Gong, and B. Wang, "Expanding lte network spectrum with cognitive radios: From concept to implementation," *Wireless Communications, IEEE*, vol. 20, no. 2, pp. 12–19, 2013.

[6] G. Szabo, D. Orincsay, B. Peter Gero, S. Gyori, and T. Borsos, "Traffic analysis of mobile broadband networks," *3rd Annual International Wireless Internet Conference, Austin, TX*, p. 22–24, 2007.

[7] "Spectrum Policy Task Force, ET Docket." http://transition.fcc.gov/sptf/files/E&UWGFinalReport.pdf.

[8] J. Mitola and J. Maguire, G.Q., "Cognitive radio: making software radios more personal," *Personal Communications, IEEE*, vol. 6, no. 4, pp. 13–18, 1999.

[9] S. Wang, L. Xie, H. Liu, B. Zhang, and H. Zhao, "Acra: An autonomic and expandable architecture for cognitive radio nodes," in *Wireless Communications and Signal Processing (WCSP), 2010 International Conference on*, pp. 1–5, 2010.

[10] D. Xu, Q. Zhang, Y. Liu, Y. Xu, and P. Zhang, "An architecture for cognitive radio networks with cognition, self-organization and reconfiguration capabilities," in *Vehicular Technology Conference (VTC Fall), 2012 IEEE*, pp. 1–5, 2012.

[11] B. S. Manoj, R. Rao, and M. Zorzi, "COGNET: a cognitive complete knowledge network system," *Wireless Communications, IEEE*, vol. 15, no. 6, pp. 81–88, 2008.

[12] M. Ahmed, S. Hailes, V. Kolar, M. Petrova, and P. Mahonen, "A component-based architecture for cognitive radio resource management," in *Cognitive Radio Oriented Wireless Networks and Communications, 2009. CROWNCOM '09. 4th International Conference on*, pp. 1–6, 2009.

[13] P. Sutton, J. Lotze, H. Lahlou, S. Fahmy, K. Nolan, B. Ozgul, T. Rondeau, J. Noguera, and L. Doyle, "IRIS: an architecture for cognitive radio networking testbeds," *Communications Magazine, IEEE*, vol. 48, no. 9, pp. 114–122, 2010.

[14] M. Mueck, A. Piipponen, K. Kalliojarvi, G. Dimitrakopoulos, K. Tsagkaris, P. Demestichas, F. Casadevall, J. Perez-Romero, O. Sallent, G. Baldini, S. Filin, H. Harada, M. Debbah, T. Haustein, J. Gebert, B. Deschamps, P. Bender, M. Street, S. Kandeepan, J. Lota, and A. Hayar, "Etsi reconfigurable radio systems: status and future directions on software defined radio and cognitive radio standards," *Communications Magazine, IEEE*, vol. 48, no. 9, pp. 78–86, 2010.

[15] A. MacKenzie, J. Reed, P. Athanas, C. Bostian, R. M. Buehrer, L. DaSilva, S. Ellingson, Y. Hou, M. Hsiao, J.-M. Park, C. Patterson, S. Raman, and C. da Silva, "Cognitive radio and networking research at virginia tech," *Proceedings of the IEEE*, vol. 97, no. 4, pp. 660–688, 2009.

[16] J. Xiao, R. Hu, Y. Qian, L. Gong, and B. Wang, "Expanding LTE network spectrum with cognitive radios: From concept to implementation," *Wireless Communications, IEEE*, vol. 20, no. 2, pp. 12–19, 2013.

[17] D. Astely, E. Dahlman, A. Furuskar, Y. Jading, M. Lindstrom, and S. Parkvall, "LTE: the evolution of mobile broadband," *Communications Magazine, IEEE*, vol. 47, no. 4, pp. 44–51, 2009.

[18] Aricent, "The LTE Network Architecture: A comprehensive tutorial." http://goo.gl/SgeDT.

[19] 3GPP TS 36.401, "Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Architecture description(Release 8)."

[20] H. Holma and A. Toskala, *LTE for UMTS – OFDMA and SC-FDMA Based Radio Access.* John Wiley and Sons, 1 edition (2009).

[21] "Long Term Evolution (LTE): A Technical Overview." http://bit.ly/19YGo0g.

[22] S. Sesia, I. Toufik, and M. Baker, *LTE - The UMTS Long Term Evolution: From Theory to Practice.* John Wiley and Sons, 2 edition (2012).

[23] 3GPP TS 36.410, "Evolved Universal Terrestrial Radio Access Network (E-UTRAN); S1 general aspects and principles; General Description (Release 11)."

[24] 3GPP TS 36.420, "Evolved Universal Terrestrial Radio Access Network (E-UTRAN); X2 general aspects and principles; General Description (Release 11)."

[25] J. M. III, *Cognitive Radio: An Integrated Agent Architecture for Software Defined Radio.* PhD thesis, Computer Communication Systems Laboratory, Dept. of Teleinformatics, Royal Institute of Technology, Stockholm, Sweden, 2000.

[26] H. Karimi, "Geolocation databases for white space devices in the UHF tv bands: Specification of maximum permitted emission levels," in *New Frontiers in Dynamic Spectrum Access Networks (DySPAN), 2011 IEEE Symposium on*, pp. 443–454, 2011.

[27] C. Prehofer and C. Bettstetter, "Self-organization in communication networks: principles and design paradigms," *Communications Magazine, IEEE*, vol. 43, no. 7, pp. 78–85, 2005.

[28] R. Zhang, Y.-C. Liang, and S. Cui, "Dynamic resource allocation in cognitive radio networks," *Signal Processing Magazine, IEEE*, vol. 27, no. 3, pp. 102–114, 2010.

[29] A. Eisenblatter, M. Grotschel, and A. Koster, "Frequency planning and ramifications of coloring," *Discussiones Mathematicae Graph Theory*, vol. 22, no. 1, pp. 51–88, 2002.

[30] A. Gamst, "Some lower bounds for a class of frequency assignment problems," *Vehicular Technology, IEEE Transactions on*, vol. 35, no. 1, pp. 8–14, 1986.

[31] R. Anthony, “Scalable and efficient graph colouring in 3 dimensions using emergence engineering principles,” in *Self-Adaptive and Self-Organizing Systems, 2008. SASO ’08. Second IEEE International Conference on*, pp. 370–379, 2008.

[32] L. Garcia, K. Pedersen, and P. Mogensen, “Autonomous component carrier selection: interference management in local area environments for LTE-advanced,” *Communications Magazine, IEEE*, vol. 47, no. 9, pp. 110–116, 2009.

[33] F. Ahmed, O. Tirkkonen, M. Peltomaki, J. M. Koljonen, C. H. Yu, and M. Alava, “Distributed Graph Coloring for Self-Organization in LTE Networks,” *Journal of Electrical and Computer Engineering*, 2010.

[34] J. Neel and J. Reed, “Performance of distributed dynamic frequency selection schemes for interference reducing networks,” in *Military Communications Conference, 2006. MILCOM 2006. IEEE*, pp. 1–7, 2006.

[35] M. Peltomaki, J. Koljonen, O. Tirkkonen, and M. Alava, “Algorithms for self-organized resource allocation in wireless networks,” *Vehicular Technology, IEEE Transactions on*, vol. 61, no. 1, pp. 346–359, 2012.

[36] M. Dillinger, K. Madani, and N. Alonistioti, *Software Defined Radio: Architectures, Systems and Functions*. Wiley, April 2003.

[37] J.-C. Chen and C.-K. Wen, “A novel cognitive radio adaptation for wireless multicarrier systems,” *Communications Letters, IEEE*, vol. 14, no. 7, pp. 629–631, 2010.

[38] GNU Radio, “USRP overview and related documents.” `http://gnuradio.org/redmine/projects/gnuradio/wiki/USRP`.

[39] “Protocol to Access Spectrum Database.” http://datatracker.ietf.org/doc/draft-ietf-paws-protocol/.

[40] P. Chadwick, “Sensitivity and range in wlan receivers,” in *Radio LANs and MANs, IEE Colloquium on*, pp. 3/1–3/5, 1995.

[41] CISCO Systems, “WLAN Channel deployment guide.” http://www.cisco.com/en/US/docs/wireless/technology/channel/deployment/guide/Channel.html.

[42] J. Koljonen, M. Alava, M. Peltomaki, and O. Tirkkonen, “Distributed generalized graph coloring,” in *Self-Adaptive and Self-Organizing Systems (SASO), 2010 4th IEEE International Conference on*, pp. 174–183, 2010.

[43] A. Vega, P. Bose, A. Buyuktosunoglu, J. Derby, M. Franceschini, C. Johnson, and R. Montoye, “Architectural perspectives of future wireless base stations based on the IBM PowerEN processor,” in *High Performance Computer Architecture (HPCA), 2012 IEEE 18th International Symposium on*, pp. 1–10, 2012.

[44] N. Kolehmainen, J. Puttonen, P. Kela, T. Ristaniemi, T. Henttonen, and M. Moisio, “Channel quality indication reporting schemes for utran long term evolution downlink,” in *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*, pp. 2522–2526, 2008.

[45] R. Basukala, H. A. M. Ramli, K. Sandrasegaran, and L. Chen, “Impact of CQI feedback rate/delay on scheduling video streaming services in LTE downlink,” in *Communication Technology (ICCT), 2010 12th IEEE International Conference on*, pp. 1349–1352, 2010.