FARAMIR
Mapping Your Radio Environment

2007 - 2013

Flexible and Spectrum Aware Radio Access through Measurements and Modelling in Cognitive Radio Systems

FARAMIR

Document Number D6.2

Prototype Description and Field Trial Results

Contractual date of delivery to the CEC:	30.06.2012
Actual date of delivery to the CEC:	30.06.2012
Project Number and Acronym:	248351 - FARAMIR
Editors:	Liljana Gavrilovska, Vladimir Atanasovski (UKIM)
Authors:	Ferran Casadevall (UPC), Jordi Pérez-Romero (UPC), Liljana Gavrilovska (UKIM), Vladimir Atanasovski (UKIM), Daniel Denkovski (UKIM), Valentin Rakovic (UKIM), Mihajlo Pavloski (UKIM), Lorenzo Iacobelli (TCS), Pascale Fouillot (TCS), Ioannis Dagres (IASA), Andreas Polydoros (IASA), Jaap van de Beek (HWSE), Erik Lidström (HWSE), Tao Cai (HWSE), Antoine Dejonghe (IMEC), Tim Farnham (TREL), Woon Hau Chin (TREL), Filipe Ramos (TREL), Mohammed Faies (TREL), Berna Sayrac (FT), Junaid Ansari (RWTH), Elena Meshkova (RWTH), Petri Mähönen (RWTH), Jad Nasreddine (RWTH), Janne Riihijärvi (RWTH), Peng Wang (RWTH)
Participants:	RWTH, IMEC, UPC, HWSE, TREL, IASA, TCF, UKIM, FT
Workpackage:	WP6
Security:	PU
Nature:	R, P
Version:	1.0
Total number of pages:	142

Abstract:

This document provides a detailed description of the FARAMIR prototypes together with the specific FARAMIR innovations, their practical application, as well as field and experimental results for all the developed solutions. The document first elaborates on the specific implementation issues regarding the generic FARAMIR Radio Environmental Map (REM) prototype (whose architecture is extensively discussed in earlier deliverables D2.4, D4.3 and D6.1) and its reliability for accurate REM creation. The document then discusses the newly developed hardware within the FARAMIR project that is able to perform fast, accurate and reliable spectrum sensing in a wide range of frequencies in a small-form factor design. Finally, the deliverable discusses the specific usages of the developed FARAMIR REM prototype in selected scenarios such as femtocell power and channel optimization, LTE usage of TVWS, advanced Radio Resource Management (RRM) and Mobile Ad-hoc NETworks (MANETs). All gathered insights are analyzed on a system engineering level showing which conclusions can be drawn from the experimental work. The document clearly shows the benefits and the potentials of the FARAMIR introduced REM prototype for various applications in future wireless networks allowing optimized network performances and optimal usage of the available resources.

Keywords: Radio Environmental Map (REM), Prototype, SCALable raDIO (SCALDIO), Femtocell, Long Term Evolution (LTE), TV White Space (TVWS), Radio Resource Management (RRM), Mobile Ad-hoc NETwork (MANET).

Document Revision History

Version	*Date*	*Author*	*Summary of main changes*
0.1	17.02.2012	UKIM	Proposed ToC
0.2	10.05.2012	UPC	Initial UPC contribution (Section 5)
0.3	31.05.2012	UKIM	Updated ToC
0.4	12.06.2012	TCS	TCS contribution (first part)
0.5	14.06.2012	UKIM	Contributions from UKIM and HWSE
0.6	26.06.2012	UKIM	Integrated latest contributions from HWSE, TRL, IASA, UKIM and IMEC
1.0	29.06.2012	Janne Riihijärvi (RWTH)	Final contributions and editing
1.0f	29.06.2012	Petri Mähönen (RWTH)	Coordinator review and approval

Table of Contents

1 Introduction

The notion of Radio Environment Map (REM), originally envisioned as a two-dimensional representation of the radio field strength, is now foreseen as a rich database or knowledge base storing various kinds of radio environmental information. This information can either be static (e.g. locations of transmitters and/or receivers, terrain model etc.) or dynamic (e.g. propagation environment, up-to-date spectrum measurements, users' activity patterns etc.) and can subsequently be used for various optimization procedures in secondary spectrum access scenarios. Therefore, the REM concept is envisioned as a powerful enabler and/or facilitator for reliable dynamic spectrum access, and more generally for improving the environmental awareness and spectral efficiency of wireless networks.

The FARAMIR project targets the analysis of the REM concept, showcasing its practical feasibility by developing a flexible and modular prototype, which is being instantiated in several different scenarios aiming to prove the REM benefits in operational networks. The developed prototype and innovations within the FARAMIR project are a distinct proof-of-concept of the REM idea (most of the previous work on REM is purely theoretical), whereas their evaluations clearly show the gained benefits from the REM usage. All developed solutions integrate the theoretical and prototyping work from WP2, WP3, WP4 and WP5 showing the pragmatic FARAMIR project approach.

This document describes the FARAMIR developed REM backend prototype and the specific (sub)prototypes and innovations that utilize the developed REM technology to perform radio environmental aware resource management. It gives implementation details on the FARAMIR related innovations and shows extensive field results from the testing of the developed prototype(s). The developed solutions follow the defined FARAMIR system architecture [1] and the necessary prototype requirements [2].

The document is structured as follows. Chapter 2 elaborates the implementation details of the REM backend technology developed throughout the FARAMIR project duration. It gives specific details on the developed components and interfaces that enable the REM construction and its subsequent usage. Additionally, Chapter 2 also discusses the reliability and the accuracy of the REM constructed with the FARAMIR's REM backend in terms of performance assessments of different Measurement Capable Devices (MCDs) and different spatial interpolation techniques utilized for the REM construction with additional results given in the appendix. Chapter 3 introduces the novel hardware developed within the FARAMIR project activities capable of fast and accurate wideband spectrum sensing. It shows the practical feasibility of a spectrum sensor within the power and cost constraints of a mobile device. Chapters 4 and 5 showcase applicable instantiations of the REM concept in real operational networks, i.e. cognitive femto-cells and LTE usage of TV White Space (TVWS). These chapters discuss the specific implementation details that enable the connection between the REM backend and the LTE equipment and provide field results that prove the benefits of using REM-assisted radio resource management. Chapter 6 elaborates the REM benefits for advanced radio resource management between differently profiled primary and secondary systems. Chapter 7 shows the applicability of the REM in mobile ad-hoc networks scenarios. Finally, Chapter 8 concludes the document highlighting the most important conclusions from the developed REM prototype and the conducted experimental work.

2 FARAMIR REM Framework

The FARAMIR REM framework represents the core part of the FARAMIR system architecture [1] responsible for gathering, storing and processing of the REM information. The REM framework is also responsible for providing REM information to the REM users, Figure 2-1. This allows various instantiations of the REM concept within different scenarios depending on the actual REM user profile. This chapter describes the implementation aspects of all REM backend components, measurement capable devices, and the different interfaces, which form the cornerstone of the FARAMIR's REM concept and foster the various usage applications described in the following chapters of this document. Additionally, the chapter discusses the developed Graphical User Interface (GUI) that allows visualization of the REM data and gives performance assessment analysis of the constructed REM reliability and accuracy.

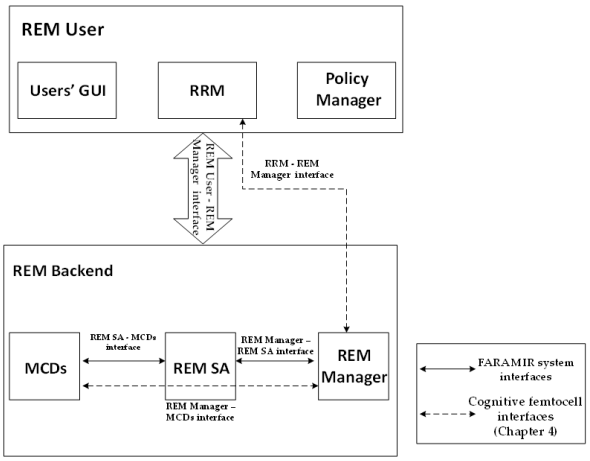


Figure 2-1: Generic FARAMIR system architecture relative to the REM backend.

The FARAMIR's REM framework comprises three main components (Figure 2-1), i.e.:

- **Measurement Capable Devices (MCDs)** - all network devices capable of performing spectrum measurements;
- **REM data Storage and Acquisition unit (REM SA)** - main REM backend storage entity capable of storing raw and processed REM data and
- **REM Manager** – REM backend component responsible for requesting measurements and extracting and processing the data stored in the REM SA.

The following subsection provides extensive implementation details on the components and their interconnecting interfaces.

2.1 Components and Interfaces

This subsection describes the implementation of the legacy MCDs, the REM SA and the REM Manager along with their associated functionalities, features and communication interfaces.

2.1.1 MCDs

The FARAMIR's REM backend supports several different types of MCDs:

- Universal Software Radio Peripheral 2 (USRP2),

- TI eZ430 RF2500,
- High end spectrum analyzers; and
- Novel FARAMIR spectrum sensing engine SCAlable raDIO (SCALDIO).

The MCDs are developed to support multiple functionalities like *MCD registration/deregistration* to the REM SA, real-time periodic and triggered *spectrum measurements* as well as remote and on-the-fly *reconfiguration*. All of the MCDs, regardless of their type, utilize the same unified MCD-REM_SA interface and protocol in order to exchange the required information with the REM_SA. Upon activation, the REM SA address and TCP port are provided to the MCD in order to create a TCP connection between them. Moreover, the initial configuration parameters of the MCD are manually specified at the startup. After the initialization, the MCD registers to the REM SA by sending its initial configuration parameters and is prepared for spectrum measurement tasks. The MCD's measurement data can be sent periodically at every sweep completion or when a trigger from the REM_SA occurs. If the MCD is to be stopped, then it deregisters from the REM SA.

In the following we provide a brief overview of the components developed in order to interface the legacy MCD solutions towards the REM framework.

USRP2 MCD. The USRP2 sniffer implementation [3] is based on a C++ sensing solution that utilizes the GNU radio basic and extended block. Its hardware, composed of a motherboard and a daughterboard, enables flexible tuning of Tx/Rx characteristics of the radio. According to its spectrum measurement properties, this device belongs to the mid-end group of spectrum sensing devices. Figure 2-2 depicts the general USRP2 spectrum sensing architecture [3]. It contains six GNU Radio and one custom made processing blocks written in C++:

1. ***Block no.1****:* creates the USRP2 source and controls the hardware (sets up sampling rate and tuning frequencies);
2. ***Block no.2****:* converts a stream of complex samples to vector of complex samples;
3. ***Block no.3****:* calculates an FFT transform on input complex samples;
4. ***Block no.4****:* calculates squared magnitude (power) on complex samples;
5. ***Block no.5****:* custom made processing block - selects between different detector types and sensing modes, initiates frequency switching;
6. ***Block no.6****:* calculates magnitude (amplitude) on complex samples;
7. ***Block no.7****:* calculates an FFT transform on input real samples.

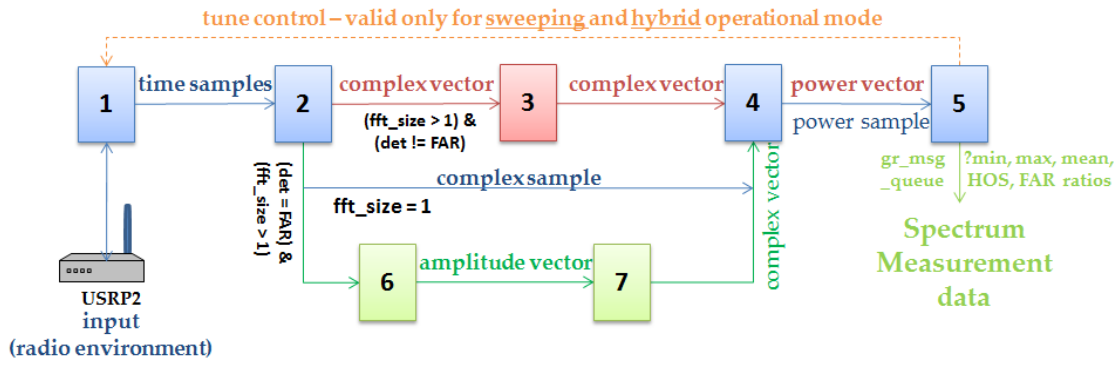


Figure 2-2: USRP2 based sensing application architecture.

Due to the relatively high processing power, this MCD implementation enables the usage of several detector types, such as *min*, *mean*, *max hold* detectors, *Higher Order Statistics (HOS)* and *FFT Averaging Ratio (FAR)* detector types, and three modes of operation, i.e. *real-time measurement* mode, *sweeping* mode and *hybrid* mode of operation. One of the greatest disadvantages of the USRP2 MCD is its non-linear input-output characteristic which requires manual calibration of the device [4].

TI eZ430 RF2500 MCD. The Texas Instruments (TI) eZ430is an USB-based MSP430 wireless development tool. It is not originally a spectrum sensing device and needs custom made software to perform sensing. Figure 2-3 depicts an implementation example of TI eZ430. The TI_sense represents a custom developed C-language based sensing application ported on the device. Application porting and RF part configuration are conducted via IAR Embedded Workbench Programming environment.

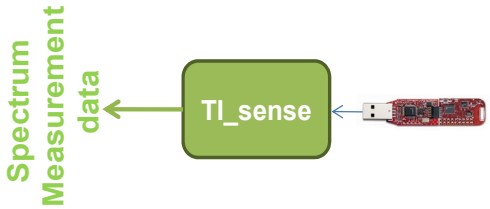


Figure 2-3: TI eZ430 based sensing application architecture.

The TI eZ430 covers the full 2.4GHz ISM band and possesses modest sensing capabilities in terms of its sensitivity and data resolution. The device has a linear input/output characteristic, thus the calibrating curve is defined only with a given offset that reflects the current state of the radio environment (i.e. deep fading state). On the other hand, the TI RF2500 has limited processing capabilities making it suitable only for energy based detection. Its spectrum measurement properties put it into the group of low-end devices [4].

2.1.2 REM SA component

The REM SA is the component in charge of storing and managing raw and processed REM information as well as active MCD data. The implementation of this component provides several functions supporting the requirements of the FARAMIR REM architecture. The storage database of the REM SA is developed as an SQL database in Microsoft SQL Server 2008 development environment and its structure is shown on Figure 2-4.

The implementation of the REM SA SQL database includes seven tables:

- MCDs;
- Measurement configurations;
- Spectrum data;
- Radio Interference Fields;
- Transmitters;
- Propagation models and
- Communication devices.

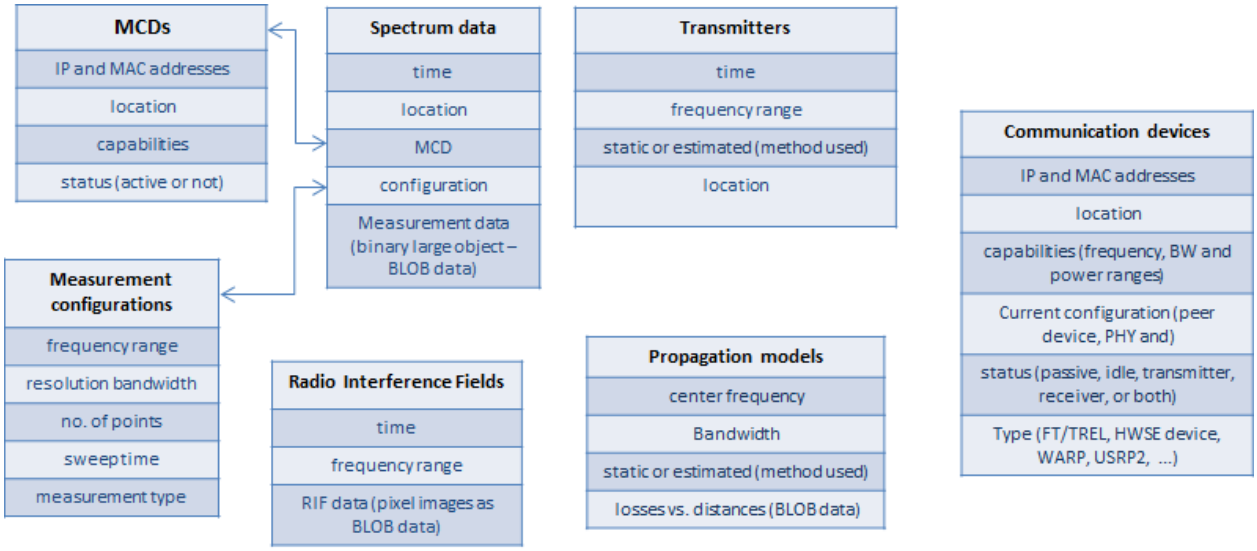


Figure 2-4: REM SA data storage structure.

The ***MCDs* table** is used to keep sensor information such as MCD addresses (IP and MAC), MCD location (the current location of the MCD expressed in Cartesian coordinate system), MCD status (Active, Idle or Off) and MCD capabilities (referring to the different types of MCDs, i.e. USRP2, TI eZ430, SCALDIO). This table carries valuable information enabling the REM backend to be aware of and track the active MCDs in the network.

The ***Measurement configurations* table** is used to store the measurement types that can be performed by the sniffer devices specified by start and end frequency, resolution bandwidth, number of measurement points, sweep time and measurement type (referring to the min, mean and max hold detectors). Each MCD (i.e. sniffer) is associated to an entry in this table corresponding to its current measurement configuration.

The ***Spectrum data* table** stores the raw measurement results as Binary Large Object (BLOB) data from the active MCDs in the network. The BLOB data approach enables the database to store large quantities of measurement information without affecting the overall performance of the REM SA component. Besides the measurement data, this table also stores information about the location (useful if the measurement is done by mobile MCD) and time the data was collected, as well as the MCD and measurement configuration that were employed to perform the given measurement.

The ***Radio Interference Field* table** stores pixel images of the Radio Interference Field Estimation (RIFE) maps for a given frequency range and time. The RIFE maps are also stored as BLOBs in the REM SA.

The ***Transmitters* table** stores information regarding the active transmitting devices in the area regarding their location (expressed in Cartesian coordinate system), operating frequency (i.e. frequency range), time (time at which they were detected as active) as well as the method of detection. The detection method can be either *static* – referring to predefined administrator input or *estimated* – using the MCDs to detect the active transmitter. In the case of estimated detection, the field also carries information regarding the type of estimation used.

The ***Propagation models* table** is used to store propagation model information such as center frequency (carrier frequency for the given propagation model), bandwidth, type of the model

(*static* – predefined or *estimated* – dynamically estimated by the MCDs) and the results (defined as the pathloss in dependence of the distance) stored as BLOB data.

The ***Communication devices* table** stores information regarding the active network devices that utilize the stored REM information, i.e. the active REM users regarding their addresses (IP and MAC), location (the current location of the device expressed in Cartesian coordinate system), capabilities (frequency range, max bandwidth, noise floor, minimal and maximal transmit power), current configuration (current transmit power and bandwidth), status (idle, receiver, transmitter, transceiver), type (LTE femtocell, LTE TWVS) etc. Similar to the ***MCDs***, this table facilitates the REM backend to be aware and serve the active REM users in the network.

2.1.3 MCD – REM SA interface

The MCD – REM SA interface provides the communication between the respective components. The communication is handled by asynchronous TCP/IP sockets, where the socket server is realized in C#. The socket server module performs the appropriate data conversion and the read/write operations in the REM SA database. The socket server can simultaneously serve a significant number of socket clients (thousands of MCD devices).

This interface includes the definitions of several messages handling the interactions between the MCD and the REM SA components. These messages along with their brief explanations are presented below.

- The `RegisterReq` and `RegisterRsp` messages are used to register a new MCD in the REM SA. The registration message carries the MCD location, addresses, and measurement capabilities in terms of the supported frequency range, bandwidth, measurement modes and detection capabilities. The response message returns the assigned MCD ID.

```
    public struct RegisterReq
    {
        public UInt16 MsgType;
        public UInt16 mcdType;
        [MarshalAs(UnmanagedType.ByValTStr,
SizeConst = MCDRemoteIFDefs.MACADSIZE)]
        public String mcdMAC;
        [MarshalAs(UnmanagedType.ByValTStr,
SizeConst = MCDRemoteIFDefs.IPADSIZE)]
        public String mcdIP;
        public Single currPosX;
        public Single currPosY;
        public Single currPosZ;
        public UInt64 startFreq;
        public UInt64 endFreq;
        public UInt64 resBW;
        public UInt16 noPoints;
        public Single sweepTime;
        public UInt16 detType;
        public UInt16 measMode;
        public UInt16 detMetric;
    }
```

```
public struct RegisterRsp
{
    public UInt16 MsgType;
    public UInt32 mcdID;
    public UInt32 groupID;
    public UInt32 confID;
}
```

- The `MeasurementReq` and `MeasurementRsp` messages are used to query/return specific measurements from an active MCD. The `MeasurementRsp` message reporting to the REM SA can be either triggered or periodic.

```
    public struct MeasurementReq
    {
        public UInt16 MsgType;
        public UInt16 useDefaultConf;
        public UInt32 mcdID;
        public UInt32 confID;
```

```
public struct MeasurementRsp
{
    public UInt16 MsgType;
    public UInt32 mcdID;
    public UInt32 confID;
    public UInt16 noPoints;
```

```
        public UInt64 startFreq;
        public UInt64 endFreq;
        public UInt64 resBW;
        public UInt16 noPoints;
        public Single sweepTime;
        public UInt16 detType;
        public UInt16 measMode;
        public UInt16 detMetric;
        public UInt16 reporingMode;
    }
```

```
        public Single currPosX;
        public Single currPosY;
        public Single currPosZ;
        [MarshalAs(UnmanagedType.ByValTStr,
SizeConst = MCDRemoteIFDefs.TIMESIZE)]
        public String timeInstant;
        // raw data follows up as doubles
    }
```

- The `ReconfigureReq` message is used to enforce a reconfiguration of a specified MCD by the REM SA (REM Manager). The `ReconfigureRsp` message returns the status of the reconfiguration (success or failure).

```
public struct ReconfigureReq
{
    public UInt16 MsgType;
    public UInt32 mcdID;
    public UInt32 confID;
    public UInt64 startFreq;
    public UInt64 endFreq;
    public UInt64 resBW;
    public UInt16 noPoints;
    public Single sweepTime;
    public UInt16 detType;
    public UInt16 measMode;
    public UInt16 detMetric;
}
```

```
public struct ReconfigureRsp
{
    public UInt16 MsgType;
    public UInt32 mcdID;
    public UInt16 recStatus;
}
```

- `LocationReq` and `LocationRsp` are messages used to query/return the location of a registered MCD device in the REM SA.

```
public struct LocationReq
{
    public UInt16 MsgType;
    public UInt32 mcdID;
}
```

```
public struct LocationRsp
{
    public UInt16 MsgType;
    public UInt32 mcdID;
    public Single currPosX;
    public Single currPosY;
    public Single currPosZ;
}
```

- The `StopMeasurement` message is used to trigger an active MCD to stop performing spectrum measurements.

```
public struct StopMeasurement
{
    public UInt16 MsgType;
    public UInt32 mcdID;
}
```

- The `DeRegisterReq` message is used by the MCD to de-register from the REM SA.

```
public struct DeRegisterReq
{
    public UInt16 MsgType;
    public UInt32 mcdID;
}
```

2.1.4 REM Manager Component

The main role of the REM Manager is to perform the processing tasks in the FARAMIR's REM backend (e.g. data fusion, node localization, propagation model estimation etc.). It is designed to be extensible due to the multiple processing modules that can be installed within. The modules can communicate among each other and are also connected to the external interfaces (REM SA –

REM Manager, REM Manager – REM User). The REM Manager implementation is developed in C# programming language (.NET Framework 3.5) and includes four modules i.e. toolboxes:

- Spatial interpolation toolbox;
- Statistical–analyses toolbox;
- Transmitter localization toolbox and
- RRM toolbox.

Spatial interpolation toolbox. The spatial interpolation toolbox is responsible for data fusion and creation of RIFE maps. It can utilize either the pure IDW or IDW modified Shepard's method interpolation technique (see [5] and [6] for a detailed description), but other interpolation techniques can also be added (such as kriging, radial basis function, polyharmonic spline, thin plate spline, triangulated irregular network, Gradient plus Inverse Distance Squared etc.). The toolbox incorporates two processing functions:

```
public int iwinterpall(ref double[,] xyz, int n, int nx, int d, int nq, int nw,
                       double xmin, double ymin, double xmax, double ymax, double
                       xres, double yres, ref double[,] interpdata)
```

- This function calculates the RIFE map and uses multiple input parameters:
  - Size of the interpolation area i.e. RIFE map (x, y coordinates – the area is always rectangle shaped)
    - double `xmin`
    - double `ymin`
    - double `xmax`
    - double `ymax`
  - RIFE resolution (number of interpolated points)
    - double xres
    - double yres
  - Input parameters for the fusion algorithm (these parameters reflect the characteristics of the measurement area)
    - ref double[,] xyz,
    - int n,
    - int nx,
    - int d,
    - int nq,
    - int nw

```
public double iwinterppoint(ref double[,] xyz, int n, int nx, int d, int nq, int nw,
                            ref double[] x)
```

- This function calculates the RIF output at a single interpolation point with coordinates provided in xyz.

Statistical–analyses toolbox. This toolbox processes statistical information such as propagation model estimation, empirical probability models (empirical probability density functions - epdf and cumulative density functions - ecdf), historical spectrum occupancy (i.e. duty cycles) etc. The toolbox comprises multiple processing functions:

```
public double[,] estimatePropagationModel(double seconds, UInt64 centerFrequency,
                                          UInt64 bandwidth, int method)
```

- This function is responsible for estimating the path loss model in the area of interest. It returns the estimation as a pathloss value in dependence of the distance based on the following input parameters:
  - double seconds (duration of the estimation)
  - UInt64 centerFrequency
  - UInt64 bandwidth (Channel bandwidth for the given estimation)
  - int method (1-direct distance based estimation, 2 - RIF based estimation)

```
public byte[] calculateAddStats(UInt32 mcdID, UInt64 startFreq, UInt64 endFreq, UInt16
                                statType, Single resSize, String startTime, String
                                endTIme)
```

- This function calculates the empirical probability models. The function returns either the epdf or the ecdf function depending on the input parameters:
  - UInt32 mcdID (from which MCD to perform the calculation)
  - UInt64 startFreq
  - UInt64 endFreq
  - UInt16 statType (epdf or ecdf)
  - Single resSzie (number of points calculated)
  - String startTime (start time of the statistics)
  - String endTime (end time of the statistics)

```
public byte[] calculateDutyCycle(UInt32 mcdID, UInt64 startFreq, UInt64 endFreq,
                                 UInt16 chunkSize, String startTime, String endTIme,
                                 Single threshold)
```

- This function calculates the duty cycle at a given point of interest. The output data of the function is the duty cycle in dependence of the time of interest specified in the input parameters:
  - UInt32 mcdID (from which MCD to perform the calculation),
  - UInt64 startFreq
  - UInt64 endFreq
  - UInt16 chunkSize (number of averaged samples)
  - String startTime (start time of the statistics)
  - String endTime (end time of the statistics)
  - Single Threshold (Threshold value for the duty cycle)

Transmitter localization toolbox. The transmitter localization toolbox is responsible for performing an ML estimation of a transmitter location [7] and estimation of its transmit power. The toolbox comprises one generic processing function:

```
public byte[] getTransmittersLocations(float ulPointX, float ulPointY, float drPointX,
                                       float drPointY, UInt16 xPoints, UInt16 yPoints,
                                       double startFreq, double endFreq, bool last,
                                       float seconds, String startTime, String
                                       endTIme)
```

- This function returns information about the location of the transmitter (in Cartesian coordinate system) as well as its estimated transmit power based on the following input parameters:
  - Size of the area used for transmitter localization:
    - float ulPointX (upper left corner on X-axis)
    - float ulPointY (upper left corner on Y-axis)
    - float drPointX (downward right corner on X-axis)
    - float drPointY (downward right corner on Y-axis)
  - Resolution of the area:
    - UInt16 xPoints
    - UInt16 yPoints
  - Frequency band of interest for the estimation
    - double startFreq
    - double endFreq
  - Number of estimations:
    - bool last (one or multiple location estimates),
    - float seconds (calculate multiple estimates on the given number of seconds),
  - Duration of the statistics
    - String startTime
    - String endTIme

RRM toolbox. The RRM toolbox provides the REM manager with the capability to perform joint channel and power allocation for multiple users that share the same spectrum availabilities. The algorithm used in this toolbox tends to minimize the aggregate system interference by distributing the closest transmitters on different unoccupied channels as well as maximizing the receiver SINR by utilizing information about the position of the transmitters and the estimates of the propagation models for the given unoccupied channels. This toolbox comprises two processing functions:

```
List<double> getFreeChannels(double startFreq, double bandwidth, int noChannels,
                             double Threshold, double dyCyThr, float major, String
                             timeHist, String timeNow)
```

- This function returns a list of the available channels based on the following input parameters:
  - Frequency band of interest for the estimation

- double startFreq
- double bandwidth

- Number of channels
  - int noChannels
- Thresholding parameters
  - double Threshold (dBm above the noise floor)
  - double dyCyThr (duty cycle threshold)
  - float major (majority voting, *K* out *N* rule)
- Duration of the statistics
  - String startTime
  - String endTIme

```
public double[,] freqPowerAllocation(double minSINR, List<double> getFreeChannels,
                                     bool useEstimatedPM, double [,] propModel, bool
                                     powerAllocation)
```

- This function calculates the channel and the power allocation to different devices. It utilizes the following input parameters for its calculations:
  - double minSINR (minimal required SINR at the receiver)
  - List<double> getFreeChannels (list of available channels)
  - bool useEstimatedPM (usage of propagation model estimation in the RRM process)
  - double [,] propModel (propagation model for the available channels)
  - bool powerAllocation (usage of power allocation in the RRM process)

2.1.5 REM SA – REM Manager Interface

This interface connects the REM Manager and the REM SA components of the REM backend. It is developed to perform standard ADO.NET database access and read/write operations based on the requirements of the REM Manager (and the REM User).

2.1.6 REM Manager – REM User Interface

The REM Manager – REM User Interface handles the communication between the respective components. The communication is performed using TCP/IP socket communication where the asynchronous socket server is realized in C# and is a part of the REM Manager. The REM Manager can serve multiple REM Users.

The REM backend implementation specifies several pairs of messages that can serve as a base for different REM facilitated cognitive network scenarios. These messages are divided into two classes, i.e.:

- generic REM User messages and
- Communication Device (CD) messages.

The generic REM User messages are used for the interactions related to the extraction of generic REM data, while the CD messages are focused on the interactions related to the RRM functionalities of the REM prototype. All messages are presented and briefly explained below.

Generic REM User messages

➢ The `ActiveMcdsReq` and `ActiveMcdsRsp` messages are used by the REM User and the REM Manager to query/return the information about the active MCDs and their settings. `McdInfo` is the structure containing the information about the MCD configuration.

```
public struct ActiveMcdsReq
{
    public UInt16 MsgType;
    public UInt16 mcdType;
}
```

```
public struct ActiveMcdsRsp
{
    public UInt16 MsgType;
    public UInt16 NoMcds;
    // McdInfo array follows
}
```

```
public struct McdInfo
{
    public UInt32 mcdID;
    public UInt16 mcdType;
    public UInt64 startFreq;
    public UInt64 endFreq;
    public UInt64 resBW;
    public UInt16 noPoints;
    public Single sweepTime;
    public Single positionX;
    public Single positionY;
    public Single positionZ;
}
```

➢ The `RemReq` and `RemRsp` messages are used to query/return the RIF field estimation by the REM Manager. The spatial RIF field for a specific frequency band (`startFreq`, `endFreq`) is returned as a matrix representing the interpolated values at the grid points in the region between the upper left location point and the lowest right location point. The `xPoints` and `yPoints` parameters set/return the x and y resolutions, respectively.

```
public struct RemReq
{
    public UInt16 MsgType;
    public Single ulPointX;
    public Single ulPointY;
    public Single drPointX;
    public Single drPointY;
    public UInt64 startFreq;
    public UInt64 endFreq;
    public UInt16 xPoints;
    public UInt16 yPoints;
}
```

```
public struct RemRsp
{
    public UInt16 MsgType;
    public Single ulPointX;
    public Single ulPointY;
    public Single drPointX;
    public Single drPointY;
    public UInt64 startFreq;
    public UInt64 endFreq;
    public UInt16 xPoints;
    public UInt16 yPoints;
    // RIF matrix of doubles follows
}
```

➢ `TransmitterReq` and `TransmitterRsp` are messages used to query/return the information about the active transmitters in the inspected area. `TransmitterRsp` returns the estimated transmitters in the range between `startFreq` and `endFreq`, where `TransmitterInfo` is the structure of the transmitter information (its location, power and frequency band of operation).

```
public struct TransmitterReq
{
    public UInt16 MsgType;
    public Single ulPointX;
    public Single ulPointY;
    public Single drPointX;
    public Single drPointY;
    public UInt64 startFreq;
    public UInt64 endFreq;
    public UInt16 xPoints;
    public UInt16 yPoints;
}
```

```
public struct TransmitterRsp
{
    public UInt16 MsgType;
    public UInt16 noTrans;
    //TransmitterInfo array follows
}
```

```
public struct TransmitterInfo
{
    public Single pointX;
    public Single pointY;
    public Single power;
    public UInt64 startFreq;
    public UInt64 endFreq;
}
```

- `DutyCycleReq` and `DutyCycleRsp` are the messages used to query/return duty cycle results from a specific MCD on a specified frequency band and time period using a specified threshold.

```
public struct DutyCycleReq
 {
     public UInt16 MsgType;
     public UInt32 mcdID;
     public UInt64 startFreq;
     public UInt64 endFreq;
     [MarshalAs(UnmanagedType.ByValTStr,
SizeConst = MCDRemoteIFDefs.TIMESIZE)]
     public String startTimeInstant;
     [MarshalAs(UnmanagedType.ByValTStr,
SizeConst = MCDRemoteIFDefs.TIMESIZE)]
     public String endTimeInstant;
     public UInt16 chunkSize;
     public Single threshold;
 }
```

```
public struct DutyCycleRsp
{
    public UInt16 MsgType;
    public UInt32 mcdID;
    public UInt64 startFreq;
    public UInt64 endFreq;
    [MarshalAs(UnmanagedType.ByValTStr,
SizeConst = MCDRemoteIFDefs.TIMESIZE)]
    public String startTimeInstant;
    [MarshalAs(UnmanagedType.ByValTStr,
SizeConst = MCDRemoteIFDefs.TIMESIZE)]
    public String endTimeInstant;
    public UInt16 noResults;
    //Results follow as array of Single
variables
}
```

- The `AvgPowerReq` and `AvgPowerRsp` messages are used to query/return the average received power per MCD in a specified frequency band. `AvgPowerInfo` specifies the structure of the results.

```
public struct AvgPowerReq
{
    public UInt16 MsgType;
    public UInt32 mcdID;
    public UInt16 mcdType;
    public UInt64 startFreq;
    public UInt64 endFreq;
}
```

```
public struct AvgPowerRsp
{
    public UInt16 MsgType;
    public UInt16 mcdType;
    public UInt64 startFreq;
    public UInt64 endFreq;
    public UInt16 noMcds;
    // AvgPowerInfo array
follows
}
```

```
public struct AvgPowerInfo
{
    public UInt32 mcdID;
    public Single result;
}
```

- The `OldRemReq` and `OldRemRsp` messages have the same meaning as `RemReq` and `RemRsp` messages, but they are used to query/return an estimated historical RIF field in the specified time period.

```
public struct OldRemReq
 {
     public UInt16 MsgType;
     [MarshalAs(UnmanagedType.ByValTStr,
SizeConst = MCDRemoteIFDefs.TIMESIZE)]
     public String timeInstant;
 }
```

```
public struct OldRemRsp
{
    public UInt16 MsgType;
    public Single ulPointX;
    public Single ulPointY;
    public Single drPointX;
    public Single drPointY;
    public UInt64 startFreq;
    public UInt64 endFreq;
    [MarshalAs(UnmanagedType.ByValTStr,
SizeConst = MCDRemoteIFDefs.TIMESIZE)]
    public String timeInstant;
    public UInt16 xPoints;
    public UInt16 yPoints;
    // RIF matrix of doubles follows
}
```

- `AddStatReq` and `AddStatRsp` are used to query/return additional statistics such as empirical PDF or CDF of the received power for specific MCD in specified frequency band and time periods.

```
public struct AddStatReq
 {
     public UInt16 MsgType;
     public UInt32 mcdID;
     public UInt16 statType;
     public UInt64 startFreq;
     public UInt64 endFreq;
```

```
public struct AddStatRsp
{
    public UInt16 MsgType;
    public UInt32 mcdID;
    public UInt64 startFreq;
    public UInt64 endFreq;
    public Single startXValue;
```

```
        [MarshalAs(UnmanagedType.ByValTStr,
SizeConst = MCDRemoteIFDefs.TIMESIZE)]
        public String startTimeInstant;
        [MarshalAs(UnmanagedType.ByValTStr,
SizeConst = MCDRemoteIFDefs.TIMESIZE)]
        public String endTimeInstant;
        public Single resSize;
    }
```

```
        public Single endXValue;
        [MarshalAs(UnmanagedType.ByValTStr,
SizeConst = MCDRemoteIFDefs.TIMESIZE)]
        public String startTimeInstant;
        [MarshalAs(UnmanagedType.ByValTStr,
SizeConst = MCDRemoteIFDefs.TIMESIZE)]
        public String endTimeInstant;
        public UInt16 noResults;
       //Results follow as array of variables
    }
```

- `PropagationModelReq` and `PropagationModelRsp` are used to query/return an estimated propagation model from the REM Manager.

```
    public struct PropagationModelReq
    {
        public UInt16 MsgType;
        public UInt64 centerFreq;
        public UInt64 bandwidth;
        public UInt16 estimationMethod;
        public bool estimateNew;
        public bool fit;
        public Single seconds;
        [MarshalAs(UnmanagedType.ByValTStr,
SizeConst = MCDRemoteIFDefs.TIMESIZE)]
        public String Timing;
    }
```

```
    public struct PropagationModelRsp
    {
        public UInt16 MsgType;
        public UInt64 centerFreq;
        public UInt64 bandwidth;
        public UInt16 estimationMethod;
        public UInt16 noDistances;
        public bool hist;
        // Propagation model follows as Dx2
matrix of doubles, D number of distances
    }
```

- The `FSDreq` and `FSDrsp` are used to query/return a measurement sweep from a specified observation point in a specified frequency band.

```
    public struct FSDreq
    {
        public UInt16 MsgType;
        public Single xCoord;
        public Single yCoord;
        public Single StartFreq;
        public Single EndFreq;
        public Uint64 NoPoints;
        public String TimeInstant;
    }
```

```
    public struct FSDrsp
    {
        public UInt16 MsgType;
        public UInt16 result;
        public UInt16 flag;
        public UInt64 FSDpoints;
        public String TimeInstant;
        //Results follow as array of doubles
    }
```

- The `TriggerReq`, `TriggerRsp`, `TriggerEvent` and `TriggerStop` are used to register, fire and stop the triggering of a specific event. An event can correspond to specific metric (RSS, SINR) surpassing a specified threshold at a given location.

```
    public struct TriggerReq
    {
        public UInt16 MsgType;
        public Single pointX;
        public Single pointY;
        public UInt16 metric;
        public UInt16 operator;
        public UInt64 threshold;
    }

    public struct TriggerRsp
    {
        public UInt16 MsgType;
        public UInt16 triggerID;
    }
```

```
    public struct TriggerEvent
    {
        public UInt16 MsgType;
        public UInt16 triggerID;
        public UInt16 status;
    }

    public struct TriggerStop
    {
        public UInt16 MsgType;
        public UInt16 triggerID;
    }
```

Communication Device related messages

- The `CommDeviceRegReq` and `CommDeviceRegRsp` messages are used to register a new CD in the REM Manager (REM SA). The registration message carries the device location, addresses and capabilities in terms of the supported transmit power, frequency range and bandwidth. The response message returns the assigned ID to the new/updated CD.

➢

```
    public struct CommDeviceRegReq
    {
        public UInt16 MsgType;
        public UInt16 deviceType;
        [MarshalAs(UnmanagedType.ByValTStr,
SizeConst = MCDRemoteIFDefs.MACADSIZE)]
        public String deviceMAC;
        [MarshalAs(UnmanagedType.ByValTStr,
SizeConst = MCDRemoteIFDefs.IPADSIZE)]
        public String deviceIP;
        public String currPosX;
        public String currPosY;
        public String currPosZ;
        public Single minPower;
        public Single maxPower;
        public Single noiseFloor;
        public UInt64 startFreq;
        public UInt64 endFreq;
        public UInt64 maxBW;
    }
```

```
public struct CommDeviceRegRsp
{
    public UInt16 MsgType;
    public UInt16 deviceID;
}
```

➢ The `CommDeviceConfRsp` is used to enforce a specific configuration (transmit power, frequency, bandwidth of operation) to an active CD.

```
    public struct CommDeviceConfRsp
    {
        public UInt16 MsgType;
        public UInt32 deviceID;
        public UInt32 receiverID;
        public Byte codeRate;
        public Byte applicationType;
        public UInt64 centerFreq;
        public UInt64 bandwidth;
        public Single power;
        [MarshalAs(UnmanagedType.ByValTStr,
SizeConst = MCDRemoteIFDefs.IPADSIZE)]
        public String destIP;
    }
```

➢ The `CommDeviceLocReq` and `CommDeviceLocRsp` are used to query/return the location of an active CD.

```
    public struct CommDeviceLocReq
    {
        public UInt16 MsgType;
        public UInt32 deviceID;
    }
```

```
public struct CommDeviceLocRsp
{
    public UInt16 MsgType;
    public UInt32 deviceID;
    public Single currPosX;
    public Single currPosY;
    public Single currPosZ;
}
```

➢ The `CommDeviceDeRegReq` is used by the CD to deregister from the REM Manager (REM SA).

```
    public struct CommDeviceDeRegReq
    {
        public UInt16 MsgType;
        public UInt32 deviceID;
    }
```

➢ The `ActiveCommDevicesReq` and `ActiveCommDevicesRsp` are used to query/return the active CDs registered in the REM SA. The device operating parameters and current status are returned in the structured format provided in `CommDevicesInfo`.

```
    public struct
ActiveCommDevicesReq
    {
        public UInt16 MsgType;
        public Byte status;
        public UInt64 startFreq;
        public UInt64 centerFreq;
        public UInt64 endFreq;
    }
```

```
    public struct
ActiveCommDevicesRsp
        {
            public UInt16 MsgType;
            public UInt16 NoDevices;
            //CommDevicesInfo array
follows
        }
```

```
    public struct CommDevicesInfo
    {
        public UInt32 deviceID;
        public UInt16 deviceType;
    [MarshalAs(UnmanagedType.ByValTStr,
    SizeConst =
    MCDRemoteIFDefs.MACADSIZE)]
        public String deviceMAC;
        public Byte status;
        public Single currPosX;
        public Single currPosY;
        public Single currPosZ;
        public UInt64 centerFreq;
        public UInt64 bandwidth;
        public Single power;
        public UInt32 peerID;
    }
```

- The `CommunicationReq` message is used to trigger communication establishment between two registered CDs. This message prompts the REM Manager to perform the RRM functions to allocate frequency, power and bandwidth to the querying pair.

```
   public struct CommunicationReq
    {
        public UInt16 MsgType;
        public UInt32 transmitterID;
        public UInt32 receiverID;
        public bool useREM;
        public bool usePowerAllocation;
        public bool useEstimatedPM;
        public UInt16 estimationMethod;
        [MarshalAs(UnmanagedType.ByValTStr,
SizeConst = MCDRemoteIFDefs.TIMESIZE)]
        public String timing;
    }
```

- `CommunicationStopReq` is used to break an ongoing communication between two active CDs.

```
    public struct CommunicationStopReq
    {
        public UInt16 MsgType;
        public UInt32 transmitterID;
        public UInt32 receiverID;
    }
```

- The `CommDeviceStatusRep` message is used by the communication device to report the current status (idle, transmitting, receiving) and operating parameters to the REM Manager (REM SA).

```
    public struct CommDeviceStatusRep
    {
        public UInt16 MsgType;
        public UInt32 deviceID;
        public UInt16 status;
        public UInt64 centerFreq;
        public UInt64 bandwidth;
        public Single power;
        public UInt32 peerID;
    }
```

After carefully scrutinizing the components and the corresponding interfaces within the REM backend, the following subsection elaborates a custom developed GUI that is used to visualize the capabilities of the REM backend technology.

2.2 Visualization

FARAMIR's GUI represents a standalone client application based on Windows Presentation Foundation (WPF) framework [8]. The main purpose of the GUI is to gather real-time information from the REM Manager and present it graphically in a user-friendly manner. The visual

representation of the REM Manager's inbuilt functionalities is also useful in testing and developing purposes.

WPF is a next-generation presentation system for building Windows client applications with improved graphical experience. It is suitable for both standalone and browser-based applications and represents a convenient solution for real-time presentation of spectrum activity. The WPF package is an integral part of the .NET Framework 4.0 [9] and can be used in combination with standard .NET elements on top of the Windows platform (Figure 2-5).

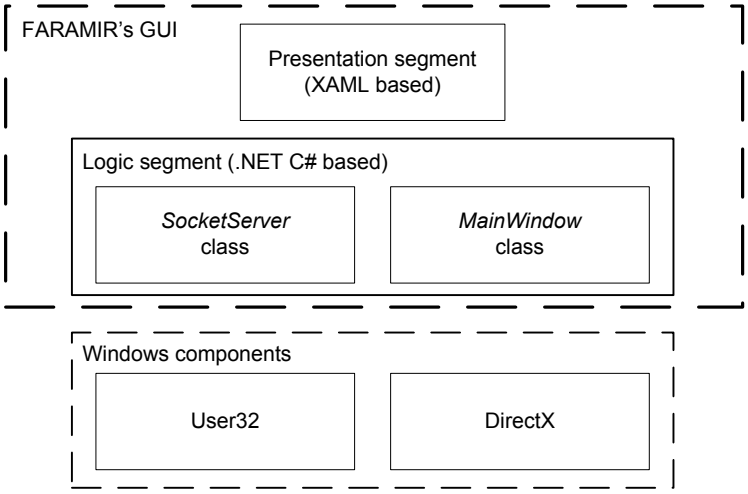


Figure 2-5: GUI architecture – built with WPF/.NET components on top of Windows components and DirectX.

One of the main advantages of WPF and the GUI is the ability to separate the *presentation* (design of the user interface) and *logic* (background functionality) segments of the application. The presentation segment is responsible for visual organization of the GUI and components' behavior in interaction with the user. It employs Extensible Application Markup Language (XAML) [10] - a declarative programming language based on XML [11]. XAML incorporates many graphical advantages of DirectX [11] and allows integration and behavior description of visual components such as 2D, 3D, animation and multimedia. The logic segment is responsible for the functionality of the visual components and management of data gathered by the REM Manager. Generally, it can employ any .NET based programming language. The GUI uses C# [13] as a high level, C-based, object oriented language. The logic segment is composed of two main classes:

- *SocketServer* (for connecting to the REM Manager and unpacking messages in the interface)and
- *MainWindow* (for control of the components' functionality and packing messages in the interface to REM Manager).

Figure 2-6 depicts the hierarchy of the GUI's logic segment.

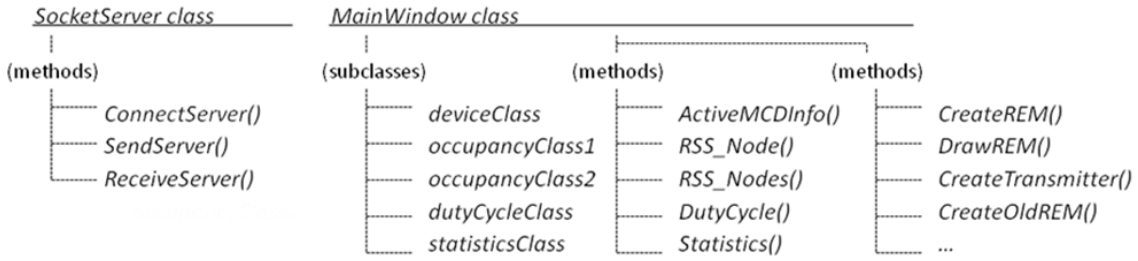


Figure 2-6: Hierarchy of the GUI's logic segment.

The GUI uses the *SocketServer* class for communication with the REM Manager. This class implements the following methods:

- *ConnectServer()* - upon startup, the GUI automatically creates an object of the SocketServer class. Then, when connecting to the REM Manager, it calls the method *ConnectServer()* including destination's IP address and port as parameters.
- *SendServer()* - the *SendServer()* method continuously checks the status of the connection and waits for a message from the *MainWindow* class. When a message is received from *MainWindow*, it resends it to the REM Manager. The following list includes the messages sent from the GUI to the REM Manager:
    - *ActiveMcdsReq* – request for a list of active MCDs and their parameters
    - *RemReq* – request for REM (sent with a repetition rate of 1s)
    - *TransmitterReq* – request for estimated position of a possible transmitter
    - *DutyCycleReq* – request for measured duty cycle by a selected MCD in a given time interval
    - *AvgPowReq* – request for the last measured received signal strength (RSS) value by a selected MCD (is sent with repetition rate of 1s)
    - *OldRemReq* – request for REM at a given time instant and
    - *AddStatReq* – request for additional statistics.
- *ReceiveServer()* - this method continuously checks the status of the connection for a receiving message from the REM Manager. Upon message reception, it unpacks the message and calls a corresponding method in the *MainWindow* class for further processing. The following list includes the messages sent from the REM Manager to the GUI:
    - *ActiveMcdsRsp* – contains a list of active MCDs (this message is always followed by *McdInfo* message)
    - *McdInfo* – contains the parameters of each MCD
    - *RemRsp* – contains a matrix with REM values
    - *TransmitterRsp* – includes a list of estimated transmitters (this message is always followed by *TransmitterInfo* message)
    - *TransmitterInfo* – holds information about each estimated transmitter

- *DutyCycleRsp* – contains calculated duty cycle for a selected MCD
- *AvgPowerRsp* – includes a list of measured RSS values by each MCD (this message is always followed by *AvgPowerInfo* message)
- *AvgPowerInfo* – holds measured data for each MCD
- *OldRemRsp* – returns a calculated matrix with REM values at a previous time instant
- *AddStatRsp* – includes additional statistics from the measurements.

An object of the *MainWindow* class is instantiated when starting the GUI and is responsible for the operation of the graphical components, the management of the received data and the packing of request messages for the REM Manager. The following subclasses manage the data received from the *SocketServer* class:

- *deviceClass* – objects of this class represent the set of MCDs and contain their measurement parameters. When the REM Manager sends *ActiveMcdsRsp* and *McdInfo* messages, the *SocketServer* class calls the method *ActiveMCDInfo()* to create a list of MCDs (*deviceClass* objects). The GUI provides MCDs information in a list or graphically in the REM screen (Figure 2-7 a, b, c);
- *occupancyClass1, occupancyClass2* – objects of these classes contain the RSS samples of a selected MCD and RSS samples of all MCDs, respectively. When the REM manager sends *AvgPowerRsp* and *AvgPowerInfo* messages, the *SocketServer* class calls the methods *RSS_Node()* and *RSS_Nodes()* to include the measurement data and present it graphically (Figure 2-7 b);
- *dutyCycleClass* – an object of this class holds the calculated duty cycle values for a selected MCD in a given time interval received with a *DutyCycleRsp* message from the REM Manager. The *DutyCycle()* method handles this type of data and presents it graphically on the GUI (Figure 2-7 c);
- *statisticsClass* – an object of this class contains the values of calculated statistics received with *AddStatRsp* message from the REM Manager. The method *Statistics()* is responsible for handling this type of data and its graphical presentation (Figure 2-7 d). Available statistics include Probability Distribution Function (PDF) and Cumulative Distribution Function (CDF) of measured RSS samples.

Figure 2-7: GUI's graphical data presentation a) REM screen with MCDs and estimated transmitter position b) Current RSS values for a single and multiple MCDs c) Calculated duty cycle for selected MCD d) Old REM and statistics graph

Additional methods in the *MainWindow* class are responsible for operation of the graphical components and packing request messages for the REM Manager. Some of the more important methods are:

- *CreateREM()* and *DrawREM()* – responsible for graphical presentation of the REM screen;
- *CreateTransmitter()* – responsible for graphical presentation of the estimated transmitter position;
- *CreateOldREM()* – responsible for presentation of previous REM screen etc.

After elaborating on the visualization of the REM backend capabilities, the following subsection discusses the accuracy and the reliability of the REM backend in terms of different MCDs' sensing performances and different spatial interpolation techniques' precision.

2.3 Performance Assessment

This subsection elaborates on the REM accuracy and reliability as a quintessential aspect for the possible subsequent usage of the developed REM backend technology in various instantiations (Chapters 4 and 5).

2.3.1 MCD Sensing Performance

Table 2-1 [4] compares the hardware and sensing related parameters of the MCDs that are implemented in the REM backend (details on the SCALDIO performance are separately discussed in Chapter 3). According to the specifications, the USRP2 devices have higher sensing capabilities making them agile and reliable legacy sensing devices.

Table 2-1: Performance of the legacy MCD devices.

Performance metric	**USRP2**	**TI eZ430 RF2500**
Frequency bands	daughterboard dependent	2.4– 2.485 GHz
Resolution bandwidths	195 kHz – 25 MHz	58 kHz – 812.5 KHz
Frequency steps	various	25kHz – 405 kHz
Frequency switching delays	<200 μs	<809 μs
Sampling type	IQ	RSSI
Sampling period	>40 ns (decimation dependent)	<310 μs (bandwidth dependent)
Sensitivity	-164 dBm/Hz (for RFX2400)	-83dBm (812 kHz) -104dBm (203 kHz)
On-board processing capabilities	50 MHz 32bit RISC CPU	16 MHz 16bit RISC CPU
On-board memory	1 MB SRAM	1K RAM/32K ROM

Figure 2-8 depicts the Receiver Operating Characteristic (ROC) curves [14] for the USRP2 for different input signals. The curves are obtained for the RFX2400 daughterboard using a center frequency of 2.401 GHz, a resolution bandwidth of 2 MHz and a receiver gain of 40 dB. It is evident thatthis configuration allows reliable detection of input signals of -94 dBm and above. The performances can be improved with higher gain settings, but the increase of the gain reduces the dynamic range of the USRP2 device. This is not suitable for a REM prototype, since the performances of the RIF estimation and localization features require high dynamic range and are strongly affected by the device non-linearities.

Figure 2-9 presents the ROC curves [14] of the TI eZ430 RF2500 device for four different cases of input power. It is evident that the TI eZ430 RF2500 device can reliably detect signals with power down to-110 dBm. The results are obtained using a resolution bandwidth of 812.5 KHz with the device's registers set to give the highest possible sensitivity. The results correspond to the most sensitive settings i.e. samples averaging in the dBm domain (Figure 2-9 presents the case of 100 samples averaging). This proves to be the most efficient mode of operation of this device in terms of spectrum sensing.

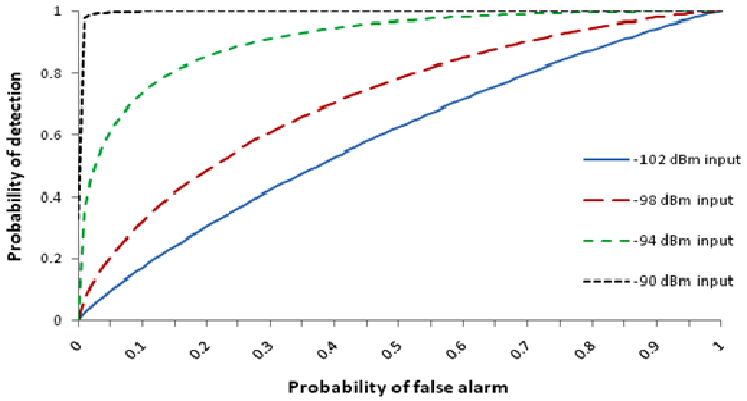


Figure 2-8: USRP2 ROC curves for input RF power of -102, -98, -94 and -90 dBm, resolution bandwidth of 2 MHz and 40 dB of receiving gain.

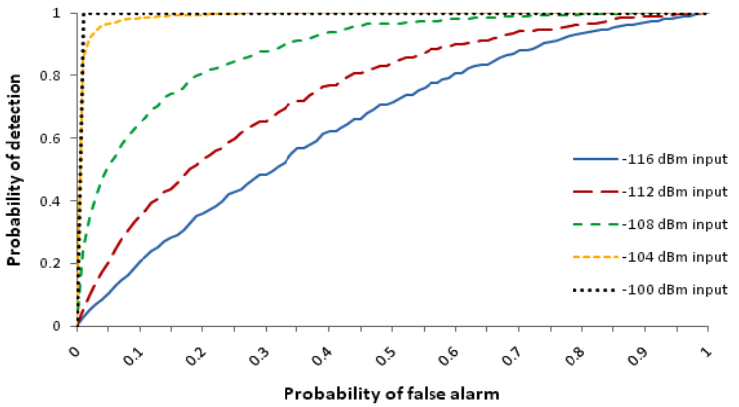


Figure 2-9: TI eZ430 RF2500 ROC curves for five different input power levels and sample averaging of 100 RSSI values in dBm domain.

The following subsection provides detailed insight into the reliability of the constructed REM.

2.3.2 Reliability of the IDW based RIF estimation

This subsection focuses on performance assessment of the Inverse Distance Weighting (IDW) based spatial interpolation, which is tightly related to the RIF estimation feature of the REM Manager. The IDW methods are simple and robust spatial interpolation methods widely used in various research areas due to their ability to work with different sensor geometries and limited number of observation points. The focus of the practical analyses here targets four IDW based interpolation methods:

- the classical IDW interpolation [5] using all available spatial RSS observations to perform the interpolation (referred as IDW method);
- a modified version of the classical IDW interpolation [5] accounting for the direction, the number and the set of considered neighboring RSS observation points and the slope of the interpolation function (referred as IDWM method).
- an enhanced version of the modified Shepard's method [15] using an inverse covariance matrix of the input values (x, y) for the weighted estimation of the nodal functions parameters (referred as MSMC) and

- additional enhancement of the MSMC method [15] using adaptive N_q - number of used nearest observations for nodal function fitting and N_w - number of used nearest observations for interpolation (referred as AMSMC).

For the purpose of practical assessment of the IDW based RIF estimation, a testbed comprising ten TI eZ430 RF2500 devices and three USRP2 devices was set up in an indoor environment (Figure 2-10). The devices were placed in a 25m^{2}classroom area having a large number of chairs and tables as obstacles and shadowers. The TI sensors were used as MCDs, while the USRP2 devices were employed as signal sources generating 5 MHz wide OFDM signals. The experiments focused on the 2.4 GHz ISM band evaluation. A total of 24 different scenarios (combinations) were tested, having each of the signal sources active as a single transmitter or in a pair of transmitters, with three possible transmit power levels, i.e. -5, -15 and -25dBm. Each inspected scenario was evaluated for 5 minutes to gain sufficient time domain statistics. It is important to note that the radio environment during the measurements was not completely controlled and some of the errors may originate from the outer interference.

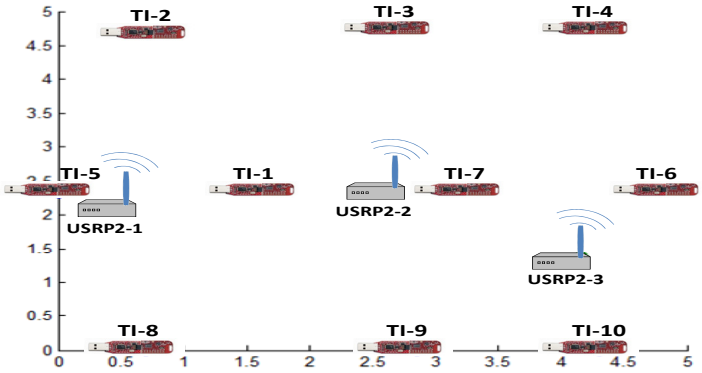


Figure 2-10: Evaluation scenario

The evaluation on each of the aforementioned methods focused on the error, the Absolute Error (AE) and the Mean Absolute Error (MAE) per sensor. The calculations of the errors were made with a sensors exclusion approach averaging over all combinations starting from nine (out of ten) active sensors down to only five active ones. The errors were calculated as residuals of the interpolated values and the values measured by the excluded sensors. Only the corner sensors with IDs 2, 4, 8 and 10 were not excluded to alleviate the border effects. The obtained results are cumulative for all 24 inspected scenarios. They are presented as box plots, where the central mark is the median, the edges of the box are the 25th and 75th percentiles and the whiskers extend to the most extreme data points not considered as "outliers". The used whisker length is 1.5 times the interquartile range.

Figure 2-11 presents the comparison between the different IDW methods in terms of the MAE per sensor for the case of nine active sensors. In addition, the figure presents the impact of the distance exponent d_{exp} in the interpolation MAE. The results prove that the IDW methods generally perform better when the distance exponent d_{exp} is equal to 1. The AMSMC method with d_{exp}=1 is the most reliable interpolation overall, because it offers the lowest and the less variable MAE. It is important to note that the MSM based methods do not perform well when the used d_{exp} is equal to 2. The IDW and the IDWM methods have similar behavior, the latter experiencing slightly lower variance of MAE, but having more "outliers".

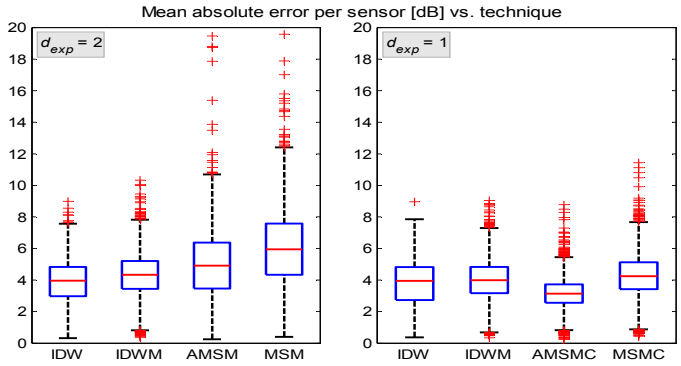


Figure 2-11: Comparison of IDW based methods in terms of MAE per sensor, impact of distance exponent for interpolation with 9 active sensors.

Figure 2-12 depicts the box plots presenting the MAE per sensor performances of the IDW interpolation methods of interest with respect to the number of active sensors in the interpolation. As verified by the testbed results, the AMSMC approach (with d_{exp}=1) provides the best MAE performances for each case with the standard deviation of MAE per sensor ranging from 1.04 dB to 1.56 dB for nine and five sensors, respectively. However, this interpolation is mostly affected by the decrease of the number of sensors. This is due to the fact that the optimization (adaptation) space of the N_q and N_w parameters is reduced with the decrease of the number of active sensors. While the classic IDW approaches are more robust to "outliers", the MSM approaches are not.

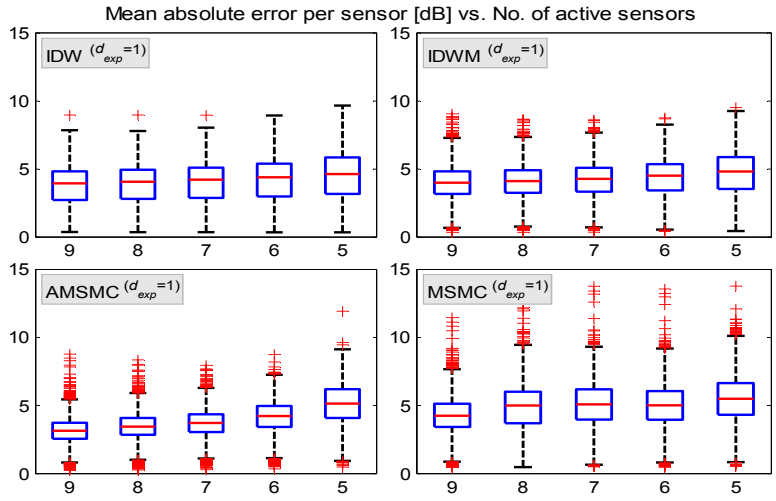


Figure 2-12: Impact of the number of active sensors to the MAE per sensor for the different IDW based methods.

Figure 2-13 presents the dependence of the AE on the location of the interpolation point. Namely, the AEs are evaluated at the locations of the excluded sensor for the case of nine active sensors. The AMSMC interpolation approach again provides the best results for most of the interpolation points (excluded sensors positions). The results show that some of the locations suffer higher interpolation errors, i.e. locations of excluded sensors with IDs 5 and 7 are mostly affected. The standard deviation of the AE at these sensors positions for the AMSMC approach is 3.74 dB and 3.44 dB, while for the classic IDW the respective values are 5.03 dB and 4.53 dB. The error has a negative bias at these locations for all tested IDW methods. This is logical, considering Figure 2-10and the positions of the signal sources – the sensors with IDs 5 and 7 are the closest ones

receiving the highest signal power. When these points are excluded and the dynamic range of the field is high (high power transmissions), the interpolation surface "down-fits" the extremes.

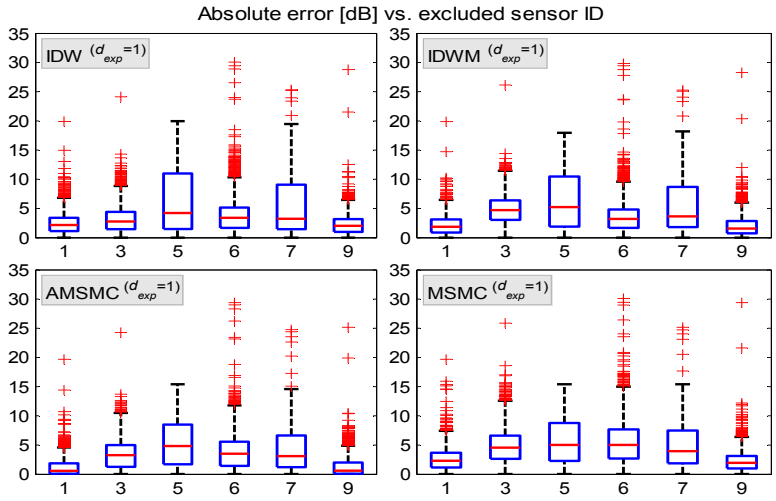


Figure 2-13: AEs at the excluded sensor ID locations for 9 active sensors.

Presented results give an insight in the reliability bounds of the IDW based methods to the problem of spatial interpolation of RIFs with a low number of observations. Out of all inspected methods, the AMSMC interpolation with d_{exp} equal to 1 has proven to offer the lowest interpolation errors. However, the main concern of the IDW based spatial interpolation is the erroneous RIF estimation in nearby transmitter areas. A reasonable approach in these cases would be to use the measurements to perform an initial transmitter location and power estimation. This information can be fed to the IDW based interpolation to reliably synthesize the RIF.

2.3.3 Reliability of RSS based Transmitter Localization

The RSS based localization is a key feature within the REM backend due to the lowest complexity of implementation and requirements in terms of the MCD capabilities. This subsection presents the reliability of several non-Bayesian LS and ML based RSS localization methods for the case of indoor environments. The results aim to provide the reliability (performance) bounds for the case of limited number of RSS spatial observations.

2.3.4 Single Source Case

The RSS based localization methods in the focus of investigation are:

- A full search ML approach [16][17] assuming log-normal shadowing, requiring a priori knowledge on the simplified propagation model parameters and performing a full search of the transmitter location using a global optimization algorithm;
- A full search LS approach [16][18] based on the a priori knowledge on the simplified propagation model parameters, performing a linear LS estimation of the transmitter location;
- An ML grid search approach [7][16] assuming log-normal shadowing, simultaneously estimating the transmitter location and propagation model parameters performing a grid search and

- An LS grid search approach [16] simultaneously estimating the transmitter location and propagation model parameters performing a grid search of the transmitter location minimizing the squared error.

The performance evaluation comprises two setups with two different MCD compositions and geometries. The first setup (Figure 2-14 a)) uses ten TI eZ430 RF2500 devices placed in more regular manner in a 25m^2 area in a classroom with many chairs and tables. A USRP2 device was used as a transmitter generating 5 MHz wide OFDM signals with a 0dBm transmit power. The measurements were conveyed for three possible positions of the transmitter in the 2.4 GHz ISM band. The scenario with transmitter Tx1 active is subsequently referred as*Sk1*, the one with Tx2 active is referred as *Sk2*, while the scenario with Tx3 active as *Sk3*. The second setup (Figure 2-14 b)) is heterogeneous employing four types of MCDs, sixteen in total, i.e.: seven TI eZ430 RF2500, seven USRP2 devices, one SCALDIO device and one R&S FSL6 spectrum analyzer. The MCDs were randomly placed in a 49m^2 area in the corner of a large hall having a lot of movement within. Measurements were made for a single position of a USRP2 transmitter, generating a narrowband 2 MHz GMSK modulated signal in the 2.4 GHz ISM band with 0dBm transmit power. This scenario is subsequently referred as *DySPAN*.

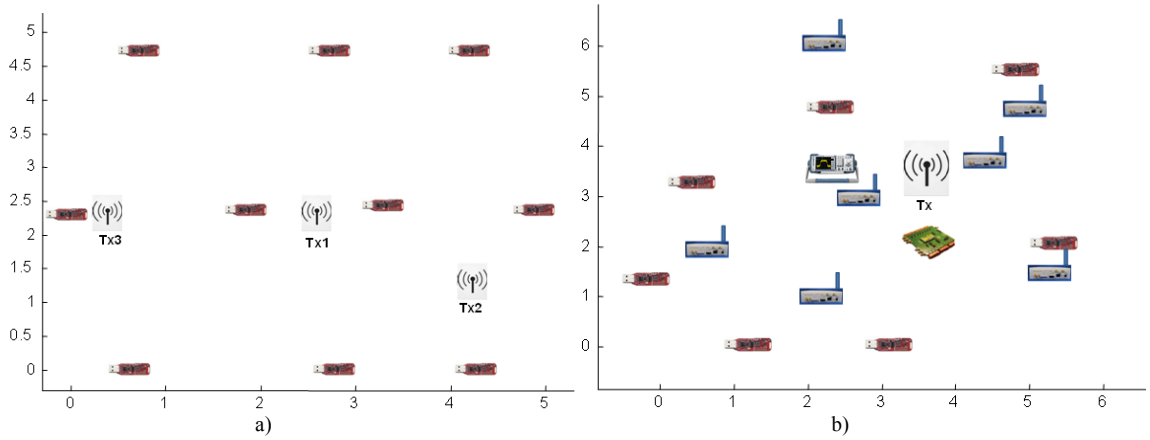


Figure 2-14: a) Homogeneous sensor setup in a 25 m^2 classroom; b) Heterogeneous sensor setup in a 49m^2 area in a corner of a hall.

The MCDs used classical energy detection in all scenarios. Every sample corresponds to 2s average of the received power. Each measurement scenario lasted for 5 min resulting in 150 measured samples per scenario. The grid search approaches were evaluated for a regular search grid consisting of 900 (30 x 30) points.

Figure 2-15 presents the results obtained for the four inspected scenarios and the four evaluated localization techniques. The results are presented as box plots of the *Squared Error* (SE), presenting the median and the 25th and 75th percentiles and the "outliers". The used whisker length is 1.5 times the interquantile range. The dashed horizontal line indicates the value of 4MMSE (Minimum Mean Squared Error), derived from the Cramer-Rao Lower Bound (CRLB), representing the 95% confidence interval of an ideal localization technique (approaching the CRLB). Figure 2-15 proves that the localization techniques generally behave better when the transmitter location is near the center of the MCD geometry rather than near the edges.

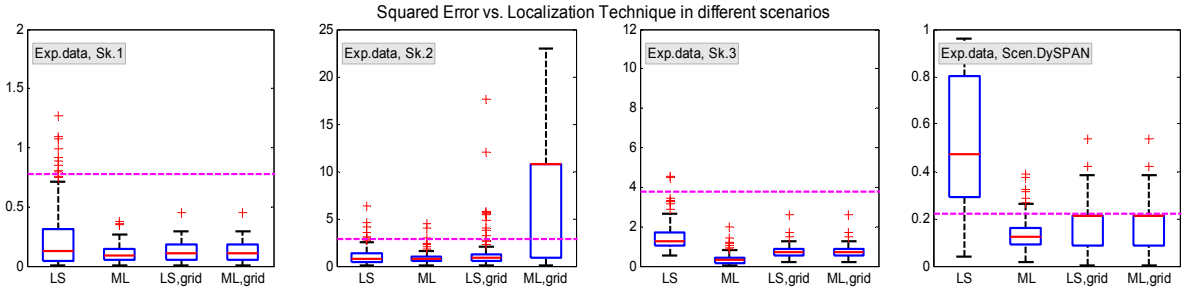


Figure 2-15: SE localization results for the different localization techniques and scenarios.

Comparing the different localization techniques, it is evident that the highest performances are experienced by the full search ML algorithm. The proposed LS-GRID algorithm provides the second best performances and stable behaviour in all scenarios. Considering that the ML full search algorithm requires the propagation model to be a-priori known and is computationally most demanding, the LS-GRID proves to be a reasonable solution for indoor and dynamic environments. The remaining two localization techniques, the LS full search and the ML grid approach experience unstable behavior depending on the scenario and the placement of the transmitter and the MCD.

Another interesting conclusion is that the *DySPAN* scenario (the only scenario with heterogeneous sensors) experiences the worst performances with respect to the theoretical limits (the CRLB). This is due to the heterogeneity of the used MCD equipment and the calibration offsets between the different types of devices causing more erroneous localizations. However, comparing the SE results of the corresponding homogeneous and heterogeneous scenarios (where the transmitter is in the centre, scenarios *Sk1* and *DySPAN*), there is still a performance increase in the latter due to the higher number of MCDs regardless of the heterogeneity of the used hardware.

Presented results provide an insight into the reliability bounds of the RSS based localization for a low number of observations. The full search ML algorithm shows superior performance in terms of SE in all scenarios. However, it requires powerful optimization algorithm, which usually is computationally exhaustive. Additionally, most of these algorithms are very susceptible to the underlying cost function, i.e. the log-likelihood that usually changes rapidly in dynamic environments. The LS-GRID can be a logical solution offering the second best performances, which can be partially attributed to the minimal amount of prior knowledge and assumptions made about the surrounding environment. The increase of the number of MCDs improves the performances with respect to the level of hardware heterogeneity.

2.3.5 Multiple Source Case

The scenario where multiple sources exist is a very challenging one and got special attention within FARAMIR. In the literature, a lot of work exists assuming the environmental noise to be additive, which is a valid model mainly for multiple acoustic sources [19][20].The target cost function for ML based estimation in this case is the sum of square error between the total mean power received by each sensor and the received measured power, a non-optimal approach under log-normal shadowing. The difficulty in describing the distribution of the received power in closed form leads to the adoption of this solution also to the log-normal case. In [20], a global optimization approach based on this cost function was proposed and a clustering solution was introduced based on the *k*-means algorithm [21] to enhance the convergence and reduce the overall complexity. Following the same modeling assumption, a quasi Expectation Maximization (EM) approach was proposed in [22]. All the above techniques assumed also knowledge of the

number of sources. This problem was first addressed in [23] where the inherent scarcity (small number of sources, large number of measurements) was exploited as prior information to improve estimation performance by suitably modifying the least-absolute shrinkage and selection operator (Lasso) in [24].

The first attempt of a true ML approach was introduced in FARAMIR [25] by the use of the approximation of the sum of log-normal variables introduced in [26]. The unknown number of sources was treated by the use of a large number of possible sources placed on a grid. The drawback of this approach was the increased complexity and some convergence issues when the number of grid points was high.

To avoid these problems, a direct approach was also followed on finding the multiple sources(search for *k*-sources without a grid [27]) by using the same approximation as in [25]. The number of sources is estimated by the use of appropriate modeling-order selection methods, such as the Akaike Information Criterion (AIC) or the Minimum Description Length (MDL) [28]. To avoid the complexity of directly maximizing the approximate Likelihood for a large number of sources, we adopted an approximate model for the description of the measured signal strength. The model is the so-called Gaussian Mixture Model (GMM), which is parameterized by the positions and the transmit powers of the unknown sources. The iterative EM algorithm is then invoked for estimating the parameter set, which defines the GMM. The complexity of the proposed algorithm grows only linearly to the number of sources, a major reduction in complexity, allowing the joint direct estimation of the number of sources, positions and power.

Performance analysis. Based on the approximation of the sum of log-normal variables proposed in [25], an approximate CRLB was derived. This is a novel result that allowed not only to assess the performance of the adopted algorithms, but also to characterize the sensor density needed to achieve the target estimation accuracy. In this subsection, we demonstrate, through simulations, the accuracy of the analysis, as well as the performance of the two methods for estimating the correct number of sources. The scenario used is the same as in Figure 2-14 (homogeneous setup), withUSRP2 *Tx*2 and *Tx*3 as active sources. Four different values of shadow fading variance are used, $\sigma^2 = (2,4,8,12)$, with path-loss exponent equal to 2 and *Tx* power equal to -10dBm for both sources. Figure 2-16 plots the MMSE performance of an full ML search assuming perfect knowledge of the number of sources, perfect initial conditions and known transmit power (in order to avoid convergence issues) and evaluates the validity of the derived bounds. The MSE performance for the position estimates and the respective CRLBs for both sources are plotted. The performance when only one of them is active is also used for comparative reasons since the analysis in this case is accurate (not an approximation). As seen in Figure 2-16 the analysis agrees with the simulation results in the case of *Tx*2. For *Tx*3, the performance is better than the respective bounds for both cases (single-multiple sources).

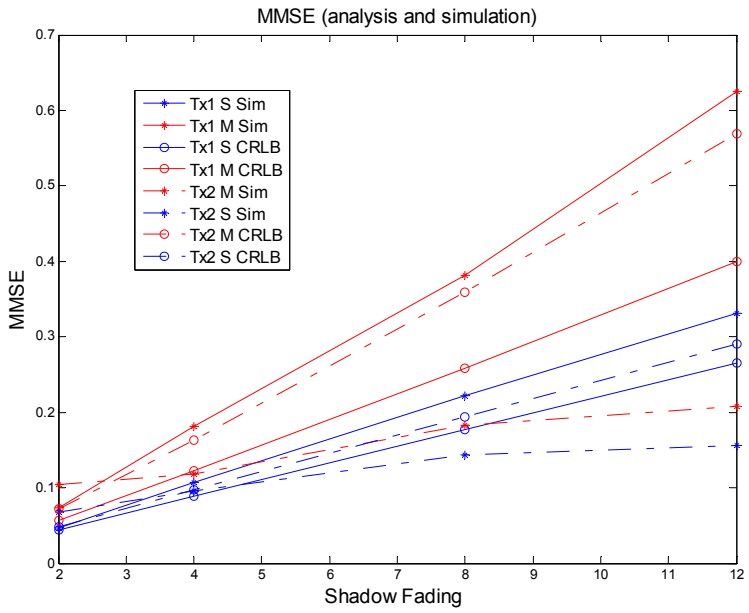


Figure 2-16: MMSE analysis and simulation results.

This behaviour can be explained by inspecting Figure 2-17. It depicts the location estimates for two cases of shadow fading (blue for $\sigma^2 = 4$ and black for $\sigma^2 = 12$). The circles around the sources indicate the 2RMMSE region based on the CRLB derivation. As we can see, the sensor near *Tx*3 works as an attractor (black hole) for the location estimates (convergence issues within this region). This has to do with the modelling assumption since the model is not valid within the region of d_0 around the source. Therefore, as long as no sensors exist within (or close by) the 2RMMSE region, also called here the convergence region, the analysis will agree with the simulations.

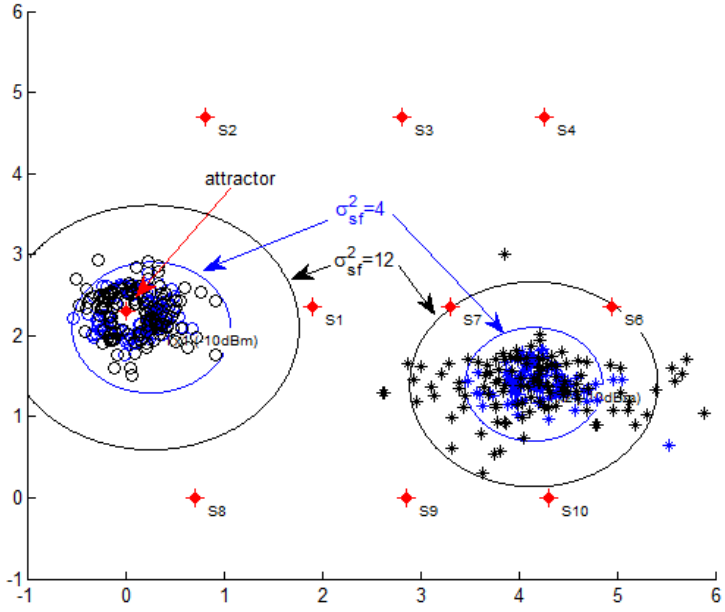


Figure 2-17: Estimation results and convergence regions.

For the same scenario, we use the EM-GMM algorithm, without the assumption of perfect *Tx* power knowledge and test the performance for estimating the correct number of sources. Perfect initial positions for the two active sources are used, since we want also here to focus only on the criterion capability to estimate the number of sources. The performance results are depicted in Table 2-2.

Table 2-2: Successful estimate percentage of the number of sources.

Method	Shadow Fading Variance			
	2	**4**	**8**	**12**
AIC	87%	98%	86%	80%
MDL	91%	98%	90%	84%

The results are indicative for the successful use of the two methods, since nearly 90% of the time the number of sources was estimated correctly.

Test-bed performance evaluation results. We conducted a practical assessment for the proposed algorithms using two scenarios, the parameterizations of which are given in Table 2-3 along with the respective CRLBs. The notation S1, S2 and PS1, PS2 is used for the two active sources and their respective transmit power in each scenario. Sx can be any of the three transmitters, *Tx*1,*Tx*2 and *Tx*3, while PSx is measured in dBm.

Table 2-3:Scenario parameterization.

ID	(PS1) CRLB	(PS2) CRLB	(S 1) CRLB	(S 2)CRLB
Scenario 1	(-10) 10	(-10) 6.9	(Tx3)3.6	(Tx2)1.6
Scenario 2	(0) 21	(0) 14.1	(Tx3)8.9	(Tx1)2.5

In this raw of experiments, we used three algorithmic choices:

- EMGMM, which is the proposed GMM based algorithm,
- ML, which is a full search of the maximum likelihood solution and
- EMGMM-ML approach, which is a full search initialized first by the proposed algorithm, all of them described in [27].

The number of sources is assumed to be known and initial positions are random within the coverage area. The initial power is also -10dB lower than the true one. The estimated propagation model has a path-loss of a=1.4 and shadow fading of σ^2=17.5 dB. We note here that at these experiments the shadow fading is not changing rapidly between successive estimates (as at the simulated experiments), since the environment is nearly static. In the extreme case of a fully static environment, the successive estimates would give approximately the same result. So, the CRLB is useful only to have an idea of the area where the cluster of successive estimates will fall. Table 2-4gives the mean of the estimated *Tx* power, the MSE of the localization error and the convergence percentage for the various choices of scenario, power estimation accumulation time and algorithmic choice. The convergence percentage is a measure of the number of estimates that fall within the convergence area (only those estimates are used for measuring the performance).

Table 2-4: Performance based on field trials.

Scenario ID, (accumulation time ms), Algorithm	Estimation Accuracy (MSE/MEAN)				
	Ptx1	**Ptx2**	**S1**	**S2**	***C %***
1,(2ms), EMGMM	-4.1	-6.3	1.8	1	98
1, (2ms) ML	-10.3	-11.9	0.5	2.5	54
1, (2ms) EMGMM-ML	-7.8	-11.2	0.6	3.7	79
1, (10ms) EMGMM	-3.8	-5.5	1.7	0.8	100
1, (10ms) ML	-11.2	-10.5	0.4	2	63
1, (10ms) EMGMM-ML	-7.1	-11.4	0.9	3.9	90
2, (2ms) EMGMM	2.8	2.1	0.3	0.7	100
2, (2ms) ML	-2.6	-4	0.5	0.8	46
2, (2ms) EMGMM-ML	-3.1	-2.7	0.5	0.8	78
2, (10ms) EMGMM	3	2.4	0.2	0.5	100
2, (10ms) ML	-2.1	-1.7	0.4	0.4	53
2, (10ms) EMGMM-ML	-1.9	-0.8	0.5	0.5	90

It is clear from Table 2-4 that:

- EMGMM exhibits high convergence rate and competitive performance in localization accuracy as compared with the highly complex ML solution;
- EMGMM tends to estimate a higher *Tx* Power (biased), which is expected due to the model approximation, c) whereas ML exhibits converge issues, but has no biased issues and finally
- EMGMM-ML shares both algorithms pros (nearly unbiased power estimates, good convergence).

Proper use of the two algorithms (EMGMM-ML) provides solutions that trade performance with complexity. In all cases, it must be noted that standard techniques for non-linear least squares minimization are the necessary ingredients for all proposed algorithms.

2.4 Summary

This chapter elaborated in details the developed FARAMIR REM backend along with visualization and performance assessment aspects. Further deployment results on the accuracy of derived radio environment maps in heterogeneous deployment scenarios are provided in the appendix. The backend technology will be used in subsequent instantiations (i.e. specific usage scenarios) providing radio environmental aware context to facilitate the radio resource management (Chapters 4 and 5). The developed REM backend is highly flexible due to its modular architecture (can be easily updated and upgraded) and the performance assessment clearly shows that it is a practically feasible approach that can provide sufficient accuracy and reliability for usage in operational networks.

3 FARAMIR Spectrum Sensing Engine

This chapter elaborates a specifically developed hardware within the FARAMIR project that acts as an MCD in the REM backend. The novelty is in the ability for fast, accurate and reliable spectrum sensing of a wide range of frequencies in a small-form factor design.

3.1 Reconfigurable Sensing Engine

One of the main goals of the FARAMIR project was to conceive a versatile spectrum sensing engine capable of covering a broad range of requirements with power/area/cost figure of merits compatible with the requirements of mobile devices. The developed sensing engine builds on a flexible analog front-end and a digital front-end for sensing, as depicted in Figure 3-1.

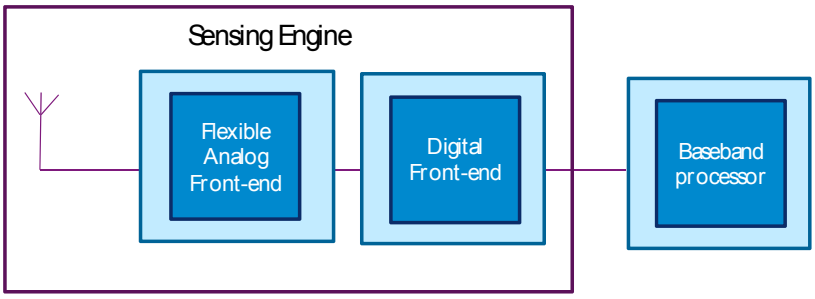


Figure 3-1: The structure of the integrated sensing engine.

The reconfigurable Analog Front-End (AFE) allows scanning the spectrum in a broad range of frequency bands. Since the basic task of a spectrum sensing engine is to scan all frequency bands for the presence of signals with several possible bandwidths, the functionality of the AFE actually requires the use of a Software-Defined Radio (SDR).

The programmable Digital Front-End (DFE) features the processing capabilities needed to analyze the signals in the scanned band for the envisioned purposes in the FARAMIR project, such as:

- Being the interface between the AFE and the baseband (front-end data and control interfaces);
- Performing the signal acquisition and coarse time synchronization for the targeted standards;
- Multiple channelization branches for simultaneous sensing and reception of different frequencies and
- Support for multi-band energy detection and feature-based sensing relying on autocorrelation.

The roadmap for the design, development and validation of the FARAMIR's spectrum sensing engine, depicted in Figure 3-2, is organized in three main phases:

- Phase 1: Standalone AFE prototype
- Phase 2: Standalone DFE prototype
- Phase 3: Integrated sensing engine prototype

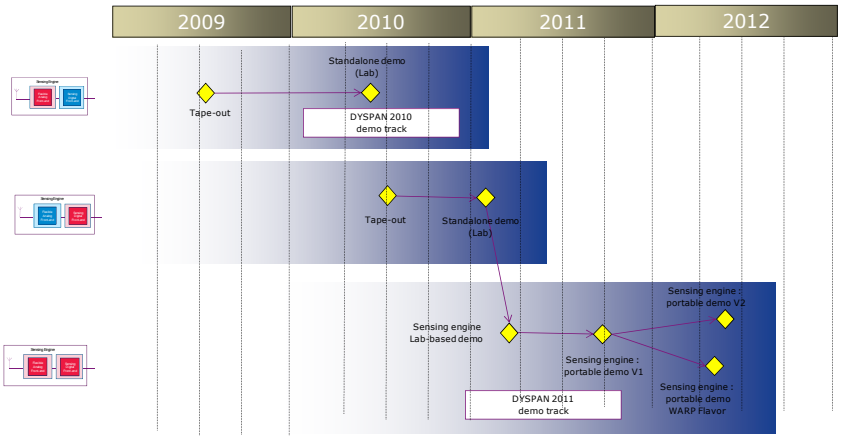


Figure 3-2: Integrated sensing engine: Prototyping roadmap.

There are two integrated spectrum sensing engines, illustrated in Figures 3-3 and 3-4, as an outcome of this roadmap. The first one includes an IMEC developed reconfigurable AFE offering maximum capabilities, whereas the second one includes an off-the-shelf WARP AFE offering maximum portability. These engines will be described in the next sections in terms of hardware and software. The interested reader is also referred to the many associated publications for further information [29]-[33].

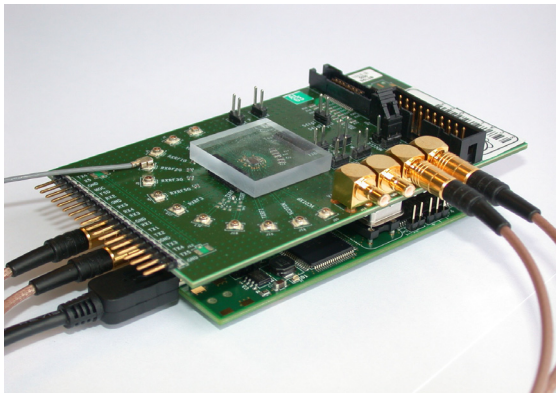


Figure 3-3: Integrated sensing engine: SPIDERv2 + SCALDIO-2b setup.

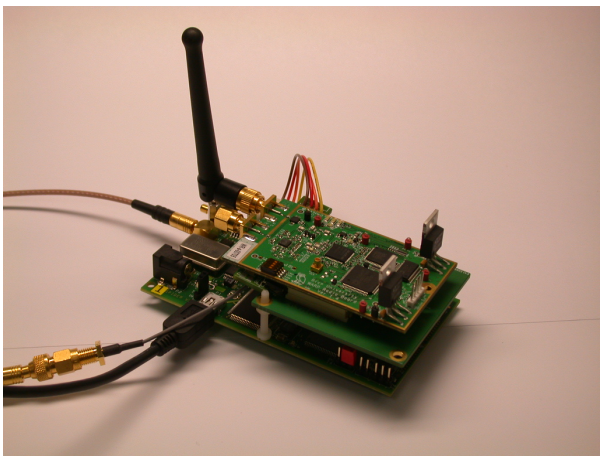


Figure 3-4: Integrated sensing engine: SPIDERv2 + WARP.

3.2 Sensing Engine Hardware

The portable spectrum sensing engine platform consists of the combination of an AFE board and a DFE board. There are two AFE solutions available, i.e. IMEC's Scaldio 2b board and Rice University's WARP board, and only one DFE, i.e. IMEC's Spider board (Figure 3-5).

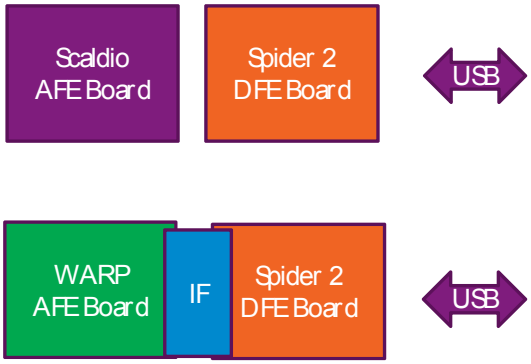


Figure 3-5: Schematic overview of hardware components.

Some functionalities within are enabled by the Spider board (the DIFFS-chip), whereas others are enabled (and thus depending on) the AFE connected to the Spider PCB. It is also possible to include functionality in the API or even in the application itself.

3.2.1 Spider PCB

The Spider PCB is a mixed signal design. It houses both a digital and an analog part. Its purpose is to provide an interface between the host PC and the rest of the platform, to provide functionality in the sensing engine and to provide power and clocking for the AFE. Two versions exist of the Spider PCB (revision 2 mainly fixes some bugs from revision 1), Figures 3-6 and 3-7.

3.2.1.1 Description

The Spider PCB can be fed in two different ways. The most straightforward ways is to use the USB power supply. This way the total power consumption of the platform is limited to 500 mA at 5 V, thus 2.5 W. If more power is required, a power jack can be plugged in. It should output 5 V DC.

The analog part of the PCB can be summarized by looking at the in- and outputs. There is only one input, namely a socket for an oscillator. The oscillator is fed with 4.8 V and its output frequency should be lower than 166 MHz. There are five outputs, i.e. three connectors for clocking and two for power. The connectors for clocking provide high-impedant clocks which are routed to two SMB-jacks and one MMCX-jack. The frequency is equal to the frequency of the oscillator and the output voltage-swing is adjustable between 2.5 V and 3.3 V. The power connectors provide a clean supply for the AFE. Additionally, there is a two-pin connector (one for ground and one for 5 V) and a four-pin connector (one for ground, one for 1.2 V, one for 1.4 V and one for 2.5 V).

The digital part can be divided in four parts as elaborated below.

First, it foresees the connection to the host PC through a USB-connection, the same that is providing the power supply as mentioned before. The chipset used is Cypress' EZ-USB FX2LP.

Second, a high speed 120-pin connector foresees an interface to connect the AFE to. One pin is reserved to indicate the output voltage of the connector. Two other pins are reserved to output a differential clock, which is a LVPECL-version of the output of the oscillator mentioned in the previous paragraph. All other 117 pins are connected to an FPGA.

Third, the PCB contains IMEC's DIFFS chip. The DIFFS chip is a DFE, the link between the AFE and the baseband platform. This baseband platform is not included in the sensing engine.

Fourth, there are some general purpose components, such as an FPGA, memory and some general purpose IO's. The FPGA is a Spartan 6 LX45. It connects all digital components on the PCB on the hardware level. Most IO voltages are fixed except for the pins connected to the AFE connector as mentioned before. Five of the 117 pins are connected to bank 0 and therefore fixed at 2.5 V. The other 112 pins are connected to bank 1 with a selectable IO voltage. Voltages of 2.5 V and 3.3 V are foreseen on the PCB and can be selected by means of a jumper switch. In principle, it is possible to connect other IO voltages as well, as long as it is supported by the FPGA. The middle pin of the jumper should then be connected to an external power supply. The design on the FPGA is depending on the AFE connected, since the connections are different for every FE, and every FE has its own specific functionality and interface. Furthermore, the FPGA code is different for the two different revisions of the Spider PCB (discussed in more details in the following subsection). Next to the FPGA, important for the sensing engine is the memory present on the Spider PCB. For revision 1, this is a 16 Mbit SRAM component, whereas for revision 2, this is a 64 MByte SDRAM component.

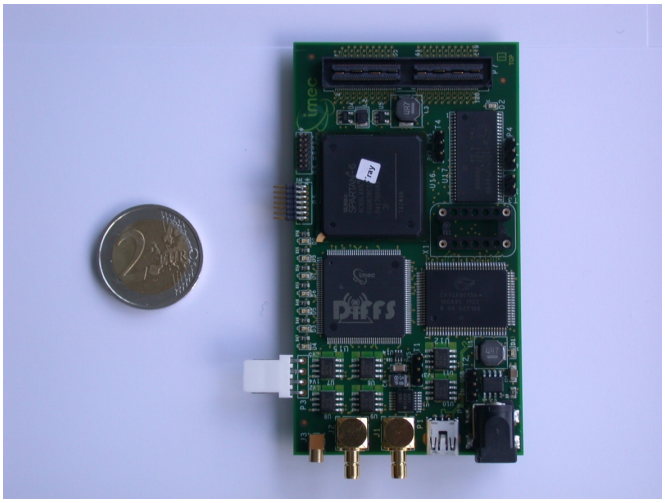


Figure 3-6: Spider (v1) PCB.

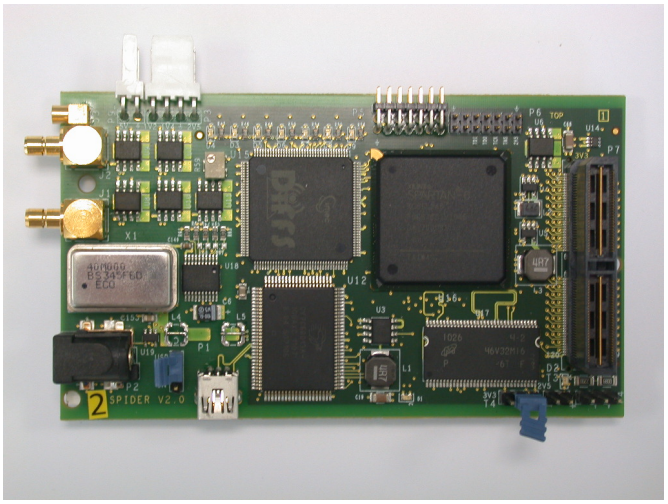


Figure 3-7: Spider v2 PCB.

3.2.1.2 FPGA Code

As mentioned in the previous subsection, there are a number of designs of the FPGA code. The design depends on the version of the Spider board, the AFE connected to the Spider board and (possibly) the desired functionality. This last dependency will not be covered in this document, since it can be very specific, e.g. it is possible to exclude the transmit functionality from a design or to change the size of the transmit- or receive-buffer.

The purpose of the FPGA is to connect all different components on the physical level. This means that for different components, different FPGA code will be needed (different pin-connection needed). Also, every component has its own specific interface running at different clock speeds, meaning that the FPGA design can differ for different components, Figure 3-8. Examples of interfaces are the DIFFS host interface, DIFFS control interface, DIFFS front end interface, SRAM interface, SDRAM interface, RX interface, TX interface, WARP interface etc. All interfaces are controlled by the Program interface and use a FIFO system to handle data streams. The translation between the different interfaces and the USB-chipset is done by the Spider Back End, or "spiderback".

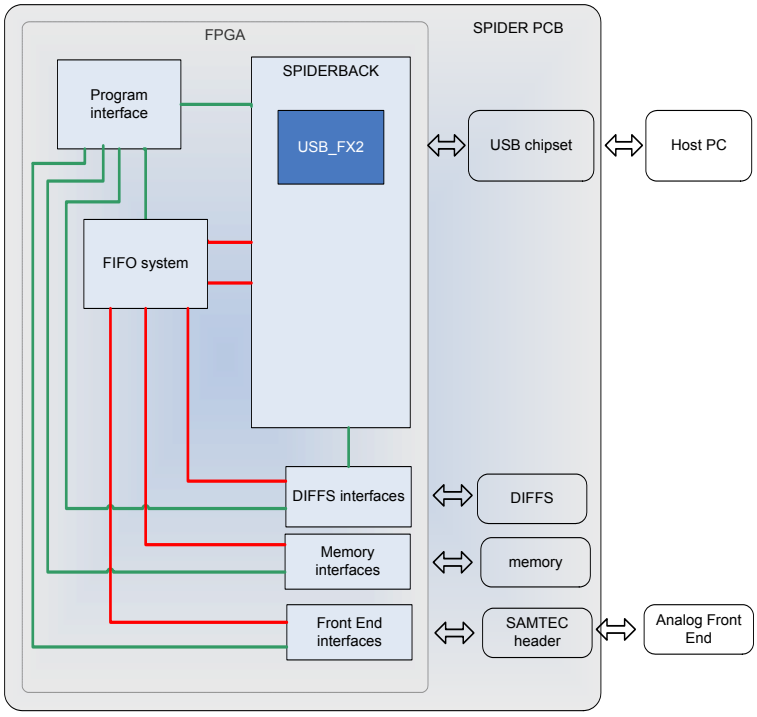


Figure 3-8: Simplified look of the FPGA design on the Spider board.

Every FPGA design requires a separate bitfile to be created. This bitfile can be uploaded to the FPGA via the USB chipset, using the ZTEX EZ-USB FX2 SDK.

3.2.2 Scaldio 2b PCB

This board (depicted in Figure 3-9) houses the Scaldio 2b chip. The Scaldio chip is in the center of the PCB under a protective acrylic glass shield. It is a flexible RF front-end: a single-chip reconfigurable receiver, transmitter and 2 frequency synthesizers in 40 nm-CMOS technology. The flexible receiver, including analog-to-digital converter, is fully software-configurable across all channels in the frequency bands between 100 MHz and 6 GHz. Its properties can be adapted to the requirements of the standards that are used.

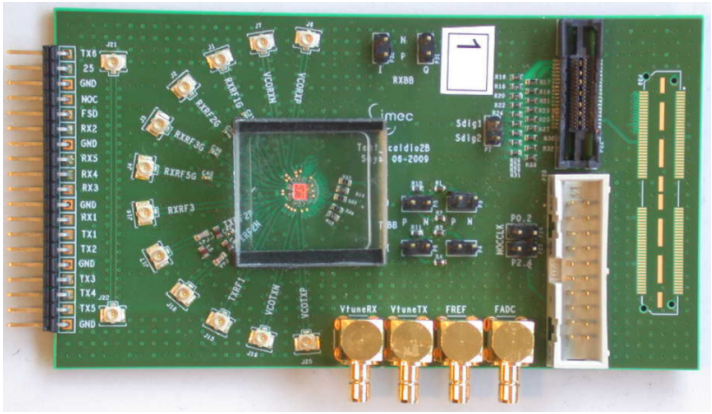


Figure 3-9: Picture of the SCALDIO 2b test PCB.

The usage of the receiver section of the chip is performed with the following available connections on the PCB:

- UFL connections to connect the antenna to the corresponding LNA;
- 120-pin QSH Samtec connector containing the digital output signals from the ADC, including a clock signal on which the data can be sampled (the data signal consists of 11 bits);
- 1 SMB jack to input the reference clock for the receiver PLL (FREF) and
- 1 SMB jack to input the ADC sampling clock (FADC).

Power is supplied to the board using the 19 pins header at the bottom of the PCB. The other connectors are not used for normal receiver operation of the chip. The Scaldio has a Network-On-Chip (NOC) for configuration. This interface consists of 5 signals that can be accessed through the Samtec connector.

3.2.3 WARP TRX PCB and WARP Radio Board

The third AFE solution is the WARP radio board designed by Rice University. The radio board is a daughtercard providing a single RF transceiver with a digital baseband interface, which is commercially available for everyone. The RF transceiver on the radio board is Maxim's MAX2829 designed for dual-band 802.11 a/g applications covering both the 2.4 GHz to 2.5 GHz band and the 4.9 GHz to 5.875 GHz band. The received analog baseband signal is sampled by an AD9248, while the RSSI output is sampled by an AD9200. The analog baseband signal for transmission is generated by an AD9777. The ADC and the DAC are designed by Analog Devices. The digital IQ and RSSI signals, as well as the digital control signals for the different components on the radio board, are available through two Hirose headers. Power is supplied to the WARP radio board through these connectors as well. The RF part, consisting of a Power Amplifier (PA), an antenna switch, bandpass filters (BPF) and baluns, connects to two SMA jacks to connect an antenna to. Three clocks have to be provided to the board: the RF reference clock is fed through an MMCX plug and must be 20 MHz or 40 MHz; the IQ sampling clock can be fed both differential and single-ended using a dedicated clock header or using the Hirose connectors and must be 40 MHz; the RSSI sampling clock is fed trough the Hirose connectors and should be 20 MHz or lower. A picture of the WARP radio board is shown in Figure 3-10.

Figure 3-10: Rice University WARP radio board.

An interface board is needed to connect the WARP radio board to the digital part of the platform. This WARP TRX PCB is designed especially for interfacing between the WARP radio board and the Spider v2 board. Two Hirose connectors are foreseen on the TRX board to connect the WARP radio board to and a SAMTEC connector is foreseen for the Spider board. The signals from the Hirose connectors are routed to the Samtec connector, some after level-shifting. A two pin header enables the connection of a 5 V power supply, which is routed to the supply pins on the Hirose connectors. A picture of a WARP TRX board connected to a Spider v2 board is shown in Figure 3-11. One can distinguish the red-black power supply cable coming from the Spider board providing the 5 V power supply for the WARP Radio Board.

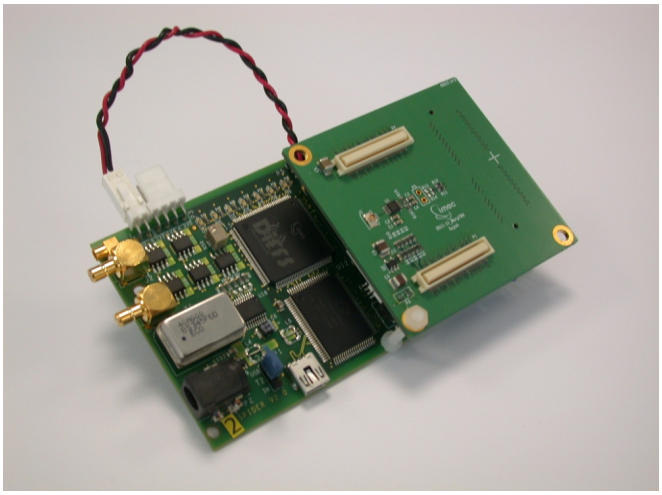


Figure 3-11: WARP TRX board connected to a Spider v2 board.

3.3 Sensing Engine API

The Sensing Engine Application Programming Interface (SE API), called API from here on, is a software library built to use the SE. The usage of the SE consists of selecting the desired mode of operation, configuring it with a set of predefined parameters and offloading results from the platform. The API therefore foresees a number of functions to allow the user to have access to all functionality of the SE while limiting the need for knowledge of the different components of the platform. Since there are several possible configurations of the platform, there are also several versions of the API available. There are a number of predefined, fixed versions of the API that can

be distributed among the users that want to use the SE, but it is also possible for the user to add specific functionality to the API. However, care should be taken when altering the API and no stable functioning of the SE is guaranteed. Either way, the API yields a number of files bundled in a package that can be used to create and build applications for the SE. This section elaborates the content of this package, the generation of the package and the structure of the API. Section 3.4 explains the usage of the package for creating applications.

3.3.1 The SE API Package

The API can come in two forms depending on how the application that will use the API will be built. If the application is written in C (or a similar programming language), the corresponding API will contain:

- A number of include files;
- A number of library files;
- Firmware for the USB chipset;
- A bitfile for the FPGA;
- Firmware for the DIFFS chip;
- Configuration files for each Scaldio;
- The FWLoader;
- A script to configure the platform and
- A script to build the executable for the application.

If the application is written in Matlab, then the corresponding API will contain:

- A Matlab executable file for Linux (.mexglx-file);
- A set of Matlab-functions for communication with the platform;
- A set of Matlab-functions for specific operations on the platform;
- Firmware for the USB chipset;
- A bitfile for the FPGA;
- Firmware for the DIFFS chip;
- Configuration files for each Scaldio;
- The FWLoader and
- A script to configure the platform.

Depending on the platform the API is intended for, the content of the package will (slightly) differ in terms of different bitfile, different USB chipset firmware, different libraries etc.

3.3.2 Structure of the SE API

The API consists of a number of elements or layers, Figure 3-12. The lowest layer foresees an interface between the platform and the remainder of the API. This means that it enables communication with the host PC running the application. There are three different interfaces developed in the API, i.e. the HAPS interface, the HOST interface and the USB interface. The HAPS

interface is intended for the lab-based setup with the HAPS FPGA board. For the same reason as explained in section 3.2, this interface is never included in the API. The HOST interface is a functional model. Therefore, no platform has to be connected to the PC. The applications built with a package based on this interface will functionally behave exactly like the SE would. The intent is to allow the user to get familiar with the functionality of the SE. However, this interface offers no exact timing behavior of the SE. Also, this interface is outputting dummy data, but in the same form as the 'real' SE would. The USB interface will be used for platforms in which the Spider board is coupled to the PC. This interface provides basic functionality to the rest of the API in the form of a special designed data structure (struct) and 10 functions. The struct holds information on the state of the interface and which USB-device is coupled to it. Note that it is possible for multiple instances of the struct to be active at the same time, since it is possible that multiple Spider boards are connected to a single PC. Four of the functions are used for data transfer to and from the Spider board, respectively for reading and writing single or multiple values. The six other functions are used for controlling the interface, namely for opening, initializing, resetting, testing and closing an interface and getting its status. The same 10 functions are also available in the Host interface, as is the struct. This means no difference is perceived between these interfaces by the higher layers of the API. Next to the source-files for the interface, each directory contains the script needed for building the corresponding library. Additionally, the USB interface directory contains the firmware source files and the scripts to build the firmware.

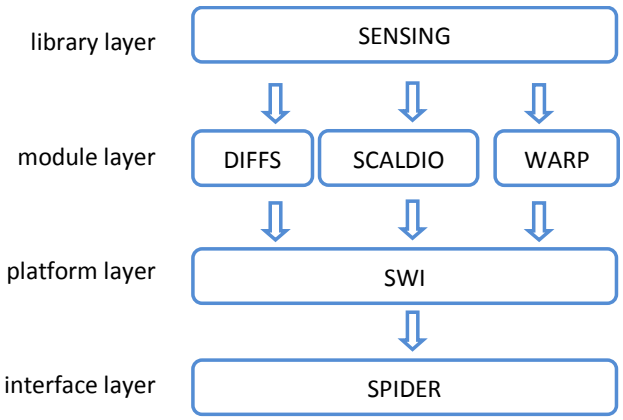


Figure 3-11: Overview of the SE API layers.

The second layer is the so called platform layer. Every platform uses the same functionality (provided by the interface layer), but can offer different functionality to the higher layers depending on the platform. Every different platform also has its own specific memory and register mapping, which is included in the platform by the use of include-files. Furthermore, the API can be built for every platform. Therefore, every directory contains the necessary scripts, which are in their turn invoking other build scripts if necessary. Examples of platforms are the host platform, the Spider Scaldio Integration (SSI) platform and the Spider Warp Integration (SWI) platform.

The third layer contains all functionality related to the different components of the platform or modules. The modules covered in the API are the DIFFS chip, the Scaldio FE and the WARP FE. For most users, these functions and everything related to it will not be visible. Moreover, the content of this layer covers all required functionality of the modules, so modifying it should be limited to extending functionality by adding new firmware for the DIFFS for example. However, care should be taken when adding new functionality as this may lead to instability or unwanted behavior and the API should be thoroughly tested before use. For every module, the available functionality is

more or less the same: there is a struct similar to the one of the interface layer, which keeps track of the state the module is in. These structs also hold another struct in which the active configuration of the module is saved. Furthermore, there are functions to open, initialize or close a module, to change the configuration, to activate it and to get results. An overview can be found in the respective include files.

The fourth and last layer contains the top level SE functionality. This layer is called the library as it foresees a library of functions that can be used in applications. For now, only one library is present, namely the sensing library. Similar to the content of the modules layer, the sensing library contains a struct, which is in fact an instance of the library and thus actually is a 'Sensing Engine'. This means each SE is a struct which keeps track of the state of the SE, the configuration of the SE and also holds the structs of the different modules that belong to the SE. This way, all information on the SE is bundled in one (instance of a) struct. Next to this struct, there are again a number of functions that in this particular case are available to the user. These functions can be used to open, initialize and close a SE, to configure it, to start or stop it and to get results from it. A prototype of the struct holding the configuration of the SE and of all available functions is available in the sensing library header file. This header file is included in the SE API package.

3.4 Applications

This section describes how the user can create applications using the SE API package. As previously explained, these applications will be built using the top level SE functionality provided by the library, in this case the sensing library. First, the section explains how the user performs general actions, such as opening or closing the SE. Then, the section explains where every mode of operation can be used for, how it has to be configured and how the result will look like. Addressed modes of operation are ADC-logging, FFT-sweeping and DVB-T cyclostationary detection. Snippets of C-code are included to clarify the actions mentioned.

3.4.1 General

There has to be at least one instance of the SE opened and initialized in order to use the SE. For every SE used in the application, a corresponding handler "se_t" has to be defined, which is a pointer to the struct corresponding to the SE. An instance of the configuration struct "se_config_s" will be needed for each SE in order to configure an SE. Declaration of two SE's is:

```
se_t my_se_1, my_se_2;
struct se_config_s my_se_config_1, my_se_config_2;
```

Opening and initialization is done as follows:

```
my_se_1 = se_open(number1, number2);
my_se_2 = se_open(number3, number4);
se_init(my_se_1, &my_se_config_1);
se_init(my_se_2, &my_se_config_2);
```

Number1 and number3 are the serial numbers of the Spider boards corresponding to the SE, to which an offset of 128 is added when using a Spider v2. For example, when number1 = 0x03, this stands for Spider v1 board number 3. When number1 = 0x87, this stands for Spider v2 board number 7. Number2 and number4 are the serial numbers of the AFE connected to the Spider board: 0 for a WARP board or the Scaldio serial number for a Scaldio. After initialization, the SE will be ready for use and the default configuration will be copied in the local configuration struct.

To configure the SE, the user can change the elements of the configuration struct and use the "se_configure" function. More details are given in the overview of the different modes of

operation. To avoid unwanted behavior when configuring the SE, the user can first check if the entered configuration is valid by using the "se_check_config" function:

```
my_se_config_1.mode = FFT_SWEEP;
my_se_config_1.first_channel = 1;
my_se_config_1.last_channel = 2;
if (se_check_config(my_se_1, my_se_config_1) == 1) {
      se_configure(my_se_1, my_se_config_1, binary1);
}
```

Binary1 is either 0 or 1, where 0 indicates a single measurement and 1 indicates continuous measurements.

The user can then start and stop the SE and fetch results from the SE as desired:

```
se_start_measurement(my_se_1);
se_get_result(my_se, result1);
se_start_measurement(my_se_1);
se_get_result(my_se, result2);
```

Result1 and result2 are pointers to a floating point value where the result will be stored. The user has to make sure enough memory space is allocated for the result depending on the configuration. For fetching the result every second for one minute using continuous measurements, following code would be used:

```
se_start_measurement(my_se_1);
int i=0;
while (i < 60) {
      se_get_result(my_se_1, result_i);
      sleep(1);
      i++;
}
se_stop_measurement(my_se_1);
```

After the user is finished using the SE, it should be closed to make it available for other users:

```
se_close(my_se_1);
se_close(my_se_2);
```

3.4.2 Modes of Operation

This subsection describes the different modes of operation and how they can be configured.

- ADC-logging of RF-data
    - The user can select the first and last channel to sweep over (between 1 and 901 for Scaldio and between 1 and 37 for WARP) and the FE gain setting. For Scaldio, BB bandwidth is fixed to 20 MHz, whereas for WARP it is configurable.
    - The output is up to 24575 complex I/Q time domain samples.
- FFT-sweeping of (part of) the spectrum (Figure 3-13)
    - The user can select the first and last channel to sweep over (between 1 and 291 for Scaldio and between 1 and 28 for WARP), the number of FFT-bins (16, 32, 64 or 128) per channel and the FE gain setting. The BB bandwidth is 20 MHz for Scaldio and 19.8 MHz for WARP.
    - The output is one power value per FFT-bin per channel.

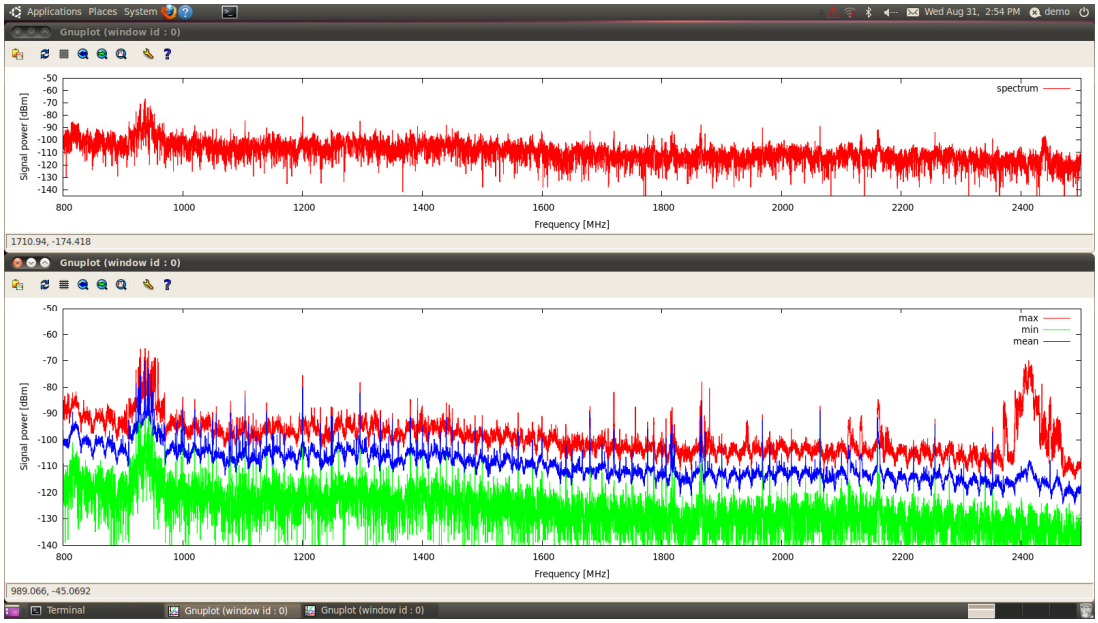


Figure 3-13: Wide-range frequency scan.

- DVB-T cyclostationary detection (Figure 3-14)
  - The user can select the first and last channel to sweep over (between 16 and 66), the number of carriers (2048 or 8192), the cyclic prefix length (guard interval 1/4, 1/8, 1/16 or 1/32) and the FE gain setting. The BB bandwidth is automatically selected according to the channel.
  - The output is a qualifier indicating the autocorrelation property of the received signal in each channel.

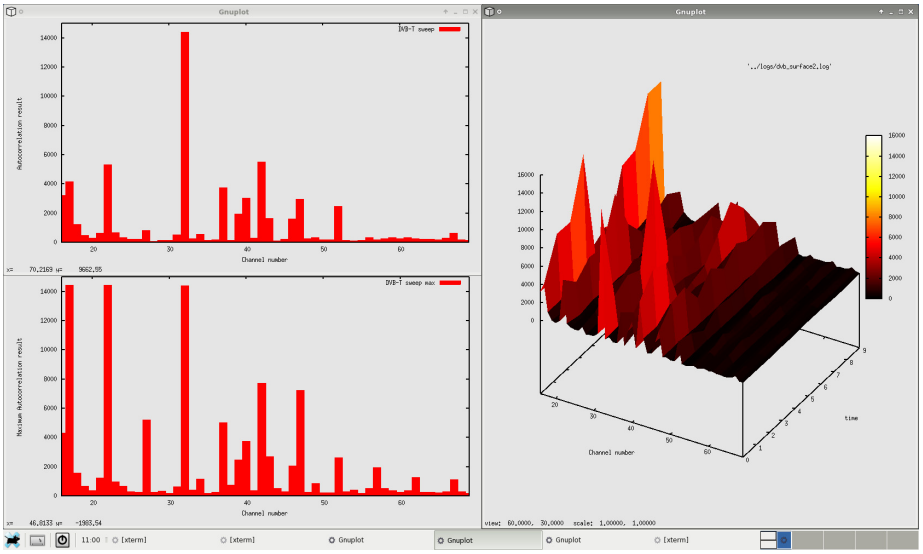


Figure 3-14: DVB-T sensing.

It is possible to add functionality to the SE API and/or the application. This can be adding a new mode to the sensing library, adding parameters to the configuration struct or implementing extended calculations or functionality in the different layers of the API.

3.5 Summary

This chapter explained in details a specifically developed hardware within the FARAMIR project along with its operational framework and sensing capabilities. The hardware itself is a demonstration of the ability to provide real-time spectrum measurements of wide range of frequencies in a small form-factor design that can be integrated in future mobile and handheld devices. The introduced hardware is integrated as an MCD in the FARAMIR's REM backend system (chapter 2) providing improved sensing capabilities and resulting in more reliable REM construction.

4 Cognitive Femtocells

This section elaborates the usage of the REM backend technology in order to provide optimal power control, channel allocation and antenna directivity for cognitive femtocells (increased intra-operator spectrum management functionality). A first stage version of this prototype was presented in its early development stages on an Internal France Telecom (FT) fair in December 2011, and the development has been continued until the present day including integration of full LTE prototype equipment for HeNB and corresponding UE.

4.1 Prototype Objectives and Summary of the Considered Scenario

In this prototype, a number of indoor deployed LTE femtocells are able to adapt their use of resources to avoid interference. The adaptations that are possible are power control, resource allocation and antenna directivity. The REM is used to assist in the Radio Resource Management (RRM) decision making and hence improve efficiency in resource utilization/optimization. Figure 4-1 depicts the overall scenario of interest.

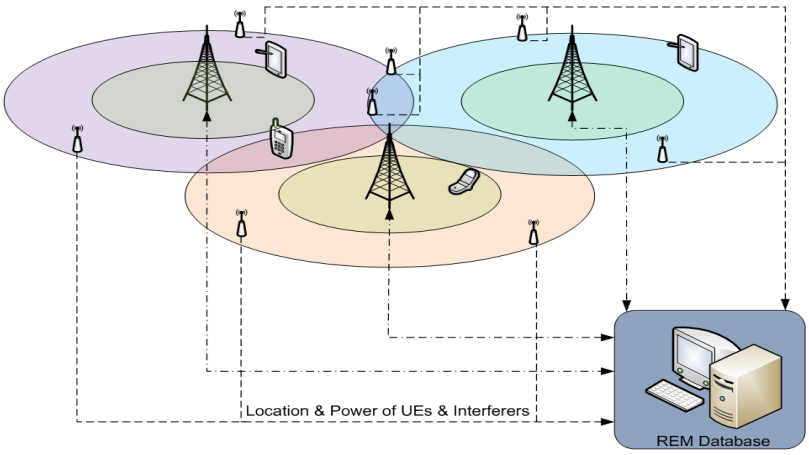


Figure 4-1: Cognitive femtocell prototype high level architecture.

4.1.1 Scenario 1 - Interference to/from neighbouring HeNB

In this scenario, the only interference is from other indoor (i.e. low power) femtocells. REM based coordination is exploited to avoid the interference that would otherwise occur between the adjacent femtocells. The neighbouring femtocells RRM functions are assumed to be "REM enabled" and hence register with REM to facilitate the coordination process. The relative localization of the UEs, in respect to the femtocells, is used to determine when interference will occur and hence make the appropriate RRM related power control/resource allocation or antenna directivity related decisions (as shown in Figure 4-2). The shadowing in the environment caused by building construction and environment (i.e. such as office walls and furniture) as well as movement of the UEs and people play an important part in this scenario and are considered by REM.

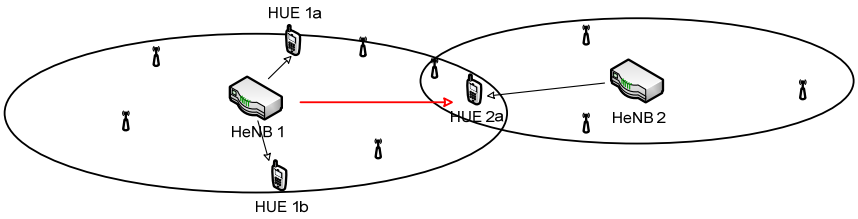


Figure 4-2: Neighbouring HeNB Interference.

4.1.2 Scenario 2 - Interference to/from Neighbouring eNB

This scenario is different from the previous one in that the interference is due to the existence of an overlay eNB cell within which the HeNB resides. In this case, the eNB is located outdoors and has significantly higher transmit power level compared to the femtocell HeNB. The potential interference victim node (UE) can be either attached to the HeNB or the eNB and the HeNB must perform the appropriate adaptation in order to prevent the interference (as shown in Figure 4-3). The building shielding effects and movement of the UEs plays an important role in this scenario and are predicted by the REM (e.g. based on neighbourhood building plans). The impact of people and smaller dynamic objects are less important in this scenario. As the eNB transmit power is expected to be high compared with the HeNB, there are two possible sub-scenarios, one when the HeNB is at the extremity of the eNB range and the other when the HeNB is near to the eNB, but shielded by a building structure (which may not shield the entire cell coverage area). Hence, in the first sub-scenario, the predominant interference victim is the UE associated with the eNB when it is affected by (i.e. close to) the HeNB in the downlink or the UE associated with the HeNB in the uplink direction, whereas, in the second sub-scenario the main victim is the UE associated with the HeNB by the downlink transmission from the eNB.

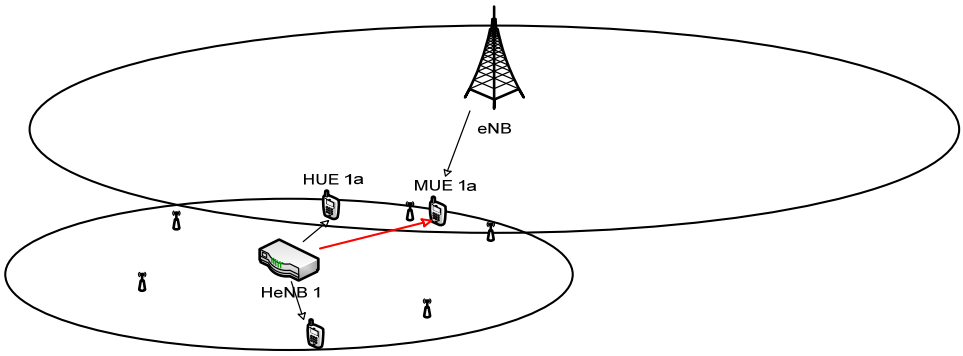


Figure 4-3: Neighbouring eNB Interference.

4.2 First Stage Prototype

In order to gain experience from the effects of the propagation environment indoors, and also to understand the key integration points between the femtocell components and the REM backend,

implementation efforts related to the scenario were split into two stages. While the actual LTE hardware and software was being prepared, WARP software defined radio boards [52] were used as femtocell and UE replacements for prototyping purposes.

The first stage cognitive femtocell prototype illustrates in Scenario 1 the optimum setting and tracking of the: 1) femtocell transmit powers[1] and 2) channel allocations by using the radio environmental information. The initial transmit power and channel allocation of a femtocell at start-up can be downloaded from the network, but they are clearly not optimal values which are adapted to the specific radio environment of the femtocell and which establish a compromise between coverage and interference. To this end, REMs can provide the necessary environmental RF information to find the optimum transmit powers and channel allocations. The exact coverage and capacity needs of the neighboring femtocells are also part of this environmental information. REM can also provide the statistical information on the propagation medium so that the optimum transmit power values and channel allocations can be calculated using a realistic propagation model. In this way, REMs are powerful enablers which can provide increased intra-operator radio resource management functionality.

4.2.1 Self-Optimization of Femtocell Transmit Power through REMs

We consider two neighboring Closed Subscriber Group (CSG) indoor LTE femtocell access points that are managed by a central unit for transmit power optimization. This means that any problems are reported to the central unit that will then instruct the two femtocells on how the transmit power need to be managed. Since femtocells are deployed and operated by the users, their ON-OFF time patterns and locations are not known beforehand. A femtocell may appear or disappear to re-appear again elsewhere without prior warning. Therefore, transmit power setting of femtocells are subject to constant optimization in order not to suffer from lack of coverage, and also from inter-femto interference in the case of CSG femtocells. It means that femtocell transmit power has to be optimized at initial start-up as well as later on, according to the turning ON-OFF and/or re-location of the neighboring femtocells. And since femtocells are installed and operated by the customers, they must be plug-and-play devices, requiring this constant optimization process to be performed autonomously.

Let us consider the case of initial powering-up of a femtocell. Since it is unaware of its surrounding environment, autonomous setting of its initial transmit power is a challenging task. If the transmit power is too low, it may not cover the whole area that it is supposed to provide service to. If the transmit power is too high, it will probably cause interference to the neighboring femtocells (interference to the neighboring macrocells is also an issue, but is not considered in this scenario). An optimum value of the transmit power should be set in order to compromise between coverage and interference.

[1] By "femtocell transmit power", we mean the DL traffic channel transmit power of the femtocell access point.

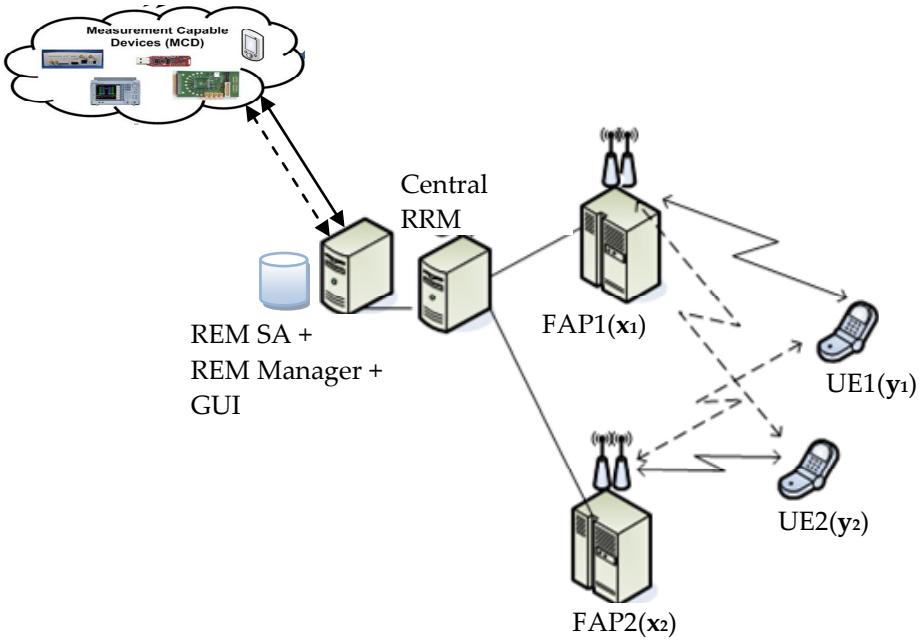


Figure 4-4: Self-optimization of femtocell transmit power.

There are two femtocells access points (FAP1 and FAP2) and two UEs (UE1 and UE2) where xi and yi denote the locations of the FAPi and UEi (i=1,2) respectively. FAP1-UE1 forms one pair and FAP2-UE2 forms the second pair, as depicted in Figure 1. Both FAPs are controlled by the Central Radio Resource Manager (CRRM). Initially, FAP1-UE1 pair is operational (ON) and FAP2-UE2 pair is OFF. We consider a video streaming application on UE1 provided by FAP1 for visualization purposes. The transmit power of FAP1 (P^t_{FAP1}) is set at a level to provide a sufficient QoS for the video streaming application (without any optimization since there are no neighbouring femtocells in the beginning).

The FAP2-UE2 pair is switched ON. Again, we consider a video streaming application on UE2 provided by FAP2.

Scenario without REM

One extreme is that FAP2 may adopt a "cautious" policy and set its transmit power (P^t_{FAP2}) to a low value, causing a coverage hole. Although we cannot visualize the coverage hole, we should observe the QoS degradation experienced by UE2.

On the other extreme, FAP2 adopts a "courageous" policy and sets its transmit power (P^t_{FAP2}) to a high value, causing interference to the FAP1-UE2 pair. Although, we cannot visualize the interference zone, the QoS degradation experienced by UE1 on the video streaming application is clearly observable.

Scenario with REM

Next, we discuss the solution with REM. FAP1 and FAP2 are both connected to a laptop which hosts the REM, specifically, the REM Storage and the REM Management functional blocks. In reality, this laptop represents a central network entity located either at the femtocell Management System (i.e. the Operation and Maintenance Center –OMC– or at the HeNb GW of the LTE architecture.

When FAP2 is switched ON, the REM Manager first verifies if the REM data is up-to-date. If not, it requests measurements from the MCDs via the REM Acquisition modules located also at the laptop as far as the demo is concerned. In reality, the REM Acquisition modules can be located at the FAPs (in case of MCDs being the UEs) or in a central node of the dedicated sensor network (in case of MCDs being the sensors) which is connected to the REM Manager. Note that although in reality, UEs will act as MCDs, in the prototype, the UEs will not act as MCDs due to the difficulty in implementing the required communication between the UEs and the FAPs. After verification of the REM, the REM Manager performs the optimization task by using the information stored in the REM Storage unit and we can observe that the video streaming application on UE1 does not suffer from QoS degradation.

4.2.2 Self-optimization of femtocell channel allocation through REMs

We consider three neighboring CSG indoor femtocell Access Points (APs) that are not coordinated through a common network management. This means that they are unaware of the presence (as well as the state and the characteristics) of neighboring APs. A UE is attached to each AP, forming three AP-UE pairs: (T1, R1), (T2, R2) and (T3, R3). (T1, R1) and (T3, R3) are close to each other and (T2, R2) is relatively far from the other two (Figure 4-5). There are two available channels, ch1 and ch2, in the RF environment.

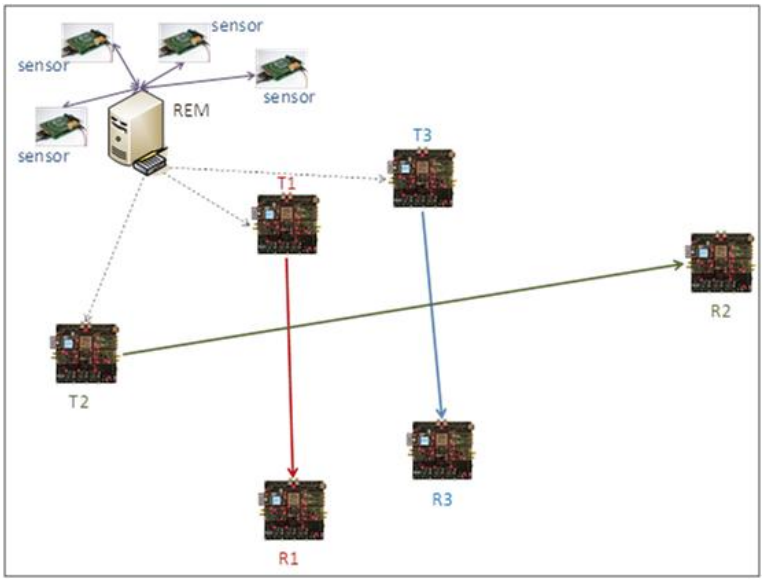


Figure 4-5: Optimized channel allocation overall setup.

Ti's and Ri's are composed of WARP boards, Ti's configured as WiFi access points, and Ri's configured as mobile terminals (connected to laptops for illustrating some video streaming applications). The REM sits on a laptop and is connected to the Ti's via a backhaul (wired) link. The sensors are of heterogeneous nature (USRP2, TI, Sun SPOT, spectrum analyzers etc.) and perform energy-based spectrum measurements on the specified frequency band (ch1, ch2, ch1+ch2).

Scenario without REM

At t = t0, (T1, R1) starts communication (Figure 4-6). It chooses a default value of transmit power, Pt, and an arbitrary channel, say ch1. A video application can be seen on the screen of T1 (i.e. the screen of the laptop connected to the WARP board). A high video quality can be observed.

Figure 4-6: Time t = t0 in the scenario without REM.

At t = t1, (T2, R2) starts communication, also with a video application on R2 (Figure 4-7) with a default transmission power Pt. Since T2 is not coordinated with T1, it chooses a channel in an arbitrary manner. If ch1 is chosen, then some glitches can be observed on the videos of R1 and/or R2. Otherwise, QoS is high on both Ri's.

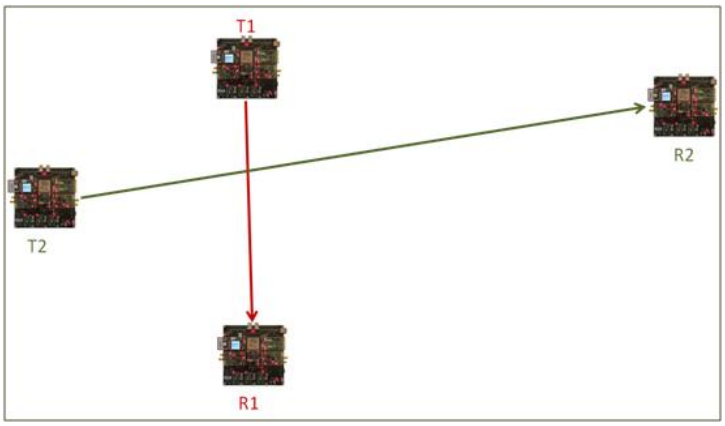


Figure 4-7: Time t = t1 in the scenario without REM.

At t = t2, (T3, R3) starts communication, also with a video application on R3 (Figure 4) and also with a default transmission power Pt. Since T3 is not coordinated with the other Ti's, it chooses a

channel in an arbitrary manner. If it chooses the same channel with (T1, R1), bad video quality will be observed on both R1 and R3. If it chooses the same channel with (T2, R2), some glitches can be observed on R2 and R3. There is no guarantee to have an optimum allocation with respect to the relative positions of the Ti's.

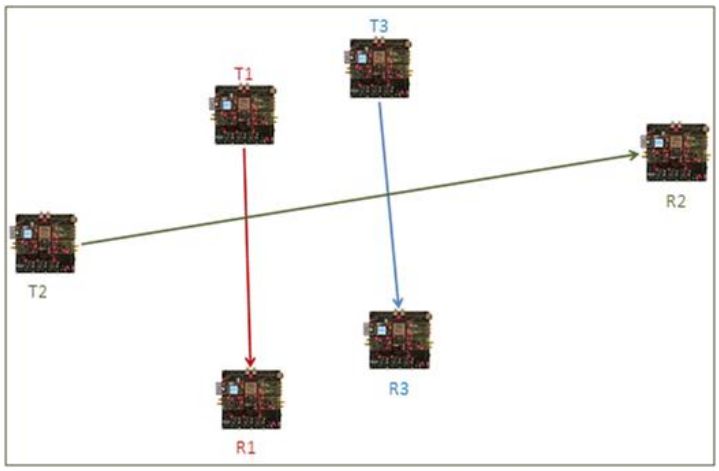
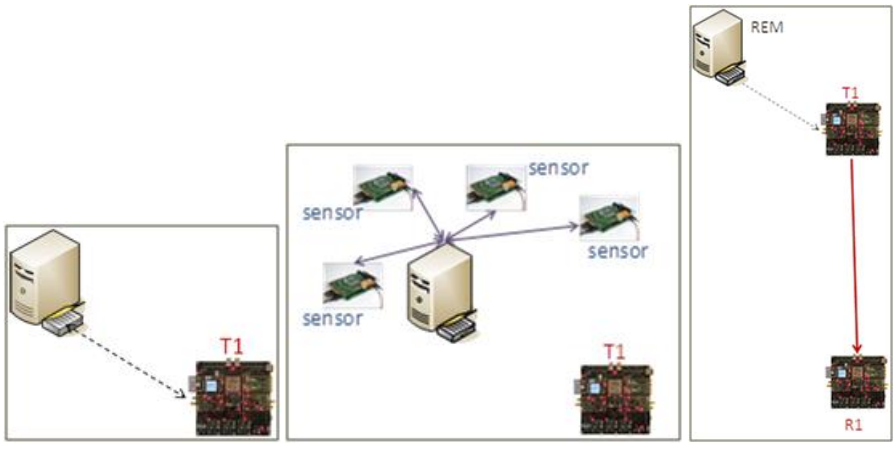


Figure 4-8: Time t=t2 in the scenario without REM.

Scenario with REM

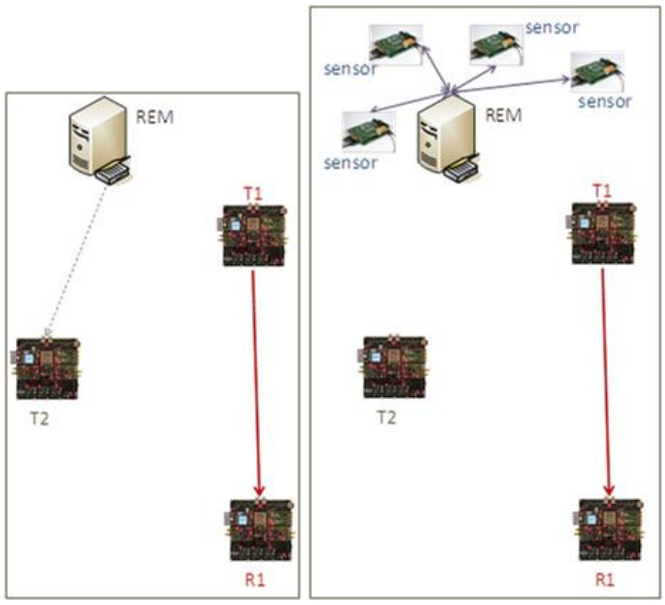
At t = t0, (T1, R1) starts communication with a default transmission power Pt. But before starting communication, T1 gets registered to the REM by sending its location and some possible RF measurements that it carries out over the two available channels ch1 and ch2 (Figure 4-9a). REM Manager checks whether the information in the REM Storage is up-to-date or not. If it is not up-to-date, it requests measurements from the sensors (Figure 4-9b) via REM Acquisition module, and updates the information/map in REM Storage. Then, it instructs a transmission configuration (which channel to use) to T1 (Figure 4-9c) and T1 starts communication with R1. A good video quality is observed on R1.



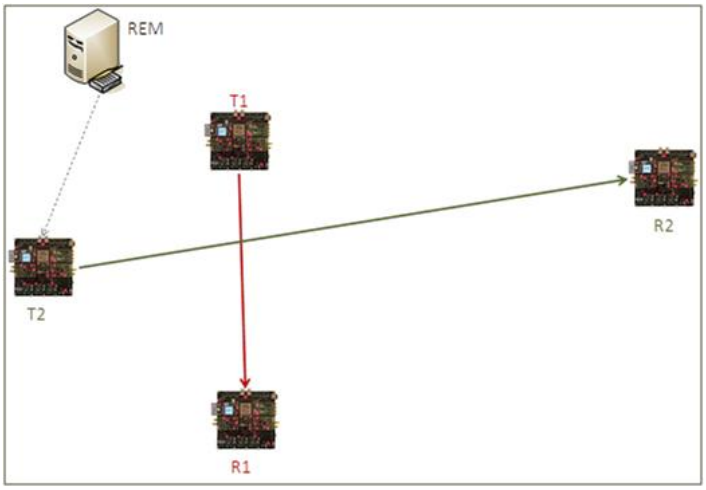
(a) **(b)** **(c)**

Figure 4-9: Time t = t0 in the scenario with REM.

At t = t1, (T2, R2) starts communication with a default transmission power Pt. But before starting communication, T2 gets registered to the REM (Figure 4-10a) by sending its location and some possible RF measurements that it carries out over the two available channels ch1 and ch2 (Figure 4-10a). REM Manager checks whether the information in the REM Storage is up-to-date or not. If it is not up-to-date, it requests measurements from the sensors (Figure 6b) via REM Acquisition module, and updates the information/map in REM Storage. Then, it instructs a transmission configuration (which channel to use) to T2 (Figure 4-10c) and T2 starts communication with R2. If ch1 was allocated to (T1, R1) at t=t0, ch2 is allocated to (T2, R2) at t = t1 and vice versa. Again, a good quality is observed on both R1 and R2.



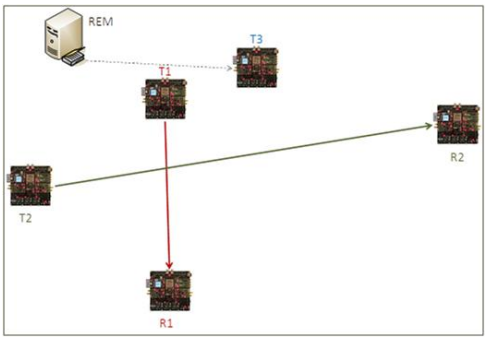
(a) **(b)**



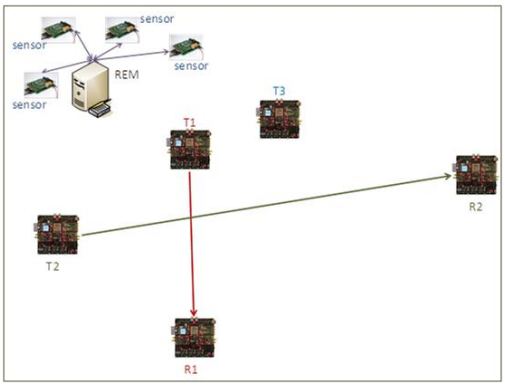
(c)

Figure 4-10: Time t = t1 in the scenario with REM.

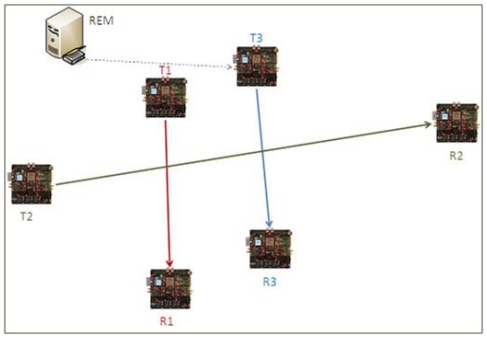
At t = t2, (T3, R3) starts communication, but before starting communication, T3 gets registered to the REM (Figure 4-11a) by sending its location and some possible RF measurements that it carries out over the two available channels ch1 and ch2 (Figure 4-11a). REM Manager checks whether the information in the REM Storage is up-to-date or not. If it is not up-to-date, it requests measurements from the sensors (Figure 4-11b) via REM Acquisition module, and updates the information/map in REM Storage. Seeing that (T1, R1) is close to (T2, R2) and (T3, R3) is relatively far, it instructs T2 to switch to the same channel as T1, and allocates the other channel to T3. In this way, an optimum allocation with respect to the locations of the Ti's is guaranteed (Figure 4-11c).



(a)



(b)



(c)

Figure 4-11: Time t = t2 in the scenario with REM.

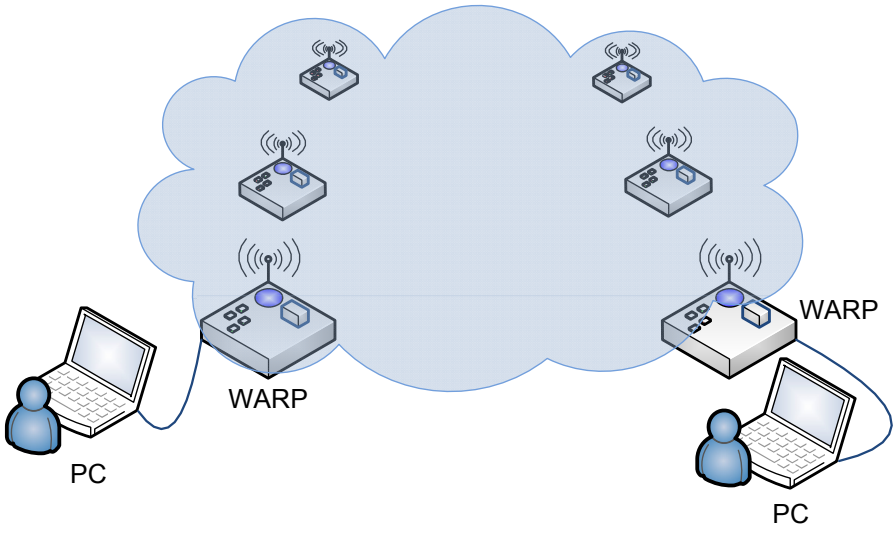


Figure 4-12: Illustration of the WARP-based femtocell prototype.

The above scenario was implemented using the REM framework and using the WARP boards as simple configurable radio interfaces as illustrated in Figure 4-12. A selected test video is streamed using RTP based on the VLC player code based, with the actual sending process also controlled through the REM GUI described in Chapter 2. In the prototype implementation a WARP board receives the Ethernet packets obtained from the connected laptop, packages them into the payload portion of the WARP radio interface packets and transmits them over the air using the radio daughter board in the selected 2.4GHz ISM band. The transmission channel and the transmit power can be chosen on per packet basis. Following this a second WARP board, as a receiver, accepts the packets and passes them to the Ethernet port where a waiting VLC client accepts the packets and starts displaying the streamed video in real-time.

The control information from the REM engine to a WARP board is passed over the UART interface. This includes the destination address, the receiver's packet detection sensitivity threshold, the transmission power, the payload modulation scheme and the communication channel. In order to enable experimenting with the first stage prototype in compact areas, SMA based RF attenuators were used to keep the WARP communication restricted to a very restricted region and to demonstrate the effects of different levels of interference from other WARP boards (at nearby and relatively far distances) in the femto-cell scenario.

The WARP software used in the first stage prototype is based on the RWTH Decomposable MAC Framework [53] (in turn based on the WARP Reference Design v16 [54]), which provides the necessary radio and MAC functionality (cf. Appendix A for API definitions) to establish communication among WARP boards for the femto-cell scenario. Therefore the channel access and air interface used differs somewhat from the actual LTE deployments (addressed in the second stage femtocell prototype), but this deviation is minor for the purposes of the above storyboard.

4.2.3 Overview of the Decomposable MAC Framework

Decomposable MAC Framework is based on the component oriented design philosophy, where a particular MAC solution is realized by binding or "wiring" a set of common MAC functionalities. These common MAC components serve as the building blocks. The concept is similar to LEGOS® as illustrated in Figure 4-13, where two different MAC schemes are realized based on the same set of common MAC components.

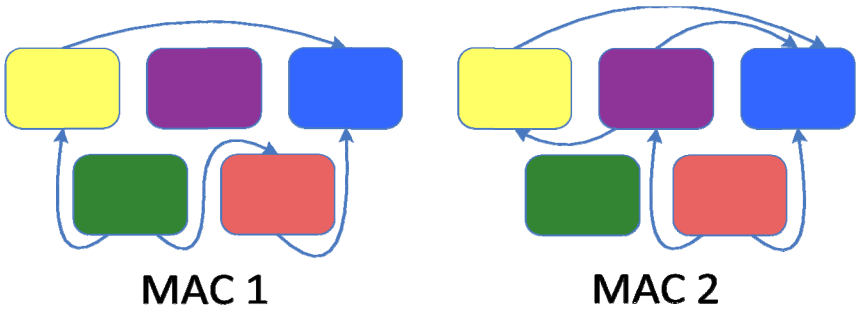


Figure 4-13: Different MAC schemes are realized by binding the same set of common MAC components.

The decomposable MAC Framework defines a list of fundamental components, which include timer functionalities, carrier sensing algorithms, radio state control functionalities, random number generator Framing and buffer management functionalities, sending and receiving frame functionalities, etc. The fundamental MAC components in Decomposable MAC Framework are implemented in hardware to allow speed gains. Generally, the fundamental components are wired together in a similar fashion across different MAC protocols. This leads to a set of secondary level components consisting of fundamental components and/or other secondary level components. Random backoff functionality is a good example of a secondary level component, which consists of timer, carrier sensing and random number generator. Other secondary components may include sending packets, RTS/CTS/DATA/ACK handshake, etc.

4.2.4 Deployment Experiences

The WARP based first stage prototype was tested extensively both during the development phase as well as subsequently in the Orange Labs research exhibition in December 2011 in which the implementation and related concepts were showcased at (see Figure 4-14). The experiences gained during these times were highly positive, showing that the REM-based approach can indeed highly reliably enable high grade of service being achieved in congested radio environment. While the used deployments were small, these results were found promising enough so as to continue the development of the prototype towards its second stage, this time incorporating actual LTE prototyping components instead of the software defined radio based solutions employed during

the first stage. The second stage of the prototype development is discussed in more detail in the following Sections.

Figure 4-14: First stage femtocell prototype at the Orange Labs research exhibition.

4.3 Second Stage Prototype: Necessary Components and Connections to the REM backend

To facilitate the "REM enabled" notion, cognitive femtocell scenarios require that the femtocell RRM, adaptive Transmit Power Control (TPC) and antenna directivity functions register with the REM backend and are able to exchange the relevant information to support the RRM optimization. The dataflow interfaces to support this are illustrated in Figure 4-15, which indicates the existence of two REM managers. The one termed UKIM REM Manager is the one elaborated in Chapter 2. The reason behind this is to facilitate the use of different models covering different aspects of the radio environment. For instance, considering the physical layout modeling (such as walls and their associated radio propagation characteristics), dynamic shadow prediction of moving objects (such as people), different interpolation techniques operating on different timescales and also optionally resource usage analysis models. It is also possible for different models to exist to cater for different spatiotemporal and frequency resolution, which is necessary in the scenario consisting of heterogeneous HeNB and eNB interference.

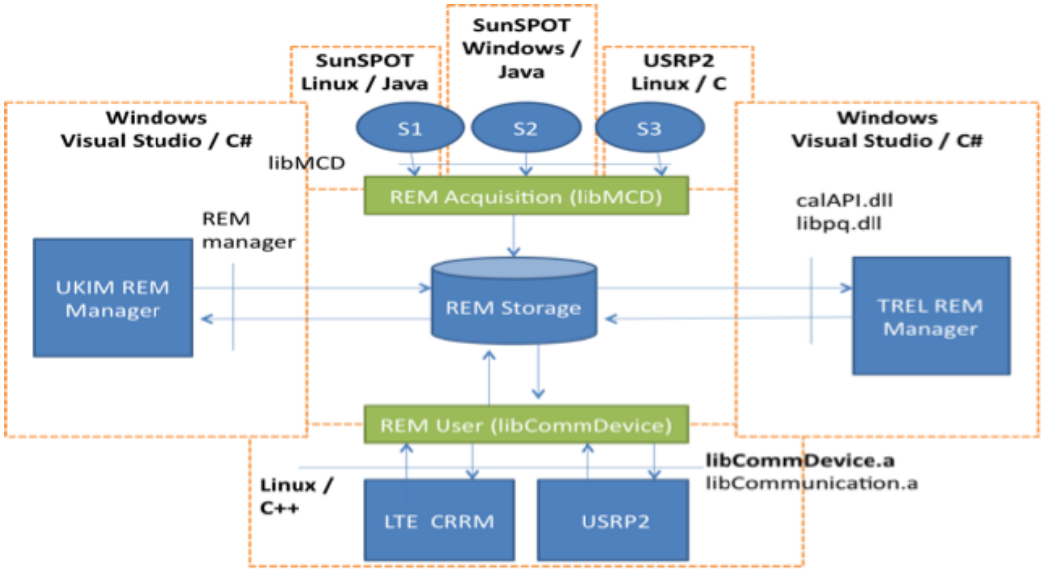


Figure 4-15: Overview of dataflow interfaces.

4.4 Prototype Functionalities

We begin the description of the second stage of the prototyping effort by briefly recalling the functionalities of the REM framework as used in the femtocell scenario. A subset of the overall REM architecture was implemented for initial prototyping purposes separately by TREL with the following functionalities. The second stage prototype was then successively integrated in the final stages of the project together with the mainline REM prototype backend described in detail in Chapter 2.

4.4.1 REM Acquisition

The MCDs sense the radio environment and process the raw data before passing it via the REM acquisition function to the REM storage. The MCD sensing functions are configurable in both time (i.e. sampling rate) and frequency and can be controlled by the REM acquisition function. The USRP2 based MCD have significantly more processing capability and frequency range than the SunSPOT sensors. As it utilizes a fast Analogue to Digital Conversion (ADC) function (of up to 100MSPs) of the Intermediate Frequency (IF) that is down-converted from Radio Frequency (RF), it has the ability to digitally process the samples in order to detect and distinguish between multiple signals in the same frequency band (i.e. on different channels or resource blocks). Hence, it permits the selection of not only the band of interest but also the sampling rate, FFT size and channelization of interest as well.

4.4.2 REM Storage

This function is provided by a Structured Query Language (SQL) relational database solution that has several tables corresponding to the different types of REM data that need to be stored. The database function permits queries and also asynchronous retrieval of data by the REM managers and REM users. The database also permits the REM manager to feed processed data back into the

database corresponding to the generated REMs. This data provides the REM characterization of the radio environment.

4.4.3 REM Manager

The REM manager function is responsible for calculating the REM using measurement based prediction models. Different types of models are supported and hence the REM can support multiple REM managers that are differentiated by the type of model or their resolution in space, time and frequency dimensions. The REM Manager also supports localization functions in order to determine relative positions (in the REM) of interferers and potential victims.

4.4.3.1 TREL REM Manager

The TRL REM manager functions are illustrated in Figure 4-16 and show that there are three main sub-functions, which are REM localization, model and interpolation. The REM indoor localization function performs clustering of MCD measurements and computation of relative path losses (and REM assisted compensation) in order to determine the relative positioning of the transmitters in the environment. It is assumed that GPS is not available to the indoor UE devices and so the localization is performed by the REM manager function. For outdoor located UEs, it is possible to utilize their GPS capabilities and hence this function becomes unnecessary.

The REM model function consists of both propagation and antenna models that determine how the radio energy is radiated in the environment. As the LTE devices can consist of directional antenna elements, this is taken into account within the corresponding antenna model. The current antenna model is based on an omni-direction or quasi-cardioid gain patterns, although other patterns can be easily added. Also, the underlying radio propagation model is based on a uniform distance relationship (exponent) that can be modified to take into account the type of environment and frequency band considered.

Finally, the interpolation function makes a prediction of the radio signal level at various points in the environment, using the REM models, in order to permit adaptation and optimization of the configuration and radio resource utilization. The REM interpolation function not only interpolates the radio signal (as per the propagation and antenna radiation models), but also interpolates the shadow fading anomalies observed based on the REM measurement data. This is estimated by comparing the REM measurement errors with respect to the expected values and the average differences obtained for each Measurement Point (MP).

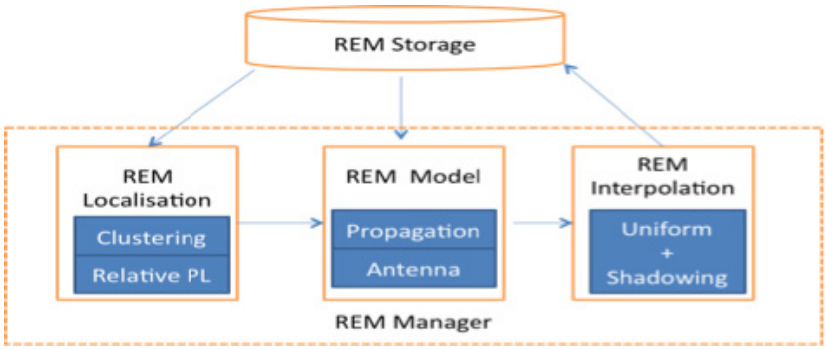


Figure 4-16: Internal functions of the TREL REM Manager

REM localization. The REM localization function uses the relative signal strength MCD measurement values and makes use of the ratio of the measurements taken from different pairs of MCDs. The REM processing that is required to determine the relative signal strengths, with respect to the MCDs, and hence also an estimate of the transmit power of the transmitters, is then given by the geometric calculation in Figure 4-17, where *a*, *b* and *d* are the measured reference signal levels and *c* represents the path loss between the adjacent APs. The assumption in this approach is that MCDs are equally spaced and each of the transmitter signals can be distinguished by the MCDs. The method of distinguishing the individual signals is by processing the MCD data using a fuzzy-c-means clustering algorithm in order to determine values, occurring over a time period of a few seconds, exhibiting similar characteristics. Hence, the noise and far away (low power) interferers are filtered out of the resulting REM model. Further to this, multiple transmitters operating on the same channels can be distinguished from each other provided that they have sufficient spatial separation or power level differences (assuming discontinuous transmission schemes, such as in LTE). Therefore, in order to resolve the signals from various sources, the measurements are clustered using a fuzzy-c-means clustering process after computing relative signal levels (*r* and *s* etc. as shown in Figure 4-17). This clusters the individual temporal measurement vectors that comprise discrete samples X = {a, r, s,....} into groups that represent the most likely determination of relative signals from the data sets. Then, the (x, y) point coordinates (in signal space) are calculated based on cluster centers v_i to determine the transmitter signals relative to the MCD reference points and predict the transmit power.

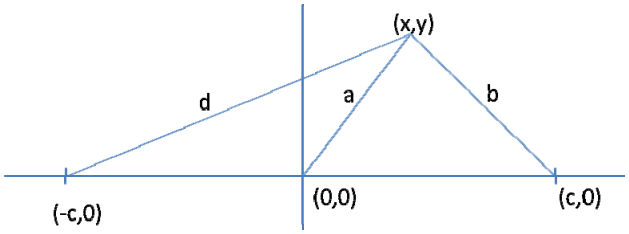
The fuzzy-c-means clustering algorithm takes the data vector set $X = \{x_1, x_2, x_3, ..., x_N\}$ and partitions it into *c* clusters such that the following objective function is minimized:

$$J\ (U, v) = \sum_{k=1}^{N} \sum_{\mathrm{i=i}}^{\mathrm{c}} (\mathrm{u_{ik}})^{\mathrm{m}}\, (\mathrm{d_{ik}})^2$$

$$u_{ik} = \frac{1}{\sum_{j=1}^{c} \left(\frac{d_{ik}}{d_{jk}}\right)^{2/(m-1)}} \mathrm{i} = 1,2, \dots, \mathrm{c\ and\ k} = 1,2, \dots, N$$

$$v_i = \frac{\sum_{\mathrm{k=1}}^{\mathrm{N}} (\mathrm{u_{ik}})^{\mathrm{m}}\, \mathrm{x_{ik}}}{\sum_{\mathrm{k=1}}^{N} (\mathrm{u_{ik}})^{\mathrm{m}}} \qquad \mathrm{i} = 1,2, \dots, \mathrm{c}$$

where U is the membership matrix and u_{ik} is value of the k^{th} data vector x_k in the i^{th} cluster, $d_{ik} = \| x_k - v_i \|$ is the Euclidean distance between data vector x_k and the cluster centroid v_i and exponent m>1. The clustering algorithm is an iterative process for minimizing the objective function $J\ (U, v)$, involving selecton of an initial random cluster membership matrix (U) where: $\sum_{i=1}^{n} \mathrm{u_{ik}} = 1$ for all k values (1..*N*). Then, the initial cluster centres v_i are computed using the above equation followed by the calculation of objective function $J\ (U, v)$, and re-computation of a new membership matrix by starting the process over again. The iterations continue until either the maximum iterations (*R*) is reached or the maximum difference between membership values, from the previous iteration, is less than a threshold amount (α).



$$a = \sqrt{x^2 + y^2}$$

$$b = \sqrt{y^2 + (c - x)^2}$$

Assuming $b = a.r$

$$r^2 = \frac{y^2 + (c - x)^2}{y^2 + x^2}$$

$$r^2(y^2 + x^2) = y^2 + c^2 - 2cx + x^2$$

$$y = \pm\sqrt{\frac{c^2 - 2cx}{r^2 - 1} - x^2}$$

$$d = \sqrt{y^2 + (c + x)^2}$$

Assuming $d = a.s$

$$s^2 = \frac{y^2 + (c + x)^2}{y^2 + x^2}$$

$$y = \pm\sqrt{\frac{c^2 + 2cx}{s^2 - 1} - x^2}$$

$$x = \frac{c(s^2 - r^2)}{2(r^2 + s^2 - 2)}$$

Figure 4-17: Relative signal space geometric calculation of coordinates (x, y)

The drawback of the basic relative signal strength approach for the REM localization function, described above, is that there is an implicit assumption that the radio propagation is uniform in space and also that the individual transmitter signals from different locations do not exhibit similar relative signal values (i.e. are collocated or occur in symmetrical locations with respect to the MCDs). The consequence of these assumptions not being valid is the loss of accuracy and ability to distinguish between different transmitters, both of which are undesirable. Therefore, an additional step is performed to attempt to compensate for the shadowing anomalies in the environment by using the output of the REM interpolation to apply a corresponding compensation (δx, δy) to the (x, y) coordinates computed by REM localization. This compensation process requires historical measurement data from different MPs and relatively slowly moving shadowing effects in order to provide a better overall localization estimate. In order to provide better determination of individual transmitters, it is required to also consider frequency sub-channels and an assumption that different transmitters will have more activity on certain sub-channels.

REM Model. The REM model function is used to minimize the number of measurements that must be taken in order to build up the characteristics of the radio environment. Therefore, the benefit that the REM model provides depends heavily on the real radio environment and how many MCDs are available within a given locality. For instance, if the radio environment is dynamic and there are few MCDs available, the use of specific REM models is more important, but in less dynamic

scenarios, with many MCD available in the locality, the use of models is less beneficial. For this reason, we consider the use of models that can take account of the highly dynamic aspects of the radio environment that cannot easily be captured with limited numbers of deployed MCDs in a locality. This is most likely to occur when sudden adaptive changes occur such as changing channel allocation, adaptive power control and antenna directivity. This necessitates an antenna directivity model that sufficiently captures the antenna configuration and the associated radiation pattern and a propagation model that represents the main distance based attenuation for the different radio channel resources.

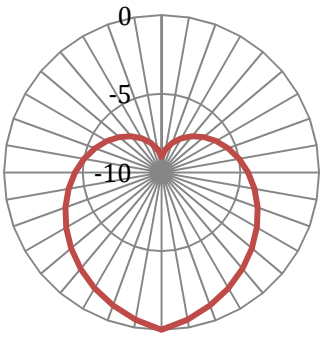


Figure 4-18: Antenna directivity model

The antenna directivity model currently used is based on a quasi-cardioid directional gain pattern (shown in Figure 4-18) or is assumed to be omni-directional. The antenna directivity can then be characterized by the gain difference (i.e. from front to back) and the angle of the beam with respect to a reference (i.e. such as magnetic North or a building elevation).

The main propagation model assumed within the REM model function is based on a simple distance dependent path-loss exponent. However, to take account of the shadowing anomalies in the environment (which occur due to obstructions), there is also a shadow fade localization and prediction model. Rather than attempt to track fast moving objects in the environment, this model is based on the historical REM measurement data that is used to correlate the spatial locations where differences occur from the measurements and the expected signal levels. These errors are then clustered (using the same fuzzy-c-means approach as within the REM localization function) to approximately localize the shadow regions in the radio environment.

REM Interpolation. The purpose of the REM interpolation function is to predict the signal levels at each of the points of interest in the local radio environment due to each of the transmitters. Therefore, the primary input for interpolation is the localization information together with transmitter power levels (and/or their predictions) and the corresponding REM models. The interpolation technique used in this instance is a uniform gradient based on point radiators and the corresponding radiation patterns obtained from the REM propagation and antenna models. The shadowing impact is currently assumed to be circular in nature with equal impact in all directions, which reflects point obstructions rather than linear obstructions (i.e. does not consider static walls etc.).

4.4.4 REM User (Communication Devices)

The REM users in the cognitive femtocell scenarios are the resource management functions of the communication devices which comprise of LTE femtocells, i.e. the RRM, the adaptive power control and the antenna directivity functions for the locality of the femtocell deployment. There are also additional USRP2 software radios that are configured to act like LTE femtocells and UE devices, although not actually conformant with the LTE standard, in order to have sufficient number of potential interference sources. The REM users register all the radio devices with the REM and also specify their corresponding configuration parameters and make requests for interference notifications from the REM. These notifications are based on thresholds set on the signal level received from certain specified transmitters at specified points in the radio environment (corresponding to the receivers that are potential interference victims). The functions used for this interface are:

- REM User to REM

```
INT initCommDevice(CommDeviceRegReq *conf, IPAD remIPAddr, unsigned short RemPort, char *commDeviceMacAddr);

INT closeCommDevice();

INT reportCommDeviceStatus(unsigned short status, UINT64 centerFreq, UINT64 bandwidth, float power, UINT32 peerID, UINT32 antennaDir);

INT requestNotify(UINT32 txId1, USHORT logical, float signalLevel, UINT32 txId2, UINT32 requestId);

INT cancelNotify(UINT32 requestId);
```

- REM to REM User :

```
INT doConfigure (const CommDeviceConfRsp *msg);

INT doGetLocation (const CommDeviceLocReq *msg);

INT processRegRsp (const CommDeviceRegRsp *msg);

INT processEventNotifyRsp(const CommDeviceEventNotifyRsp *msg);
```

Upon receiving a notification of a signal level threshold condition matching from the REM, the REM users perform an appropriate adaptation which is either to change the transmit power level, steer the antenna in an appropriate direction or reallocate resources. Therefore, several triggers are registered with the REM in order to determine the most appropriate action to take. The REM user function also informs the REM of the configuration change via the reportCommDeviceStatus function.

4.5 Field Results and Discussion

After explaining the specific details on the cognitive femtocell prototype, this subsection provides field testing results and a discussion of the potentials and the benefits of the REM technology.

4.5.1 TREL REM Manager

4.5.1.1 REM localization

The REM localization function has been tested with different MCDs and transmitter characteristics. The results of a test with USRP2 MCDs and a USRP2 transmitter are shown in Figure 4-19. The red

line indicates the smoothed trajectory of the transmitter through the test environment and the blue dots correspond to individual location predictions made at fixed time intervals. There is some variance around the smoothed trajectory caused by the multipath and dynamic variations of the environment that cannot be eliminated entirely by the clustering process.

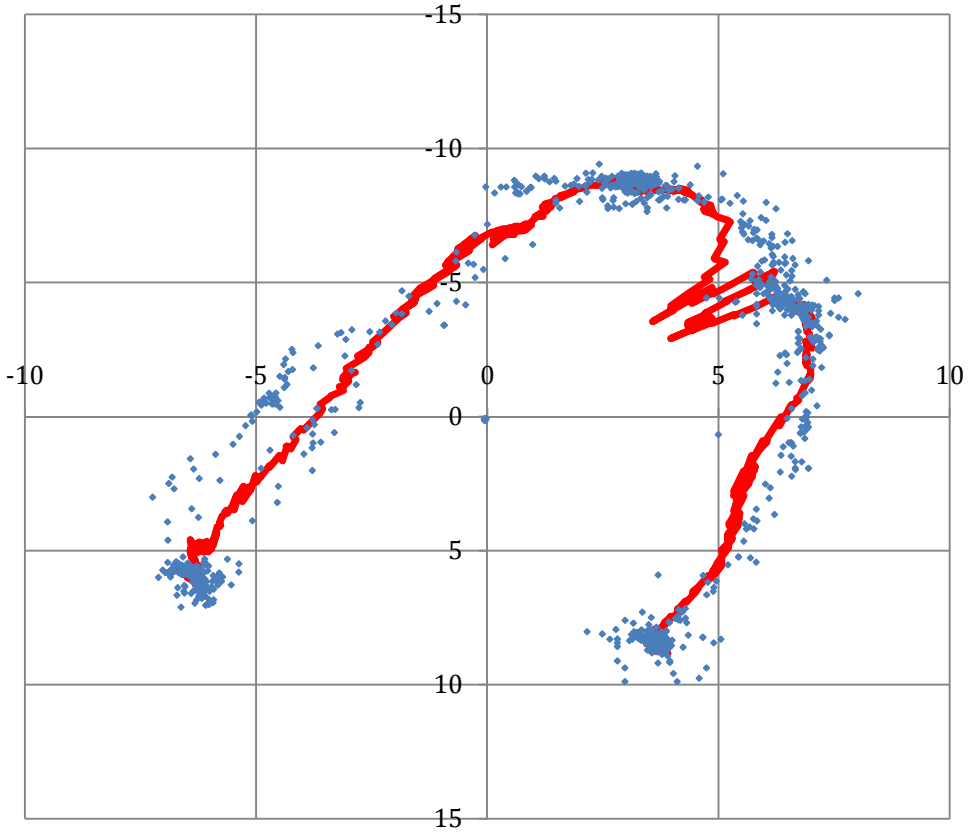


Figure 4-19: REM localisation test with USRP2s.

4.5.1.2 REM Model & Interpolation

The REM models used to estimate the signal level (and hence predict interference levels) from the different transmitters at the victim locations are illustrated in Figure 4-20. The blue dots correspond to the communication devices (i.e. radios) in the environment and the two transmitters are located at a and b with the other devices being inactive on the selected channel. In this case, the signal levels of the transmissions from devices a and b are predicted at locations c, d and e. The dark shadows indicate the predicted shadowing from people or objects in the environment.

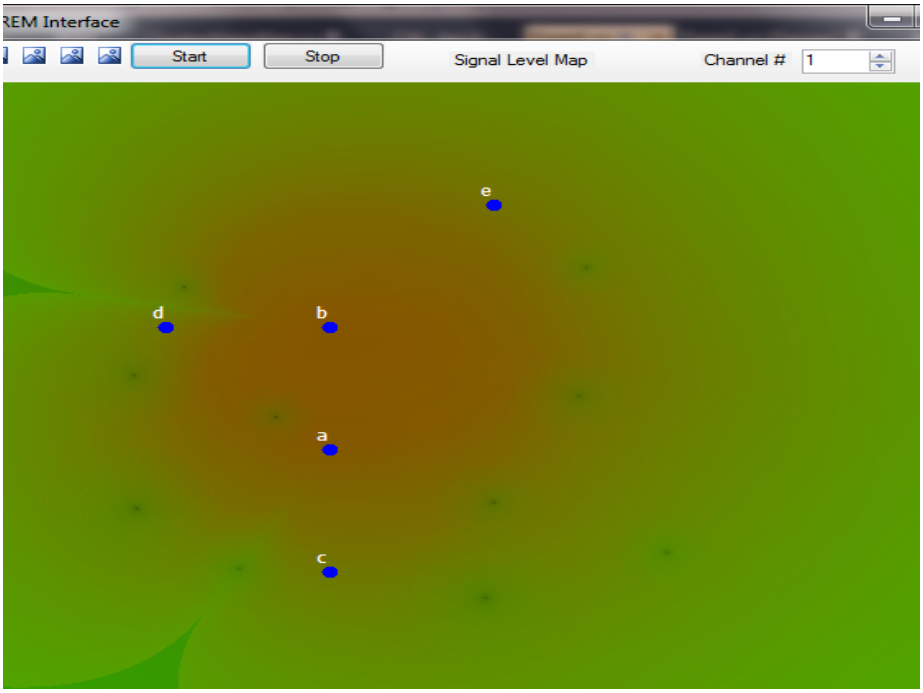


Figure 4-20: REM with directional antenna model and shadowing localization

4.5.2 TREL REM Users

eNode B. The eNodeB employed is a picoChip PC9608 LTE FDD eNodeB as shown in Figure 4-21. The device will be communicating with the test UE wirelessly on LTE Band 13 spectrum.

The eNodeB will be controlled by a central radio resource manager, which accesses the REM manager to obtain:

- Location of UE,
- Interference strength of different frequencies at UE location.

Using these, the CRRM will perform antenna control, using the directional antennas shown in Figure 4-22. Additionally, the CRRM will perform resource block allocation accordingly to the signal strength information to either maximize SNR or reduce interference to others.

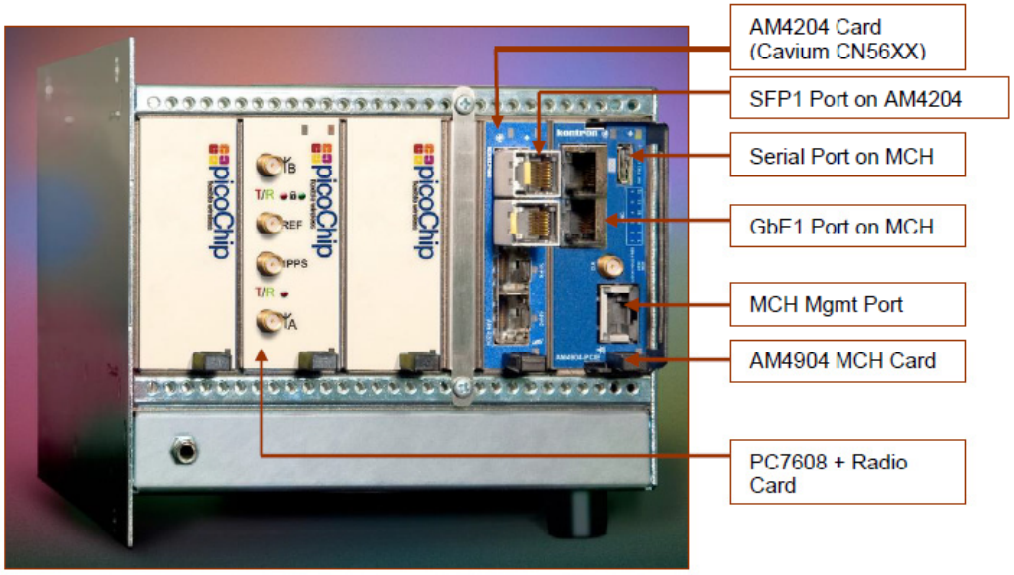


Figure 4-21: picoChip PC9608 LTE FDD eNodeB.

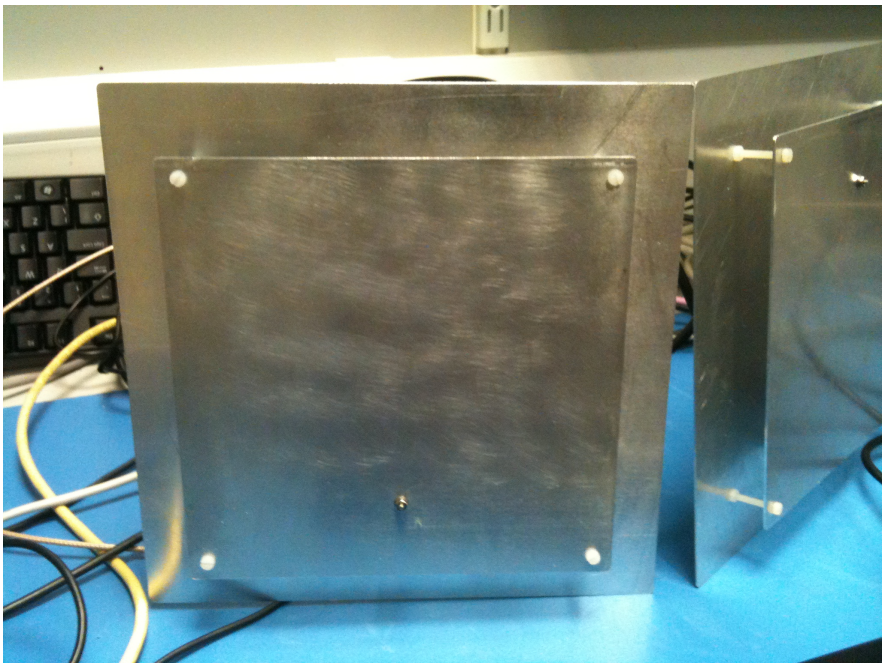


Figure 4-22: Directional antenna for 750 MHz.

Test UE. The test UE deployed is an Aeroflex TM500 Test UE capable of supporting up to 300 Mbps downlink and 75 Mbps uplink, as shown in Figure 4-23. The test UE will connect to the eNodeB and receive video streamed from the eNodeB as a visual indicator of the quality of the wireless link.

Figure 4-23: eNode and Test UE setup.

As discussed above, these prototype components have been integrated into a whole enabling experimentation and testing in an indoor femtocell scenario including the use of directional antennas to shape the interference patterns of femtocell based on the geolocalized measurements and location information stored in the REM. Within the tests carried out with the prototype setup, the potential of using the REM data for interference management in this manner was clearly demonstrated. Changing the direction of the antenna based on the localized position of the UE can be used to reduce the interference to adjacent femtocells quite significantly (values of 10-15 dB were observed in the tests, with higher values easily been obtainable with improved antenna construction and beamforming techniques) while at the same time improving the SINR at the UE itself. The consortium members are currently carrying out more extensive tests as well as studying the efficiency of different resource management algorithms based on the measurement data sets collected during the testing of the second stage prototype.

5 LTE Usage in TV White Spaces

A number of recent feasibility studies have addressed the question of deploying an LTE network in the UHF TV band white space [34][35]. The driving idea is that existing cellular network technologies could be adapted to operate in an opportunistic fashion and that this approach would allow for swift and smooth exploitation [36]. Both a coverage extension in rural areas and a capacity enhancement by small-cell operation are being considered as use cases.

The studies showed that such an opportunistic operation and the associated intersystem interference comes with high protection requirements with respect to the existing TV infrastructure [37]. In particular, such a system is generally believed to require a form of cognition in the sense that it needs a much higher level of *environmental awareness* than existing cellular deployments where a regime of exclusive licensing dominates.

This chapter focuses on the usage of the FARAMIR REM backend technology to facilitate LTE network deployment in the white spaces of the UHF TV band. REMs store and process a variety of spectrum measurement data provided by sensors and enable inference of environmental characteristics such as locations of transmitters, prevailing propagation conditions and estimates of the spectrum usage over time and space [38]. REMs are not only able to store relevant information about the primary TV transmitters (and other primary transmission devices as wireless microphones), but also about *other* secondary devices (LTE base stations or other transmitters) that exploit the TV white spaces under the reigning licensing conditions.

This particular FARAMIR prototype shows how intelligent guidance through a REM can enable the operation of an opportunistic cellular network in the spectral white spaces of the upper UHF TV bands. A prototype system consisting of a sensing device, a generic REM architecture along with an LTE Release 9 compliant base station illustrates how environmental awareness is obtained, reasoned about and exploited to control the actions of a communication network. All aspects of the cognitive cycle are covered. The architecture is modular in the sense that interfaces are generic and minimal.

Figure 5-1 illustrates the prototype system architecture and Figure 5-2 the LTE equipment in the laboratory of Huawei Sweden specifically developed to showcase the benefits of the FARAMIR innovations.

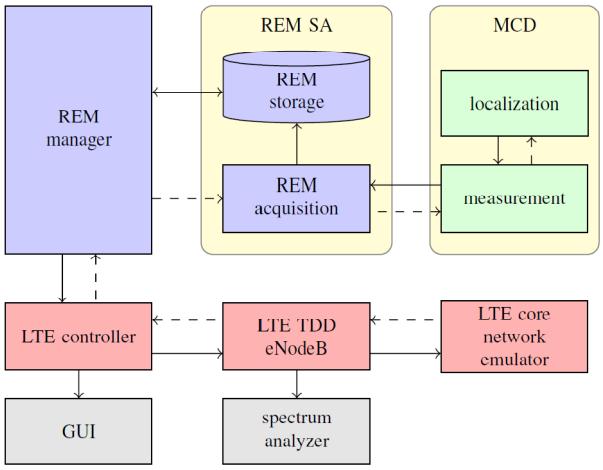


Figure 5-1: Prototype system architecture.

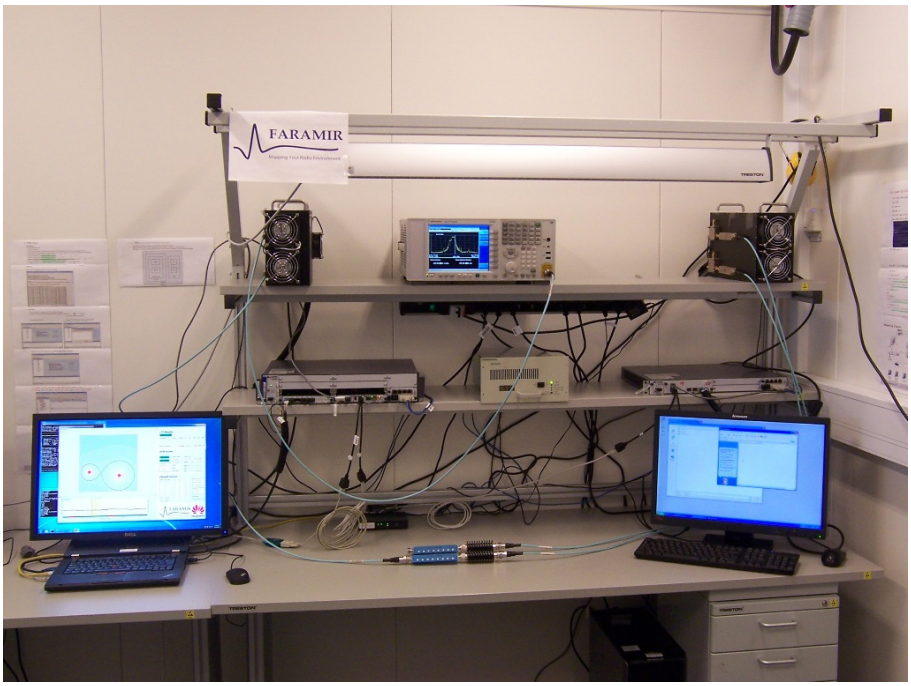


Figure 5-2: LTE prototype system.

5.1 Prototype Goals and Description

The overall goal is to demonstrate REM-based, dynamic spectrum access of an LTE network operating a 10 MHz channel in the 686-790 MHz UHF TV band. This section describes in more details the proposed demonstration setup.

The *essential loop of events* in the LTE subsystem of the demonstrator is:

1) The LTE controller queries the REM-manager for field-strength information;
2) Upon reception of such a query from the LTE controller, the REM manager in turn queries the REM's SA module for the relevant field-strength information;
3) The REM manager replies to the LTE controller with the relevant field strengths;
4) Based on the relevant field strength information, the LTE controller's Cognitive Resource Manager (CRM) maintains and updates a list of spectrum opportunities (TV channels that are "clear" for secondary transmission). If necessary, the LTE controller instructs the eNodeB to perform spectrum handover (moving its spectral content from one TV channel to another);
5) The LTE system performs the relevant spectrum handover and
6) GOTO 1).

The prototype system illustrates in various ways how environmental awareness can be exploited. In the course of the above events, relevant signals, spectra and optimization parameters in both the REM-manager and the LTE-controller are visible/accessible in three different ways, i.e:

- Through an advanced GUI - relevant information of the REM and the LTE controller are available on a monitor. This allows an operating engineer to control the behavior of the CRM (the algorithms it deploys) and other parts of the system.
- Through a spectrum analyzer – measurements of the transmitted secondary signal's power spectrum. The instances of frequency handover are clearly traceable on its display and associated to simultaneous information in the LTE controller's GUI.
- Video streaming. A secondary receiver maintains an end-to-end IP-based connection with a network server. During the course of events, a video is streaming from this server over the LTE radio link to the prototype UE, where it is visible on a computer monitor. During the instances of spectrum handover the quality of this service is unaffected.

The above-described functional skeleton is cast in two concrete example set-ups of the system, i.e. a small-region scenario with real-time measurements and a large-region scenario with emulated measurements. In both set-ups, following the above chain of events, the environmental information stored in the REM is reported to the LTE controller, where it causes the CRM to instruct a TV band handover of the LTE-system.

5.1.1 A Small-Region Scenario with Real-Time Measurements

In the first set-up, the prototype illustrates how real-time sensing information, obtained through a sensing device in the immediate vicinity of the secondary transmitter, can be exploited. The purpose of this prototype constellation is to establish a proof-of-concept of the complete REM-architecture in a real-time environment. Operating in a "live" environment, the set-up contains all components of the system architecture including a dedicated MCD, relevant REM architectural components and REM User components in the form of the secondary LTE hardware. The MCD picks up the actual TV band characteristics reigning at the prototype's location in the 700 MHz band and the LTE system adapts its spectral occupation to this situation through the available measurements in the REM.

The spatial component of the REM is irrelevant for this purpose and with only one sensing device present the REM collapses to the trivial one-point map. The relevant measurements appear in the form of power spectrum measurements at the single location of the LTE transmitter.

Since the transmission characteristics of a typical TV network do not change very fast and other secondary transmitters are typically not present under today's regulatory regime, a separate transmitter (for instance in the form of a USRP2 device) can be used to establish a sudden appearance of another transmitter in the vicinity of the LTE transmitter.

At some point during the demonstration, and depending on the availability of a spectrum license, this USRP2 transmitter (not shown in Figure 5-1) starts transmitting in the particular TV channel operated by the LTE controller and thus invokes the chains of events described in the previous section, which eventually results in the evacuation of this TV channel by the LTE system.

5.1.2 A Large-Region Scenario with Emulated Measurements

In the second setup, the prototype system illustrates how real-life wide-area TV network characteristics are imported from the REM and incorporated in the secondary LTE system. The scale of the covered region (many square miles) prevents the use of real-time MCDs in the prototype set-up. Therefore, we exploit *pre-recorded field strength maps* in the REM, synthesized offline based on the available TV network characteristics (TV transmitter power, height, location, directivity etc.) for a set of predefined regions along with appropriate radio channel models.

The pre-recorded data in the REM reflects some sudden changes in the radio field, either as a result of TV operators carrying out deployment changes in the TV network or as a result of other secondary devices appearing or disappearing in the eNodeBs vicinity.

Whenever these sudden changes occur, the field strength in the LTE system's location changes and therefore the chains of events described in the previous section is invoked, which eventually results in the spectrum handover by the LTE system. Again, this handover is visible on the LTE controller's GUI and on a spectrum analyzer, while the service over the LTE link is unaffected.

5.2 Necessary Components and Connections to the REM Backend

The prototype and the corresponding demonstration has been carefully designed to enable experimentation with different TV network scenarios, both through the emulation of different countries' TV networks, as well as real-time operation. The actual prototype comprises two separate subsystems, i.e. an LTE part maintaining a video service in the UHF TV band comprising an LTE eNodeB and LTE UE (Figures 5-3 and 5-4) and a REM backend part that collects, infers and provides data and information about the radio field in the relevant TV bands in the neighborhood. Both subsystems are connected by a dedicated LTE controller, which interacts with the REM (obtaining relevant information), invokes a resource/spectrum management module (taking suitable spectrum decisions) and subsequently instructs the LTE base station to change its spectrum occupation. Any spectrum handover processes are transparent to the video link being maintained in the network.

Figure 5-3: LTE eNodeB hardware - baseband board (left), transmission board (middle) and RF module (right).

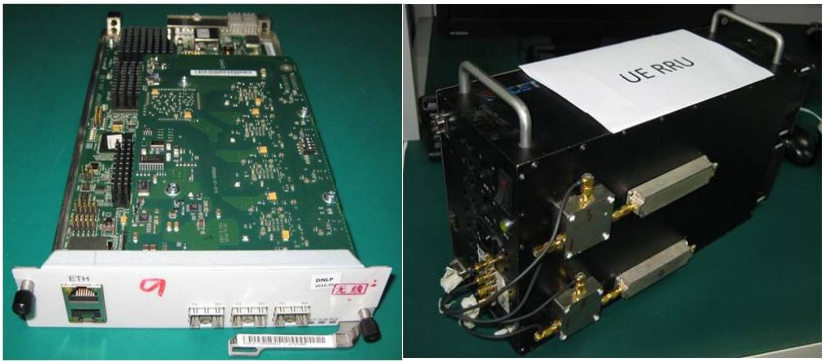


Figure 5-4: LTE UE hardware - baseband board (left) and RF module (right).

5.3 Prototype Functionalities

Beyond the REM backend hardware and software described in Chapter 2, the various hardware and software components of the secondary opportunistic LTE system in this prototype that provide an end-to-end system include:

- LTE eNodeB prototype, compliant with 3GPP's Release 9 of the LTE standard, comprising:
  - a transmission board
  - a baseband board for the LTE's 10 MHz TDD mode
  - an RF-module with two antennas for the upper UHF band, 686-790 MHz

  The RF module connects with an optical fiber cable to the baseband board.
- LTE UE prototype, compliant with 3GPP's Release 9 of the LTE standard, comprising:
  - a baseband board for LTE's 10 MHz TDD mode
  - an RF-module with two antennas for the upper UHF band, 686-790 MHz

  The RF module connects with an optical fiber cable to the baseband board.
- Virtualized core network software -module including a Mobility Management Entity (MME), Serving Gateway (SGW) and Packet-data-network Gateway (PGW).
- PC acting as a network server with video-on-demand support and
- PC for visualizing streaming video received by the UE.

Figure 5-5 shows how the above components are physically connected in the prototype setup, whereas Table 5-1 lists its main characteristics. The following subsections describe in more details the essential functionalities of the prototype system.

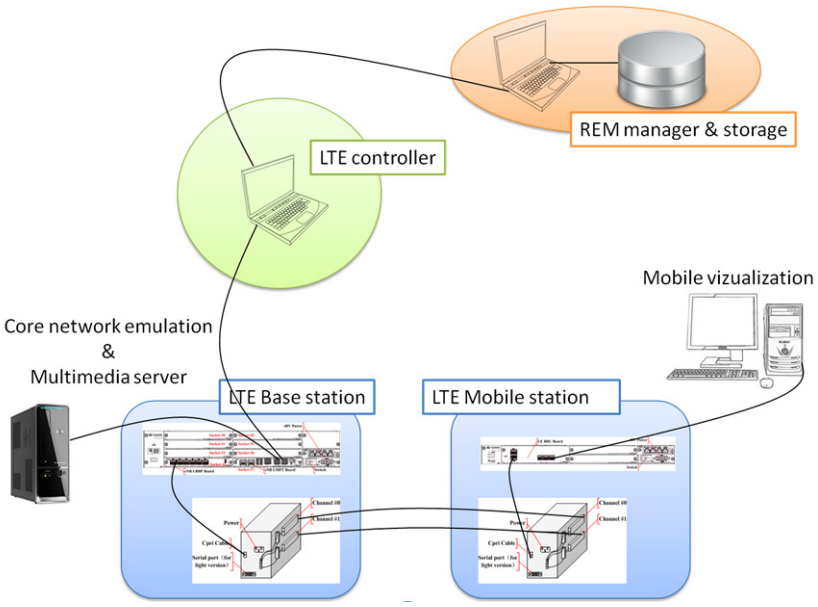


Figure 5-5: LTE system components and physical connections.

Table 5-1: LTE prototype system characteristics.

Standard compliancy	**LTE , release 9**
duplex mode	TDD, normal cyclic prefix
bandwidth mode	10 MHz (2 adjacent TV channels)
TDD mode	subframe configuration 2
MIMO mode	2x2 SFBC
eNodeB output power	30 mW (15 dBm)
TV channels	48-60
frequency range	686-790 MHz
frequency handover	handover during TDD quiet
frequency granularity	500 kHz
UL/DL PHY data rate	*ca.* 5/15 Mbps
DL end-to-end data	*ca.* 6 Mbps video streaming
core network	virtual, software emulated

5.3.1 Data Transmission Functionality of the Secondary LTE-TDD System

Upon switch-on, the TDD-LTE eNodeB establishes a connection with the core network and establishes a cell using a 10 MHz bandwidth segment in the upper UHF band. When this cell has been set up, the LTE UE can access it according to the LTE Release 9 specifications. Following the successful access procedure, an end-to-end IP connection is established. The UE can now access the network server over a transparent IP network and, for instance, tune to a video stream from this server. A PC attached to the UE baseband serves to control and visualize this connection.

The LTE connection follows the TDD mode of operation with a normal length of the cyclic prefix and a UL/DL subframe configuration 2 (with a 5 ms switching point periodicity), Figure 5-6. This frame structure essentially means that 64% of the time is dedicated to downlink traffic, 23% of the time is dedicated to uplink traffic and 13% of the time is used as a "silent switching period". This switching period is 0.64 ms (essentially 9 OFDM symbols have been blanked from the switching).

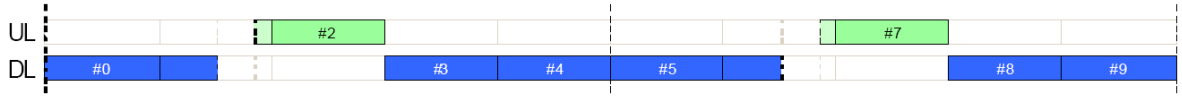


Figure 5-6: TDD subframe configuration 2. Each time slot of 1 ms is either dedicated to the uplink, to the downlink, or to both where a silent guared time separates the downlink and uplink portions

Both the eNodeB and the UE transmit with roughly 30mW (15dBm) in total over the two antennas. In the absence of an RF-license for the TV band, we deploy the LTE link over a set of RF cables along with a channel attenuation filter of about 75 dB. In this setup, we have measured raw data rates in the order of 15 Mbps in the downlink and 5 Mbps in the uplink, well in line with the partitioning of the time resources in the adopted TDD mode (it is also in line with the theoretical maximum throughput figures of roughly 18.5 Mbps and 6.6 Mbps[2]). With an RTP downlink streaming protocol, we can stream 6 Mbps video without noticeable loss of the service quality.

[2] This TDD mode has 9000 OFDM symbols per second in the downlink and 3200 OFDM symbols per second in the uplink. With each OFDM symbol consisting of 600 subcarriers modulated by 6 bits (64QAM), the maximum data rate is upper bounded by 9000x600x6=32.4 Mbps in the downlink and, similarly, 11.52 Mbps in the uplink. This data rate is then reduced because of coding (15/16), pilot

5.3.2 Spectral Agility of the Secondary LTE-TDD System

A critical feature of the TDD LTE prototype system is that it can perform a spectrum handover *during the 0.64 ms TDD switching period* described above. Specifically, the system maintains a built-in protocol where the eNodeB can decide to evacuate its operating frequency channel and move its cell to another frequency. The UE's connected to this eNodeB will join this spectrum handover which is seamless and transparent in the sense that ongoing traffic is not interrupted and does not experience a reduction of the service quality. To this end, the eNodeB instructs all the connected UE's to perform a spectrum handover to another frequency well ahead of the actual handover time instant. There is a proprietary protocol between the LTE baseband and the LTE RF front-end including these instructions (when to do the handover and to which frequency).

The spectrum handover is guaranteed to be finished within 2 seconds from the time of decision at the baseband. The actual handover is performed within a TDD switching period and hence does not affect the systems throughput negatively.

5.3.3 Functionalities of the LTE controller

The LTE controller is a software module, implemented in Python 2.7, which incorporates the following main functions.

- ***Provide a real-time interface with the REM backend***. The purpose of this function is to allow the LTE controller to acquire necessary information on which sensible spectrum management decisions can be based. Through a number of messages in a FARAMIR-developed protocol, the LTE controller can query for (and receive back):
    1. A power spectrum in a certain location or
    2. An entire radio field strength map for one specified frequency.

    Table 5-2 lists the detailed message structure used in this interface.
- ***Manage and control the spectrum usage*** of the secondary LTE system through the operation of proper CRM algorithms. In the current version of the LTE controller, the CRM allows the operation of three algorithms to manage and control the LTE system's spectral occupation:
    1. "***Stay at selected***". Complete control is in the hands of the human operator (through the GUI). No automated resource management algorithms are invoked.
    2. "***Stay at allowed***". The CRM compares the latest power spectrum received from the REM to a threshold value set by the human operator (through the GUI). Only when the average received power in the particular TV band operated by the LTE system exceeds this threshold (indicating a primary or other secondary transmitter is operating the same TV channel and close to the LTE eNodeB), the CRM instructs the LTE system to evacuate its TV channels. It will instruct the LTE system to move to the particular TV channel with smallest average received power as reported in the latest report from the REM.
    3. "***Jump to best***". The CRM periodically compares the latest average received power in all TV channels and instructs the LTE system to move to the particular TV

overhead (85%), control channels (80%) and H-ARQ retransmissions (90%). Therefore, maximum data rates are roughly 57% x 32.4= 18.5 Mbps in the downlink and 57% x 11.52= 6.6Mbps in the uplink.

channel with smallest average received power as reported in the latest report from the REM.

The basis of the above second and third spectrum managing approaches is the power spectrum in the location of the LTE eNodeB as reported by the REM. The radio field strength in other locations (the other kind of information provided by the REM, see above) is not used in the CRM algorithms. More advanced CRM algorithms may well use such information, especially in the case where other secondary *mobile* radio transmitters can be tracked in the REM and there future location can be predicted and anticipated on in an appropriate CRM algorithm.

- ***Provide a real-time interface with the LTE eNodeB baseband.*** The LTE controller maintains a simple proprietary interface with the eNodeB. In the current version of the LTE controller, this interface only supports a connection message along with a message instructing that a spectrum handover has to be carried out and to which frequency. The granularity of the frequency allocation is 500 kHz.
- ***Provide a (graphical) interface with an operator***. Figure 5-7 illustrates the LTE-controller's main GUI window. The left hand side shows the two kinds of information provided by the REM in separate figures. The bottom figure illustrates the power spectrum over all relevant TV channels in the 700 MHz band in the location of the LTE system's eNodeB, while the top figure shows the entire spatial characteristics of the radio field strength provided by the REM in one of the TV channels in the 700 MHz band. In each of these figures, the threshold value used by the CRM is indicated with a black line. In regions or channels where the field strength exceeds this threshold, secondary transmission is not allowed (also indicated with a red color in the bottom power spectrum plot). Those locations or channels where the field strength is smaller provide "white space opportunities" (and indicated with a green color in the bottom power spectrum plot). The yellow point in the upper figure and the yellow bar in the bottom figure indicate the location and TV channel currently operated by the LTE system.

Table 5-2: REM interface protocol messages.

message	**comment**	**Relevant fields**
CommDeviceRegReq	request to register to REM	deviceID
CommDeviceRegRsp	registration response	deviceID
		register
FSDreq	request for a FSD	xCoord, yCoord
		StartFreq, EndFreq
		NoPoints
		TimeInstant
FSDrsp	FSD response	FSDpoints
		TimeInstant
		Values
RemReq	request for a REM	ulPointX, ulPointY
		drPointX, drPointY
		startFreq, sendFreq
		xPoints, yPoints
		TimeInstant
RemRsp	REM repsonse	ulPointX, ulPointY

		drPointX, drPointY
		startFreq, sendFreq
		xPoints, yPoints
		TimeInstant
		REM
CommDeviceDeRegReq	request to de-register	deviceID

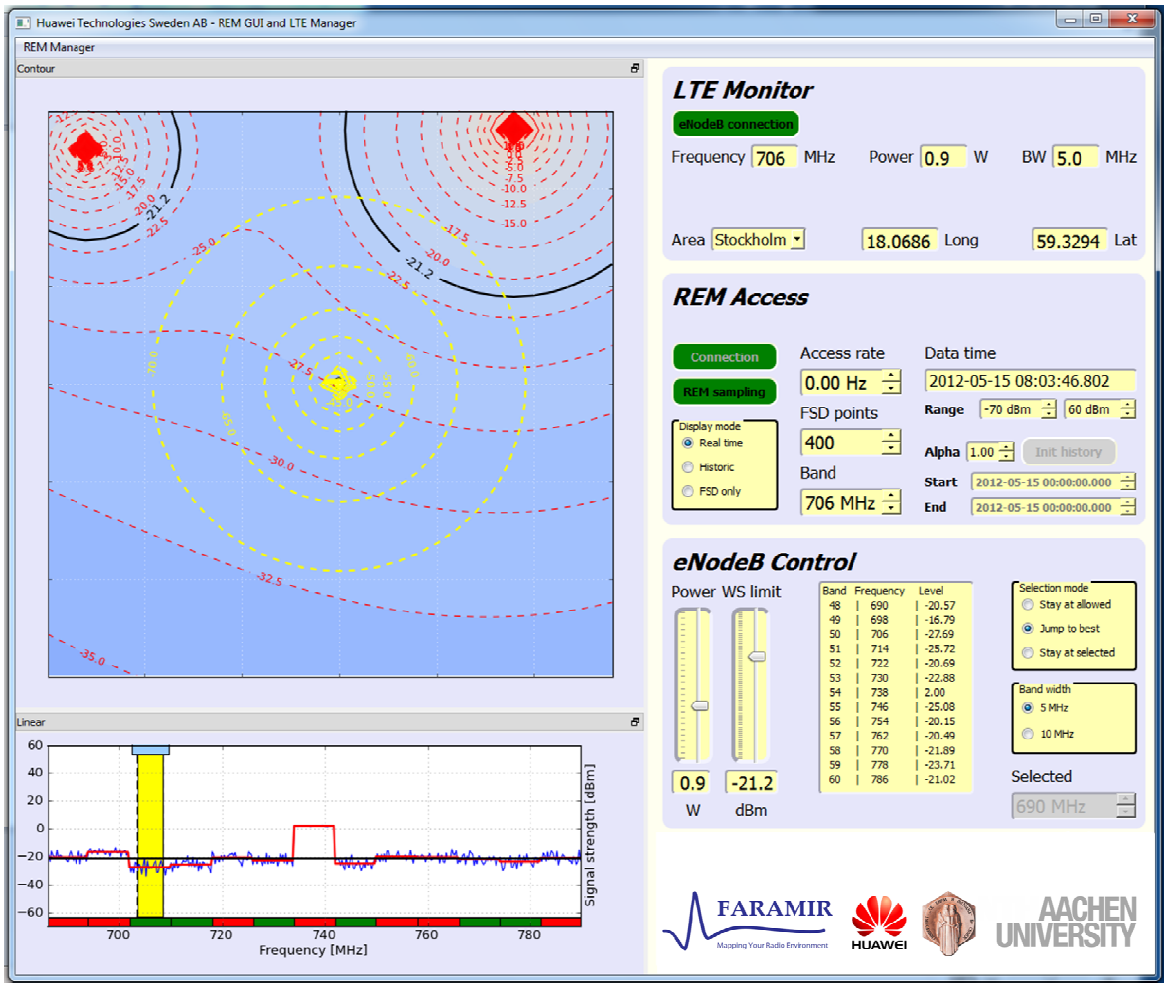


Figure 5-7: GUI of the LTE controller.

5.4 Discussion

The various interfaces of the prototype system have been tested extensively. Essentially all the above-described functionality has also been tested and has been confirmed to work properly.

Figure 5-8 and Figure 5-9 illustrate a typical operation of the prototype system. The top images in Figure 5-8 show the 10 MHz LTE carrier power as measured by a spectrum analyzer before and after a spectrum handover, while the bottom images show the associated snapshots of the LTE controller GUI. Note the change of the spatial characteristics of the radio field (two other transmitters have become active) that caused the LTE system to change its frequency.

During this process, a video is streaming, without interruption, from a network server through the virtual core network and the LTE eNodeB to the LTE UE and the attached PC. Figure 5-9 shows the monitor attached to this PC with the active streaming service.

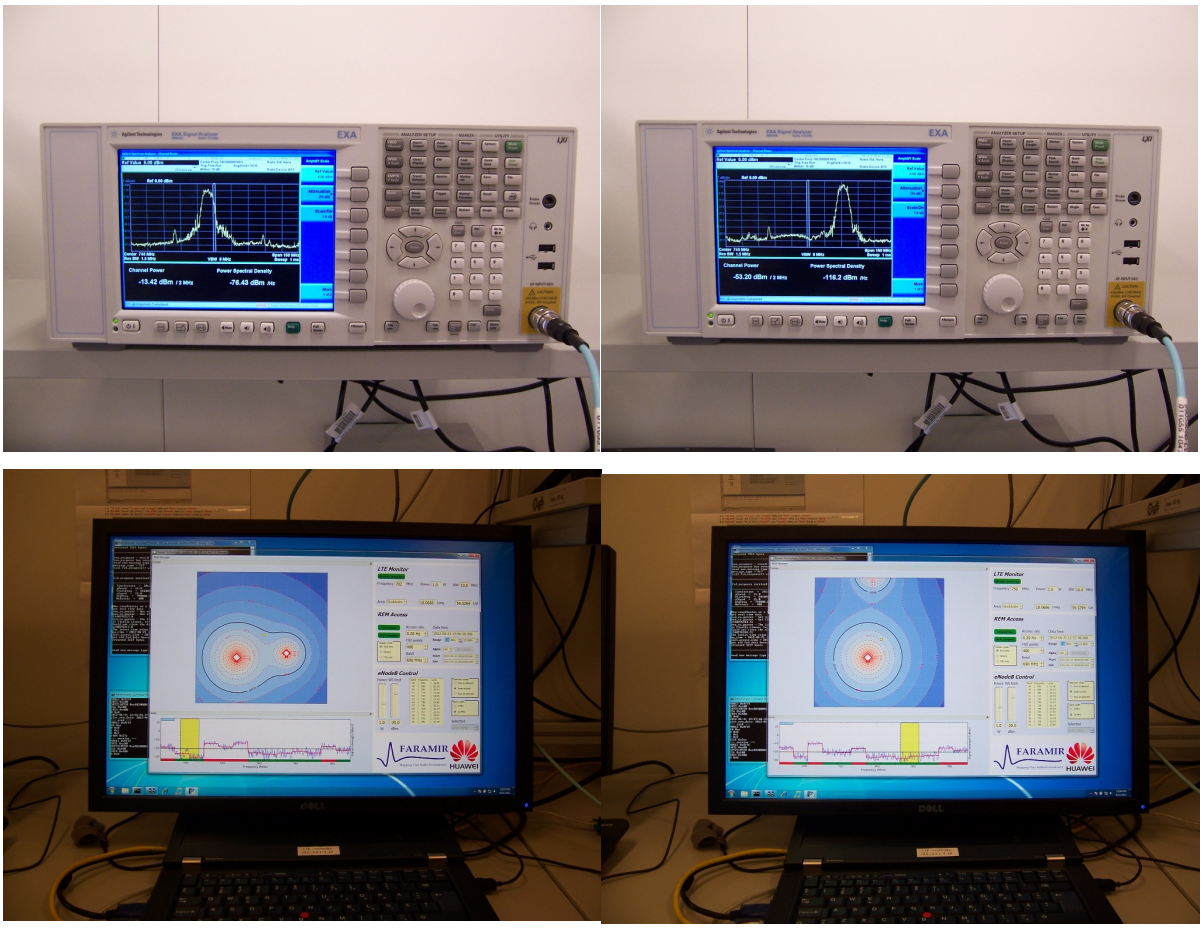


Figure 5-8: Illustration of the operation of the REM controller.

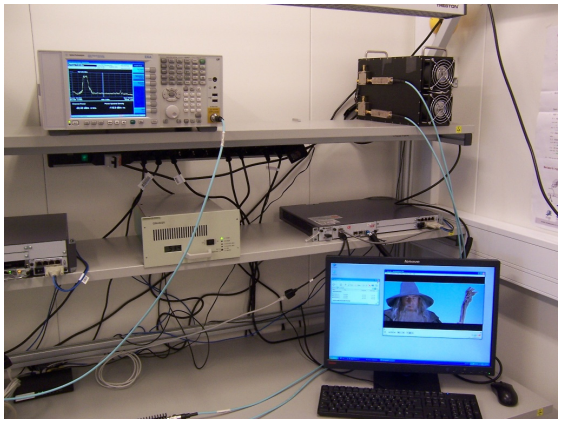


Figure 5-9: Illustration of the UE receiving streaming video during frequency handover operation.

In many ways the prototype described in this chapter constitutes a sample, alternatively a status report, of a development process that spans over a longer time. In this perspective, the following is a list of deliberate omissions and potential improvements of the LTE controller that we are planning to address in future versions.

1. The LTE system (the eNodeB) allows to monitor a number of key characteristic parameters, such as the center frequency, the bandwidth mode, the transmitted power etc. The current version of the LTE controller does not incorporate this monitoring functionality. The GUI has already a reserved pane for this in the top right, but in the current version the values are simple echoes from the control pane.
2. The LTE system (the eNodeB) allows for the control of its output power. The LTE controller does not exploit this yet although this is a typical functionality that can be of value in a secondary deployment of the system.

One aspect of the prototype system is known to have issues. The access of the UE into the cell operated by the eNodeB is unreliable. This is related to a known bug in this version of the prototype and will be addressed in a next version. Proper access of the UE (and a proper establishing of a secondary communication link) is therefore not guaranteed. Despite this instability, we have been able to extensively evaluate and test the spectrum handover capabilities when cell access was properly established. During these sessions, we have confirmed that the frequency handover does not affect the service quality and is indeed transparent to the LTE service. This represents a crucial proof-of-concept of the benefits and the potentials of the FARAMIR's REM technology in LTE usage in white space scenarios.

6 Advanced REM-Based RRM Prototype

The main objective of this prototype is to validate and exploit the dynamicity of the REM contents to improve the RRM algorithms in the context of cognitive radio. In particular, the prototype envisages the optimization of the spectrum selection policy when secondary users, with different service profiles, access opportunistically a set of channels during the inactivity periods of primary users. The selection will be done taking as input the statistical characterization of the primary user activity that will be stored in the REM and the secondary user service profiles.

This particular prototype does not rely on the usage of the REM backend technology developed and elaborated in chapter 2 per se. Rather than that, it completely follows the developed architecture and the associated functionalities of the REM backend in a part emulated part developed, standalone prototype.

6.1 Prototype Storyboard

Cognitive Radios (CRs) have been introduced as a mean to reuse the spectrum holes left unused by legacy (primary) services under the strict constraint of not interfering them. As a result of the opportunistic nature of this kind of radio access, the behaviour and perceived performance of the cognitive users (secondary network) depends on the spectrum occupancy pattern of the primary system. Thus, the availability of realistic and accurate, yet simple, spectrum occupancy models can provide significant benefits on the performances of the secondary network.

Certainly, focusing on the time perspective, some empirical measurements [39] have shown that, in addition to the expected daily/weekly periodicity, some degree of correlation between consecutive activity (ON) / inactivity (OFF) periods of the primary users behaviour exists [40], [41]. The amount of correlation depends on the band of interest and the considered traffic conditions. Then, the exploitation of the primary-user statistical patterns, capturing intra-channel dependence structures potentially exhibited by primary systems, could be used in order to improve the specific task of spectrum selection when secondary users, with different service profiles, access opportunistically a set of channels during the inactivity periods of primary users.

The aim of the envisaged prototype is to demonstrate and validate that the spectrum selection technique, which uses as input the statistical characterization of the primary user activity stored in the REM, performs much better, in terms of Spectrum Handover (SpHO) rate, than other spectrum selection techniques not using this information. Notice that that every SpHO will have a cost in terms of signalling requirements and service degradation in terms of latencies for the secondary users, then the target of the optimization (minimization) of the number of SpHOs of the secondary network clearly results in a better network performance perception from the viewpoint of the secondary users.

6.1.1 Considered Spectrum Selection Technique

The basic idea behind the envisaged spectrum selection is to choose the best sensed-as-free channel for secondary operation (according to a given criterion). As a basic principle, the knowledge retained in the REM about the primary user traffic will be used to estimate the remaining free-time for each of the sensed-as-free channels. Generally speaking, the primary user statistics stored in the REM can be first-order metrics such as means or conditional probabilities or higher-order metrics such as variances or correlation functions. The prototype elaborated in this chapter considers first order metrics.

Particularly, for each declared idle channel "i", in a given time, it is assumed that the REM Manager is keeping track of the duration of the last ON_i period as well as the so-far observed duration of the current OFF_i period ($Idle\_C_i$). The remaining OFF period (Rem_T_i) at a given time instant can be estimated by subtracting the so-far observed availability time ($Idle\_C_i$) from an estimation of the actual OFF period given the last observed ON period as follows:

$$\mathrm{Re}m\_T^i = E(OFF_i / ON_i) - Idle\_C^i \tag{6.1}$$

In case that no dependency is observed between consecutive ON/OFF periods the statistic reduces to $E(OFF_i / ON_i) = E(OFF_i)$.

Once the estimate of the sensed-as-free channels remaining free-time is available, the spectrum selection criterion takes into account the characteristics of the secondary service requesting a channel, in terms of its Mean Holding Time (MHT) and chooses a channel whose remaining time fits with the MHT of the requested service. That is, the criterion could be formulated as:

$$i^*_{selected} = \arg\min_i \left| \beta_i \times \mathrm{Re}m\_T^i - MHT_m \right| \tag{6.2}$$

where β_i is a compensation factor that should be adjusted for each channel based on the accuracy level in the estimation of OFF periods (Rem_T^i).

Finally, it is worth to mention here that, as a reference for comparison, a criterion consisting in a random selection among the idle channels will also be considered. Notice that this could be a reasonable criterion in case that there would not be a REM supporting the cognitive system and providing knowledge about primary users.

To compare the secondary system performances obtained using these two different spectrum selection criteria, the spectrum handover rate metric will be used. It is defined as:

$$SpHO\_rate = \frac{n^o\_spectrum\_HO}{T_r - T_s} \tag{6.3}$$

being T_r the current emulation time and T_s the starting emulation time.

In addition to that, other additional markers have been incorporated in the prototype to control the secondary user performances. These additional markers are: the blocking and dropping probability, which take into account in some way the load of the secondary network, as well as the spectrum handover probability that indirectly measures the trade-off between the secondary network load and the availability of sensed channels as free.

6.2 Necessary Components

6.2.1 Overview of the REM-Based RRM Prototype Implementation

The basic architecture of the advanced RRM REM-based prototype is shown in Figure 6-1, where we can identify the following entities:

1. A set of sniffing devices (Measurement Capable Devices or MCDs) sensing the received power level of the different RF channels in different sites of the demonstration area.

2. The REM_SA unit collecting the measurements obtained by the different sniffers. The connection between the sniffers and the REM_SA can be implemented, for instance, through an Internet connection.

3. For every RF channel considered in the demonstration, the REM manager computes the average value of the idle time. The set of these average values is the REM parameter to be transmitted to the emulated SCR System.

   Notice that in the envisaged prototype, the REM_SA and the REM manager are allocated in PCs #1 and #2, respectively.

4. The Secondary Cognitive Radio (SCR) System is a software-emulated part of the prototype, which runs together with the REM manager in PC#2. This emulated part aims at providing an accurate and realistic framework of the envisaged secondary system where the performance of the RRM algorithms and procedures could be fully assessed and evaluated. From the point of view of the emulation, this SCR system is composed by a set of secondary users, which generate traffic sessions following an exponential distribution statistic, characterized by the call arrival rate, i.e. the number of calls per minute and a secondary cognitive base station, which interacts with the secondary users.

   Due to the software nature of this emulated secondary networks, it is also necessary to send from the REM Manager to the emulated part the channel state of the involved RF channels. Notice that in a real system this information from the REM Manager would not be necessary, because the secondary devices would perceive themselves the channel state.

   Remark: The prototype assumes that the secondary network is deployed in a small area of the coverage area of the primary network. Therefore, all the secondary users as well as the secondary base station perceive the same state (idle or busy) of the different RF channels involved in the subsequent demonstration.

   In particular, a Matlab-based platform is used as an appropriate software instrument for developing and implementing this SCR system due to its simplicity, versatility, instrument control and communication capabilities and wide repertory of powerful mathematical and statistical functions as well.

5. The obtained results of the demonstration are shown in a specific GUI, indicated as PC#3 in Figure 6-1. The GUI provides an intuitive and user-friendly interface enabling the user to set-up specific emulation parameters, such as the location/area, spectrum band and time period, number of secondary users, traffic load etc., as well as to assess and validate the envisaged RRM algorithms. The GUI ensures that the emulation parameters and user's request are valid. The SCR system performances are basically provided as time evolution graphical representations. Particularly, the GUI shows:

   - Spectrum HO probability for both random and REM based algorithms;
   - Number of spectrum handovers per second for both random and REM based algorithms and
   - Dropping and blocking probabilities for both random and REM-based algorithms.

Communication among all elements of the prototype is accomplished by means of bidirectional point-to-point UDP connections.

Notice that the envisaged architecture of the advanced RRM REM-based prototype is fully compliant with the mandatory requirements of the generic FARAMIR REM backend prototype, identified in chapter 2 as well as with the communication protocols defined in [42],[43].

Finally, it is worth to mention here that the envisaged spectrum selection algorithm runs in parallel with a reference algorithm, based on a random selection among the idle channels, in order to compare the advantages provided by the REM information in the performances of the RRM algorithms[3].

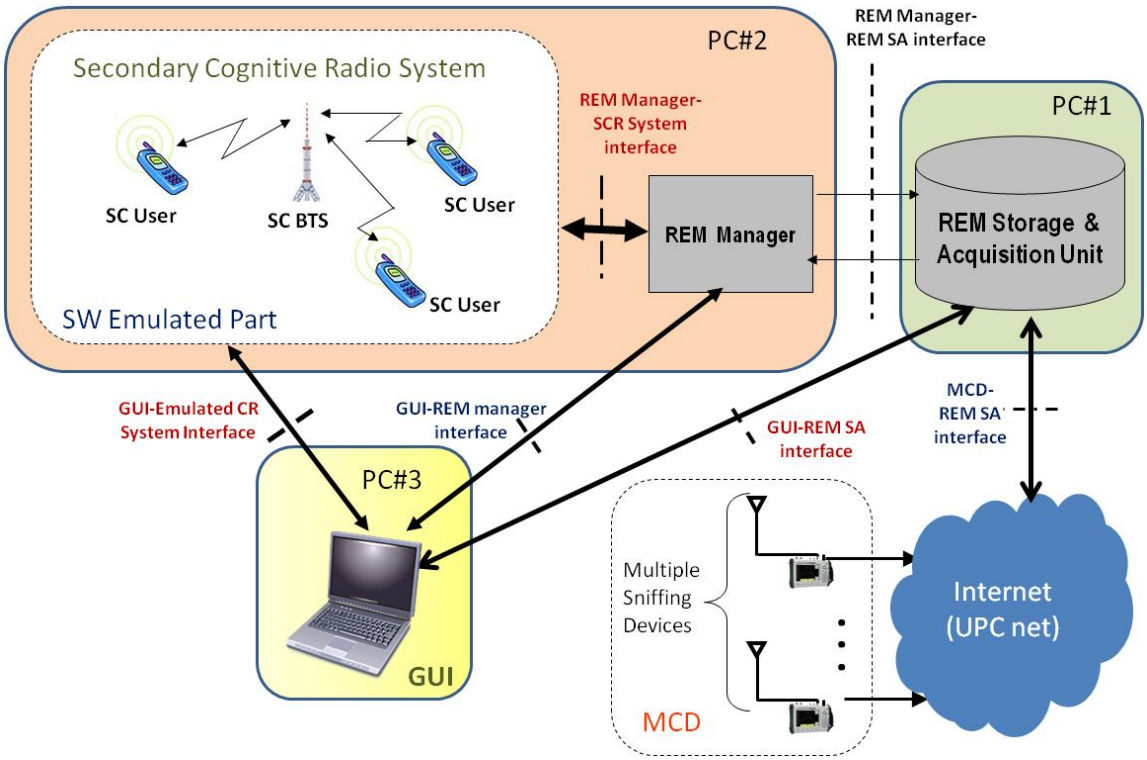


Figure 6-1: Advanced REM-Based RRM prototype architecture.

6.2.2 Implementation Issues

Each entity/element of the advanced RRM REM-based prototype architecture is implemented by means of a dedicated device. MCDs are based on Spectrum Analyzers (SPAs) providing raw spectrum data while the rest of the elements (i.e. REM SA, REM Manager, SCR system and GUI) are implemented by means of three dedicated computers running a tailor-made software script under Matlab. All the elements of the prototype (SPAs and PCs) are connected to Universitat Politecnica de Catalunya (UPC) intra-network by means of regular Ethernet connections (see Figure 6-2). Alternatively, for portable demonstration purposes, the elements of the prototype could be interconnected by means of a switch. Communication among the elements of the prototype is accomplished by means of bidirectional point-to-point UDP connections.

The implemented prototype has been envisaged and designed to include several MCDs. In the current implementation, there is one MCD (i.e. one SPA) since in practice one single MCD is

[3]Both algorithms receive the same information about the channel state (Idle or busy) of the involved RF channels and user behavior, but the random algorithm does not receive the REM information (average value of the idle time of the available RF channels)

enough for most practical demonstrations. All software modules are implemented in Matlab and run under Windows.

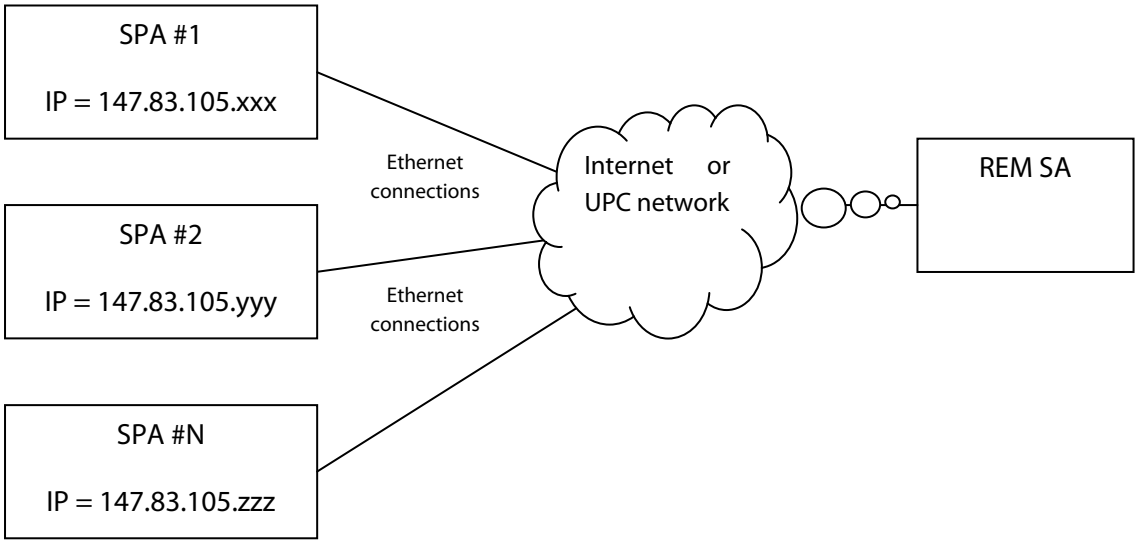


Figure 6-2: RRM prototype hardware architecture.

6.2.3 Communication Interfaces among the Prototype Elements

Communication among the elements of the prototype is accomplished by means of bidirectional point-to-point UDP connections established during the start up phase (i.e. when the prototype modules are switched on) between each pair of entities needing to communicate. UDP connections are unidirectional meaning that a bidirectional communication between two elements A and B requires the establishment and configuration of two UDP connections in the start-up phase, one connection A→B and another connection B→A. UDP connections are established to the IP addresses specified in the `IP_ADDRESSES.txt` file with arbitrary UDP port numbers that ensure the avoidance of conflicts between UDP ports at different connections. The following shows an example of the file's contents:

```
DREM_GUI=147.83.105.194
REM_MANAGER=147.83.105.195
REM_SA=147.83.105.196
```

For each entity (i.e. PC running the corresponding entity) it is necessary to specify the PC's IP address so the appropriate UDP connection can be established and configured in order to reach each element of the prototype. These IP addresses must correspond to the real IP addresses of the PCs and it is important to check that they can be reached and that the UDP port numbers employed in the code are not blocked by the network or by any of the software firewalls that might be running in each PC. The MCD/SPA also needs an IP address to be reached, which need to be configured in the SPA (refer to the SPA user's guide, [44], to obtain more information on how to do this). Note that the SPA's IP address does not need to be specified in this file since this is provided as an input parameter that can be configured by the user in the GUI before starting the prototype's execution.

The communication among the elements of the testbed is accomplished by means of UDP connections except in the case of the communication link between the REM SA and the MCD/SPA,

which is based on a VISA/TCP-IP connection[4]. Note that the latter requires that the Matlab's Instrument Control Toolbox is installed in the REM SA computer along with an appropriate VISA driver, which is not included with Matlab. The National Instruments NI-VISA driver v4.4.1 is currently employed [45].

6.2.3.1 MCD – REM SA Communication Interface

MCDs are controlled remotely from the REM SA computer, which commands the SPAs to initiate a new sweep, monitors the SPAs' states and retrieves the measured data every time a new sweep is completed. The data retrieved by the REM SA from the MCDs is stored in a structure called SPA_data. The structure contains the same number of elements as the number of MCDs/SPAs specified. Each element contains 2 matrices, one called 'power' and other called 'time', which store the amplitudes and timestamps for each MCD/SPA as illustrated in the figure below.

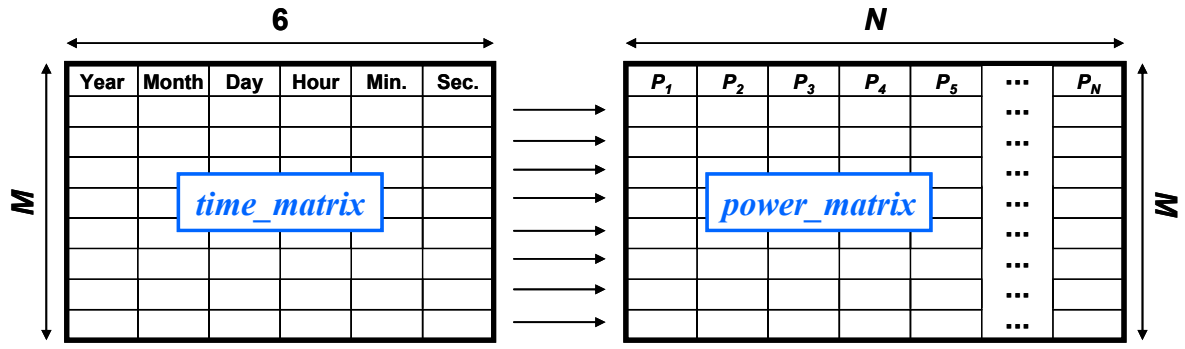


Figure 6-3: Data formats.

Each row corresponds to one sweep of the SPAs. Both matrices can store data for the last *M* sweeps performed by the SPAs. Every time a new sweep is completed, the oldest sweep (row) in each matrix is removed and the last sweep is added to the matrices. The power matrix contains *N* columns, one for each channel within the measured spectrum band, while the time matrix contains 6 columns to store time stamps in YYYY-MM-DD-HH-MM-SS format. The *m*-th row of the time matrix contains the time stamp for the sweep of the *m*-th row stored in the power matrix. The power matrix can be configured to store the absolute power values measured in each channel in dBm or, alternatively, the binary busy/idle states of the channels (derived from the power measurements by a means of a power thresholding technique).

The power and time matrices in the REM SA are periodically updated and passed to the REM Manager, which makes use of these spectrum data to compute the REM. The REM data is passed to a network/algorithm emulator (also implemented in the REM Manager PC) that, based on the REM data, performs some simulation and returns the simulation result. The simulation result is then sent to the GUI PC, which displays the results to the user and updates the simulation results periodically.

The communication interface between the MCD/SPA and the REM SA computer is based on a VISA/TCP-IP connection. The establishment of this communication link is fully specified in the application note by Anritsu [46]. All the communication is accomplished by means of commands in SCPI format. Detailed information on the set of commands available in the employed SPA (Anritsu MS2721B) can be found in [47].

[4] The VISA (Virtual Instrument Software Architecture)library allows connecting test & measurement equipment via various hardware interfaces.

6.2.3.2 REM SA – REM Manager Communication interface

The REM SA provides periodically, after every SPA's sweep, the captured power and time data to the REM manager, which stores the received data in appropriate matrices following the same format specified above.

The REM SA may send two different message types to the REM manager, a time message and a power message. The time message is a string starting with the character 'T' (which identifies the message) followed by six numerical values (in string format) indicating the year, month, day, hour, minute and second of the sweep. The power message is sent immediately after the time message and is composed of a string starting with the character 'P' (which identifies the message) followed by 551 numerical values (in string format) indicating the 551 power values (in dBm) measured in the last SPA's sweep.

6.2.3.3 Communication interface GUI – REM SA

The DREM GUI module provides a graphical interface to the prototype's user in order to configure all the relevant aspects (parameters) of the prototype, control the prototype execution (connection with MCD/SPA, start and stop) and display the statistical results of the prototype's real-time emulation. The different message types that may be sent from the DREM GUI module to the REM SA are as follows:

Connect to MCDs ('C'). Indicates to the REM SA that the user has requested the establishment of the corresponding communication links with the MCDs/SPAs. The REM SA module then initializes the communication and sends the list of configuration parameters to the MCD/SPA. The employed configured parameters have been selected based on extensive methodological studies and provide an appropriate configuration that leads to reliable results under a wide range of frequency bands. Since these configuration parameters are general enough and no modification is expected, they have been hardcoded in the source code. In case that any parameter needs to be changed, this can be easily done in the source code.

Change frequency band ('B'). Indicates to the REM SA that the user has requested the modification of the operating frequency band. The received string command is in the format 'BXXX-YYY', where XXX indicates the start frequency of the new band and YYY indicates the stop frequency of the new band. Based on the received string command, the REM SA module extracts the frequency range to be measured and sends appropriate commands to the MCDs/SPAs in order to switch to the new start and stop frequencies.

Start testbed execution ('A'). Indicates to the REM SA module that the user has pressed the start button and therefore that the prototype, and thus the other modules, are running. This causes the module to switch to the 'running' state, in which the module performs the required set of operations during the normal prototype execution.

Stop testbed execution ('O'). Indicates to the REM SA module that the user has pressed the stop button and therefore that the prototype, and thus the other modules, have been halted. This causes the module to switch to the 'stop' state, in which the module does not perform any operation and the internal configuration parameters can be reconfigured from the REM GUI interface.

Disconnect from MCDs ('D'). Indicates to the REM SA that the user has requested the termination of the corresponding communication links established with the MCDs/SPAs. The REM SA module then terminates the established communication links with the MCDs/SPAs and releases the employed computer resources.

Finish execution ('X'). Indicates to the REM SA that the user has closed the GUI window. This command stops the REM SA module. For a new execution of the prototype, the REM SA module needs to be launched again.

6.2.3.4 GUI – REM Manager Communication Interface

The GUI module provides a graphical interface to the user for configuration of all relevant aspects (parameters) of the prototype, control the prototype execution (connection with MCD/SPA, start and stop) and display the statistical results of the prototype's real-time emulation. The different message types that may be sent from the DREM GUI module to the REM manager are:

Change frequency band ('B'). Indicates to the REM manager that the user has requested the modification of the operating frequency band. The received string command is in the format 'BXYYY-ZZZ', where X identifies the specified MCD/SPA, YYY indicates the start frequency of the new band and ZZZ indicates the stop frequency of the new band. Based on the received string command, the REM manager module extracts the new frequency range being measured by the REM SA and MCD modules and takes it into account when performing computations and displaying results. The new frequency band is applied in any subsequent computation until a new modification is requested.

Change channelization ('C'). Indicates to the REM manager that the user has requested the modification of the channelization. The received string command is in the format 'CXXXX' where XXXX indicates the new channelization (i.e. radio frequency bandwidth of the measured signals), in kHz, to be applied in the subsequent computations and displayed results. The new channelization is applied in any subsequent computation until a new modification is requested.

Change decision threshold ('T'). Indicates to the REM manager module that the user has requested the modification of the power decision threshold to be applied in the processing of the power data provided by the REM SA module. Measured power values are compared to this threshold in order to determine which channels are busy/idle. The received string command is in the format 'TXXX' where XXX indicates the power decision threshold specified in dBm. The new decision threshold is applied in any subsequent computation until a new modification is requested.

Change number of CR users ('N'). Indicates to the REM manager module that the user has requested the modification of the number of DSA/CR users to be emulated. The received string command is in the format 'NXXX' where XXX indicates the number of DSA/CR users. The new number of DSA/CR users is applied in any subsequent computation until a new modification is requested.

Change CR user arrival rate ('R'). Indicates to the REM manager module that the user has requested the modification of the call arrival rate for the DSA/CR users emulated in the prototype. The received string command is in the format 'RXXX' where XXX indicates the new call arrival rate (λ) in number of calls per minute. The new arrival rate is applied in any subsequent computation until a new modification is requested.

Change CR mean holding time ('H'). Indicates to the REM manager module that the user has requested the modification of the mean holding time for the DSA/CR users emulated in the prototype. The received string command is in the format 'HXXX' where XXX indicates the new mean holding time ($1/\mu$) in number of seconds per call. The new mean holding time is applied in any subsequent computation until a new modification is requested.

Start prototype execution ('A'). Indicates to the REM manager module that the user has pressed the start button and therefore that the prototype, and thus the other modules, are running. This causes the module to switch to the 'running' state, in which the module performs the required set of operations during the normal prototype execution.

Stop prototype execution ('O'). Indicates to the REM manager module that the user has pressed the stop button and therefore that the prototype, and thus the other modules, have been halted. This causes the module to switch to the 'stop' state, in which the module does not perform any operation and the internal configuration parameters can be reconfigured from the REM GUI interface.

Finish execution ('X'). Indicates to the REM manager that the user has closed the DREM GUI window. This command stops the REM manager module. For a new execution of the prototype, the REM manager module needs to be launched again.

6.2.3.5 Communication Interface between Emulated CR System – REM GUI Statistics

The REM GUI statistics module is a software module that is run in the GUI computer and in parallel with the REM GUI module. The purpose of this module is to receive the statistics and results from the simulation performed in the secondary cognitive network emulation module and display such results in a separate window. The results are received periodically, stored in the corresponding matrices and employed to update the graphs, which display not only the most recent value of the display parameters, but also, in some cases where it is appropriate, the history thereof. The different message types that may be sent are:

RANDOM blocking probability ('B'). Sends to the DREM GUI statistics module the last value computed for the system's blocking probability when the RANDOM algorithm is employed. The received string is in the format 'BXXX' where XXX indicates the RANDOM blocking probability, computed as:

RANDOM_nof_blockings_counter/RANDOM_nof_blockings_total

REM blocking probability ('C'). Sends to the DREM GUI statistics module the last value computed for the system's blocking probability when the REM algorithm is employed. The received string is in the format 'CXXX' where XXX indicates the REM blocking probability, computed as:

REM_nof_blockings_counter/REM_nof_blockings_total

RANDOM dropping probability ('D'). Sends to the DREM GUI statistics module the last value computed for the system's dropping probability when the RANDOM algorithm is employed. The received string is in the format 'DXXX' where XXX indicates the RANDOM dropping probability, computed as:

RANDOM_nof_droppings_counter/RANDOM_nof_droppings_total

REM dropping probability ('E'). Sends to the DREM GUI statistics module the last value computed for the system's dropping probability when the REM algorithm is employed. The received string is in the format 'EXXX' where XXX indicates the REM dropping probability, computed as:

REM_nof_droppings_counter/REM_nof_droppings_total

RANDOM SpHo probability ('H'). Sends to the DREM GUI statistics module the last value computed for the system's SpHo probability when the RANDOM algorithm is employed. The received string is in the format 'HXXX' where XXX indicates the RANDOM SpHo probability, computed as:

RANDOM_nof_SpHo_counter/RANDOM_nof_SpHo_total

REM SpHo probability ('I'). Sends to the DREM GUI statistics module the last value computed for the system's SpHo probability when the REM algorithm is employed. The received string is in the format 'IXXX' where XXX indicates the REM SpHo probability, computed as:

REM_nof_SpHo_counter/REM_nof_SpHo_total

RANDOM SpHo rate ('P').Sends to the DREM GUI statistics module the last value computed for the system's SpHo rate when the RANDOM algorithm is employed. The received string is in the format 'PXXX' where XXX indicates the RANDOM SpHo rate in terms of the number of SpHo per second, computed as:

RANDOM_nof_SpHo_counter/(simulation_current_time-simulation_start_time)

REM SpHo rate ('Q'). Sends to the DREM GUI statistics module the last value computed for the system's SpHo rate when the REM algorithm is employed. The received string is in the format 'QXXX' where XXX indicates the REM SpHo rate in terms of the number of SpHo per second, computed as:

REM_nof_SpHo_counter/(simulation_current_time - simulation_start_time)

RANDOM channel for a random CR user ('Y'). Sends to the DREM GUI statistics module the current channel assignment for a random CR user (the same for the whole emulation) when the RANDOM algorithm is employed. The received string is in the format 'YXXX' where XXX indicates the current channel allocated to the CR user:

secondary_users_channel_RANDOM(1)

REM channel for a random CR user ('Z'). Sends to the DREM GUI statistics module the current channel assignment for a random CR user (the same for the whole simulation) when the REM algorithm is employed. The received string is in the format 'ZXXX' where XXX indicates the current channel allocated to the CR user:

secondary_users_channel_REM(1)

Current channel states ('S'). Sends to the DREM GUI statistics module the current state for each primary channel. The received string is in the format 'SXXX' where XXX is a string where each character denotes the state for the corresponding channel, which may be zero (the primary user is not occupying the channel – idle channel) or one (the primary user is occupying the channel – busy channel).

Current user states ('U'). Sends to the DREM GUI statistics module the current state for the traffic of each secondary user. The received string is in the format 'UXXX' where XXX is a string where each character denotes the state for the corresponding user, which may be zero (the secondary user has no data to transmit – inactive session) or one (the secondary user has some data to transmit – active session).

Mean OFF periods ('M'). Sends to the DREM GUI statistics module the current value of the average length of idle periods for each primary channel. The received string is in the format 'MXXX' where XXX is an element string where each element indicates the mean length of the idle periods (in seconds) for the corresponding channel.

Current OFF periods ('T'). Sends to the DREM GUI statistics module the current value of the time period for which each primary channel has been idle up to the reporting moment. The received string is in the format 'TXXX' where XXX is an element string where each element indicates the

length of the current idle periods (in seconds) for the corresponding channel. A value equal to zero means that the channel is currently busy.

Start prototype execution ('A'). Indicates to the DREM GUI statistics module that the user has pressed the start button and therefore that the prototype, and thus the other modules, are running. This causes the module to initialize all the internal variables and data structures employed to store the statistic results sent by the REM manager module.

Stop prototype execution ('O'). Indicates to the DREM GUI statistics module that the user has pressed the stop button and therefore that the prototype, and thus the other modules, have been halted. This causes the module to close the separate figure window where the statistical results are shown.

Finish execution ('X'). Indicates to the DREM GUI statistics module that the user has closed the DREM GUI window. This command closes the separate figure window where the statistical results are shown and stops the DREM GUI statistics module. For a new execution of the prototype, the DREM GUI statistics module needs to be re-launched.

6.2.4 SCR System Emulator

The aim of the SCR system emulator is to emulate in real time the behavior of the algorithm presented in section 6.1. This emulator is integrated in the same PC as the REM manager. The emulator is updated after every sweep received by the REM manager from the REM SA module and the corresponding statistics are updated as well and then reported to the DREM GUI statistics module making use of the aforementioned interface messages. The algorithm's internal variables are detailed below.

6.2.4.1 Variables Associated to the Activity of the Primary System

- nof_channels_in_band: Number of RF channels within the measured band, which is computed based on the frequency range and channelization specified by the user in the REM GUI.
- current_channel_states: Vector that stores the current state for each of the channels within the measured spectrum band. Zero means idle, one means busy channel.
- already_off_time: Vector storing, for each of the channels within the measured spectrum band, the time (in seconds) that the channel has been in the idle state since it was released the last time. A value equal to zero means that the channel is currently in the busy state.
- mean_OFF_periods: Vector storing, for each of the channels within the measured spectrum band, the time (in seconds) that the channel has been in the idle state since the simulation started. For each channel, this vector stores the average value of the idle periods, which is updated when an idle period disappears (i.e. when the channel becomes busy after a period of inactivity) by means of a recursive FIR filter:

$$mean\_OFF\_periods(x) = \alpha * mean\_OFF\_periods(x-1) + (1-\alpha) * already\_off\_time(x);$$

  where the new mean value mean_OFF_periods(x)is updated[5] based on the previous mean value mean_OFF_periods(x-1) and the duration of the most recent idle period

[5]**Note:** *The vector mean_OFF_periods is initialized to zero. This means that for channels that have always been idle in the emulation, the corresponding values within the vector are never updated and this affects the channel selection procedure meaning that these channels are never selected. To avoid this problem, when making the channel selection decision, the*

already_off_time(x), making use of a weighting factor α, which is currently configured as $\alpha = 0.7$

- last_time_stamp**:** Stores the time stamp when the last sweep was received. This time stamp is the time information provided by the REM SA module and is used to keep track of the primary spectrum activity and compute the length of idle periods as they appear/disappear. The time stamp is a vector with six numerical values indicating the year, month, day, hour, minute and second of the time stamp.

6.2.4.2 Variables Associated to the Traffic of the Secondary System

- nof_users: Number of DSA/CR users in the system, as specified by the prototype user in the REM GUI window.
- user_arrival_rate_calls_min: Secondary call arrival rate, in number of calls per minute, as specified by the prototype user in the REM GUI window.
- MHT_sec_call: MHT of the DSA/CR users' calls, in number of seconds per call, as specified by the prototype user in the REM GUI window.

6.2.4.3 Variables Associated to the Internal States of the Secondary Users

- secondary_users_state: Vector storing, for each DSA/CR user, the state of its internal traffic source. Zero means that the user has no data to transmit, while one means that the DSA/CR has some data to transmit.
- secondary_users_next_state_change: Vector including, for each DSA/CR user, a numerical value that represents a time stamp in seconds (see help of the clock function in Matlab), which points to the time instant when the state of internal traffic source for every DSA/CR user needs to be changed.
- secondary_users_channel_RANDOM: Vector including, for each DSA/CR user, the currently allocated channel when the RANDOM algorithm is employed. A value equal to 0 indicates that the CR user has no channel allocated because there is no data to transmit. A value equal to –1 indicates that the CR user has no channel allocated because the CR user has been blocked or dropped. Another integer value, greater than zero, represents the number of the channel currently allocated to every user. In this case, the numerical values are in the interval [1, nof_channels_in_band].
- secondary_users_channel_REM: Vector including, for each DSA/CR user, the currently allocated channel when the REM algorithm is employed. A value equal to 0 indicates that the CR user has no channel allocated because there is no data to transmit. A value equal to –1 indicates that the CR user has no channel allocated because the CR user has been blocked or dropped. Another integer value, greater than zero, represents the number of the channel currently allocated to every user. In this case, the numerical values are in the interval [1, nof_channels_in_band].

6.2.4.4 Flowchart of the Internal States for a CR user

Figure 6-4 shows the flowchart of the possible internal states for each CR user and the possible sequences of states depending on the system's conditions. The values of relevant internal status for each state are also shown.

---

vector is copied into another variable and in the copied vector the values equal to zero are replaced by a large number (100000) in order to represent an infinite value.

The first state corresponds to the case where the CR user has no data (session is inactive) and therefore has no channel allocated. When a new session starts, the user has data to transmit, but no channel is allocated to this user. If there are no free channels at that moment, the user is blocked and remains in this state until the session finishes (the CR user waits for the next session). However, if there are free channels (i.e. channels not used by primary or secondary CR users), the channel selection algorithm (RANDOM or REM-based) is executed and a free channel is then selected for the user. The CR user makes use of that channel as long as the primary system does not claim it again. When a primary user reappears in a channel being used by a CR user, then the CR user is moved to another channel if there are free channels (the channel selection algorithm is executed again and the counter of spectrum handovers is updated) or dropped if there are no free channels and remains in this state until the session finishes (the CR user waits for the next session). When the CR user session finishes and there is no more data to transmit, the allocated channel is deleted and then the CR user waits for the next data session.

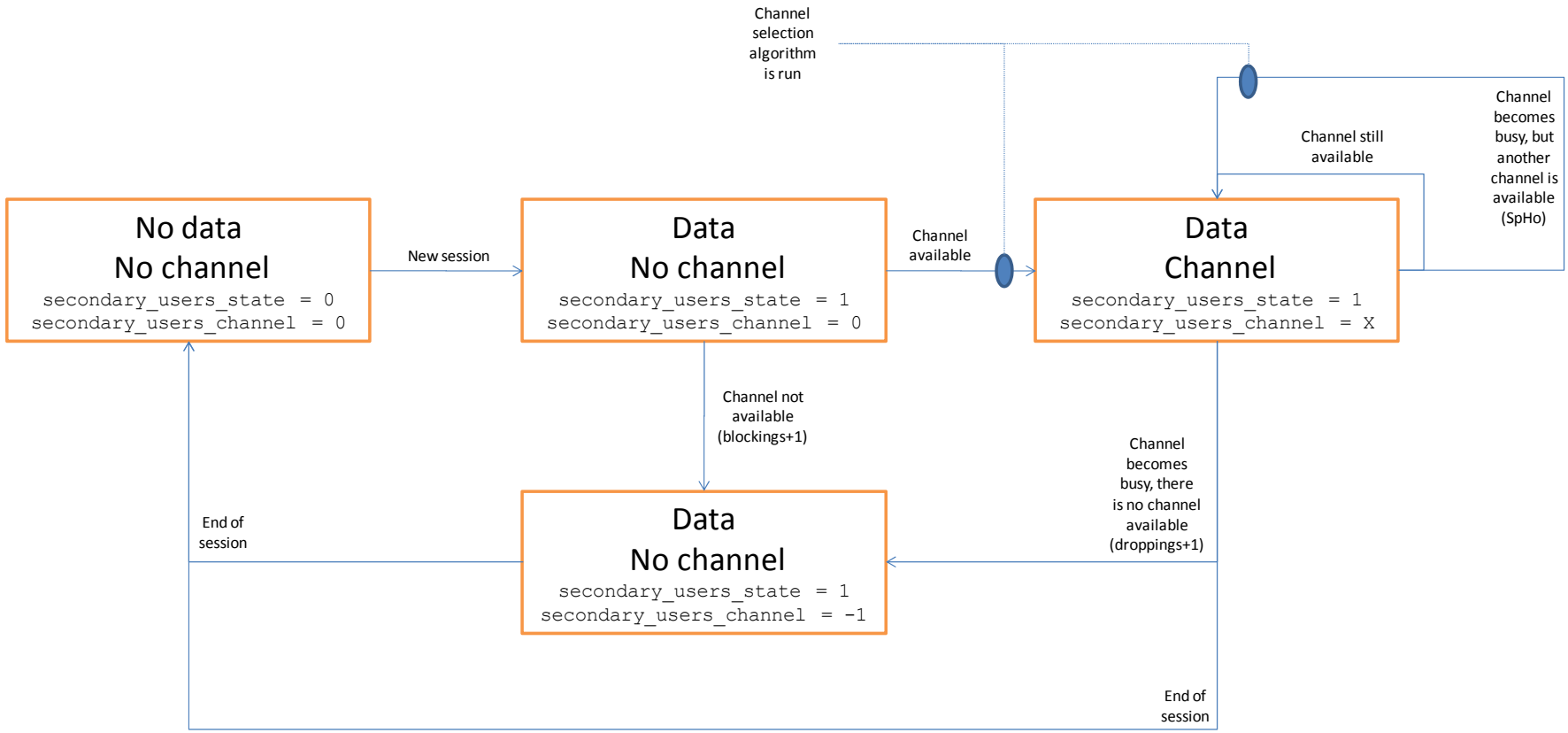


Figure 6-4: Flowchart of internal states for a CR user.

6.3 Prototype Functionalities: User Guide

Once the prototype is running, a GUI window appears allowing configuration and control of the execution of the prototype (Figure 6-5). This subsection explains (in a user guide manner) the usage of the GUI and the manipulation with the prototype.

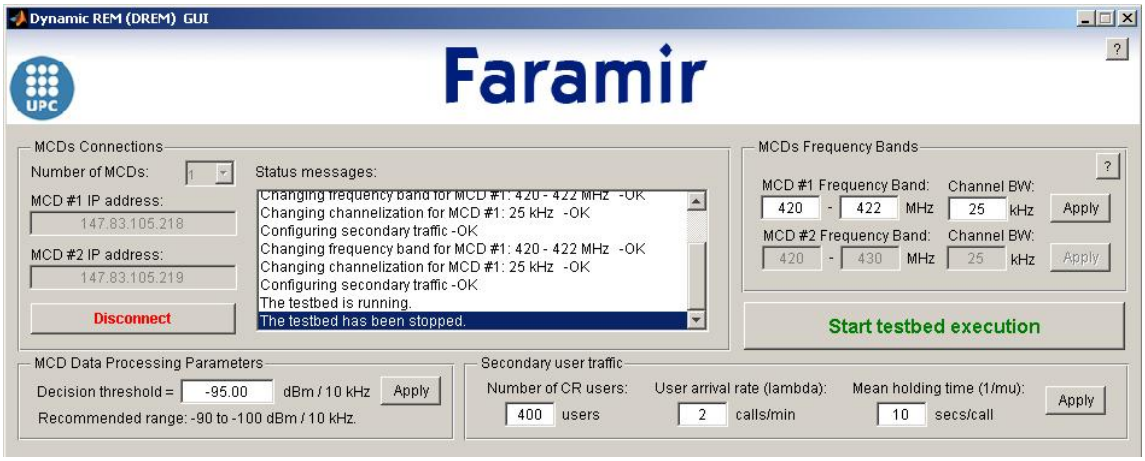


Figure 6-5: Screenshot of the "Configuration" window shown in the GUI computer

The following prototype parameters must be set-up in this GUI window:

1. In order to run the prototype, the "MCD IP address" must be specified (this IP address must be also configured in the corresponding MCD/SPA).Then, press the "Connect" button and the prototype will try to connect with the MCD/SPA (some messages will appear in the "Status messages" textbox). When the connection is established, then the rest of controls are enabled.
2. Next, the "MCD Frequency band" and the "Channel BW" (i.e. channel radio frequency bandwidth or channelization for that band) are selected. Once "Apply" button is pressed then the MCD devices will be properly configured and several confirmation messages will appear in the "Status messages" window.
3. The power "Decision threshold" (in dBm/10 kHz) to apply to the power measurements provided by the MCD/SPA (recommended values are from –90 dBm / 10 kHz to –100 dBm / 10 kHz) is specified. Once the "Apply" button is pressed, several messages will appear in the "Status Messages" window confirming that these parameters have been properly configured.
4. As final configuration step, the "Secondary user traffic" parameters (i.e. "Number of CR users", "User arrival rate (lambda)" and "Mean holding time (1/mu)") are configured, and settings taken into use with the "Apply" button. Once again, some messages will appear in the "Status Messages" window confirming that these parameters have also been properly configured.

With all the parameters configured, pressing the "Start testbed execution" will initiate the prototype functionality. The window showing the time evolution of different parameters of the emulated "Secondary Cognitive Network" (see Figure 6-8) will appear in the GUI computer along with other windows in the REM SA (see Figure 6-6) and REM MANAGER computers (see Figure 6-7).

The REM SA computer (Figure 6-6) show the history of the last 200 sweeps performed by the MCD/SPA and the corresponding powers measured at each point (in dBm/10 kHz) in the format of a waterfall plot. The REM MANAGER computer (Figure 6-7) shows the result of processing the same data with the power decision threshold specified in the DREM GUI window by the user. The black color denotes busy samples while the white color denotes idle samples. The parameters shown in all windows will update periodically upon the arrival of a new sweep from the MCD/SPA.

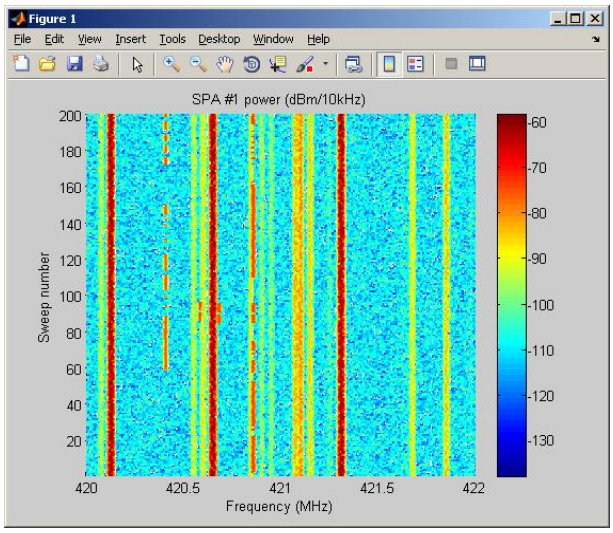


Figure 6-6: Screenshot of the REM SA computer

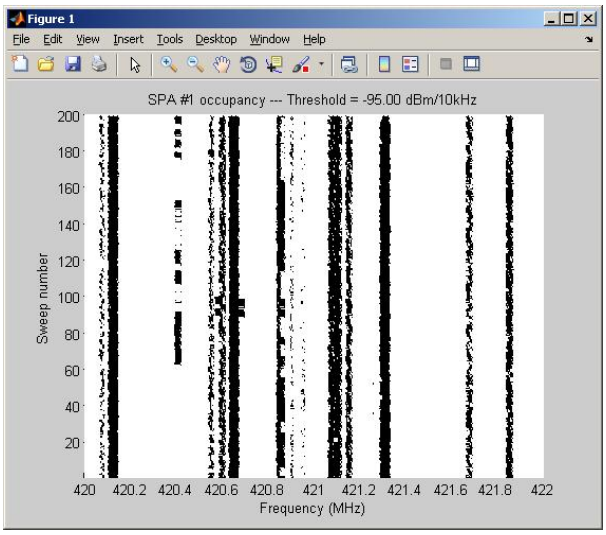


Figure 6-7: Screenshot of the REM Manager computer.

The statistics and the results of the emulation/simulation of the prototype are displayed by the REM GUI statistics module in a separate window. Figure 6-8 shows a screenshot of the window with the statistic results. The statistics shown in each graph are discussed below, assuming that the graphs are numbered left-to-right and top-to-bottom.

- Graph 1 shows the state (busy or idle) for each primary channel (channel numbers are shown in the abscissa axis).
- Graph 2 shows the length/duration (in seconds) of the current idle period for each primary channel (channel numbers are shown in the abscissa axis). For channels that have always been idle in the simulation, the value corresponds to the simulation time. For channels that are currently busy, the value shown is equal to zero.
- Graph 3 shows the history of the last 50 values for the blocking probability obtained when the RANDOM (RND) and REM algorithms are employed.
- Graph 4 shows the state (busy or idle) for each secondary user's traffic (CR user numbers are shown in the abscissa axis). A busy value indicates that the corresponding CR user has some data to transmit (session is active) while an idle value indicates that the corresponding CR user has no data to transmit (session is inactive).
- Graph 5 shows the average value for length/duration (in seconds) of idle periods for each primary channel (channel numbers are shown in the abscissa axis). For channels that have always been idle in the simulation, the value shown is equal to zero.
- Graph 6 shows the history of the last 50 values for the dropping probability obtained when the RANDOM (RND) and REM algorithms are employed.

- Graph 7 shows the history of the last 50 values for the channel currently allocated to a randomly selected CR user (the CR user is the same for the whole simulation) when the RANDOM (RND) and REM algorithms are employed. A value equal to zero means that the CR user has no channel allocated because there is no data to transmit for that user (the session is inactive) while a value equal to –1 means that the CR user has no channel allocated because the CR user is either blocked or dropped.
- Graph 8 shows the history of the last 50 values for the probability of spectrum handover (SpHo) computed over all the CR users in the system.
- Graph 9 shows the history of the last 50 values for the spectrum handover (SpHo) rate in terms of number of SpHo per second computed over all the CR users in the system.

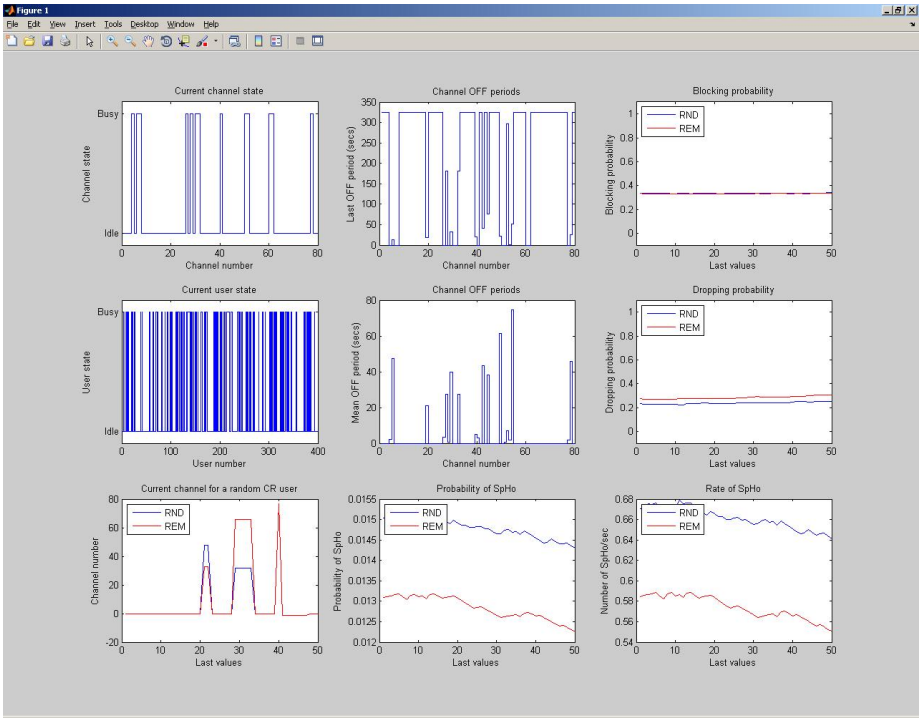


Figure 6-8: Screenshot of the prototype statistics.

Finally, in order to stop the emulation, press the "Stop testbed execution" button in the GUI window (see Figure 6-5). At this point, it is possible to change the configuration parameters and launch a new emulation. To stop the emulation and finish the prototype execution, first press the "Stop testbed execution" button, then the "Disconnect" button and then close the DREM GUI window. The statistics window and the execution of modules in the other Matlab windows will finish.

6.4 Field Results and Discussion

Taking into account previous measurements performed in the UPC-Campus Nord and reported in [48], the band ranging from 420-430 MHz was selected for demonstration. This band has an average occupancy around 60% and has channels with high or very high occupancy, but also channels with medium and low occupancy as shown in Figure 6-9 in terms of power values (max, min and average), instantaneous evolution of the temporal spectrum occupancy and duty cycle. Notice that the selected band basically corresponds to the downlink TETRA system. However,

thanks to the emulated nature of the envisaged secondary cognitive network, we are able to evaluate the performances of the proposed algorithms considering true real time channel behavior without interference to the primary system, in this case the TETRA System.

Taking into account that the channel bandwidth of the TETRA system is 25 KHz there are 40 channels in each MHz. Then, in order to maintain the emulation time bounded while still managing a large enough number of secondary users, the band selected for the trials was 421-422 MHz. As it can be seen in Figure 6-10 (a), there is a channel (around 423.1 MHz) permanently occupied with received levels around -70 dBm, another channel also with high occupancy around 421.1 MHz, but with lower signal levels (around -80 to -90 dBm) and eight channels exhibiting an ON-OFF (occupied-free) behavior to a greater or lesser extent with signals levels ranging from -90dBm to -100 dBm. The rest of the channels have levels lower than -100dBm with occasionally levels around -90Bm. Then, if the threshold for declaring a channel occupied is fixed to -100dBm/10KHz, the channel occupancy behaviour shown in Figure 6-10 (b) is obtained. From the figure we can realize that a good mix of channels is obtained, some of them being almost all the time ON, some others with significant variation of the ON-OFF patterns and some other free channels or with large OFF duration. That is, the selected RRM algorithms will be evaluated in a realistic environment that exhibits some flexibility for secondary transmissions.

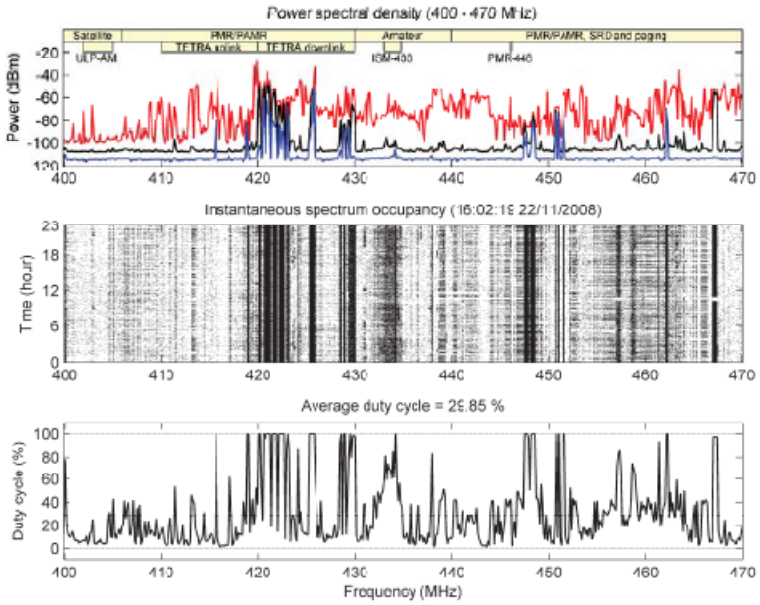
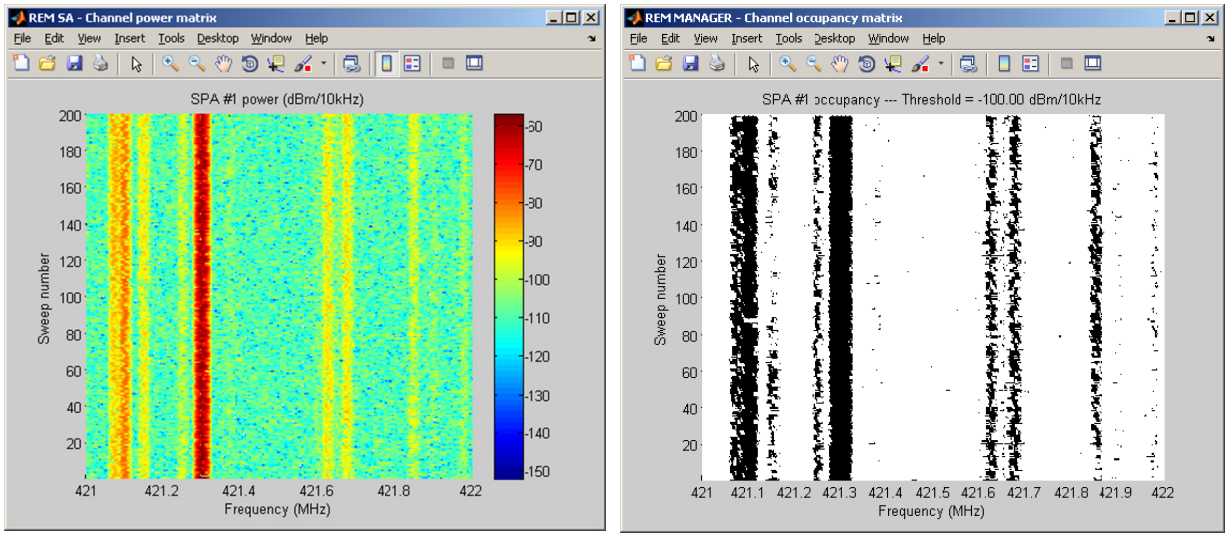


Figure 6-9: Example of the measured spectrum occupancy in terms of Power Spectral Density (PSD), the upper graph of the figure showing the minimum, maximum and average values; Instantaneous evolution of the temporal spectrum occupancy in the middle graph (black colour indicates a busy channel while the white colour means idle channel) and duty cycle in the lower graph.

In addition to that, the secondary traffic per user will be initially configured to 2 calls/min and 20 sec/call, which is a load level of 0.66 Erlangs per user (120 calls/hour * 20 sec/call/3600sec/hour),

high enough to load the system with a reasonable number of secondary users[6]. The number of secondary (CR) users can be varied between 10 and 100 in order to force various load levels.



(a) (b)

Figure 6-10: Power measurements and channel occupancy of the selected band for the trials.

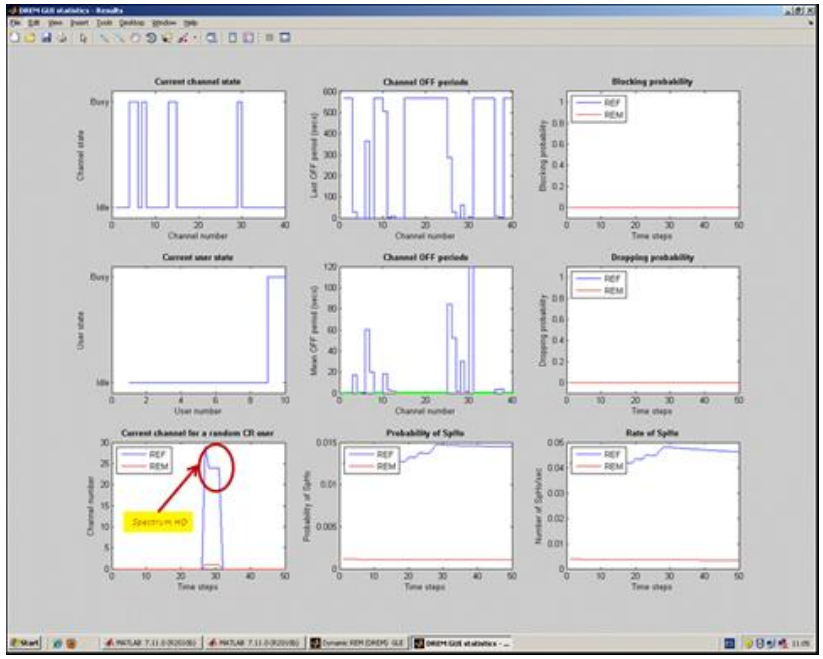
Figure 6-11shows, as an example, four snapshots of the GUI considering 10 users CR (Figure 6-11a), 40 CR users (Figure 6-11b), 70 users (Figure 6-12 (a)) and 90 users (Figure 6-12(b)).From the observation of the figures, we can highlight the graphic located in the lower left corner where one can see, as an exemplary case, how a spectrum handover event happens on the channel assigned to a secondary reference user when considering the random channel allocation algorithm (denoted as REF in the figures). Looking at the central graphic, we can see the average duration of the OFF periods for the various channels considered in the trial (numbered from 1 to 40 in the figure). It can be seen that some channels are free all the time (shown as a green line in the figure) and a few (between 7 and 10) show ON/OFF activity periods, although some of them with average OFF periods as long as 1200 s (Figure 6-11c) or 300 s (Figure 6-11a). Looking at the figure in the bottom centre and the one located in the lower right corner, where the probability of spectrum handover and spectrum handover rate are shown, it can be seen that the REM-based algorithm has better performance than the reference algorithm. This improvement is particularly significant, both in terms of probability and rate, in the middle range of the considered number of users (e.g. 40-70), which is reasonable. For a small number of secondary users, there are a large number of free channels and in general few spectrum handovers happen, whereby the advantage provided by the REM-based algorithm with respect to reference is small. Moreover, when the number of secondary users is large, although there is a greater number of spectrum handovers, there is also a greater chance that a secondary user is forced to make a spectrum handover without finding any free channel so this will lead to a dropping, with the corresponding increase in the dropping probability (see the evolution of the figure located at central row right column[7]). This implies that the probability and the rate of the spectrum handovers for both algorithms tend to be similar,

[6] Remember that the emulation time needed to reach a stable steady state behavior increases with the number of users.

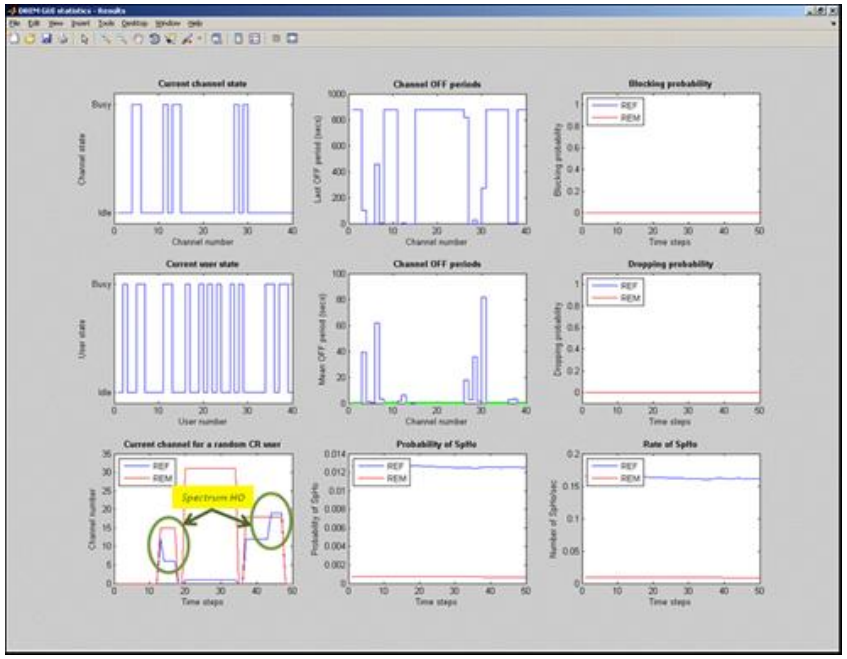
[7] When the number of users increases we move from a dropping probability near to zero, in the case of 10 and 40 users, to a dropping probability still small (around 0.5%) for 70 users, up to finally reach a probability of 20% for 90 users.

although the REM based algorithm slightly improves the performance with respect to the reference algorithm.

Finally let us mention here that increasing the number of secondary users also increases the blocking probability, in similar terms to what happens with the dropping probability. This is reasonable because the level of traffic generated by each user is in the order of 0.67 Erlangs. Then, when working with only 10 users (total traffic around 6.6 Erlangs), there are in average enough number of free channels so that all secondary users are able to access the secondary cognitive network. However, when the number of users significantly increases, for example up to 90 users, the network traffic on average is so high (about 60 Erlangs approximately) that although the available free channels range on average between 30 and 35, they are not enough to absorb the generated traffic, thereby increasing the blocking probability up to values around 15-20%.

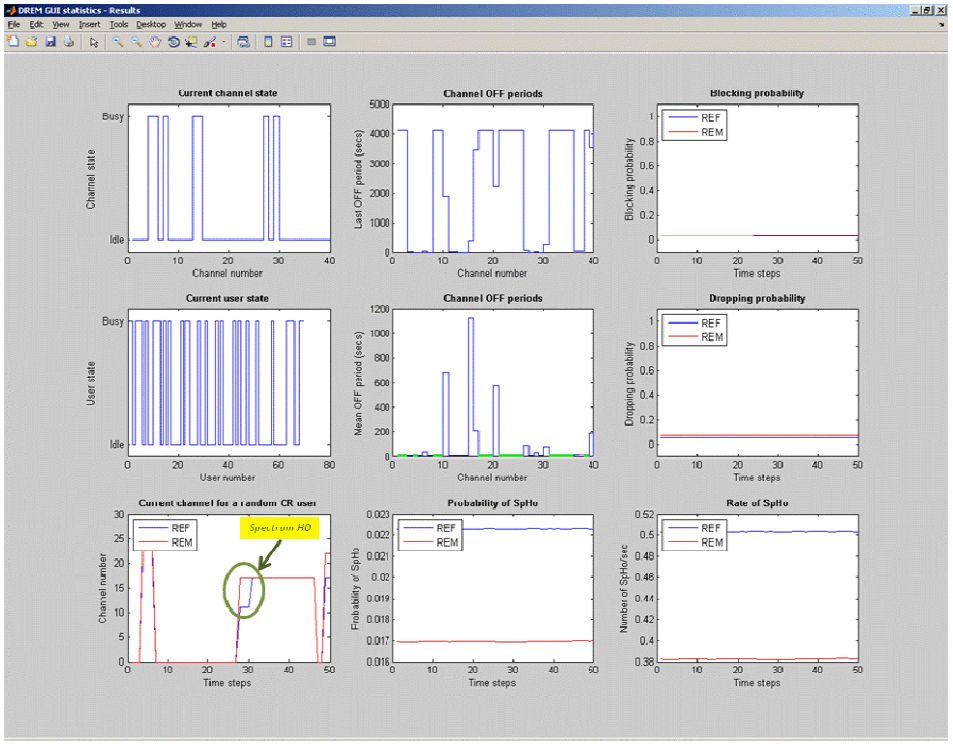


(a) 10 Secondary Users

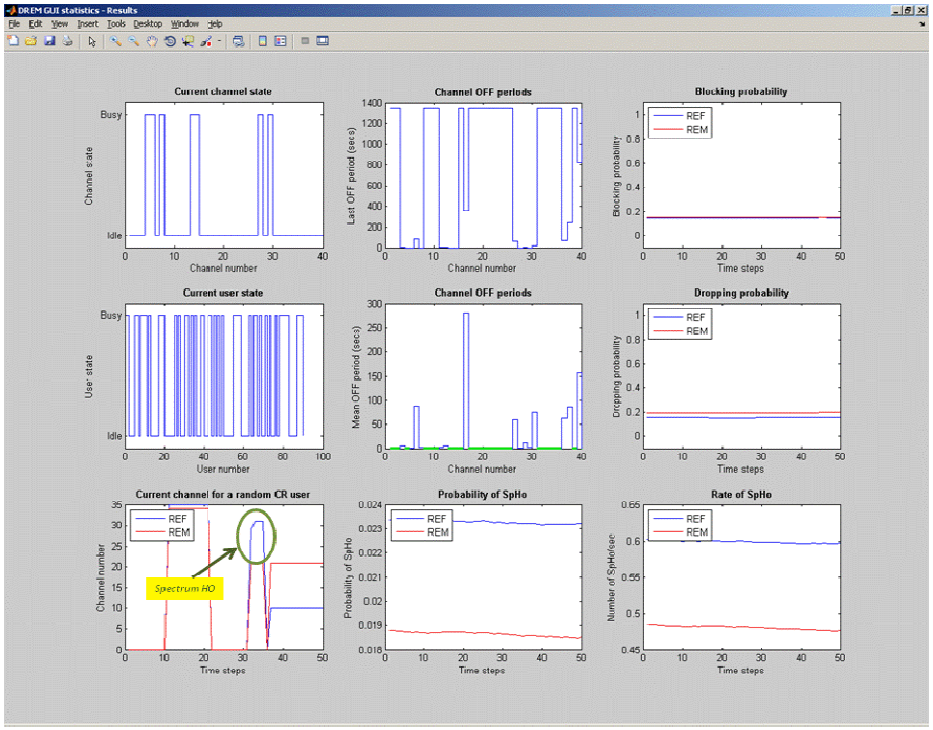


(b) 40 Secondary Users

Figure 6-11: GUI Snapshots considering different number of secondary cognitive users.



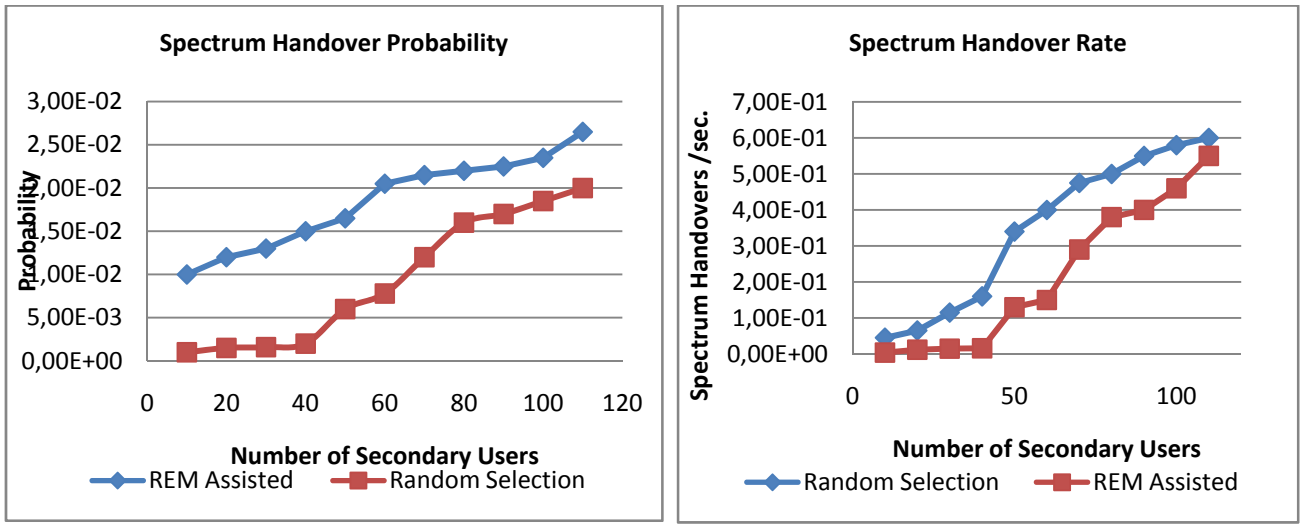
(a) 70 Secondary Users



(b) 90 Secondary Users

Figure 6-12: GUI Snapshots considering different number of secondary cognitive users

In order to provide a more accurate validation of the improvements provided by the REM based algorithm versus the random algorithm, chosen as a reference, different sessions of 30 minutes are emulated for different numbers of secondary users (ranging from 10 to 30). This was done for 5 consecutive days and the results obtained for the same number of secondary users were averaged in order to obtain the spectrum handover, blocking and dropping probabilities as well as the rate of spectrum handovers. The averaged results are shown in Figure 6-13 and Figure 6-14.



(a) (b)

Figure 6-13: Spectrum handover probability and spectrum rate versus the number of Secondary Users.

Performing, for each load level, 30-minute sessions allows us to ensure that the results are sufficiently stable becausethe emulator has reached the steady state. Moreover, averaging the results over a period of five consecutive days allows us to observe a sufficiently large number of states of the channel as to guarantee a small error in the results shown, that is thereis an accurateenough statistical characterization.

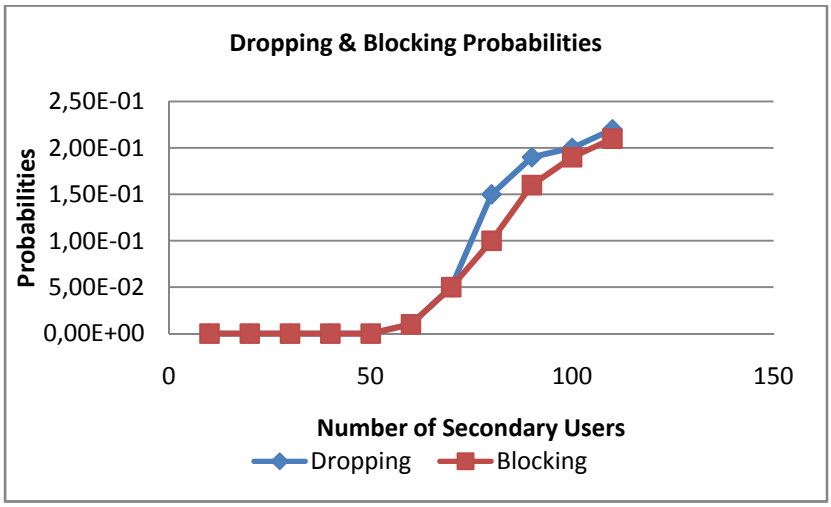
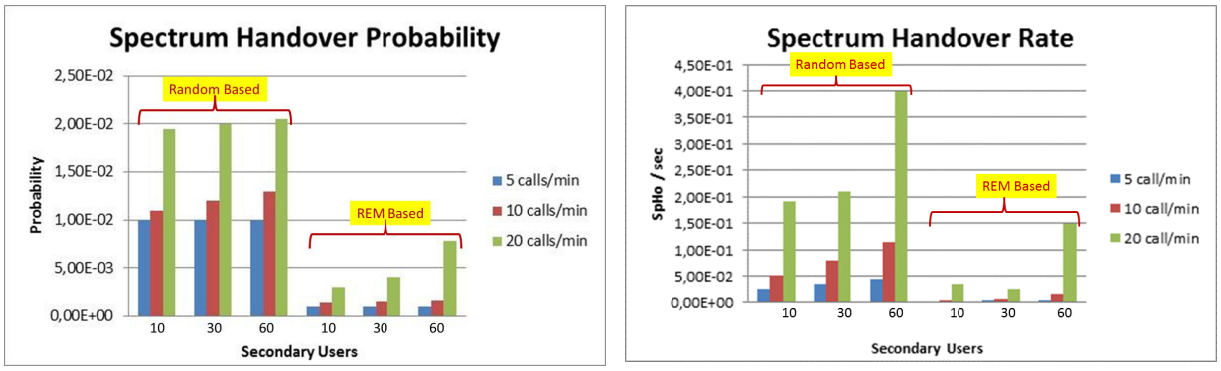


Figure 6-14: Blocking & dropping probabilities versus the number of secondary users.

Looking at the results shown in Figure 6-13, we can see how the REM-based algorithm always has a better performance than the algorithm based on random selection of channels, in both terms probability and rate of spectrum handovers. The greatest gain of the algorithm occurs when the number of users is approximately between 40 and 80 users, generating a medium system load, which is reasonable. For few cognitive users, there are enough free channels available and in general the number of spectrum handovers will be low. On the opposite, with high load, that is a large number of cognitive users, the dropping and blocking mechanisms eliminate secondary users of the cognitive system due to the lack of free channels, reducing the influence of the REM when selecting the channel. This is clearly stated in Figure 6-14, where the blocking and dropping probability evolution versus the number of cognitive users is shown. It must be firstly stressed that there is no significant difference in terms of dropping and blocking probabilities between the REM and the random algorithms. For this reason, Figure 6-14 only shows the evolution of these probabilities for the case of the REM algorithm. As shown in the figure, when the number of secondary users is above 60, both rates grow significantly, being slightly higher than the dropping rate. That is, from the viewpoint of system management it is meaningless to increase beyond 60 (approximately 45 to 47 Erlangs) the number of secondary users since the quality of service perceived by them would be too low.



(a) (b)

Figure 6-15: Spectrum handovers & spectrum rate versus the number of secondary users.

Finally, the influence in the system performance of the call rate has also been evaluated. In particular low (10 cognitive users), medium (30 cognitive users) and medium-high (60 cognitive users) traffic loads have been considered. For these numbers of cognitive users, three call rates have been assumed, namely: 5calls/min, 10calls/min and 20 calls/min. On the other hand, the secondary network system performances used for comparison between the REM based and the random based RRM algorithms are the spectrum handover rate and probability.

The obtained results show that once again the REM-based algorithm always outperforms the random based one. As expected, when the call rate increases, the number of spectrum handovers also increases, with more significant increment when moving from 10 to 20 calls/min. There are no significant performance differences when considering 10 and 30 cognitive users, that is for low medium traffic, being the performance differences more relevant when moving to medium-high traffic loads (60 cognitive users).

In summary, based on the obtained results, it can be concluded that:

1) The blocking and dropping probabilities are quite similar for the reference and REM algorithms because they depend on the number of CR users and the volume of traffic that they generate as well as their relation to the number of primary users and their spectrum

activity. The channel selection algorithm has little or no impact on the obtained blocking and dropping probabilities.

2) For a high number of secondary CR users (70 users in our study), the blocking and dropping probabilities will be relatively high. In this case, the blocking and dropping of CR users will have an important impact on the system's performance and for this reason, for a large number of CR users, the number of SpHo is less relevant. It is thus observed that the SpHo probability and SpHo rate of the reference and REM algorithms are quite similar for a large number of CR users, even though REM always outperforms the reference.

3) For a low number of secondary CR users (10 to 40 users in our study), the blocking and dropping probabilities will be relatively small (quite close or equal to zero). In this other case, the blocking and dropping of CR users will have little or no impact on the system's performance and for this reason, for a small number of CR users, the number of SpHo becomes relevant. It is observed that the REM algorithm significantly outperforms the reference algorithm in terms of SpHo probability and SpHo rate when the number of CR users is kept at reasonable levels with respect to the amount of primary channels over which the CR system operates. This demonstrates how the construction of a REM based on radio environment information and its exploitation by a CR system can lead to notable performance improvements.

7 REM Assisted Resource Management in MANETs

The FARAMIR introduced REM concept may also provide vital benefits for Mobile Ad-hoc NETworks (MANETs). This chapter explores the possibility to use the REM to improve RRM activities and to verify the compatibility of the REM concept with the specific constraints of MANETs. Additionally, the chapter also targets validation of the REM based architecture robustness in scenarios with high number of nodes.

Similarly as the prototype introduced in chapter 6, this chapter focuses solely on simulation-based study of the REM facilitated problem in MANETs. However, the essential architectural components and their interactions (as elaborated in chapter 2) are closely followed.

7.1 Prototype Scope and Focus

The identified scenario of interest is a natural disaster scenario. In case of an earthquake, a tsunami or other kinds of natural disasters, several rescue actors must quickly deploy their units in the disaster area in order to reduce the impact of the natural event in order to help wounded people and to limit collateral damages. Therefore, different kind of rescue units should be deployed in the area such as firemen, police, ambulances, specific intervention units etc. In each of these units, members should be able to communicate among them to effectively act in the area. This requires radio resources that cannot be provided by existing wireless network infrastructures.

If we take for example the case of an earthquake, most likely the telecommunication infrastructures in the area are damaged reducing their capability to operate and the available network capacity. Furthermore, the residual network capacity is quickly saturated by the traffic requests of survivors in the disaster area that try to call for help or to check if friends and relatives are unwounded and by the local intervention units. Hence, it is clear that in a similar scenario the rescue actors that arrive in the area should deploy their own wireless networks (i.e. a MANET) in order to allow units to communicate. A major issue will then be to guarantee the coexistence of the MANET deployed by the different rescue actors that could share the same radio resources and that must avoid to mutually interfere. In this sense, the REM concept can be a helpful mean. Therefore, this chapter concentrates on how the RRM problem can be solved in similar scenarios and in which way the use of REM can help.

Notice that, with respect to other scenarios, here the performance of the networks is strictly related to the number of nodes involved. Hence, this demonstration is, for certain aspects, complementary with the prototyping work done for the other scenarios by allowing to test and validate the REM concept and the algorithms developed in the other WPs when the number of nodes and networks in the area is relatively high.

In WP2, a system architecture specific for MANETs, derived from the general REM-based system architecture, was proposed, while in WP5 several algorithms and techniques to take advantage of the REM concept for RRM operations have been analysed. Here we try to validate the work done in other WPs by implementing REM and full protocol stack nodes to build an OMNET based, bit level, system demonstration.

7.2 Necessary Components

To apply the REM concept to our scenario some characteristics of MANETs that differentiate them from infrastructure-based networks should be taken into account.

In particular, the construction of the network is dynamic and the network topology can evolve in time with nodes that can cover different functionalities in the network in different moments. Moreover, the lack of infrastructures requires specific protocols to maintain in place the network and to exchange signalling information.

In the FARAMIR system architecture presented in deliverable D2.3 ([43]) and finalized in deliverable D2.4 ([49]) the general network element (or node) structure was introduced. According to the description of the network element provided, each node belongs to a class that identifies its capabilities and determines which functionalities it can cover in the network.

For MANETs in emergency scenarios, three main classes have been defined to distinguish among the capabilities of nodes:

- **Class 0**: the node does not have measurement capabilities and it does not implement Local REM (LREM) and/or RRM functionalities;
- **Class 1**: the node has measurement capabilities, but is not capable to cover the LREM role in the network and it does not implement RRM functionalities.
- **Class 2**: the node implements LREM and RRM functionalities and so it can cover the role of Network Head (NH) of the network.

Notice that the three classes defined are not exhaustive of all kind of functionalities that a node can have, but allow to clearly differentiate the main actors of each network.

The kind of networks that we consider are typically composed by both vehicles (such as ambulances, police cars etc.) and walking units (like doctors, policemen etc.). Clearly, the physical characteristics of a node (computation power, transmission range etc.) are different if the network is integrated in a vehicle or if it consists of a radio in the bag pack of an agent. Thus in addition to the classes previously presented, a further differentiation is done (going from three to six possible classes) to distinguish if the node is a vehicle (class 0a/1a/2a) or a pedestrian (class 0b/1b/2b).

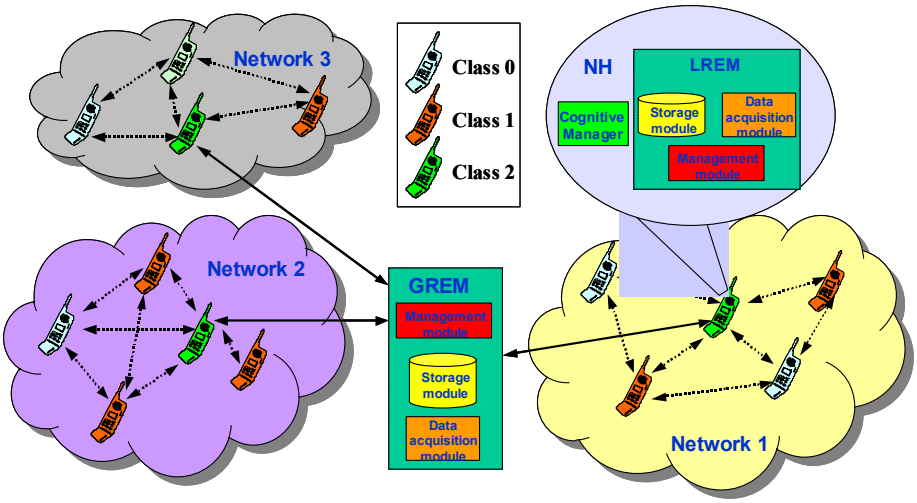


Figure 7-1: Example of nodes organized in networks following FARAMIR system architecture.

7.2.1 Scenario Description

The scenario is composed by several nodes organised in networks deployed in the same area. The structure of the nodes and the construction and the exploitation of the REM are conceived in order

to take into account the specific constraints related to the MANETs and to follow the FARAMIR system architecture, Figure 7-1.

In the beginning, users of each network elect a NH among the nodes belonging to class 2. The NH will then specialize in coordination of sensing activities and in RRM operations. From the REM point of view, the NH acts as the LREM of the network implementing REM SA functionalities to pilot sensing in the network and to collect measurement data from active MCDs, as well as REM Management functionality in order to analyse information retrieved both from LREM SA and Global REM (GREM) and to interact with modules in charge of RRM operations. Moreover, the NH will be the only node of the network that communicates with the GREM in order to report local measurements and network information and to obtain useful data to pilot radio resource allocation in the network.

7.2.1.1 Node Architecture

Different nodes can belong to different classes, but their architecture is common. However, some modules are implemented only in nodes that belong to a determined class.

The general internal architecture of each node follows a layered structure as presented in Figure 7-2.

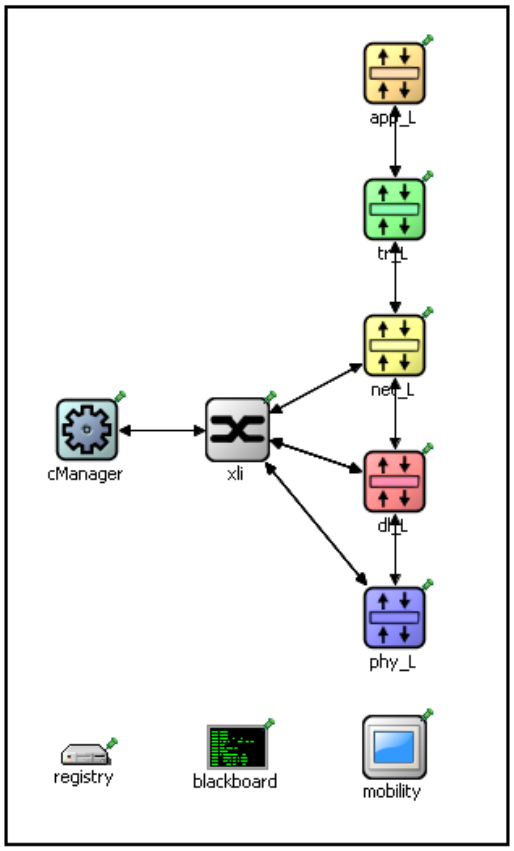


Figure 7-2: Node internal architecture.

The node architecture realized to implement the REM concept in the networks includes a block called cManager that is composed by the REM modules and the cognitiveManager module. The latter is in charge of the RRM activities for the channel selection of the network and of the analysis and exploitation of the REM provided data.

The cManager block should then interact with different stack layers, in particular with the physical layer, which includes the measurement module and the MAC layer in order to provide inputs for intra-network resource allocation operations. To allow these interactions, a cross layer interface (XLI) has been implemented in the node architecture.

More details on cManager and its interactions with layers are provided in paragraph 7.2.2.2.

7.2.1.2 REM Definition

The nature of the networks considered leads to the definition of a two level REM architecture as described in deliverable D2.4 ([49]):

- at each network level a LREM manages all the measurement activities and data analysis to provide useful information for RRM activities;
- at area level a GREM collects data from the LREMs of the networks present in the area and redistribute useful information on the area.

From implementation point of view, the node in the network that acts as LREM should be able to interact with the GREM in order to register the presence of the network in the area, to report collected information and to obtain data for RRM activities. The GREM can be implemented as an external entity or it can be even integrated in one of the networks, e.g. it could be a specific unit of the local emergency forces.

During activities of the networks, both the LREM and the GREM should be updated and so signalling information should be exchanged. To carry the signalling messages that need to be sent over the air, two possible solutions can be adopted, i.e. the use of a dedicated signalling channel or the use of the frequencies opportunistically chosen for user data transmissions. The advantage to have a dedicated channel to signalling messages is to avoid the risk of interference, but this requires a strong assumption on the existence of a proprietary channel in each network or to reduce the possible data rate dedicating some frequencies to signalling only activities. On the other side, to map the signalling messages on the frequencies opportunistically chosen for data communications makes them sensible to interference.

For the demonstration, we opted for an intermediate solution, i.e. we assume the existence of a dedicated, low data rate, signalling channel between the GREM and the NHs of the networks in the area, but we define a specific MAC frame and protocols that allow to map intra-network signalling messages (LREM related messages, but also resource allocation and network coordination messages) directly on the frequencies selected for user data transmissions with a limited impact on the network data rate. Next paragraphs will analyse in more details the exchanged messages and the interactions of the modules.

7.2.2 Implemented Interfaces and Exchanged Messages

7.2.2.1 Intra-Network Messages and MAC Implementation

The messages that carry the information exchanged in each network between nodes are transmitted through a TDMA scheme. Different kinds of slots are characterized at MAC level and they are organized in a MAC frame.

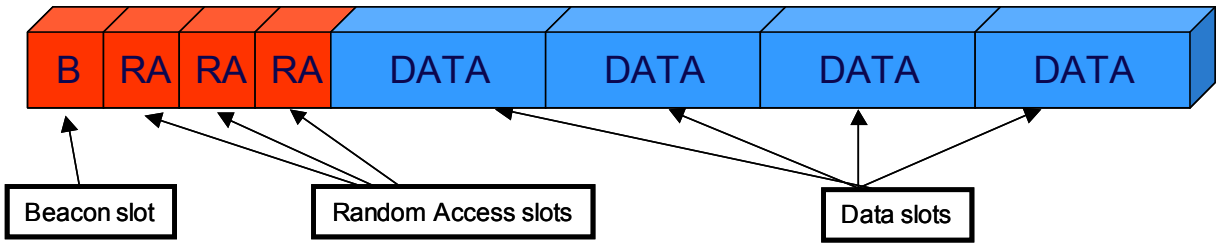


Figure 7-3: MAC frame implementation structure.

Three kinds of slots are identified:

- *Beacon slots*: used by the NH to initialize the network, to transmit resource allocation information to users and to transmit LREM orders to MCDs;
- *Random Access (RA) slots*: used by the nodes of the network to indicate their presence, to discover their neighbours, to send resource requests to the NH and to report measurements to the LREM and
- *Data slots*: used to transmit user data and to perform measurement operations.

Each MAC frame starts with a beacon slot, followed by a fix number of RA slots and a fix number of data slots. In the two first types of slots, only signalling messages are sent. The messages that can be exchanged in the slots are beacon messages, hello messages and data messages.

Beacon messages. The NH broadcasts beacon messages at the beginning of each MAC frame during the reserved beacon slot. Information included in beacon messages permits to all users to keep synchronisation with NH and to receive resource allocation information and measurement orders. Table 7-1 shows the format of beacon messages.

Table 7-1: Beacon message structure.

Node ID	Node class ID	NH/LREM ID	Current channel ID	Next Channel ID	Resource allocation list	Sensing orders list
Node identifier	Class of the node identifier	Identifier of NH/LREM of the network -> same that Node ID	Identifier of channel in use during starting MAC frame	Identifier of channel in use during next MAC frame	List of data slots allocated to each data flow. The format is: source Node ID, destination Node ID, data slot number, Max power available	List of sensing orders for MCDs. The format is: MCD ID, starting time, duration, channel to sense

Hello messages. Hello messages have multiple purposes in the network such as update the neighbour tables of nodes, bring requests of resources to transmit data and contain results of measurement operations to update the LREM. These messages are sent during RA slots, i.e. in each MAC frame, each node of the network selects randomly a RA slot to send a hello message. Due to the contention access of RA slots, hello messages can undergo collisions. Table 7-2 shows the format of hello messages.

Table 7-2: Hello message structure.

Node ID	Node class ID	Node position	NH/LREM ID	Current channel ID	Next Channel ID
Node identifier	Class of the node identifier	x, y coordinates	Identifier of NH/LREM of the network	Identifier of channel in use during current MAC frame	Identifier of channel in use during next MAC frame
Neighbours' list		**Resource request list**		**Sensing results list**	
List of neighbours. The format is: **neighbour node ID, position (x,y)**		List of communication requests.The format is: **destination Node ID, data size, priority**		List of sensing results. The format is: **channel ID, starting time, duration, Max power detected, Average power detected**	

Data messages. Data messages correspond to user useful traffic. The networks built for demonstration use UDP/IP protocols and so the user traffic is composed by UDP packets. According to the needs of the rescue actors, the data sent can have different nature and different importance. To take into consideration this aspect, we build a system that can support two different kind of traffic priorities: real time and best effort.

An example of the different kind of messages exchanged is provided in Figure 7-4.

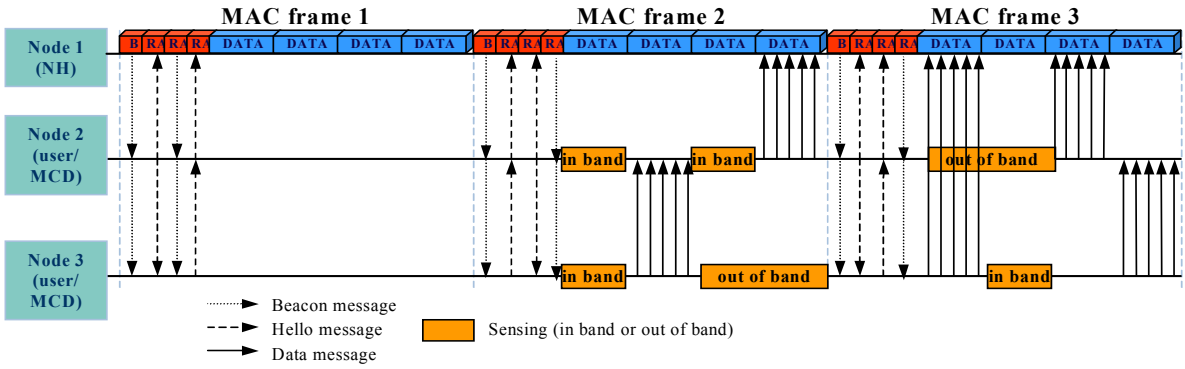


Figure 7-4: Example of messages exchanged between a NH and two nodes of class 1 of the same network.

7.2.2.2 CManager Interfaces and Exchanged Messages

In each network, the NH is the responsible of LREM and RRM operations. The module in the NH node architecture in which the LREM operations are controlled and REM data are analysed and exploited for RRM operations is the cManager. Figure 7-5 represents the internal structure of the cManager module.

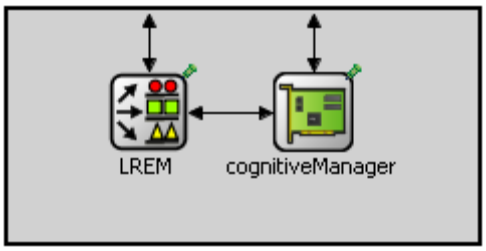


Figure 7-5: cManager module

The cManager module includes two sub-modules, the LREM module and the cognitiveManager module. The first one includes the REM sub-modules: the LREM SA and the LREM Manager. The LREM SA sub-module is in charge to guide the measurement operations and to stock the measurement results and the data received from the GREM. The LREM Manager sub-module is in charge to analyse the received data and measurements before to stock them, and to interact with the cognitiveManager module and with the GREM.

The cognitiveManager module uses the data provided by the LREM Manager to select the frequency channel for network traffic needs and to provide to the MAC layer the right inputs to allocate resources to the nodes that want to communicate inside the network. Moreover, it reports data on the network and on the RRM choices to the LREM Manager that can stock them in the LREM SA and report them, in addition to measurement results, to the GREM.

Figure 7-6 describes the messages related to the cManager activities and exchanged between the different modules in the NH and with other nodes.

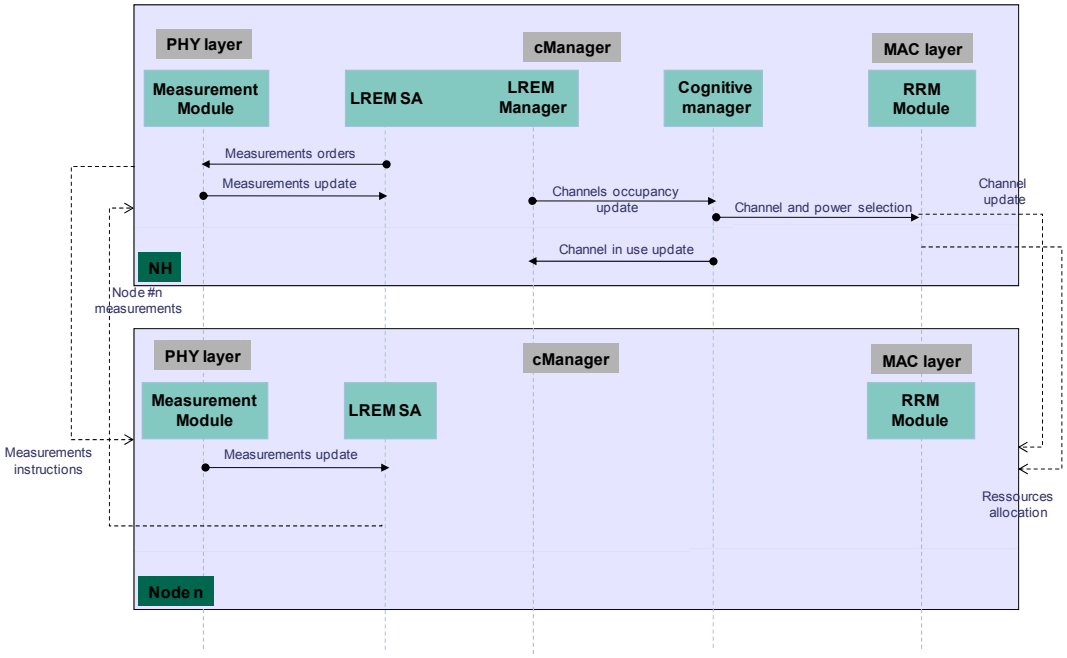


Figure 7-6: cManager messages.

Notice that in addition to the NH, the nodes in the network that also have sensing capabilities (classes 1 and 2) implement some reduced functionalities of the LREM. In particular, the messages between the LREM and the physical layer to report measurement results are exchanged.

7.2.2.3 GREM - NH Interfaces and Exchanged Messages

To improve their awareness of the radio environment in the area, the networks, through the NH, exchange messages with the GREM of the area. These signalling messages are exchanged using a dedicated low data rate channel. To take into consideration the limited data rate available and to uniform the amount of messages exchanged avoiding peaks, we chose to send updates from GREM to LREM and vice versa periodically, with the GREM that is in charge to regulate the frequency of the updates.

The periodic exchange of messages avoids excessive traffic in high interference situations when several networks can require at the same time updates from GREM quickly saturating the signalling channel. Moreover, the GREM can decide to increase or to reduce the frequency of the updates according to the number of networks in the area. The signalling channel access is organized following a TDMA scheme piloted by the GREM. Figure 7-7 represents the details of messages exchanged between the LREM Manager module of the NH of a network and the GREM.

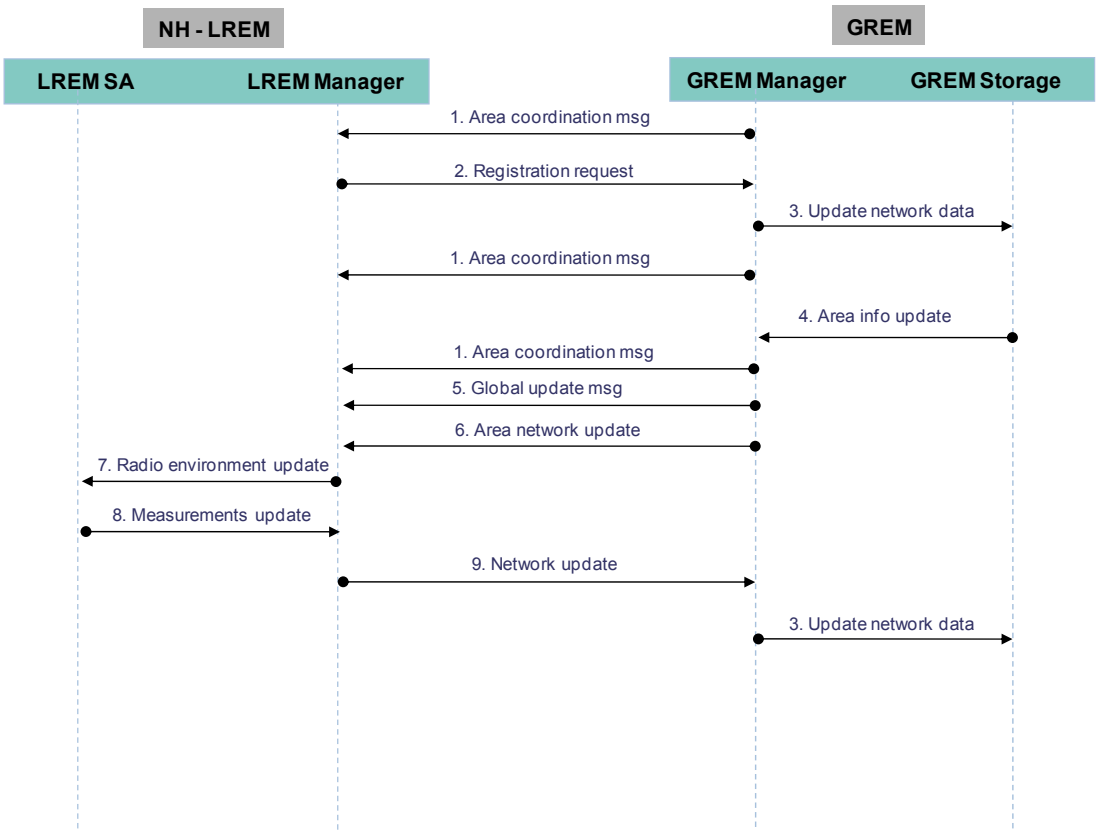


Figure 7-7: LREM – GREM messages.

The procedure for a network to register at the GREM starts with the network listening for an area coordination message from the GREM in which it can know when it has to send a registration request. Once the slots for this purpose are identified, it randomly selects one of them to send its registration request. The GREM, when receiving the registration request, updates its data on the networks present in the area that is memorized at the GREM Storage module. Hence, it will include, in the next area coordination message, the id of the new network and the slot dedicated for it to receive specific updates from the GREM and the one available for the LREM to send updates to the GREM.

7.2.2.4 Network Construction and Organization

Network is composed of a set of one and two-hop neighbour nodes. To make sure of network reactivity, all nodes are at most one-hop distant from the NH, which avoids forwarding mechanisms for signalling information. The network construction procedure described hereafter assumes that nodes are already synchronized. Different techniques for synchronization of nodes are available in the literature, e.g. this can be done through the use of GNSS [50].

Three main actions are required for the network construction: nodes have to provide information on themselves to their neighbours, identify their neighbours and elect a NH. These actions are based on the use of hello messages sent in RA slots. At each frame, nodes send a hello message that contains their identifiers, their class ID and their one-hop neighbour tables. When receiving a hello message, a node updates its neighbour table from the received message so that all topology changes are updated at each frame. Finally, the receiver identifies its NH that is defined dynamically following the procedure described below.

Each receiver, if it belongs to a class that enables to act as LREM (class 2), compares its number of neighbour with its neighbours' one. The election of the NH is based on the number of connections a node has. The node with the highest number of neighbours is elected NH. In case of equality, the node with the highest ID is chosen.

At the network initialization, all nodes have an empty neighbour table. They thus send a hello message with an empty table in the first frame. At the end of the first frame, all nodes have received hello messages from their one-hop neighbours (except if a collision has occurred in an RA slot). All nodes consequently know how many neighbours they have and compare this number to the number of neighbours their neighbours have indicated in their table (meaning zero since the first table sent is empty). As a consequence, all nodes consider themselves as network head for the second frame. They all send a beacon on a chosen channel. During this new frame, all nodes transmit their updated neighbour table. At the end of this second frame, each node knows both its and its neighbours' number of neighbours. Those that do not have the higher number of neighbours give away their NH status and stop transmitting beacons.

7.2.3 Implemented RRM Algorithms

The choice of the applied RRM algorithm has an effect not only on the performance of the networks, but it also impacts the REM activities. According to the implemented RRM strategy, the amount of information that has to be collected at LREM and GREM level can change.

Deliverable D5.2 ([51]) presents different strategies both for frequency channel selection, to avoid inter-network interferences, and for intra-network resource allocation in order to satisfy transmission requests of users taking into account different levels of priority of data traffic. It also proposed to extend the capabilities in terms of data rate of the system using a TDMA/OFDMA frame.

For the MANET demonstration, we decided to implement a TDMA/OFDM MAC frame, but we kept the possibility to have different priority data to transmit. The NH, when attributing resources to the different links, takes into account the priorities giving more resources to data transmissions with higher priority.

From inter-network point of view, there are two different algorithms implemented (EGADIA and CPSA fast) in order to test the behavior of the system when the RRM strategy is more or less dependent on information from the GREM. The following subsections briefly explain the two algorithms.

7.2.3.1 EGADIA Algorithm

The EGADIA algorithm is an extension of the GADIA algorithm proposed in [51]. The GADIA algorithm is a greedy algorithm operated at each network (or cluster) level and that can permit to solve the channel allocation problem under specific assumptions. In practice, the algorithm selects the frequency channel in which the network exhibits the least interference by assuming that the distance of two networks is much higher than the diameter of a network and so that the level of interference is uniform in the network.

With the proposed extension of GADIA algorithm (EGADIA), we relax the constraint on distance of networks and we consider the real level of interference experimented by nodes of the networks. In order to account for the various possible interference levels, we estimate the interference at all nodes and take the maximum interference in order to protect all the links. Then we select the channel that leads to the minimal maximum interference. The resulting algorithm is the following:

Algorithm 1: EGADIA: Extension of GADIA algorithm to account for close networks/clusters

At each cluster j
  For $c = 1, N_c$
    - measure interference level perceived at each node k on channel $c \rightarrow \{I_{j,k}^c\}_{k=1,K_j}$
    - compute $I_j^c = \sup_{k=1,K_j} I_{j,k}^c$
  End
  Take channel $c_{\text{op}}(j) = \arg\min_{c \in \{1,\ldots,N_c\}} I_j^c$

This algorihtm requires that interference of neighbour networks is collected at LREM level and is available for RRM operations. No specific information is required from the GREM to take a decision on the frequency channel to use. Notice that the EGADIA algorithm does not provide any guarantee on SINR experimented by each network.

7.2.3.2 CPSA Fast Algorithm

CPSA fast algorithm is a distributed strategy to allocate resources that allows networks to act autonomously choosing channel and power to satisfy their communication needs, but, at the same time, avoiding excessive interference on neighbors. The decision is taken at NH level using both measurements on interference and potential SINR level retrieved by the LREM and data obtained from the GREM on the presence of other networks in the area and on their activities and requirements.

With the collected information, when the channel in use is no longer available, the NH computes the maximum power available on each channel (P_{AV}). This power level is the maximum power that a user of the network can use on the corresponding channel without generating harmful interference on neighbor networks. It is computed using the following formula:

$$P_{AVi,j} = \min_u P_{AVi,j,u} = \min_u \frac{I_{TH_u} - P_{noise}}{\alpha_{j,u}(d_{u,j}, h_{i,j,u})} \quad (7.1)$$

where I_{TH_u} is the maximum interference accepted by network *u* users and $\alpha_{j,u}$ is the attenuation between the networks *j* and *u* that depends on their distance and on the coefficient of channel *i*.

Once the maximum available power on each channel is computed, the NH proceeds to rank the channels according to the following Utility Function (UF):

$$UF_{ij} = \frac{P_{AVi,j}\alpha_j(r_j, h_i)}{I_{TOT,i}} \quad (7.2)$$

where α_j is the attenuation value expected for communications inside the network between NH and farthest user and $I_{TOT,i}$ is the total interference detected on channel *i*.

Furthermore, the following condition related to the minimum capacity constraint is tested for each channel:

$$P_{AVi,j}\mu_2 \geq P_{\min}\mu_1 \quad (7.3)$$

where $\mu_2 < 1$ is a protection margin to reduce the risk of harmful interference on neighbours due to inaccurate estimation of the channel, P_{min} is the minimum transmission power that NH has to employ to fulfill the SINR constraint at user in worst conditions and $\mu_1 > 1$ is a margin to protect the transmission. Once the channels have been ranked following the UF, the NH selects the channel with highest UF for which condition (7.3) is satisfied and transmits, in the next beacon message, the decision on the channel and on the maximum power to use to nodes of the network.

Algorithm 2: CPSA fast

Check SINR and Pav_of channel in use
If (SINR < SINRmin)
  If (condition (7.3) cannot be fulfilled adapting power in use)
    Rank channels according to (7.4)
    If ((7.3) can be respected on best channel)
      Select best channel
      Use min Pav*μ_1 on channel
    Else
      Select random channel
      Use Pmax on channel

Once the channel is selected, the NH keeps checking if the SINR on it is still acceptable and if $P_{AVi,j}$ has changed. If necessary it can proceed to adapt its power respecting the following constraint:

$$P_{AVi,j} \geq P_{used} \geq P_{\min}\mu_1 \quad (7.4)$$

If a P_{used} that respects this constraint does not exist, the NH starts a selection process looking for a new channel.

7.3 Prototype Functionalities

7.3.1 Configuration and Scenario Definition

The developed prototype is configured using files that contain the description of the different scenarios of interest, Figure 7-8. In addition, some parameters (such as the resource allocation algorithm) can be directly configured at the beginning through a graphical interface.

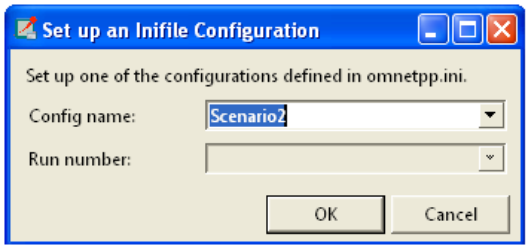


Figure 7-8: Configuration window.

During the demonstration it is possible to follow graphically the evolution of the scenario and the messages exchanged inside each node and within different nodes, Figure 7-9.

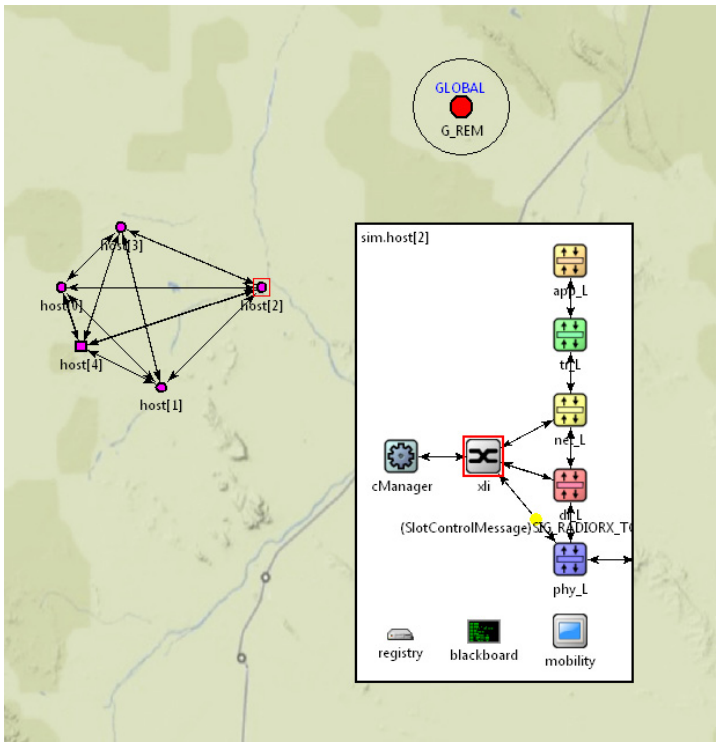


Figure 7-9: Screen shot of a message exchange inside a node.

7.3.2 Wireshark Message Analysis

To analyze the messages exchanged, we use wireshark. This software allows interpreting the bits exchanged among nodes in order to analyse the traffic and the signalling messages transmitted. Specific scripts have been developed to interpret the new messages implemented in the FARAMIR system.

7.4 Results and Discussion

The validation of the full FARAMIR architecture and the related protocols, as well as the REM and RRM strategies performance, comprises a two-step procedure. First, the full system with a relatively low number of networks and with low data rate is tested. Then, denser scenario with more networks and more traffic requirements is analyzed.

7.4.1 Low Number of Networks

This scenario comprises three networks with a number of nodes in each of them between five and seven. The nodes are deployed in the disaster area where they organize themselves in networks electing the NHs, Figure 7-10.

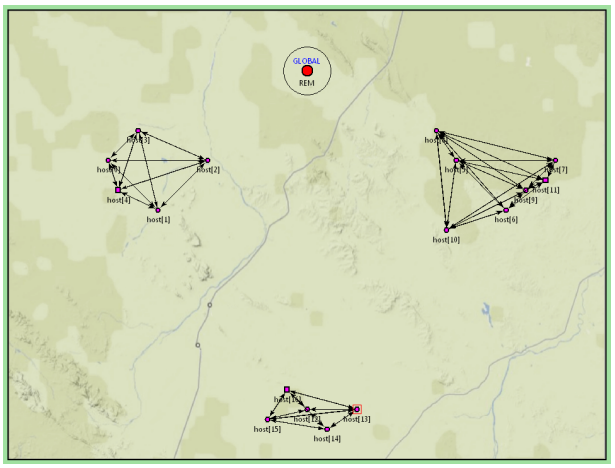


Figure 7-10: Positions of nodes in the area. The squares represent the NH and the arrows link the neighbour nodes. The color of the nodes represents the channel in use (at the beginning all the networks use the same frequency channel).

The starting frequency channel used by the nodes is the same for all networks. Once the networks are established, UDP traffic between nodes is exchanged in each network and the nodes begin to move making the networks come closer and start interfering each other (Figure 7-11).

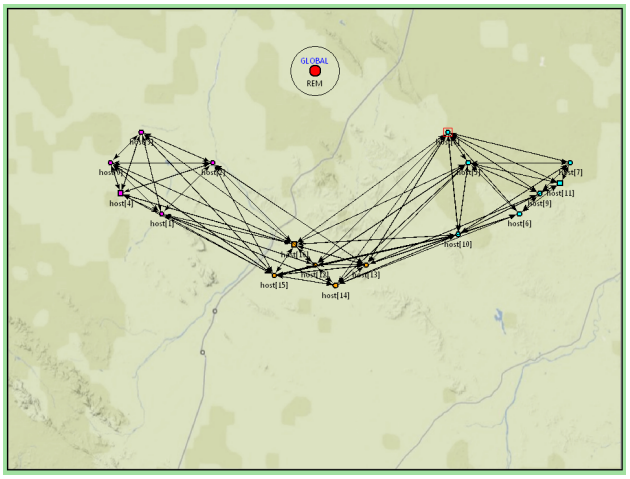


Figure 7-11: Interfering neighbours of nodes when the networks come closer.

The frequency channels are localized in the band between 400 and 600 MHz (TV white spaces) and we build the MAC frame structure in order to have a signaling ratio of 22%. The users in the networks move at a maximum speed of 3 km/h.

The first objective is to see if the intra-network signaling protocol is sufficiently robust to allow networks to apply radio resource management strategies to avoid interferences on date traffic. Figure 7-12 shows the results in terms of loss of UDP traffic using the proposed MAC frame and

signaling protocol, with respect to the use of a dedicated signaling channel. The RRM strategy is the same for both cases and is based on the EGADIA algorithm for channel selection. In addition, the figure also shows the UDP traffic data rate that the network will have without applying an opportunistic channel selection strategy based on radio environment awareness.

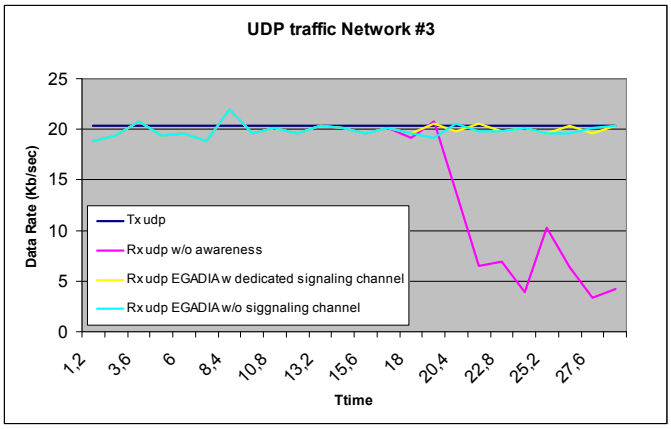


Figure 7-12: UDP data rate in Network #3.

It is evident that the UDP data rate is seriously impacted by interference when no radio environment awareness is exploited. On the other side, the use of the EGADIA algorithm allows to avoid interference with similar performance for the use of a dedicated signaling channel and of the proposed MAC frame structure and signaling protocol.

7.4.2 High Number of Networks

This scenario increases the number of networks up to seven in the same area. Moreover, the amount of data to transmit in each network including different level of priorities in data transmissions is also increased. The frequency band and all other assumptions about the nodes are the same as in the previous subsection.

Figure 7-13 depicts the starting positions of the nodes, before their organization in networks and the choice of frequency channel to use. Figure 7-14 depicts the positions of the nodes at the end of the scenario where the colors represent the frequency channels in use in each network.

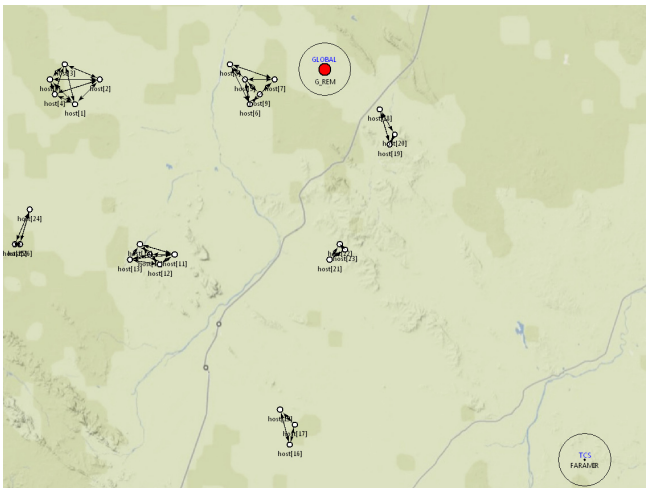


Figure 7-13: Positions of nodes in the area at the beginning of the scenario.

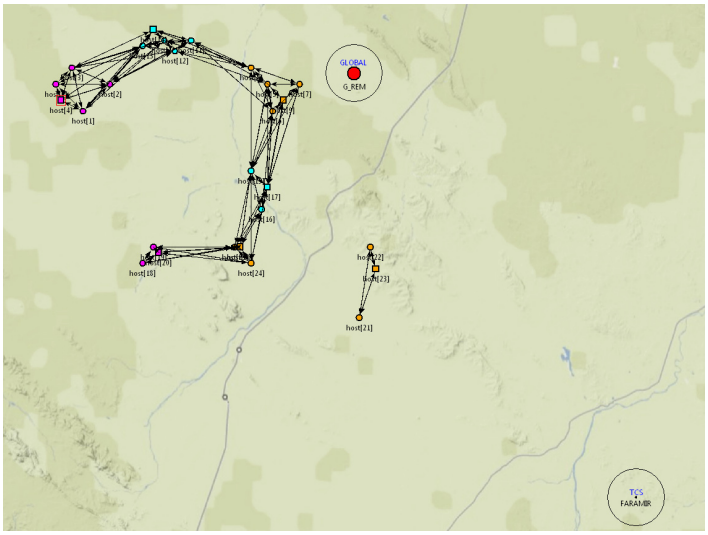
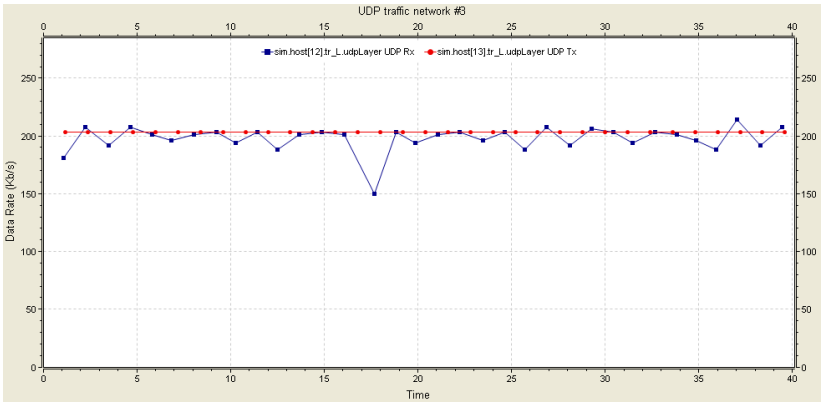


Figure 7-14: Positions of nodes in the area at the end of the scenario.

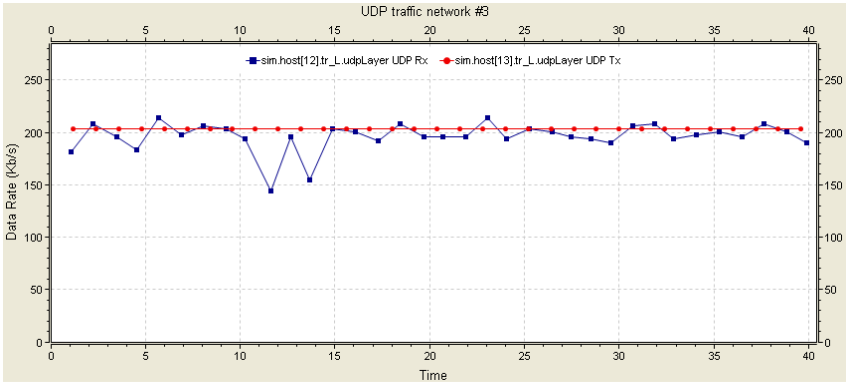
The scenario is tested using both the EGADIA algorithm for channel selection and the CPSA fast algorithm. Moreover, for CPSA fast algorithm we select two different values for the interval between the transmission of two GREM updates in order to evaluate the impact of accuracy in the knowledge of radio environment situation on the performance of the algorithm.

Figures 7-15 and 7-16 show the results in terms of UDP traffic applying EGADIA and CPSA fast with 10 and 3 seconds as frequency of updates from GREM. The red curves are the UDP packet transmitted, while the blue ones are the corresponding packets received. The analysis is done at transport level, so there are some differences between data transmitted and date received generated by the insertion of packets in queues before their transmission on the channel.

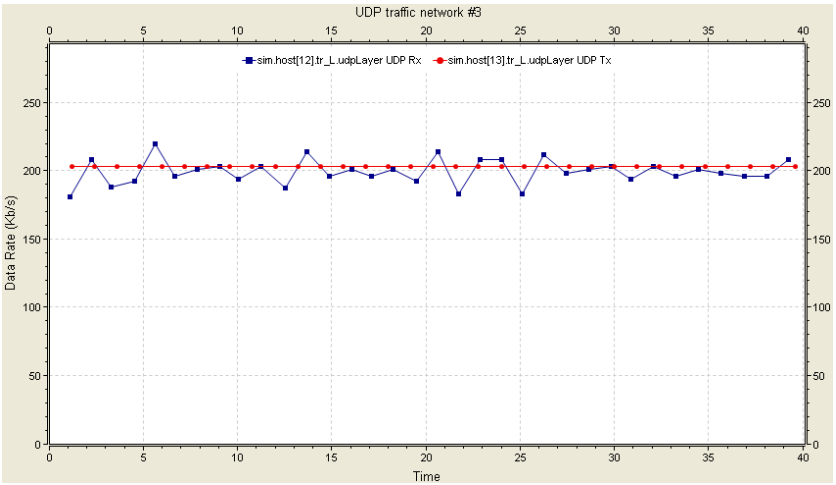
The figures show that EGADIA algorithm has good performance in this scenario in terms of UDP traffic, while CPSA fast algorithm is strongly dependent on the frequency of the updates received from the GREM. If the updates are frequently received (every 3 seconds) from the GREM, the results obtained with CPSA fast are even better than the ones of EGADIA algorithm. However, increasing the interval between two updates, the performance will degrade reducing the amount of UDP traffic correctly received.



(a)

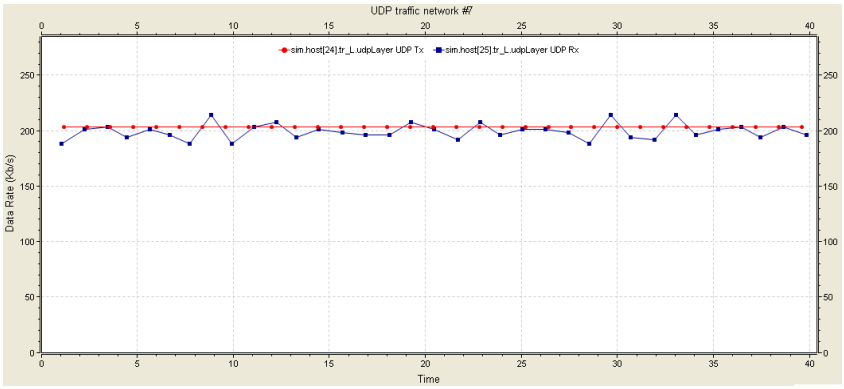


(b)

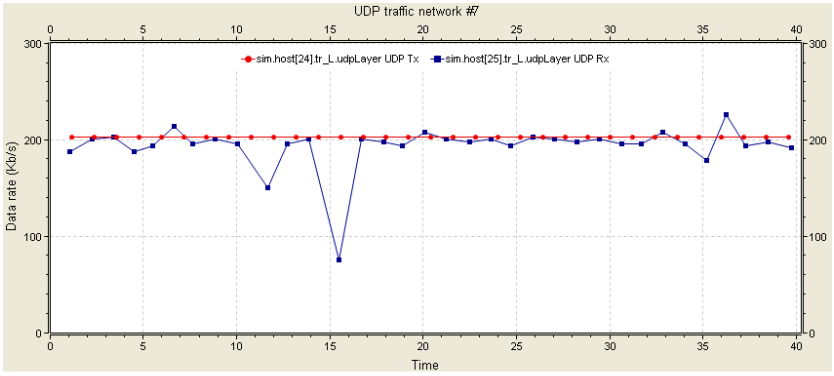


(c)

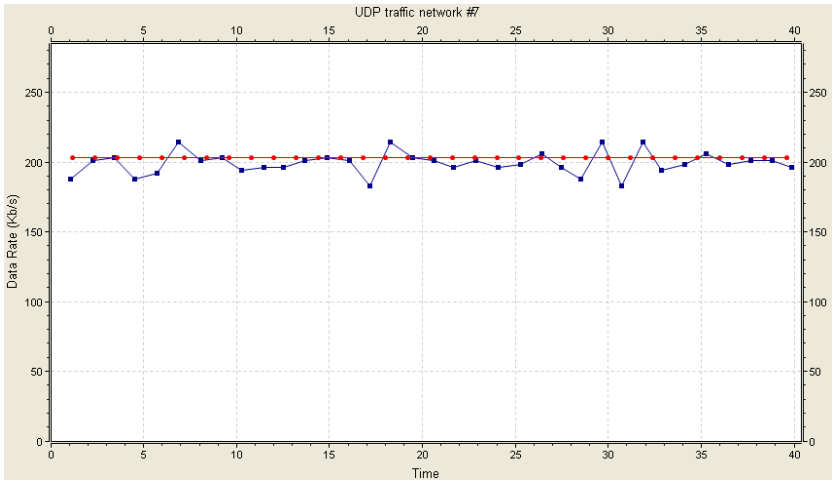
Figure 7-15: UDP data rate between two users of Network #3 when applying (a) EGADIA algorithm, CPSA fast algorithm with updates from GREM each 10 seconds (b) and CPSA fast algorithm with updates from GREM each 3 seconds (c).



(a)



(b)

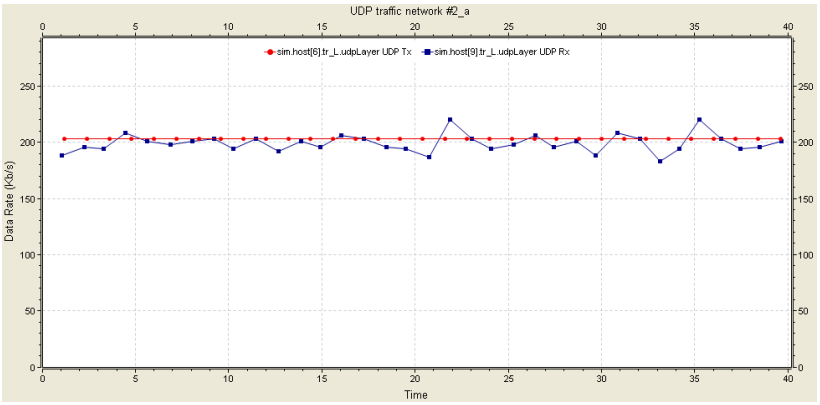


(c)

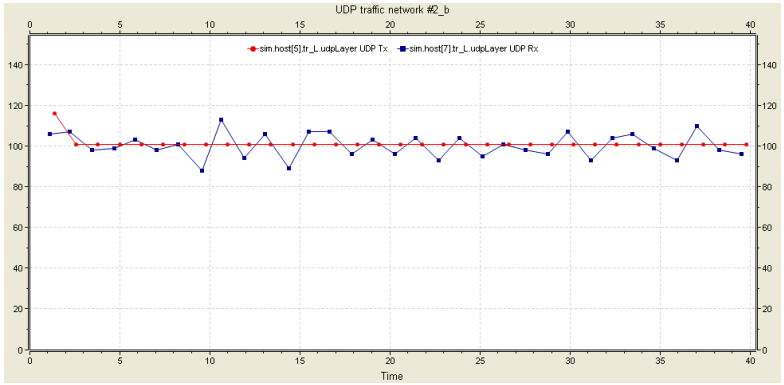
Figure 7-16: UDP data rate between two users of Network #7 when applying (a) EGADIA algorithm, CPSA fast algorithm with updates from GREM each 10 seconds (b) and CPSA fast algorithm with updates from GREM each 3 seconds (c).

Figure 7-17 shows traffic in network #2 where three fluxes of data with different priorities were transmitted. The transmission of the traffic flux with highest priority is the last one to start. The

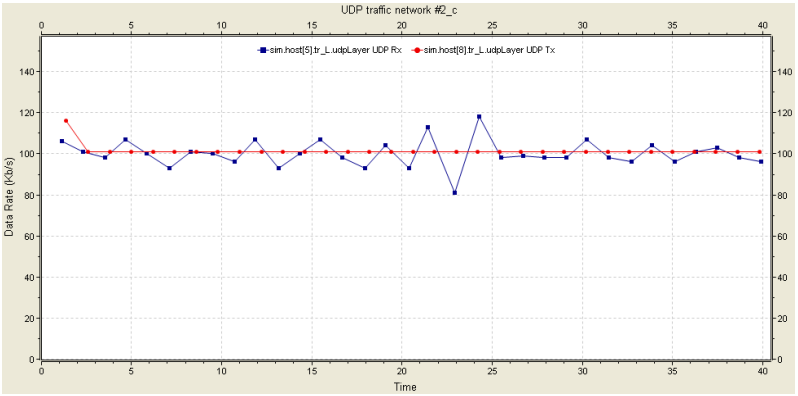
figures are related to the use of the EGADIA algorithm, but similar results have been obtained with the CPSA fast algorithm.



(a)



(b)

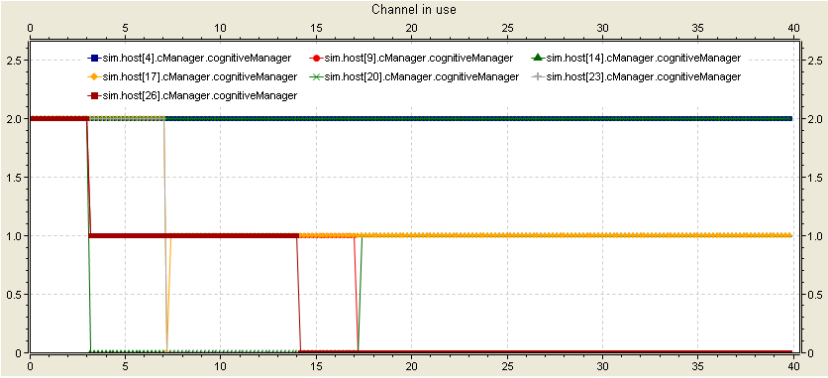


(c)

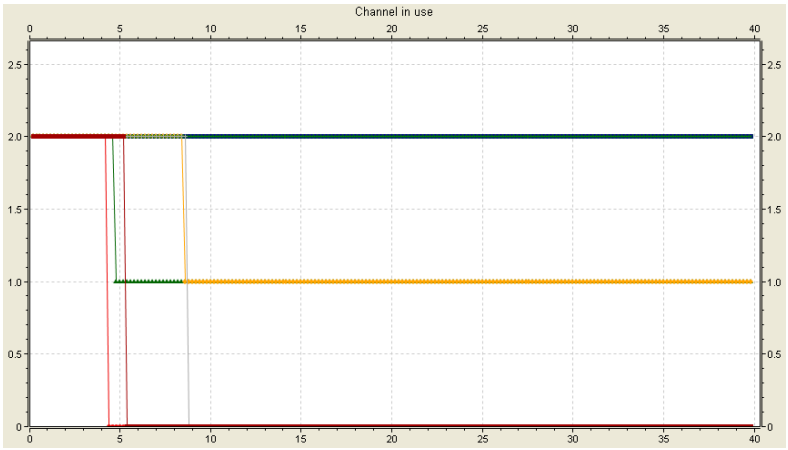
Figure 7-17: Data rate of three different fluxes of data in the same network. In (b) and (c) the data exchanged are low priority data, while in (a) they are high priority data. Transmission (a) starts later with respect to transmissions (b) and (c). Looking at red curves in (b) and (c) it is possible to see the reduction of resources allocated by NH when (a) starts.

As evident, when a node requires high priority traffic in a network where other links are already active, the NH proceeds to reduce the resources allocated to less important links to guarantee full data rate to the one with the higher priority.

Finally, Figure 7-18 compares the number of channel changes of the networks during the scenario for EGADIA and CPSA fast (with updates from GREM every 3 seconds).



(a)



(b)

Figure 7-18: Changes of frequencies channels in use by each network when applying (a) EGADIA and (b) CPSA fast algorithms. The x axis represents time, the y axis the possible channels (0, 1 and 2) and each colored curve represents the channel changes of a network.

As expected (and already showed by high-level simulations presented in [51]), EGADIA algorithm is inferior to CPSA fast, requiring 7 changes of frequency channels as opposed to the 5 changes required by CPSA. Additionally, CPSA converges more quickly to a stable configuration.

7.4.3 Summary

The prototype discussed in this chapter intends to validate the developed REM based architecture and protocols for MANETs. The results confirmed the possibility to integrate the REM concept in MANET scenarios in order to improve the effectiveness of the networks and to guarantee a better use of the spectrum. Moreover, the analysis of the performance of the RRM algorithms in scenarios with full implemented nodes allowed to confirm the considerations and the simulation results presented in [51].

8 Conclusions

Future wireless networks inevitably yield increased network performance and more optimized usage of the available network resources in order to accommodate the ever-increasing users' needs. Additionally, the development of novel wireless network solutions emphasizes the necessity of embedding self-x network features such as (re)configuration, adaptation, optimization etc. This imposes new challenges on the wireless networks designers, engineers and researchers looking for the best possible technology able to deliver the mentioned requirements. The wireless networks community currently witnesses increased efforts in the areas of DSA and CR, which can potentially overcome the spectrum scarcity problem and facilitate fair and optimal usage of the most important wireless networks resource, i.e. the wireless spectrum.

The FARAMIR project focuses on the concept of Radio Environment Map (REM), which allows increased radio environmental awareness for future wireless networks. The REMs are a key enabling solution for practical deployment of DSA and CR networks. They are envisioned as anchors able to collect, store and process valuable radio field information fostering benefits for all involved stakeholders, i.e. operators (e.g. minimization of drive tests, optimized network planning, increased fault detection etc.), regulators and dedicated public bodies (e.g. tracking compliance to regulations, estimation of the frequency planning effectiveness etc.) and, finally, users (e.g. diversified services, higher QoS etc.).

This document elaborates in detail the FARAMIR innovations in the field of REMs throughout the entire project duration. FARAMIR envisions REMs as technical enablers, thus a proof-of-concept of its feasibility and deployment was set as a benchmark. In this sense, FARAMIR proposed, designed and prototyped a novel, REM-based, backend technology that is able to fulfil the REM requirements. The technology relies on a complementary combination of a database-based and sensing-based approaches along with a modular and scalable REM processing unit that represents the cornerstone for the potential REM beneficiaries supplying them with accurate and up-to-date radio field information. The developed solution is flexible and transparent allowing its easy integration and deployment in operational networks. Additionally, FARAMIR has also developed a specific low-cost hardware solution for wideband spectrum sensing in order to prove the possibility of integrating spectrum sensing in handheld mobile devices and, also, show the complementarities of the sensing with the database-based approaches (e.g. spectrum sensing can provide up-to-date information in a centralized database facilitating propagation models estimation, spectrum occupancy estimation etc.)

The potential beneficiaries of the REM backend technology are manifold. This document elaborates in details specific instantiations of the proposed REM approach in various prototypes that target hot research topics today (such as femtocells, LTE in TVWS etc.). The prototyped solutions were tested in order to practically demonstrate the benefits of the REM usage. They clearly show the practical feasibility and application of the FARAMIR REM concept and its ability to efficiently deal with the imposed challenges in front of future wireless networks.

A. Deployment Results for Indoor REM Construction

In addition to the results presented above, several additional field trials and test deployments have been conducted in order to better understand the fundamental issues related to construction of accurate radio environment maps. Results from outdoor measurements and experiments and selected results from indoor measurements are summarized in the project deliverable D3.3. In addition to those, the project consortium as conducted extensive experimental campaigns related to indoor REM construction. In this appendix we briefly discuss one of these campaigns.

We have carried out the testbed deployment in a typical office space to study the implications of device sensitivity, sampling rates, and the spatial dependencies on REMs in a representative indoor environment. As described in the introduction our testbed fulfils the major requirements enabling the experimental research of indoor radio environment maps. These are: the diversity of the environment (e.g. different structures of the walls, diverse furniture placements in the rooms) and the space covered by the testbed; heterogeneity of spectrum sensors; support for different user activity and mobility patterns; and operation over multiple frequencies, enough to accommodate several independent channels.

In our deployment setup, we have used 60 TelosB nodes, 20 USRP2 boards and 8 WARP boards. Figure A-1 shows the testbed deployment map in the office building consisting of five rooms. We have also deployed embedded PCs interfaced to these devices. The embedded PCs are part of the office LAN infrastructure so that the measurements can be controlled centrally through a remote machine.

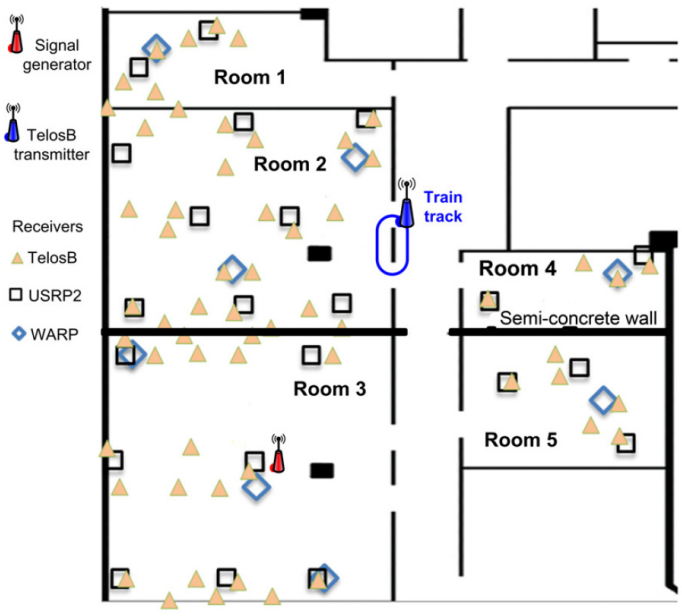


Figure A-1: The deployment map of different devices in an area of 12 m × 20 m.

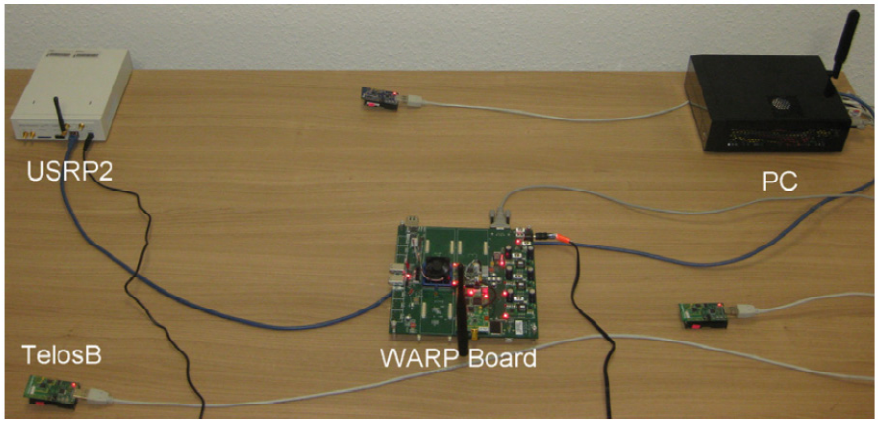


Figure A-2: Snapshot of a typical setup of the MCDs on an office table with a WARP board, a USRP2 board and three TelosB nodes attached to a PC. All the devices are uniquely accessible remotely via the LAN.

Figure A-2 shows a typical deployment setup on an office table. All the considered spectrum sensors or MCDs operate in 2.4 GHz ISM band. The WARP SDR boards and USRP2 boards have a sensing bandwidth of 22MHz and 20MHz respectively. The TelosB nodes operate as MCDs with sensing bandwidth of 5 MHz. It should be noted that WARP and TelosB boards report the spectrum readings directly in terms of RSSI values, whereas USRP2 boards use two input channels for I and Q samples, which are then converted on the host PC into the power readings. Our choice of MCDs clearly falls into three distinct classes with different sensitivity levels and hardware capabilities. WARP board is a high end SDR platform providing extremely fast spectral data samples, USRP2 is a medium grade SDR platform, while TelosB represents a low-end device for spectrum sensing.

We have carefully profiled and then calibrated to the linear operating range the considered types of spectrum sensors by feeding a referenced signal from an Agilent E4438C signal generator directly over a coaxial cable. We have observed that the inconsistencies and biases among the devices of the same type are acceptably low. We also observed that USRP2 devices show strong non-linear behaviour for low received powers, especially for levels below -75 dBm. Compared to the external monopole antennas for USRP2 and WARP boards, TelosB nodes have obviously slightly higher attenuation due to inverted F microstrip antenna.

As signal sources we used a programmable Agilent's E4438C signal generator and TelosB nodes. In this work for the continuous transmissions the E4438C generator was configured to produce QPSK modulated signal with 20 Msps and transmission power of 20 dBm. TelosB nodes were used to generate multi-source and moving signals, as well as periodic ON-OFF patterns following non-uniform distributions. These signals were produced using the test mode of CC2420 radio transceiver chip on TelosB. The chip provides a continuous 5MHz wide signal with most of the power concentrated in a bandwidth of 2 MHz. The transmit power was set to 0 dBm.

Indoor propagation characteristics play one of the central roles in a number of application scenarios for REMs. Typically the received power levels indicate a high variance in the measurements obtained by closely located sensors of the same, as well as of the different, types. This is due to the effect of propagation multipath, as one can observe non-linear and even non-monotonically decreasing function of the distance. However, our results also show that despite these complications time averages of received powers can be quite accurately estimated, especially if further information about the radio environment, such as building floor plans, is available.

As an illustration we fit the propagation data obtained from the testbed to the linear equation

$$Pi(dBm) = K - \alpha \log10(d_i) - \beta C_i,$$

where Pi(dBm) is the power received by node i, d_i is the distance between the sender and node i, and C_i is any further covariate value for node i to be used in the model. We have experimented with linear fitting (a) using only information on the distances d_i to the transmitter, and (b) adding C_i as the number of walls between the sender and the node i. All walls are assumed to be homogenous with β corresponding to the drop in the received power per wall. This could obviously be generalized further by enriching the fitting process or the assumed prior knowledge on the building structure. The results for the fitting process are illustrated separately for different MCD types in Figure A-4.

Our experiments have shown that the linear fitting with information only on the locations of the receivers resulted in average estimation errors of 5.68 dB, 5.71 dB and 7.19 dB for WARPs, USRP2s and TelosB nodes respectively. The additional data on the number of walls between the transmitter and the receivers reduced the respective errors to 3.78 dB, 5.3 dB and 6.8 dB. These results indicate that, as expected, the classical propagation models do not fit well the indoor environment and more complex, maybe even non-linear models, have to be dynamically constructed or tuned to be applicable to the particular indoor conditions should high accuracy be needed. However, the results also indicate that for rough interference management decisions suitably calibrated propagation models constructed dynamically from measurements also provide much higher accuracies than predefined models (for which errors of tens of dBs are typically reported).

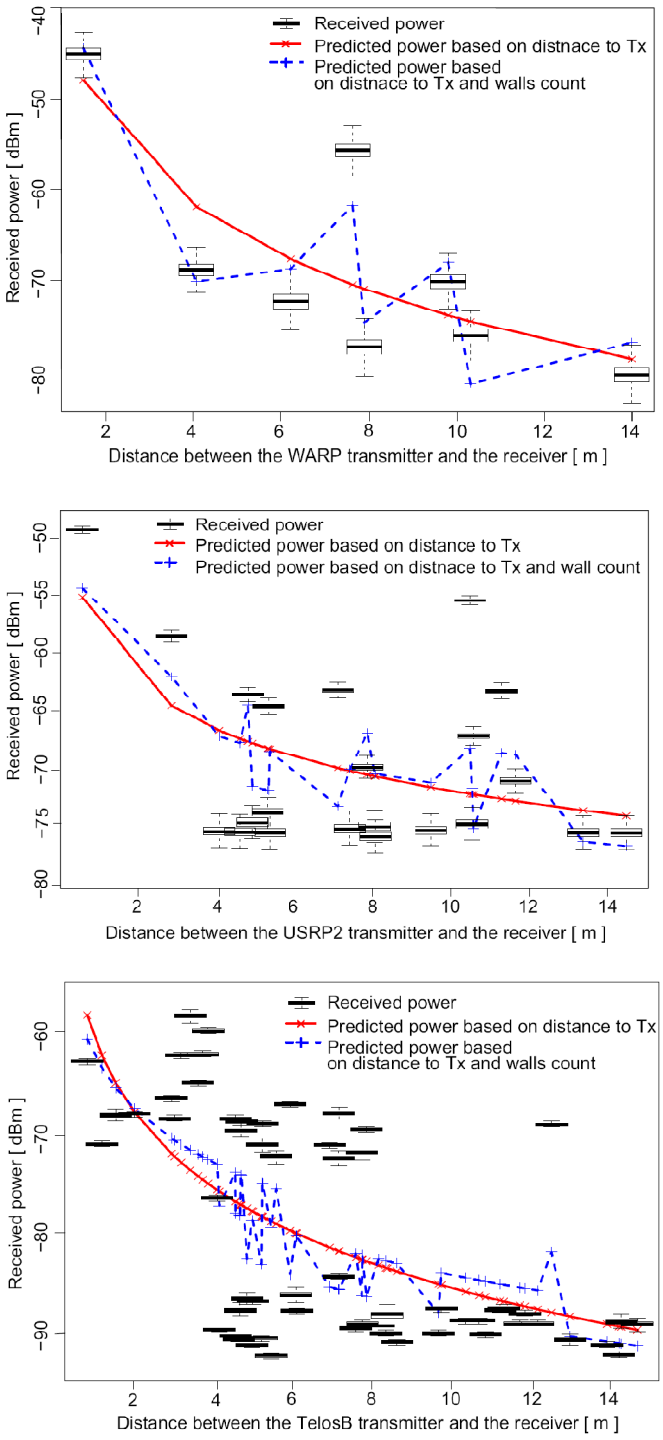


Figure A-4: Boxplots of the received power levels for the spectrum sensors at different distances to the transmitter, and results of linear fitting using the information on distances to the Tx and the walls. The figures show (from the top) the data for WARP boards, USRP2 devices and TelosB nodes.

We shall now illustrate results regarding the temporal structure of the power levels as observed by spectrum sensors. As discussed in the earlier deliverables (such as D3.3, D5.2 and D4.1) the basic characterizations of a duty cycled transmitter are the durations of the active (ON) and inactive (OFF) periods. The accurate estimation of the duty cycles, as well as ON-OFF activity periods can significantly enhance the channel sensing and selection processes for DSA networks.

Our testbed allows detecting signals with fairly high temporal resolution that is appropriate to record both packet-level and session-level behaviors in the packet-based networks, for example IEEE 802.11g wireless networks. We have also programmed the TelosB nodes to generate energy levels with timings accurately following various ON-OFF distributions. These signals are useful to systematically study in practice the applicability of various sensing algorithms. The minimal duty cycle duration for these signals is 100 ms.

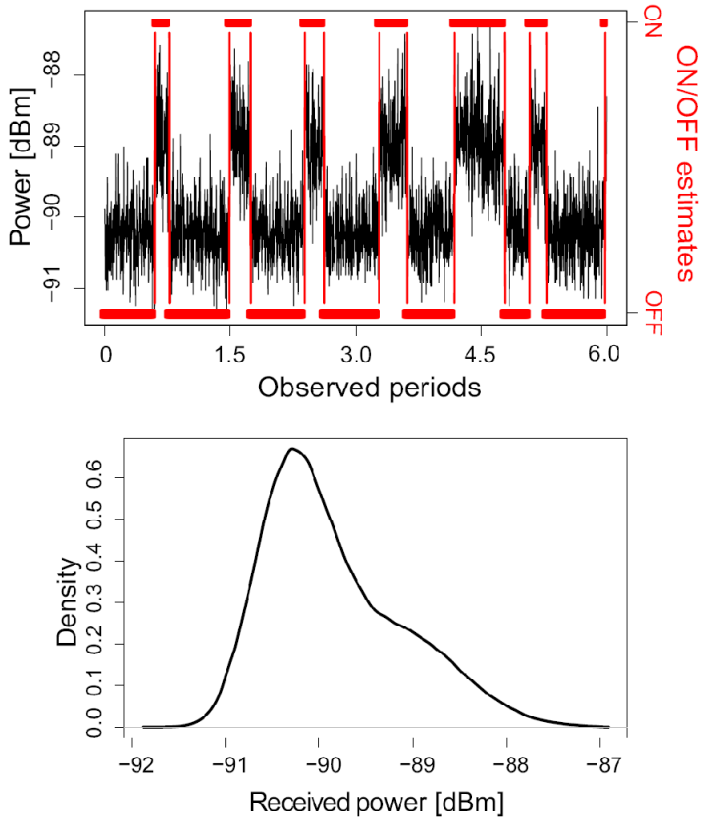


Figure A-5: The output of a WARP board receiving a weak signal, located at a distance of 11m from the transmitter across the semi-concrete wall. The top figure shows the received power and the estimated ON-OFF states of the transmitter. The lower figure portrays the estimated density of the readings.

Figure A-5 shows ON-OFF activity patterns obtained from a single transmitter, which is a TelosB node generating a narrowband 5MHz signal with ON-OFF periods following a gamma distribution. The duty cycle of the signal is 25%, with average ON period of 1.25 seconds and variance of 20%. The results are obtained from two WARP boards placed at different distances from the transmitter. As shown in the figure the density function of the received power is not bimodal in an obvious way, and additional processing is needed to estimate the distributions of the received power corresponding to ON and OFF periods for the weak signal. Depending on the application scenario mechanisms of different complexity have to be applied for data post-processing. The figure shows results from using Hidden Semi-Markov Model (see [40] for a discussion on use of Semi-Markov models and their variants for modeling spectrum use over time) for this purpose.

As a summary, the results from the testbed and associated trials discussed in the main text as well as in this appendix show that constructing accurate indoor radio environment maps including rich information about the properties of the propagation environment and activity patterns of transmitters is indeed feasible. However, care must be applied when choosing the appropriate data processing solutions, as well as in dealing with the heterogeneity of the used MCDs. For latter, the modular REM architecture developed in the project provides powerful tools that can be applied in plug-and-play fashion to a number of application scenarios.

Glossary and Definitions

Term	*Description*
ADC	Analog-to-Digital Conversion
AE	Absolute Error
AFE	Analog Front-End
AIC	Akaike Information Criterion
AMSMC	Adaptive Modified Shepard's Method Covariance
API	Application Programming Interface
BLOB	Binary Large OBjects
BW	BandWidth
CD	Communication Device
CDF	Cumulative Distribution Function
CMOS	Complementary Metal-Oxide-Semiconductor
CR	Cognitive Radio
CRLB	Cramer-Rao Lower Bound
CRM	Cognitive Resource Manager
CRRM	Cognitive Radio Resource Manager
DAC	Digital-to-Analog Conversion
DFE	Digital Front-End
DREM	Dynamic Radio Environmental Map
DSA	Dynamic Spectrum Access
DVB-T	Digital Video Broadcasting - Terrestrial
ECDF	Empirical Cumulative Distribution Function
EM	Expectation Maximization

Term	*Description*
EM-GMM	EM - Gaussian Mixture Model
EMGMM-ML	EMGMM – Maximum Likelihood
eNodeB	evolved Node B
EPDF	Empirical Probability Distribution Function
FAR	FFT Averaging Ratio
FARAMIR	Flexible and spectrum Aware Radio Access through Measurements and modelling In cognitive Radio system
FE	Front-End
FFT	Fast Fourier Transform
FIFO	First In First Out
FPGA	Field-Programmable Gate Array
GMM	Gaussian Mixture Model
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GREM	Global REM
GUI	Graphical User Interface
HeNB	Home eNodeB
HOS	Higher Order Statistics
ID	IDentifier
IDW	Inverse Distance Weighting
IDWM	Inverse Distance Weighting Modified
IF	Intermediate Frequency
IP	Internet Protocol
LNA	Low Noise Amplifier

Term	*Description*
LREM	Local REM
LS	Least Squares
LTE	Long Term Evolution
MAC	Medium Access Control
MAE	Mean Absolute Error
MANET	Mobile Ad-hoc NETwork
MCD	Measurement Capable Device
MDL	Minimum Description Length
MHT	Mean Holding Time
ML	Maximum Likelihood
MMCX	Micro-Miniature CoaXial
MME	Mobility Management Entity
MMSE	Minimum Mean Squared Error
MP	Measurement Point
MSM	Modified Shepard's Methods
MSMC	Modified Shepard's Method Covariance
NH	Network Head
NOC	Network-on-Chip
OFDM	Orthogonal Frequency Division Multiplexing
OFDMA	Orthogonal Frequency Division Multiple Access
PA	Power Amplifier
PC	Personal Computer
PCB	Printed Circuit Board

Term	*Description*
PDF	Probability Distribution Function
PGW	Packet-data-network GateWay
QoS	Quality-of-Service
RA	Random Access
REM	Radio Environmental Map
REM SA	REM data Storage and Acquisition unit
RF	Radio Frequency
RIF	Radio Interference Field
RIFE	Radio Interference Field Estimation
RND	RaNDom
ROC	Receiver Operating Characteristic
RRM	Radio Resource Management
RSS	Received Signal Strength
RSSI	Received Signal Strength Indicator
SCALDIO	SCALable raDIO
SCR	Secondary Cognitive Radio
SDK	Software Development Kit
SDR	Software Defined Radio
SDRAM	Synchronous Dynamic Random-Access Memory
SE	Sensing Engine
SGW	Serving GateWay
SINR	Signal-to-Interference-Noise-Ratio
SMB	SubMiniature version B

Term	*Description*
SPA	SPectrum Analyzer
SpHo	Spectrum Handover
SQL	Structured Query Language
SRAM	Static Random-Access Memory
TDD	Time Division Duplex
TDMA	Time Division Multiple Access
TETRA	TErrestrial Trunked RAdio
TI	Texas Instruments
TVWS	TV White Space
UDP	User Datagram Protocol
UDP/IP	User Datagram Protocol / Internet Protocol
UE	User Equipment
UHF	Ultra High Frequency
UPC	Universitat Politechnica de Catalunya
USB	Universal Serial Bus
USRP2	Universal Software Radio Peripheral 2
UF	Utility Function
VISA	Virtual Instrument Software Architecture
WARP	Wireless open-Access Research Platform
WP	WorkPackage
WPF	Windows Presentation Foundation
XAML	eXtensible Application Markup Language
XML	eXtensible Markup Language

References

[1] Deliverable D2.4: Final System Architecture. EC FP7-248351 FARAMIR Project. December 2011.

[2] Deliverable D6.1: Prototype Requirement Analysis. EC FP7-248351 FARAMIR Project. February 2011.

[3] D. Denkovski, V. Atanasovski and L. Gavrilovska, "Efficient Mid-end Spectrum Sensing Implementation for Cognitive Radio Applications based on USRP2 Devices," *COCORA2011*, Budapest, Hungary, April 2011.

[4] D. Denkovski, V. Rakovic, M. Pavloski, K. Chomu, V. Atanasovski and L. Gavrilovska, "Integration of Heterogeneous Spectrum Sensing Devices Towards Accurate REM Construction," *IEEE WCNC 2012,* Paris, France, April 2012.

[5] D. Shepard, "A two-dimensional interpolation function for irregularly-spaced data," *Proceedings of the ACM National conference*, 1968, pp. 517-524.

[6] R. J. Renka "Multivariate interpolation of large sets of scattered data," *ACM Transactions on Mathematical Software*, 14(2), 1988, pp. 139–148.

[7] R. K. Martin, R. W. Thomas, "Algorithms and bounds for estimating location, directionality, and environmental parameters of primary spectrum users," *IEEE Transactions on Wireless Communications 8(11)*, pp. 5692-5701, 2009.

[8] A. Nathan. *Windows Presentation Foundation Unleashed*. Sams Publishing, December 21, 2006.

[9] A. Mackey, *Introducing .NET 4.0: With Visual Studio 2010*, Apress ed. 1, February 1, 2010.

[10] M. Dalal and A. Ghoda, *XAML Developer Reference*, Microsoft Press, December 22, 2011.

[11] J. Fawcett, L. R. E. Quin and D. Ayers, *Beginning XML*, Wrox ed.5, July 10, 2012.

[12] O. Holthe, *DirectX 11 Programming*, Grid Publishing, 2011.

[13] J. Albahari and B. Albahari, *C# 4.0 in a Nutshell: The Definitive Reference*, O'Reilly Media ed. 4, February 10, 2010.

[14] Deliverable D3.2: Spectrum Sensing Engine and Prototype Measurements. EC FP7-248351 FARAMIR Project. June 2011.

[15] D. Denkovski et al., "Reliability of a Radio Environment Map: Case of Spatial Interpolation Techniques," *CrownCom 2012*, Stockholm, Sweden, June 2012. (accepted)

[16] D. Denkovski, M. Angjelicinoski, V. Atanasovski and L. Gavrilovska, "Practical assessment of RSS-based localization in indoor environments," *IEEE MILCOM 2012*, Orlando, Florida, 29 Oct-31 Nov, 2012. (submitted).

[17] S. M. Kay, *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*. Prentice Hall, 1993.

[18] L. Lin and K. Yang, "Best Linear Unbiased Estimator Algorithm for Received Signal Strength Based Localization," *European Signal Processing Conference (EUSIPCO 2011)*, Barcelona, Spain, September 2011.

[19] S. Xiaohongeng and H. Yu-Hen, "Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks," *IEEE Transactions on Signal Processing*, vol.53, no.1, January 2005, pp. 44- 53.

[20] J. Nelson, M. Hazen and M. Gupta, "Global Optimization for Multiple Transmitter Localization," *IEEE MILCOM 2006*, October 2006.

[21] J. Nelson, J. Almodovar, M. Gupta and W. Mortensen, "Estimating multiple transmitter locations from power measurements at multiple receivers," *IEEE International Conference on Acoustics, Speech and Signal Processing 2009 (ICASSP 2009)*, April 2009.

[22] T. Hastie, R. Tibshirani and J. Friedman, *The Elements of Statistical Learning*, New York: Springer-Verlag, 2001.
[23] J. Bazerque and G. Giannakis, "Distributed Spectrum Sensing for Cognitive Radio Networks by Exploiting Sparsity," *IEEE Transactions on Signal Processing*, vol.58, no.3, March 2010, pp.1847-1862.
[24] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *J. Roy. Stat. Soc., Series B*, vol. 58, no. 1, 1996, pp. 267–288.
[25] N. Miliou, A. Moustakas and A. Polydoros, "Interference Source Localization and Transmit Power Estimation Under Log-Normal Shadowing," *European Signal Processing Conference (EUSIPCO 2011)*, Barcelona, Spain, September 2011.
[26] L. Fenton, "The sum of log-normal probability distributions in scattered transmission systems," *IRE Trans. Commun Systems*, vol. 8, 1960, pp.57-67.
[27] I. Dagres, A. Polydoros, D. Denkovski, M. Angjelicinoski, V. Atanasovski and L. Gavrilovska, "Algorithms and Bounds for Energy-based Multi-source Localization in Log-normal Fading," *IEEE GLOBECOM 2012 Workshop on Green Internet of Things (G-IoT)*, 2012. (submitted)
[28] P. McKenzie and M. Alder, "Selecting the optimal number of components for a Gaussian mixture model," *IEEE International Symposium on Information Theory*, 27 Jun-1 Jul 1994.
[29] S. Pollin, E. Lopez, A. Antoun, P. Van Wesemael, L. Hollevoet, A. Bourdoux, A. Dejonghe and L. Van der Perre, "Digital and Analog Solution for Low-power Multi-band Sensing," *IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, Singapore, 2010.
[30] A. Dejonghe et al., "Versatile spectrum sensing on mobile devices?," *IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, Singapore, 2010.
[31] S. Pollin et al., "An integrated reconfigurable engine for multi-purpose sensing up to 6 GHz," *IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, Aachen, Germany, May 2011. (demo)
[32] P. Van Wesemael et al., "Performance Evaluation of Sensing Solutions For LTE and DVB-T," *IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, Aachen, Germany, May 2011.
[33] S.Pollin, L. Hollevoet, F. Naessens, P. Van Wesemael, A. Dejonghe and L. Van der Perre, "Versatile sensing for mobile devices: cost, performance and hardware prototypes," *ACM Mobicom - coronet workshop (invited)*, ACM, Las Vegas, Sep 2011.
[34] Deliverable D2.2: Scenario definitions. FARAMIR EC FP7-248351 project. August 2010.
[35] Deliverable D1.1: Models, Scenarios, Sharing Schemes. QUASAR EC FP7-248303 project. June 2010.
[36] M.I. Rahman, A. Behravan, H. Koorapaty, J. Sachs and K. Balachandran, "License-exempt LTE systems for secondary spectrum usage: scenarios and first assessment", in *Proceedings of the 2011 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, Aachen, Germany, May 2011.
[37] Federal Communication Commission (FCC), "Second Memorandum Opinion and Order In the Matter of Unlicensed Operation in the TV Broadcast Bands, Additional Spectrum for Unlicensed Devices Below 900 MHz and in the 3 GHz Band," September 23, 2010, Document 10-174.
[38] V. Atanasovski, J. van de Beek, A. Dejonghe, D. Denkovski, L. Gavrilovska, S. Grimoud, P. Mähönen, M. Pavloski, V. Rakovic, J. Riihijärvi and B. Sayrac, "Constructing Radio Environment Maps with Heterogeneous Spectrum Sensors," in *Proceedings of the 2011 Symposium on Dynamic Spectrum Access Networks (DySPAN 2011)*, Aachen, Germany, May 2011.
[39] R. I. C. Chiang, G. B. Rowe and K. W. Sowerby, "A quantitative analysis of spectral occupancy measurements for cognitive radio," *IEEE 65th Vehicular Tech. Conf. (VTC 2007 Spring)*, April 2007.

[40] M. Wellens, J. Riihijärvi and P. Mähönen, "Modelling primary system activity in dynamic spectrum access networks by aggregated ON/OFF-processes," *Fourth IEEE Workshop on Networking Technologies for Software Defined Radio Networks (SDR 2009)*, June 2009.
[41] M. López-Benítez and F. Casadevall, "Modeling and Simulation of Time-Correlation Properties of Spectrum Use in Cognitive Radio,"*6th International ICST Conference on Cognitive Radio Oriented Wireless Networks (CrownCom 2011)*, Yokohama, Japan, 31May- 03 June 2011.
[42] Deliverable D6.1: Prototype Requirement Analysis. EC FP7-248351 FARAMIR Project. February 2011. (restricted)
[43] Deliverable D2.3: Preliminary System Architecture. EC FP7-248351 FARAMIR Project. December 2010. (restricted).
[44] ANRITSU: Spectrum Master™ MS2721B, MS2723B, and MS2724B, Programming Manual.
[45] NI-VISA Support - National Instruments (http://sine.ni.com/psp/app/doc/p/id/psp-411/lang/en)
[46] ANRITSU Application Note: "Using MATLAB applications with MS268xA signal analyzer", Application note number MS269xA_EF6101 http://www.anritsu.com/en-us/downloads/application-notes/application-note/dwl3601.aspx.
[47] ANRITSU: "Programming manual for MS272XB series spectrum analyzers", http://www.anritsu.com/search/en-US/DownloadsSearch.aspx?site=us_www2_downloads&q=MS272xB&sort=date:D:L:d1&filter=p
[48] M. López-Benítez and F. Casadevall, "Spectrum Survey in Urban Environment: UPC Campus Nord," Barcelona, Spain, *Technical Report*. December 2010 (available in http://spes.upc.edu/)
[49] Deliverable D2.4: Final System Architecture. EC FP7-248351 FARAMIR Project. December 2011.
[50] R. Scopigno and H.A. Cozzetti, "GNSS Synchronization in Vanets," *3rd International Conference on New Technologies, Mobility and Security (NTMS)*, Cairo, Egypt, 2009.
[51] Deliverable D5.2: Evaluation of Selected Measurement-based Techniques. EC FP7-248351 FARAMIR Project. February 2012.
[52] WARP website http://warp.rice.edu/trac
[53] J. Ansari, X. Zhang, A. Achtzehn, M. Petrova and P. Mähönen, "A Flexible MAC Development Framework for Cognitive Radio Systems", Proceedings of IEEE Wireless Communications & Networking Conference, Cancun, Mexico, 2011.
[54] WARP reference design v16.1 http://warp.rice.edu/trac/wiki/OFDMReferenceDesign