Network performance & Quality of service in data networks involving spectrum utilization techniques

By
Hassan Ali Fadel Hassan Mohammed Darwish

School of Engineering and Sustainable Development
Faculty of Technology

De Montfort University

A thesis submitted in partial fulfilment of the requirements for the degree
of
Doctor of Philosophy
May 2017

To my father's soul

ABSTRACT

This research has developed technique to improve the quality of service in wireless data networks that employ spectrum utilization techniques based on Cognitive Radio. Most multiple dimension implementations focus on maximizing the Successful Communication Probability SCP in order to improve the wireless network utilization. However this usually has a negative impact on the Quality of Service, since increasing the SCP leads to increasing signal interference and Packet Loss, and thus network performance deterioration. The Multiple Dimension Cognitive Radio technique is a new technique, proposed in this thesis, that improves the Cognitive Radio Networks (CRN) efficiency by giving opportunity to secondary users (Unlicensed users) to use several dimension such as time, frequency, modulation, coding, and antenna directionality to increase their opportunity in finding spectrum hole.

In order to draw a balance between improving the networking utilization and keeping the network performance at an acceptable level, this thesis proposes a new model of multiple dimension CR which provides a compromise between maximizing the SCP and network throughput from one side and keeping the QoS within the accepted thresholds from the other side. This is important so as to avoid network performance degradation which may result from the high user density in single wireless domain as a result of maximizing the SCP. In this research, a full Cognitive Radio model has been implemented in the OPNET simulator by developing modified nodes with the appropriate coding which include basic functionality. The Purpose of this model is to simulate the CR environment and study the network performance after applying the controlled multi dimension technique presented here. The proposed technique observes the channel throughput on TCP (Transmission Control Protocol) level, also QoS KPIs (Key Performance Index) like Packet Loss and Bit Error rate, during the operation of the CR multi dimension technique and alerts the system when the throughput degrades below a certain level. The proposed technique has interactive cautious nature which keeps monitoring the network performance and once find evident on network performance deterioration it takes corrective action, terminates low priority connections and releases over utilized channels, in order to keep the performance accepted.

DECLARATION

I declare that the work described in this thesis is original work undertaken by me for the degree of Doctor of Philosophy at De Montfort University, Leicester in the United Kingdom. The work done in this research is my own, original work and that all sources used have been referenced and cited.

Hassan Ali Fadel

Date: May 2017

ACKNOWLEDGEMENT

First of all, I am grateful to Almighty God for giving me the power and the will to finish this work, and enabling me to continue this fantastic experience.

I'm very thankful and grateful to my first supervisor Professor Alistair Duffy for his guidance, patience and encouragement, also for being always understanding and supportive. Working with Professor Alistair was best part in my PhD journey in DMU.

I would also like to thank all Faculty of Technology supervision members specially Professor Raouf Hamzaoui and Dr Cristian Serdean, also the staff of Graduate School Office of De Montfort University.

My sincere regards and thanks to Mr. Andy Drozd (ANDRO Computational Solutions, LLC) for giving me the chance to make use of his work and publications, also giving all support through his team.

Last, but not the least, I am very thankful to my parents for being the permanent source of will and perseverance in my life, also my wife and my lovely kids for their support and patience throughout the journey of my PhD.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

CRN	Cognitive Radio Network
MD-CRN	Multiple Dimension - Cognitive Radio Network
C-MD-CR	Controlled Multiple Dimensions Cognitive Radio
SCP	successful Communication Probability
PU	Primary User
SU	Secondary User
TRA	telecommunications Regularity Authority
DSA	Dynamic Spectrum Access
QoS	Quality of Service
URSI	International Union of Radio Science
ASA	Authorized Shared Access
LSA	Licensed Shared Access
PA	Priority Access
SCP	Successful Communication Probability
TH	Transmission Hyperspace™
TCP	Transmission Control Protocol
BER	Bit Error Rate
PLR	Packet Loss ratio
SNIR	Signal to Noise and Interference Ratio
SONET	Synchronous Optical Network
WDM	Wavelength-division multiplexing
OSI	Open Systems Interconnection
ISO	International Organization of Standardization

CHAPTER ONE

INTRODUCTION

This chapter discusses the wireless spectrum capacity problem and introduces the basic definition and principles of opportunistic techniques; it also reviews the history of the cognitive radio techniques and its modern implementation. The chapter also explains the impact of cognitive radio on the network performance and proposes technique to maintain the quality of service.

1.1 Motivation and problem statement

Electronic communication is one of humanity's greatest inventions, that has changed many aspects of life in the developed and developing world since it appeared during the 19th century as result of the research and development work of Alexander Graham Bell, Guglielmo Marconi and others [1,2]. The evolution of this technology to wireless communication has opened the doors for many applications in fields such as mobile telephone networks, cellular data networks, satellite communication, medical applications, Television broadcast, military applications [3], sensor networks, and it is a popular topic of discussion with both the "Internet of Things" [4] and "Smart Grid" [5].

With increasing utilization, it was found that the main resource, the frequency spectrum, needed to be planned efficiently and to be controlled by single entity authorized to assign different frequency bands to users and applications. Starting 1900s some countries like the UK introduced telecom regularity bodies to regulate and control the usage of the telecommunication resources including the frequency spectrum [6], and to manage the spectrum assignment among different purposes and grant the related licenses, however the modern TRA (Telecommunication Regularity Authority) organizations were introduced in many countries in the last 20 years for same purpose also to regulate the commercial relation between the service providers and users, especially with the proliferation of users and applications, particularly since the 1990s, where internet access technologies and mobile networks introduced many applications that depend on wireless communication [7,8]. New and untraditional spectrum utilization techniques are required to make use of the available resource and optimize its capability to serve more users. Since the nineties the TRA organizations used to follow strict spectrum regulations which sometimes involve auctions to have effective utilization of spectrum and also revenue source for governments, for example the Canadian government has introduced spectrum auction in July 2008 for 105 MHz spectrum which raised 4.25 billion dollars [9].

ITU ICT reports [10] show that number of mobile broadband users is growing rapidly and exceeding the fixed broadband user growth rate. Figure 1-1 shows the mobile broad band population penetration in the period 2008 to 2012, where we can notice that some countries like Asia Pacific countries have penetration approaches 100% of population.

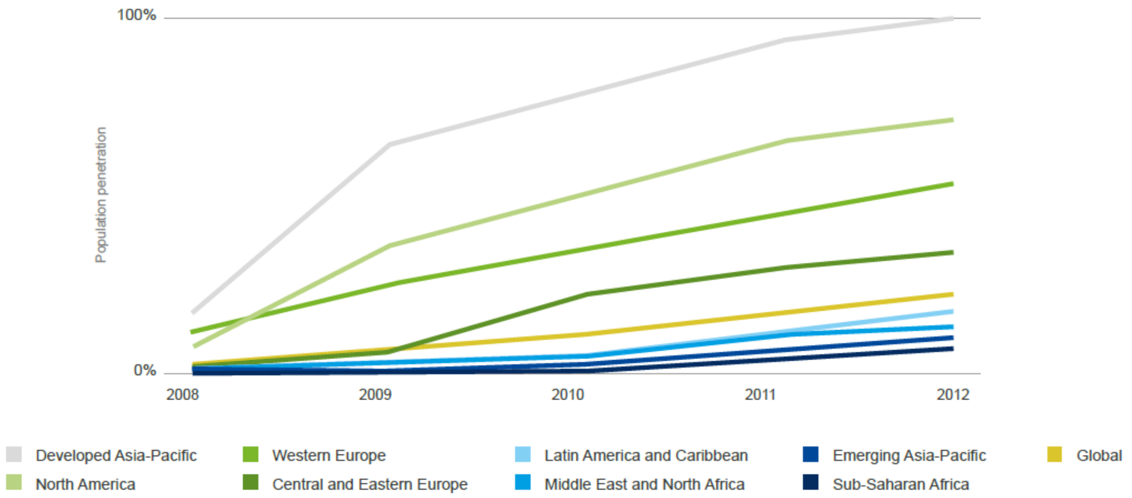


Figure 1-1: Mobile broadband population penetration – ITU ICT report [10]

According to the ITU ICT reports number of commercial mobile broadband subscribers worldwide in 2013 exceeded 1930 million subscribers, which indicates significant load on wireless resources worldwide. Figure 1-2 shows the growth of mobile broadband connections from 2011 and expected growth till 2018.

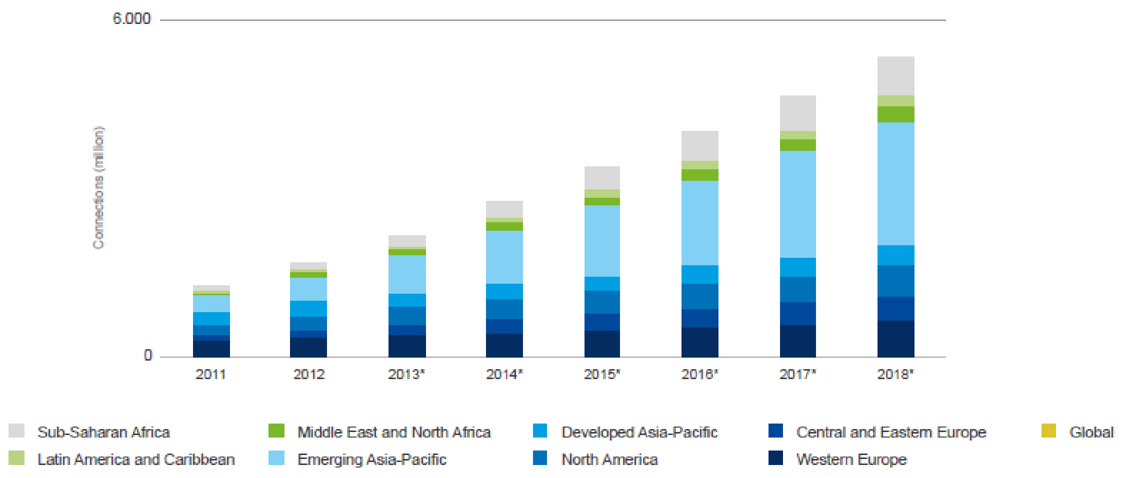


Figure 1-2: Global Mobile broadband connections – ITU ICT report [10]

In traditional wirless planning the spectrum is divided into bands where each band consists of a group of frequencies, and these bands are usually licensed to carry certain services, such as wireless regional area networks WRAN, cellular phone network, broadcast, and satellite, where this design help avoiding harmful interference between different services and users. Usually the spectrum band is allocated to a certain user or specific service however the spectrum measurements and statistics show that if part of the spectrum is used the rest is usually unused which means low spectrum utilization. The spectrum utilization techniques, that decide who uses which portion of the spectrum, and how; will play important role in developing the future wireless networks. The recent trends in wireless communication systems are migrating from centralized management systems into the self organizing systems, where the self-organizing networks consists of wirless/fixed nodes that can instantaneously establish and adapt ad hoc networks [11]. Self-organizing networks depend on Dynamic Spectrum Access (DSA) policy where free spectrum can be allocated and re-assigned to unlicensed users.

This introduces opportunistic usage the spectrum resources that are not utilized by licensed users.

When the spectrum is underutilized, some frequency bands are not fully used and this leads to have spectrum holes. The spectrum hole is defined as a range or band of frequency assigned to a licensed user that has full rights to access this frequency band at any time. These type of users are called “primary users” and have highest priority to access the frequency band which is related to their licenses, and they also called incumbent users. The spectrum holes are considered a waste of spectrum resources and most of modern spectrum applications are trying to avoid the existence of such holes with no proper utilization, however it can happen, at a certain time and in a geographic location. The opportunistic spectrum utilization techniques give a chance to other users, who may not be licensed like the primary users, to access the spectrum during the inactivity of the primary user (PU) to make use of the spectrum holes. These type of users are called “secondary users” (SU) and usually they are not licensed to access the targeted frequency band or they have lower privileges. Such unauthorized access to the spectrum holes improves the spectrum utilization significantly although it seems from the first glance that it infringes on the rights of the primary user because they use same frequency bands of the primary users but not in same time slot as explained later in chapter 2. There are several techniques to improve the spectrum utilization by finding the spectrum holes and exploiting them, one of them is the cognitive radio concept [11].

In a world crowded with wireless applications/users, the cognitive radio concept has become an interesting research area. The cognitive radio concept was developed basically to solve the spectrum utilization problem and allow the secondary users to access and exploit the licensed spectrum, where it is usually implemented to serve group of users (both primary and secondary users) who are connected to single or several controllers This combination of users and controllers are usually located in single domain which introduces Cognitive Radio “Network” as illustrated in figure 1-3. In typical wireless or mobile networks the controller is the base-station which serves all users and receives requests from different users to establish new connection and allows secondary user whenever the primary user is inactive. The network controller grants the access to the secondary users as long as they do not cause interference to primary users, or affect the quality of service (QoS) of the primary users. The QoS is defined here as the overall performance of the wireless user and network, where this performance is measured based on different factors like throughput, data error/loss, Signal to Noise Ratio (SNR), and other factors. The assessment of the QoS differs according to the operation environment and network nature, for example when we assess the performance of data network we may consider TCP throughput and packet loss beside SNR.

Figure 1-3 shows that these users can connect to Cognitive radio Network whether it has infrastructure, for example dedicated base station capable of connecting Cognitive Radio secondary users, or no infrastructure is available, where primary network access can act as infrastructure node capable of connecting Cognitive Radio secondary users to the network controller [12].

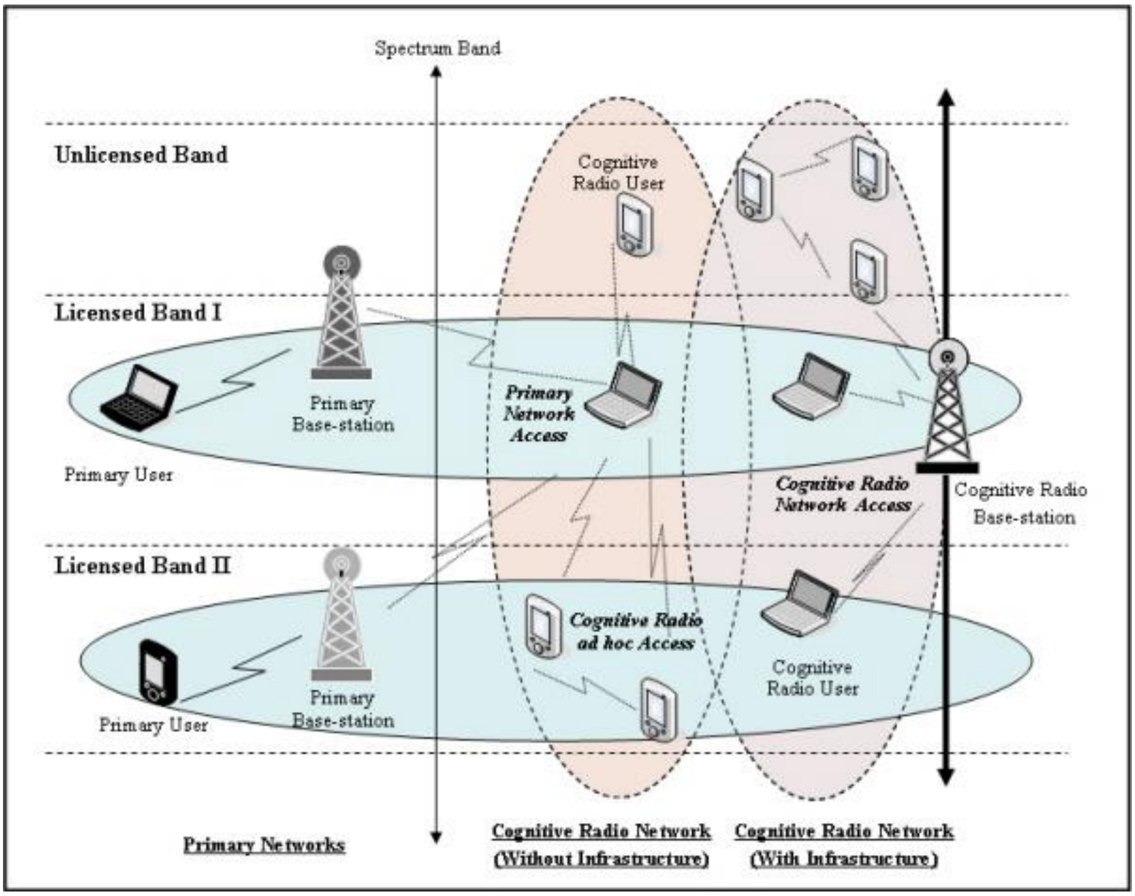


Figure 1-3: architecture example of Cognitive Radio Network [12]

In large scale networks different technologies are combined and integrated to facilitate the information flow from one to another point. Lately bottlenecks in the wireless domain became irritating issue when integrating wireless network with other networks that run technologies with high capacities where booming capacities above 100 Gbps are now achievable [13]. The integration between wireless and wired networks requires sort of consistency in capacities and speeds in order to make use of the high capacity of the wired networks. The work done by the organizations involved in the development of the 100 Gigabit Ethernet (IEEE HSSG, ITU-T SG15 and OIF) indicated that there must be consistency between the transport technologies which integrate with the 100 Gigabit

Ethernet. On the other hand the spectrum utilization measurements, for example the measurements done by Zulfiquar Sayed, Bell Labs and Lucent Technologies [14] showed that wireless networks are facing challenges due to aggressive demands specially over the previous years it was also shown that wired networks transport technologies could introduce transport solutions like SONET and WDM that are capable of supporting such aggressive demands, while wireless networks still cannot afford such high throughput due to several limitations one of them is the limitation on spectrum utilization [15,16].

On the other hand efficient spectrum utilization techniques are required to improve spectrum management and substantially increase the wireless spectrum utilization. This is important because one of the topics for research is identifying better opportunities for internetworking between wireless networks and fixed networks, for example optical transmission backbones which use technologies like WDM increased the line rate from 2.5Gbps to 10Gbps, and recently to 40Gbps [17], also IP routers and layer 3 switches started using 100G Ethernet ports. It is important to note that the consistency in network capacities between the fixed side and wireless side opens the doors for virtually unlimited integration solutions and topologies [18, 19].

In the 1990s, cellular mobile communication expanded significantly, and many applications were developed based on mobile data networks, this included commercial applications related to data services. Recent research indicates that cellular networks are usually over utilized, however other spectrum bands, for example related to radio broadcast, military applications and paging services, are insufficiently utilized [20, 21, 22].

Different aspects related to spectrum management were discussed in the International Union of Radio Science (URSI) Radio Science Bulletin report No 354 [23] which indicated the importance of spectrum management especially in regions which have significant low utilization. The report demonstrated the measurements done by M. Mehdawi and N. Riley [24] showed that many bands consists of unused spectrum specially between 1 and 2 GHz also in TV band, however the GSM 900 and GSM1800 bands always show high utilization. The measurements show that mean occupancy ratio over whole band was as low as 11%, while we have some bands like GSM bands suffer from high utilization that approaches 100% in business peak hours.

In the Federal Communications Commission FCC "Report of Spectrum Efficiency Working Group" [25] a measurement for spectrum utilization in an approximately 700 MHz block was demonstrated to show the variation in spectrum use, where the lower frequency bands had lower utilization, unlike the higher frequencies, also it shows that some bands have continuous occupancy for example television broadcast and others are dynamically occupied. The spectrum utilization shown in Figure 1-4 varies based on the geographical area as measured in 700 MHz block below 1 GHz in Atlanta (A), New Orleans (B) and San Diego (C) as indicated in the Federal Communications Commission Spectrum policy Task Force report [25].

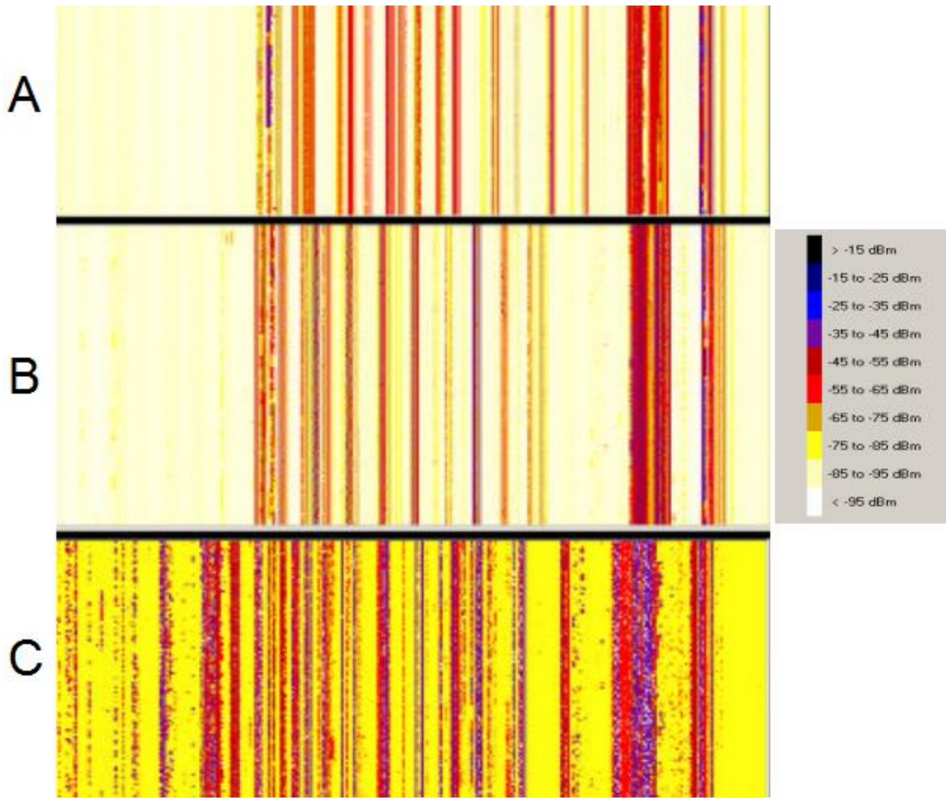


Figure 1-4: FCC Report of Spectrum Efficiency [25] – Spectrum utilization measured in Atlanta (A), New Orleans (B) and San Diego (C) shows occupancy of 700 MHz of spectrum below 1 GHz which varies based on the geographical area

Some spectrum measurements were done recently by mobile network operators and manufacturers indicate that there is tremendous growth in broadband applications data, where some operators and manufacturers estimated the wireless traffic from mobile networks and data applications will grow more than 10 times between 2012 and 2018 [26, 27, 28] where it is expected to exchange more than 6000 Peta Bytes per month by smartphones in 2018. Figure 1-5 shows spectrum occupancy measured by Ericsson in different frequency bands, and indicates the unfair distribution of occupancy against different bands, as mobile frequency bands (900, 1800 and 2100 MHz) have high power density due to high spectrum utilization.

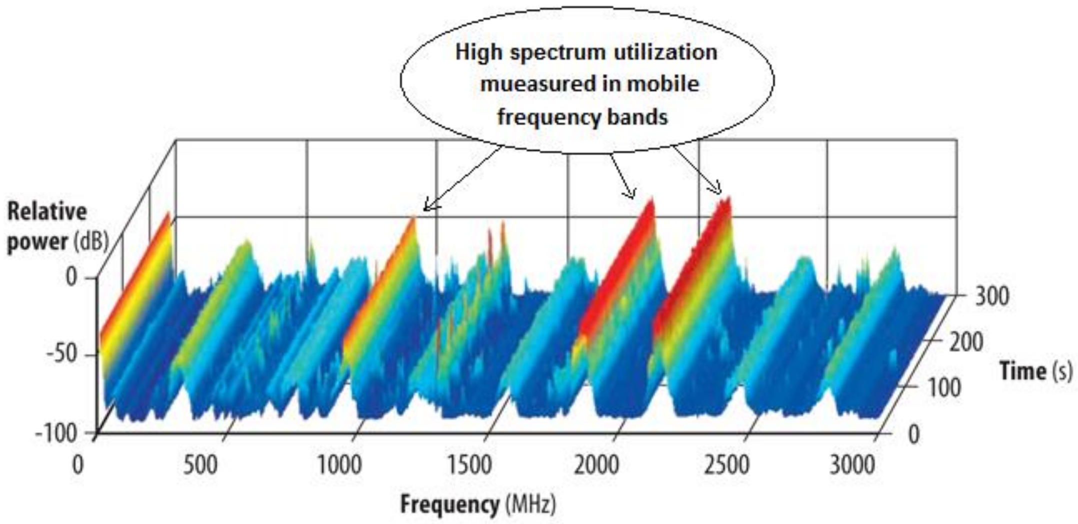


Figure 1-5: Measured spectrum occupancy indicates high utilization in mobile frequency bands – Ericsson publications [26]

With increasing demand for more spectrum capacity the following solutions were discussed: increasing the spectrum bands, improving the related network controllers like wireless network base stations, and improving spectrum utilization. Due to the limitations on the first and second solutions, most researchers are interested in the third solution as there are varieties of applications that can achieve it. One of the most important techniques to improve the spectrum utilization is the opportunistic concept, which is technique of inspecting bands of spectrum to find spectrum holes and then make use of them by secondary users. The opportunistic systems concept started to attract the researchers as it introduces solution for the spectrum utilization problem based on the available resources.

In general the new trends in spectrum sharing techniques can be divided into two categories. The first category is dynamic spectrum sharing where the spectrum holes are assigned with limited rights, and the other category depends on licensed shared access with different priority levels. The two categories vary in the degree and methodology of control, from full rights to opportunistic dynamic shared access [29].

William Lehr highlighted in his MIT paper [30] that the new trend in DSA planning is having different types of licensed users, where each user can access the spectrum based on his own priority to have the access. In such trend multiple network operators may share same spectrum with different user access type like ASA (Authorized Shared Access), LSA (Licensed Shared Access), and PA (Priority Access) in order to provide several access rights and priorities to their users. Usually the network operator applies such procedure to protect the licensee from harmful interference which affects its communication quality, and many operators considers this as solution to compromise between the exclusive licensed spectrum which provides predictable high quality of service along the operation time, and the unlicensed spectrum which is usually used in insensitive application like public WLAN networks.

Joseph Mitola introduced the cognitive radio concept in 1998 in a seminar at the Royal Institute of Technology in Stockholm, as a new untraditional wireless technique that tries to improve the spectrum utilization [31]. The main idea behind this opportunistic technique is finding the unused spectrum channels (spectrum holes) and assign them to low priority users, and the main two challenging tasks in cognitive radio are spectrum sensing (sensing the wireless environment to find spectrum holes), and then making decisions wither to use these holes or not based on the spectrum knowledge.

This concept was proposed by Joseph Mitola in 1998, published in 1999 [32] and it was improved in following years as it became very attractive research area. The CRN system was introduced as an intelligent wireless communication system that is aware of its network status, learn the changes and adapts to incoming RF demand by sensing the spectrum holes and use it to establish new connections in real time. In typical CRN process the key issues are awareness of network status, ability of learning new status and adaptivity for change, reliability, and spectrum utilization efficiency [33].

Spectral efficiency is very important factor in future wireless communication systems, as it targets higher utilization with accepted service performance. CRN systems are mainly used with lower priority secondary systems that search the spectrum to find holes and then filling the discovered spectrum holes of unused licensed spectrum with their own transmission. Frequency bands which are not fully utilized by the primary users introduce opportunity for the secondary users who wish to send over the licensed channel.

The most useful feature in CRN systems is the capability to sense the operating environment, identify changes and adapt accordingly in real time [34, 35]. This interactive approach allows the CRN to permit/deny the spectrum access to certain users in certain time slots. CRN systems are very familiar in military applications [36] where the primary users use the spectrum on daily basis for limited time, hence the CRN allows reusing the spectrum in the free hours by secondary users which increases the utilization and allows secondary users to find communication channels, however the primary users keep the higher priority in case they need the channel [37].

Some Network operators in differet countries have already deployed Cognitive Radio Networks to enhance the network utilization, for example Ministry of Transport and Communications of Finland, and Ministry of industry and some Wireless providers in Canada [38]. An interesting tool was developed by M.Vieira and L.Mello which uses the spectrum occupancy by primary users in Brazil as input data to simulate the performance of the cognitive radio system based on real time spectrum status [39].

Several research focused on the coexistence environment between cognitive and primary radio systems [40] where this environment can avoid interference. One of the most important transmission techniques in CRN systems is spectrum underlay technique, which allows the coexistance of both primary and secondary users in same time, where primary users are protected from the interferance by applying spectral masks on secondary user transmission, by applying set of rules and frequency band pass filters that limit adjacent channel interference, and ensure that the generated interference is below the noise level allowed by the primary user, however the drawback of the underlay technique is its restriction on power level and communication range.

Another CR approach is the overlay approach where the system allows secondary users to coexist in same spectrum with primary users and share with them their messages in order to control and reduce signal interferance. A trade-off is considered between the interferance between primary and secondary signals, and the accepted level of transmitted power for the secondary user. While the third CR approach is called the Interweave mode and it allows the secondary user operation only when the primary user is absent.

Hybrid schemes using combination of the mentioned approaches may present good and efficient techniques to improve the secondary users transmission rate and throughput. This will be discussed in details in this thesis.

	Underlay	Overlay	Interweave
Simultaneous Transmission	Secondary user can transmit simultaneously with Primary user. A predefined level of interference is considered to control the transmission power level of the secondary user.	Secondary user can transmit simultaneously with Primary user.	Secondary user can transmis when Primary user is absent.
Transmission power	Secondary user can transmit only below the power level which does not cause interference with primary user	Secondary user can send at any power level that does not cause interference, and it can overhear primary user messages in order to adapt its power level.	Secondary user can transmit any power at primary user Inactivity.

Table 1-1: Cognitive radio operation techniques

During cognitive radio operation the system searches the spectrum holes by detecting the unused frequencies and sensing the radio frequency spectrum, and runs a process to identify channel occupancy includes channel state information estimation at the receiver side. After the spectrum sensing process the power level of the transmitter is selected carfully in order to avoid signal interference between primary and secondary users. The spectrum sensing and channel identification processes are done by the receiver while the third task is carried out at the transmitter side, based on the feedback between the receiver and transmitter.

The cognitive radio technique can be extended to cognitive networks, where a system of an intelligent multiuser wireless communication system capabile of analyzing the radio environment and identifing the spectrum holes, adapts to changes in the environment, facilitates communication between different users, and applies resource management techniques [11], where the cognitive radio networks can serve many applications like vehicles mobility, medical applications and commercial internet [41, 42]. The main tasks of the cognitive radio system include: local radio spectrum analysis, spectrum sensing and channel identification, and finally Dynamic spectrum management and controling the transmitted power.

On the other hand Cognitive Radio does not solve the spectrum utilization problem in other applications like mobile data networks where huge number of users need to work simultaneously with no restrictions on operation time so a more efficient technique is required to improve the utilization while allowing a huge number of PUs and SUs to operate simultaneously. Although Cognitive Radio techniques improve the spectrum utilization and helps secondary users to make use of the unutilized spectrum, the conventional CRN which depends on the inactivity period of the primary user does not satisfy the demand of today's cellular networks which need to serve huge number of users simultaneously.

1.2 Current research and objectives of this thesis

Most of research undertaken in the Cognitive Radio area has focused on the spectrum utilization problem, which considered the key issue to be improving the SCP (Successful Communication Probability) by introducing new techniques that employ the CRN concept with some modifications that help CRN achieving higher SCP. One of these techniques was introduced by Andrew Drozd who focused on employing multiple transmission dimensions (i.e. time, frequency, and antenna directivity) to increase the opportunity of the secondary users to have wireless channel access, whether the primary user exists or not. The technique introduced by Androw Drozd is commercially known as Transmission Hyperspace™ (TH) [43], and it was introduced for the first time in 2006 in the 17th International Zurich Symposium on Electromagnetic Compatibility [44]. This technique focused on employing the CRN concept with some modifications that help the cognitive Radio network able to accept higher number of users hence achieving higher SCP. The TH technique employs multiple transmission dimensions and make use of the opportunistic nature of the Cognitive Radio network to find spectrum hole for the secondary users to have wireless channel whether the primary user exists or not.

In the study done by Drozd & Ozdemir [44, 45, 46] they introduced a new spectrum management and utilization paradigm based on mathematical optimization of several spectrum dimensions including frequency, time, antenna directionality, modulation, space, polarization, and other dimensions that allow secondary users to operate in single CR domain with primary users without interference or with minimal interference. They derived the probability of successful communications with different densities of CR primary and secondary users [47], and they showed that using multiple dimensions in cognitive radio networks improves the successful communication probability significantly. This analysis depends completely on monitoring the SCP as a metric to evaluate the usage of different dimensionalities. The analysis done in the above research showed the SCP value in different scenarios based on its dependency on some other factors like distance and bandwidth. The research outcomes are explained in chapter 2 of this thesis.

Multiple dimensions technique in CR networks (MD-CRN) may introduce an efficient solution for the spectrum utilization problem and CRN system limitations. Both PUs and SUs can operate with reasonable performance and few restrictions. This means that the same spectrum resources (like frequency band) can be used to allow a higher number of users, and this introduces good solution for cellular networks operators who need to provide more services without increasing the spectrum resources. Some research was done in CR Multiple Dimensions (MD-CR) areas and proved their efficiency in improving the spectrum utilization. These researches focused on studying and monitoring the SCP (Successful Communication Probability) behaviour among selecting different MD-CR parameters (like Frequency, time, space). However most of this research did not investigate deeply the impact on network performance (in terms of Quality of Service) after deploying the MD-CR technique, and improving the related SCP and spectrum utilization [48, 49, 50].

Along the previous years some research introduced good analysis for evaluating and improving the Successful Communication Probability in each case, but it did not evaluate the related Quality of Service and network performance in the multi dimensions scenario. Electromagnetic interference which may result during the SCP optimization causes losses in quality of service as indicated in [44] which affect the user performance in the RF space, hence this shall be corrected by QoS aware techniques.

This research project tries to maintain balance between improving the SCP and in same time preventing the degradation of network quality of service, it studies the impact of improving the spectrum utilization in the cognitive radio environment, and proposes a technique to balance the need to maximize the SCP and keeping the network performance and QoS within the accepted bounds [51].

1.3 Contribution of the research

The previous work focused on studying the SCP (Successful Communication Probability) and its relation with some important factors like Rx-Tx distance, number of primary users and channel bandwidth, and highlighted the SCP optimization problem and studied how to optimize the SCP value. However, QoS was not considered in this optimization, and it is important to study the impact of optimizing the SCP on the wireless network performance specially in packet networks, and assess the network performance and quality of service in terms of factors such as throughput, packet loss and delay after applying the spectrum utilization techniques.

This research studies the impact of the techniques used to optimize SCP on network Quality of Service. In many cases the SCP has a good value but the service quality is unacceptable. The network QoS is be analysed in the MD-CR case (employing dimensions like frequency, time, and modulation) in order to check the related QoS and hence decide if SCP optimization will be acceptable in this case or not. The investigations will show the impact of using the spectrum utilization techniques on overall network performance and Quality of Service. Mainly network throughput, packet loss and bit error rate will be studied in each MD-CR case as major QoS factors that affect the network operation performance.

The investigations will show the impact of using the spectrum utilization techniques on overall network performance and quality of service). It is expected to find some parameters affected negatively (performance degradation) while other parameters affected positively (performance enhancement). Based on these parameters an algorithm is proposed to keep the channel efficiency (in terms of throughput and QoS) above certain quality margin. This algorithm may be implemented in form of a real-time technique to adapt the network capability of delivering an accepted quality of service.

The contribution of this research is to study and provide a solution for monitoring and controlling the quality of service in the networks that employ spectrum utilization techniques based on concepts developed for Cognitive Radio, where these networks usually focuses on optimizing the successful communication probability regardless the impact on quality of service. The research makes the following contribution to knowledge:

1-Analysing the effect of using the multiple dimensions concept in Cognitive Radio Networks.

2-Proposing a technique that prevents degradation in Cognitive Radio Networks that employ multiple dimensions techniques.

3-Providing reliable simulation environment that can simulate Cognitive Radio Networks with multiple dimension operation.

4-Designing QoS aware CR networks that can compromise between the spectrum utilization trend and quality of service constraints and requirements specially in data networks which allow voice and real time applications.

The proposed technique is discussed in next section, with a flowchart diagram that shows how it prevents QoS degradation.

1.4 Proposed technique / methodology

Cognitive radio ad hoc networks usually aim to increase the wireless network utilization where finding and hunting the spectrum resources is the main target of this technique, as the resources may become available from time to time. The effect of maximizing the SCP we may include undesired phenomenon like increasing the interference level and losing the high throughput provided by conventional CRN. In some experiments it was found that deterioration in communication quality due to the continuous efforts to increase the SCP by the network [52], and it was recommended not to exceed certain level of interference while the network trying to maximize the utilization in order to keep the channel throughput at accepted level. This should be considered when discussing how to increase the CRN SCP.

In a Cognitive Radio environment the TCP performance is affected by many factors such as activities of the Primary Users, sensing signalling overhead and related errors, also the frequent interaction between the transport layer and the lower layers which occur as result of flow of messages between TCP transport layer and lower layers during the switching between different statuses like channel availability and unavailability.

Some research such as the TCP throughput analysis [53, 54] recommended tailoring a special TCP model for the Cognitive Radio networks that is capable to work in the related tough operational conditions. The main issues which need focus in order to improve the TCP performance in CRN are the complex and tough interaction between TCP transport layer and lower layers, also on huge overheads and imperfect spectrum sensing.

In this research, a new technique is proposed to manage the compromise between increasing the SCP and increasing the network throughput and keeping the QoS above the accepted thresholds, thus avoiding network performance deterioration which may result from the high user density in single wireless domain as a result of maximizing the SCP. In this research, a full Cognitive Radio model has been implemented in the OPNET simulator by developing modified nodes with the appropriate coding which include basic functionalities. The purpose of this model is simulating the CR environment in order to study the network performance after applying the multi dimensions technique. The proposed technique observes the channel throughput on TCP level during the operation of the CR multi dimensions technique and warns the network controller when the throughput degrades below certain level. Network controller is defined as central network element that can control and monitor network operation, for example base station in wireless network. Similar concepts were proposed to divide the operator spectrum into time-frequency slots where each secondary user submits bid to network controller then limited users are selected based on evaluating the submitted bids [55 - 59] however these proposals didn't extend to monitor the performance during the simultaneous operation of the primary and secondary users, while other research proposed Markov approach to achieve the required QoS requirements [60]. OPNET was selected as a powerful simulation tool suitable to simulate cognitive radio networks, where it can afford flexible simulation environment which can be adapted by modifying the C++ code of the simulator module. NetSim was used to verify OPNET simulation results as it afford readymade module for cognitive radio networks however OPNET is still more powerful in terms of simulator capabilities and stability.

The proposed technique tries to make balance between optimizing the SCP, increasing the network throughput and keeping the quality of service within the accepted thresholds. It executes the following sequence in order to compromise between maximizing the SCP and keeping the QoS in accepted range:

1. The network controller assigns different priority levels to the users. Primary users always have priority 1, while secondary users are assigned priorities 2 and priority 3.
2. When a user asks for the channel, the network controller checks the user's priority level. Users with priority level 1 are being allowed immediately with no further checks.

3. When priority 2 or 3 user asks for channel, the network controller checks the possibility of applying multiple dimensions CR operation mode, calculates the estimated interference with the existing primary users, and then decides on the validity of the new connection.
4. In the event that the new connection is valid, the network controller estimates the related QoS parameters (throughput, packet loss, delay, ..) and decide on the validity of the connection.
5. If the new connection satisfies the QoS requirements, the network controller allows the related user to establish the new connection.
6. During the network operation, the network controller keeps observing the QoS parameters and checks their values continuously.
7. If the network controller finds QoS degradation in a certain channel, the network controller terminates the involved priority 3 user.
8. After terminating priority 3 users the network controller re-check the QoS parameters based on short observation period.
9. In case the QoS parameters values do not satisfy the QoS pre-defined thresholds, the network controller terminates priority 2 user’s sessions.
10. After proper operation period, where the network controller finds that QoS pre-defined thresholds are satisfied and no interference is observed, it allows activating priority 2 and 3 user’s sessions again and goes through same cycle again starting step 4.

The following flowchart in Figure 1-6 shows the proposed process which employes MD-CR to improve the spectrum utilization, and in same time tries to gurantee certain level of QoS for the MD-CR environment. This technique tends to control the behavior of the multiple dimension cognitive radio technique in order to guarantee a certain level of quality of service. The proposed setup assumes three levels of priorities; accordingly the network controller classifies the users in three groups. C-MD-CR (controlled multiple dimension cognitive radio) is the proposed name for this technique.

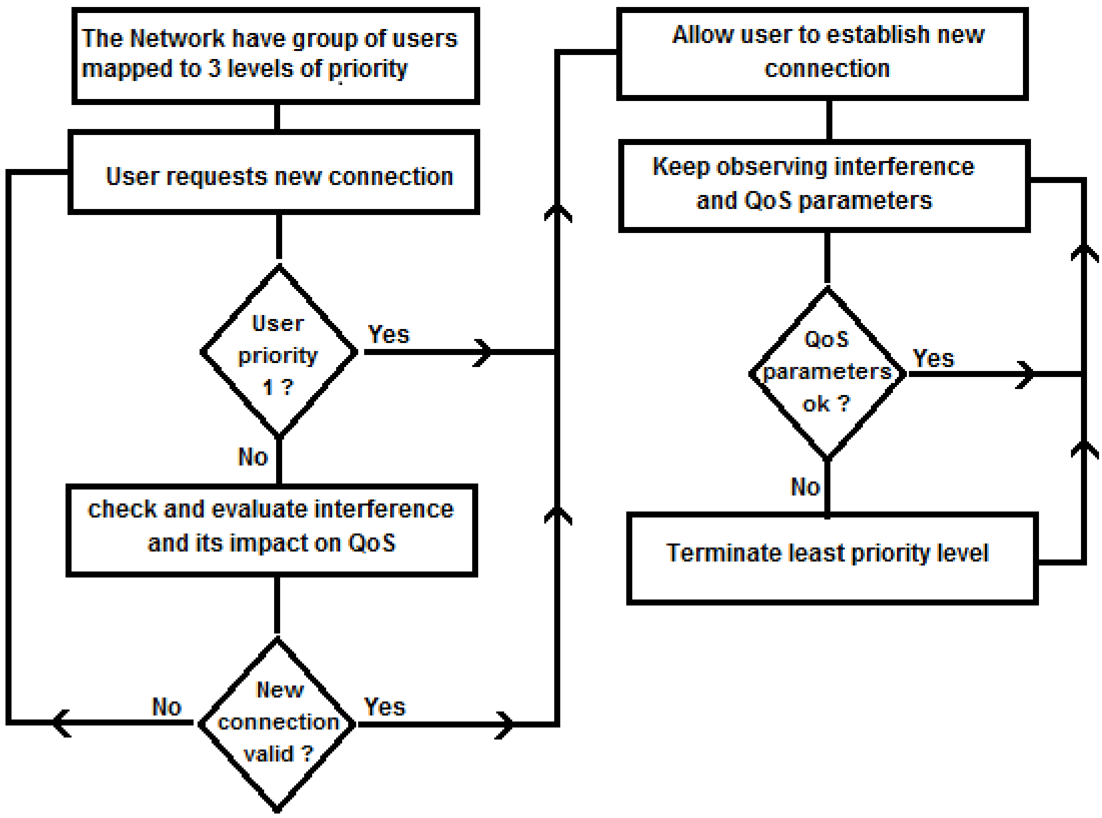


Figure 1-6: Proposed connection setup process to guarantee QoS level

1.5 Thesis Outline

This thesis is organised in seven chapters and is structured as follows:

Chapter 2	Reviews OSI model, TCP standard and Cognitive Radio standard, then discusses different performance issues related to Multiple Dimensions Cognitive Radio Networks.
Chapter 3	Introduces two simulation tools (OPNET and NetSim) that can simulate Cognitive Radio Networks.
Chapter 4	Evaluates the Quality of Service in multiple dimensions CRN and investigates the performance degradation.
Chapter 5	Proposes new technique to enhance the quality of service in multiple dimensions CRN and examines it through performance simulation.
Chapter 6	Concludes the research findings and suggests related future work.

CHAPTER TWO

TRANSMISSION CONTROL PROTOCOL AND COGNITIVE RADIO NETWORKS

This chapter reviews some Internet Protocol standards like the Open System Interconnect (OSI) model and the Transmission Control Protocol (TCP) basics, then discusses the basics of the cognitive radio standard technique also the multiple dimension cognitive radio technique and how it is evaluated in previous work, and then discusses the TCP operation in cognitive radio networks and how its throughput is evaluated, and finally the relation between packet loss and SNR.

2.1 The OSI model and TCP standard

In the following sections the OSI model and TCP protocol are reviewed in order to support the analysis presented in the following chapters.

2.1.1 The OSI model

The Open Systems Interconnection model (OSI) was developed by the International Organization of Standardization (ISO) to standardize the internal structure of the communication systems that operates in digital data networks. The model was built in 7 layers to simulate the data flow starting the user application to the physical interaction between the network components [61]. The seven layers model shown in figure 2-1 assumes that this queue presents at both sender and receiver sides to organize the data flow at each side, while connected through physical media that can be LAN network exchanges data in form of electrical signals [62].

The data flow at the sender node starts at top layer (application layer) toward the bottom layer (physical layer) and continues through the physical medium between the nodes till reaching the receiver node then flows from bottom to top layer. The OSI model considers encapsulation concept where the user data is generated at the top layer by the user application then goes through lower layers where it is encapsulated in larger Protocol Data Units (PDU), for example the PDU of the Transport layer is called segment in case the transport layer protocol is TCP and called Datagram in case UDP protocol, while the PDU of the Network layer is called Packet and the PDU of the Data link layer is called Frame. The PDU at the physical layer is typically the smallest data unit which is the bit. For each level the two entities at both sender and receiver belong to same level exchange PDUs to exchange level related data in form of header and payload. Every time layer N at the sender produces PDU it is forwarded to layer N-1 where a new header is added to this PDU to form new PDU related to layer N-1 and so on. This process is called encapsulation where each layer generates PDU and forwards it to next layer to encapsulate it in new PDU, while the process is reversed at the receiver side.

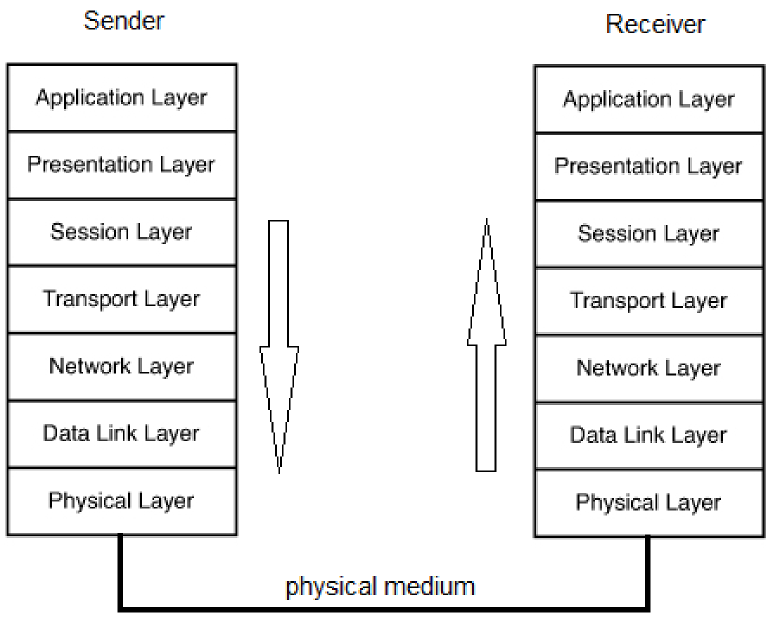


Figure 2-1: ISO OSI model

Physical, DLL and Network layers depend on node hardware implementation while remaining layers depend on firmware, software and user applications. Here are the functionalities of the different layers:

1. Physical layer: interface to physical medium, deals with raw bits, electrical and mechanical transmission system, and frequencies in wireless networks. It can

understand up to binary digits (ones and zeros) while can't understand structured data units like packets and frames.

2. Data link layer: responsible of encapsulating the data in frames, and ensure reliability of frames delivery between two end nodes, this requires applying techniques for error detection and correction, flow control and synchronization. The Data link layer contains two sub-layers: the Media Access Control (MAC) layer which is responsible of controlling how nodes access the network based on physical addressing, and the Logical Link Control (LLC) layer which is responsible of data encapsulation, error control and synchronization.
3. Network layer: manages multi node network and deals with related issues like logical addressing, routing and packets delivery. This layer isolates the upper layers in the OSI model from dealing with physical and transmission aspects by providing reliable logical addressing scheme.
4. Transport layer: ensure end to end delivery by employing techniques for error recovery, retransmission and flow control, also it is responsible of data encapsulation in segments.
5. Session layer: define the communication structure between different user applications, also it is responsible of establishing and terminating data sessions between different user applications that interoperate together.
6. Presentation layer: mainly translates the data format from the network service environment to user application environment in order to present the data in proper format understandable by the application, this includes for example data encryption and compression.

7. Application layer: responsible of end user interface and functionalities like web browsing, remote access, resource allocation, and user authentication.

Some services are performed by two or more layers. These services are called Cross-Layer functions, for example Cross MAC and PHY Scheduling which is important service in wireless networks where the packets are transmitted only in certain channel conditions. The MAC layer retrieve the channel state information from the physical layer and schedule packet transmission accordingly, this improves the data throughput [63].

2.1.2 The TCP standard

The transmission control protocol (TCP) is considered one of the most reliable transport protocols, where it can provide reliable error check and control technique, also assured ordered stream of segments, which make it very popular with IP networks [64]. The TCP is always in comparison with the UDP (User Datagram Protocol) which can reduce the datagram delivery time ignoring the link reliability.

Due to its high reliability the TCP in combination of the IP protocol forms conceptual model called the Internet Protocol Suite which presents set of communication protocols used in internet [65]. The internet Protocol Suite describes all aspects related to internet protocol for example how to encapsulate the data and add the header of each layer. TCP uses smart technique to control the stream rate and size which called TCP Sliding Window flow control protocol, where the receiver specifies the window size representing the data size that can be buffered, the sender can send only up to this size of data then wait for acknowledgement [63].

TCP segment header consists of mandatory control data used to deliver the segment to its destination and control the related flow errors. Figure 2-2 illustrates the building blocks of segment header. The source and destination fields represent TCP address which is called TCP port and it is used by source and destination to identify the TCP flow. The sequence and acknowledgement numbers are used to monitor the correct sequence of group of sequential segments and detect any missing segment. The data offset indicates the header size in words where each word is 32 bits block of data, and the flags are 1 bit flags indicate some control attributes for example Congestion window

Reduced (CWR) flag is used by the sender as a response to indicate that a request for congestion control was received and a proper window reduction was done. The checksum field is used in error detection for both header and date bytes, while the urgent pointer is used as an offset to indicate the position of the urgent data inside the data block while the URG flag should be set to indicate that the segment carries urgent data.

TCP uses efficient congestion control mechanism where it controls the rate of the data transmitted by the sender in order to keep the flow at each link with minimal congestion or collapses. The receivers and senders use the acknowledgement to infer the link status and accordingly take the right action, also RTO (retransmission timeout) is calculated carefully to optimize the extra traffic exchanged over the network.

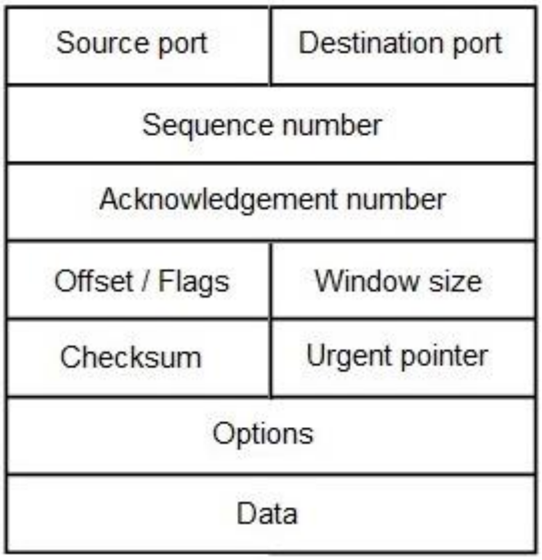


Figure 2-2: TCP/IP segment header

The TCP/IP is the most widely used architecture and it is considered as standard model for internet/intranet communications [66].

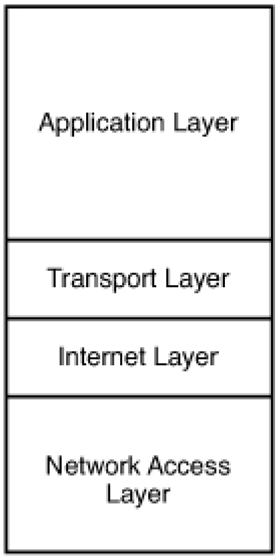


Figure 2-3: TCP/IP model

The TCP/IP protocol stack with its 4 layers is considered special case of the ISO OSI model [67], where some layers are merged to present internet oriented model. It includes 4 layers as shown in figure 2-3, and here are some major differences between the OSI and TCP/IP models:

- OSI was built as protocol independent model to present generic model, while TCP/IP model was design based on specific protocols.
- The network layer in the TCP model supports connectionless services while in OSI model it supports both connectionless and connection oriented.
- Application, Presentation and Session layers in OSI model are merged in single Application layer in TCP/IP model.

- Data Link and Physical layers in OSI model are merged in single layer (Network Access) layer.

The TCP performance is evaluated based on several metrics like throughput, packet loss rate, bit error rate. The packet loss ratio is the ratio between the lost packets to the overall transmitted packets, which usually resides in range less than 10^{-5}. Using the simulated Packet loss ratio we can conclude the Bit Error Rate from the following relation, where PLR is the Packet loss ratio, BER is the Bit Error rate, and PL is packet length in bits:

$$PLR = 1 - (1 - BER)^{PL}$$

In wireless networks the SNR (Signal to Noise Ratio) is considered as indication for the wireless link quality. The relationship between packet loss and SNR can be deduced based on the standard graphs which were created based on the experimental measurements as explained in several references for example the text book “Digital Communications, Fundamentals and Applications” [68]. From the graphs we can deduce the following:

- Most measurements indicates that the packets performance is excellent when the SNR value is above (30 – 25) dB.
- The packets performance is still acceptable (lower quality, but acceptable performance) when the SNR is around 20 dB.
- The packet performance is unaccepted (very low quality) when SNR is much less than 20 dB.

More details and measurements are illustrated in section 2.4.3 and in appendix A.

2.2 Cognitive Radio Standard

Cognitive Radio Networks were designed primarily based on IEEE 802.22 standard [23] which was targeting originally the rural broadband wireless access, with the possibility to give the un-licensed users an opportunity to access the spectrum.

In OSI (Open System Interconnection) model the physical layer (PHY) is responsible of the relationship between the devices and the physical transmission medium where it represents the data in form of electrical signals, while the Media Access Control layer (MAC) is controlling the method and permission that the devices in the network access the medium to transmit its data [61]. The PHY layer in a Cognitive Radio Network is optimized to deal with long channel response times, while the MAC layer provides compensation for long delays. The PHY transport depends on OFDM and OFDMA mechanisms [70], modulation techniques supported are QPSK, 16QAM and 64QAM. The MAC layer in this standard is connection oriented with QoS support, also CR functionalities support is provided by MAC for example dynamic and adaptive scheduling of quiet periods, and control messages between CR user and Base Station (for example: channel change request, incumbent presence) [71, 72, 73].

The following features and capabilities are supported by IEEE 802.22 WRAN standard:

- Spectrum sensing

  Spectrum sensing capability to detect spectrum gaps
- User registration and tracking

  Registration database that shows users connected to each network and their movements from network to other.

- Incumbent Database Service

  Used to give authorization to users who request accessing the spectrum

- Geo-Location

  Feature that calculates user location in support of GPS

- Self co-existance

  User existence in several WRAN cells with several base stations and interference sources.

- Channel management

  Includes several regulations like limitations on transmission power, and accepted level of interference.

2.3 Multiple dimensions in Cognitive Radio Networks

The multiple dimension Cognitive Radio concept [44] focuses on employing multiple dimensions as orthogonal transmission dimensions like time, frequency, coding, antenna directivity, to increase the opportunity for secondary users to have wireless channel access and thus increase spectrum efficiency. In conventional CRN only one dimension is used to differentiate between users (for example time). However in multiple dimensions CRN we have the option of employing more dimensions, including antenna directivity and coding, hence increase number of radio users in limited area with higher spectrum utilization efficiency [74].

The analysis done by Andrew Drozd and Onur Ozdemir used Successful Communication probability, SCP, as measure for multiple dimension CRN efficiency, also this was evaluated by Zhao [75] where the SCP was used to find the spectrum holes. The general form for the SCP based on the assumption of using an omnidirectional antenna and where users operate with a single frequency channel, then time and frequency diversity were considered to improve the SCP, and finally antenna directionality was considered as third dimension. The analysis compares the SCP in case only one dimension is used (Time or Frequency Diversity – Equations (1) and (2) and in case two dimensions are used (both Time and Frequency Diversity – Equation number (3)). The following equations were driven based on the path loss model as function in transmitted power, number of consecutive timeslots, number of frequency channels, and maximum number of hops; and they show that using multiple dimension in cognitive radio networks improves the successful communication probability significantly, where the calculated SCP in equation (3) has greater value than the SCP calculated in each case of single dimension as shown in (1) and (2). Number of hops refers to number of intermediate devices the signal pass between source and destination, where it is assumed single hop in this project for simplicity.

$$P_T^h \approx 1 - \left(1 - P^h\right)^T \qquad (1)$$

$$P_F^h \approx 1 - \prod_{f=1}^{F}\left(1 - P_k^f\right) \qquad (2)$$

$$P_{T,F}^h \approx 1 - \left(1 - P_F^h\right)^T \qquad (3)$$

Where:

P_T is the transmitted power

T is number of consecutive time slots

h is maximum number of hops

F is number of available frequency channels

Andrew Drozd and his research team showed that using multiple dimensions of the spectrum resources leads to more spectrum opportunities and utilization, and they named this multiple dimensions technique commercially the “Transmission hyperspace™”. It is clear that such multiple dimensions technique introduces an efficient method to improve the spectrum utilization, where spectrum resources can be used by higher number of users and this is considered good solution for crowded networks (example cellular networks). However, the Quality of Service must be considered when using multiple dimension techniques. In some analysis and simulations it was noticed that electromagnetic interference level may vary during the operation of the multiple dimensions CRN, which affects the quality of service as explained in [44], the electromagnetic interference has direct impact on user quality of service, and existing researches avoided the assignments that are likely to cause unacceptable losses in QoS [76], in addition it would be useful to study the impact on QoS and consider corrective techniques.

Recent techniques were proposed to solve the quality of service problem in the multiple dimension cognitive radio networks in order to balance between the spectrum utilization and quality of service, for example priority queuing [77], and other techniques like the controlled MD-CR which will be discussed later in chapters 3 and 5.

2.4 Evaluating TCP Throughput in Cognitive Radio Networks

The performance of the Transmission Control Protocol TCP plays important role in the overall performance of the Cognitive Radio Networks which employs wireless technique to exchange data. Unfortunately there is a lack of investigations and numerical analysis which evaluate the TCP performance in Cognitive Radio Networks, For example one of the few researches [53] studied the TCP throughput in cognitive radio environment and recommended considering cross-layer optimization problem with respect to lower layers parameters, however it didn't study the impact on TCP throughput due to electromagnetic interference in CRN. It found that the TCP throughput in CRN is affected by several factors for example the activities of the active PUs have direct influence to the SU TCP throughput, also the spectrum sensing errors and frequent interruptions done by PUs to SU transmissions. In addition to the previous factors which are related to CR operation, we may consider another factor related to strong interaction between the TCP transport layer and lower layers as the TCP layer is responsible of assuring the correct data delivery in environment with frequent interruptions, while TCP layer is responsible of end-to-end delivery. All previous factors affect the operation of the transport layer and reduce the TCP ability to assure end-to-end delivery, which leads to frequent retransmissions, hence reduced throughput [78].

Jian Wang [53] analysed the TCP performance in Cognitive Radio Networks and to applied changes to the TCP model in order to accommodate the challenges it faces in the CR environment. Wang found that the PU activities lead to a dynamic channel availability environment which affects normal TCP operation, hence he proposed a cross layer optimization problem in order to maximize the TCP throughput based on lower layer parameters. He found that the most important issues that affect the TCP throughput in CRN are the exhausting interaction between the TCP and lower layers due to the channel unavailability, where the SU is affected by frequent interruptions by PU, and the TCP suffers from session time-out hence re-transmits the data several times, which leads to less throughput [79]. The second issue is the effect of spectrum sensing as it wastes time in the sensing process also due to the signal collision between PUs and SUs in case of imperfect sensing [80, 81].

2.4.1 TCP throughput evaluation model

In conventional Cognitive Radio and Multiple Dimensions Cognitive Radio Networks the SNR and TCP throughput can be deduced based on the following considerations:

- SNR is obtained based on the received signal from both SU and PU.
- SNR can be used to obtain the related Packet Loss Rate from the standard relation graphs and tables as explained in Appendix A.
- TCP throughput can be calculated as function of Packet Loss rate.

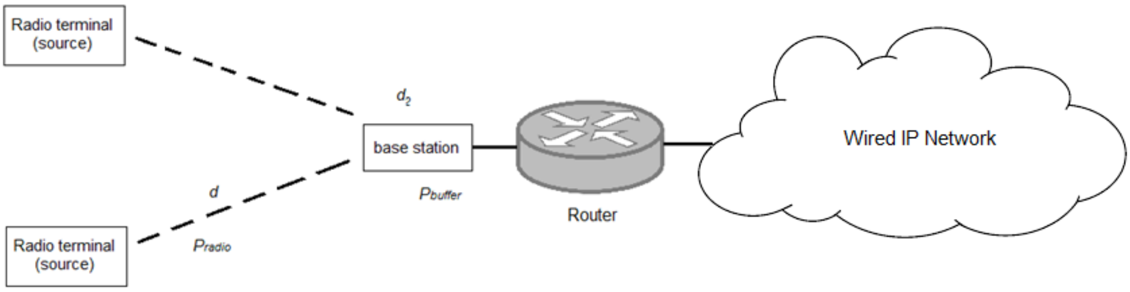


Figure 2-4: Simple model combines both fixed and radio networks

A simple network model illustrated in figure 2-4 is used in evaluating TCP throughput in cognitive radio networks where the wireless network is integrated with wired network, and TCP throughput is measured at the integration point which is wireless network controller (base station) or router. The distance between the user and the base station plays major role in the SNR value also in TCP throughput of the Secondary User. The suitable distance resides in range that avoids the interference with the Primary User and avoids the bad SNR as indicated in equation (4). The effect of shadowing and multipath fading are neglected.

$$d_s \geq d_{SU} \geq d_I \qquad (4)$$

Where:

d_s	is distance which represents bad SNR
d_{SU}	is suitable distance between SU Tx and Rx
d_I	is distance which represents interference to the PU

Usually, Cognitive Radio performance evaluation introduces the Successful communication probability SCP in terms of the received power when several parameters such as distance vary with the user mobility also according to the network dynamic conditions [82].

2.4.2 SNR analysis in Cognitive Radio Networks

It is important to study the relation between signal to noise ratio (SNR) and the quality of service metrics in wireless and cognitive radio network environments [83, 84]. The following relation explains how the Signal to Noise and Interference ratio (SNIR) is calculated in the conventional Cognitive Radio mode as discussed in [48]. The SNIR of user *n* on carrier (frequency channel) *k* is given by:

$$SNIR_n(k) \approx \frac{p_n(k)L_n(k)}{N_0 + \sum_{l \in N \cup M, l \neq n} p_l(k)L_l(k)}$$

Where:

n is identification number of transmitter-receiver pair

P is the power of user n (transmitter-receiver pair) n

L is the path loss between transmitter-receiver pair n is given as $L_n = \frac{C}{f^2 d_i^\alpha}$

f is the sub-band carrier / channel frequency

α is the attenuation constant

d is the distance between transmitter n and receiver n

C is constant

Primary transmitter-receiver pairs are numbered as $m \in (1,...,M)$

Secondary transmitter-receiver pairs are numbered as $n \in (1,...,N)$

Gausian channel with zero mean and variance N_0 is assumed

It is assumed that the path loss in the received power is the dominant loss factor, hence the effects of shadowing and multipath fading are neglected [85].

2.4.3 The Relation between Packet Loss & SNR

The relation between Packet Loss rate and SNR can be deduced from the standard graphs which were deduced from experimental measurements as explained in several references for example the text book “Digital Communications, Fundamentals and Applications” [68], and as discussed in Appendix A.

From the graphs we find the following observations:

- Most measurements indicates that the packets performance is excellent when the SNR value is above (30 – 25) dB.
- The packets performance is still acceptable (lower quality, but acceptable performance) when the SNR is around 20 dB.
- The packet performance is unaccepted (very low quality) when SNR is much less than 20 dB.

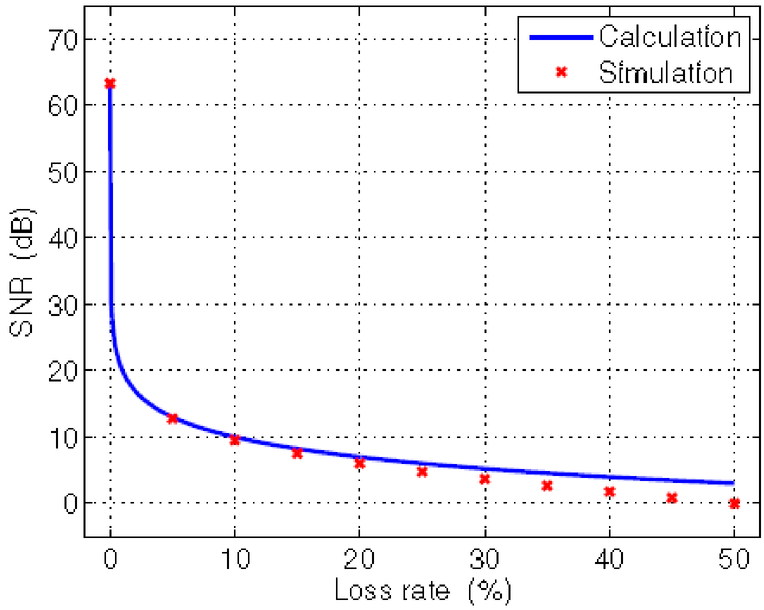


Figure 2-5: example of the relation Packet loss rate vs SNR, Department of Electrical & Computer Engineering, Southern Illinois University [86]

Figure 2-5 shows relation between the SNR and loss rate in collaborated circuit and Network operations. This analysis was done in Department of Electrical & Computer Engineering, Southern Illinois University, USA [86]. More measurements are illustrated in Appendix A.

2.4.4 TCP throughput

The following analytical model evaluates the TCP performance in wireless networks. The main characteristic used in the evaluation is the TCP traffic throughput, which is driven below based on Masahiro's analysis explained in [87], where an analytical model for TCP throughput in wireless networks is built based on modelling the communication errors related to data link layer.

p is the Packet loss rate / probability

b is number of packets per ACK (usually 1 or 2)

T_o is Time out (re-transmission time out)

L is Packet length

N is Packet transmission interval

G is offer load

W is TCP average window size

n is number of terminals

RTT is Round Trip time (radio delay + buffering delay)

Usually in the wireless networks environment, values of round trip time RTT and re-transmission time-out T_o is influenced by the packet loss characteristics due to the frame re-transmission at the Data Link Layer [88]. The Data Link Layer throughput can be expressed as:

$$S_{DLL} = \frac{nW}{NL} \exp\left(\frac{-nW}{N}\right)$$

$$E[d] = \sum_{i=0}^{\infty} (i+1)NL(1-S_{DLL})^i S_{DLL}$$

The RTT is combination of the radio delay + the buffering delay, hence:

$$E[RTT] = \sum_{i=0}^{\infty} (i+1)NL(1-S_{DLL})^i S_{DLL} + d_2$$

$$E[T_o] = E[RTT] + 4\sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \left| E[d]_i - E[d]_j \right|$$

Hence TCP throughput can be expressed as:

$$S_{TCP} = \frac{1}{RTT\sqrt{\frac{2bp}{3} + T_o \min(1,3\sqrt{\frac{3bp}{8}})\, p(1+32p^2)}} \qquad (5)$$

where $p = p_{radio} + p_{buffer}$

Equation (5) introduces TCP throughput characterized by RTT, To, and *p* which mainly depends on packet loss and its related errors, for example bit error rate.

2.5 Conclusion

This chapter discusses the cognitive radio networks standards, also the basics of the multiple dimension cognitive radio and the analysis of the successful communication probability. In section 2.3 it discusses how to evaluate the TCP throughput and SNR mathematically, and how to evaluate the SNR accordingly.

CHAPTER THREE

NETWORK SIMULATORS

This chapter introduces two network simulators which were used in evaluating the multi dimension cognitive radio technique. The chapter explores the simulator structure and how it was modified to simulate multi dimension cognitive radio networks, it also verifies this modification by running a simple simulation scenario.

3.1 Simulating Cognitive Radio environment using the OPNET modeler®

OPNET is a software simulation application produced by Riverbed, and it can simulate networks behaviour, operation in real time, also monitors the network performance efficiently [89]. OPNET provides flexible environment that can be adapted to simulate different techniques and scenarios. It was used in this research project to simulate multi-dimensional cognitive radio networks after code modification. OPNET can work with OSI model, from layer 7 to the modification of the most essential physical parameters. The main component of the OPNET simulation application is OPNET Modeler®, which is software component based on the VC++ programming language and it contains a suite of several protocols and different communication technologies.

OPNET licenses used in this research project was granted by Riverbed to DeMontfort University within OPNET University Program, which was developed by Riverbed to improve the academic research, also to encourage R&D activities in Electrical Engineering, Communication Networks, Computer Science, IT systems and related disciplines. OPNET modeler V17 is used in this analysis, however OPNET modeller in general does not provide ready-made modules for Cognitive Radio or multi dimension CRN simulation, where special code adaptation and modification shall be done in OPNET to allow CR simulation.

3.1.1 OPNET modeler structure

The OPNET modeler is using clear node structure consists of several layers of simulation utilities to run any simulation. The first layer in this structure is the object layer which consists of the network different objects (like routers, servers, user devices) that connects together to form graphical topology. Figure 3-1 illustrates the object level configuration, where each object consists of sub-layer called “Node Model” which presents several blocks of protocols and functions connected together to give the object functionality.

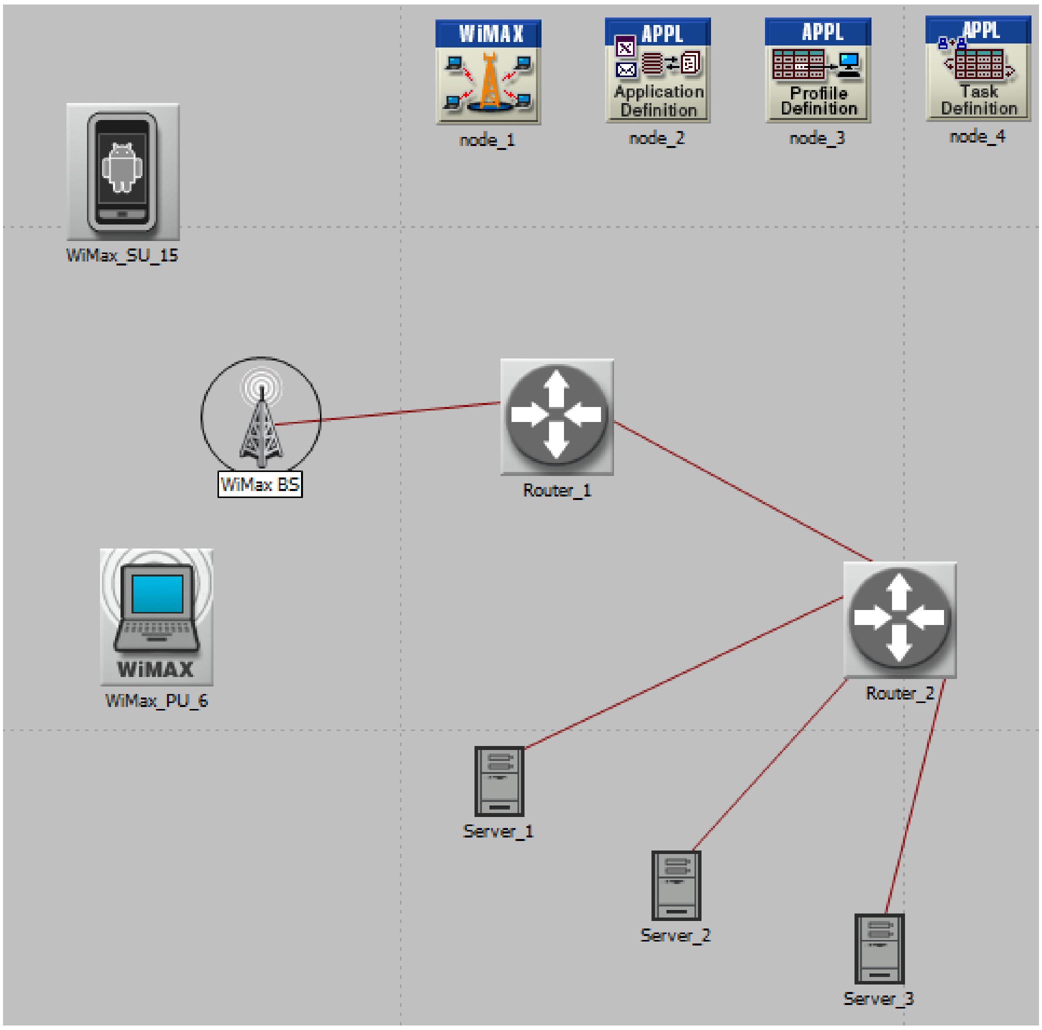


Figure 3-1: OPNET object level [89]

Figure 3-2 and figure 3-3 show the Node Model of WiMax user and Base Station. the node model consists of transmitter and receiver modules connected to the antenna module, also WiMax MAC layer which is responsible of providing addressing also channel access control technique that enables several network nodes to communicate within a multi node network for example Local Area Network LAN. In the user model more nodes related to higher layers are included for example IP, TCP and application layers, also CPU node is included to process the data flow in this model.

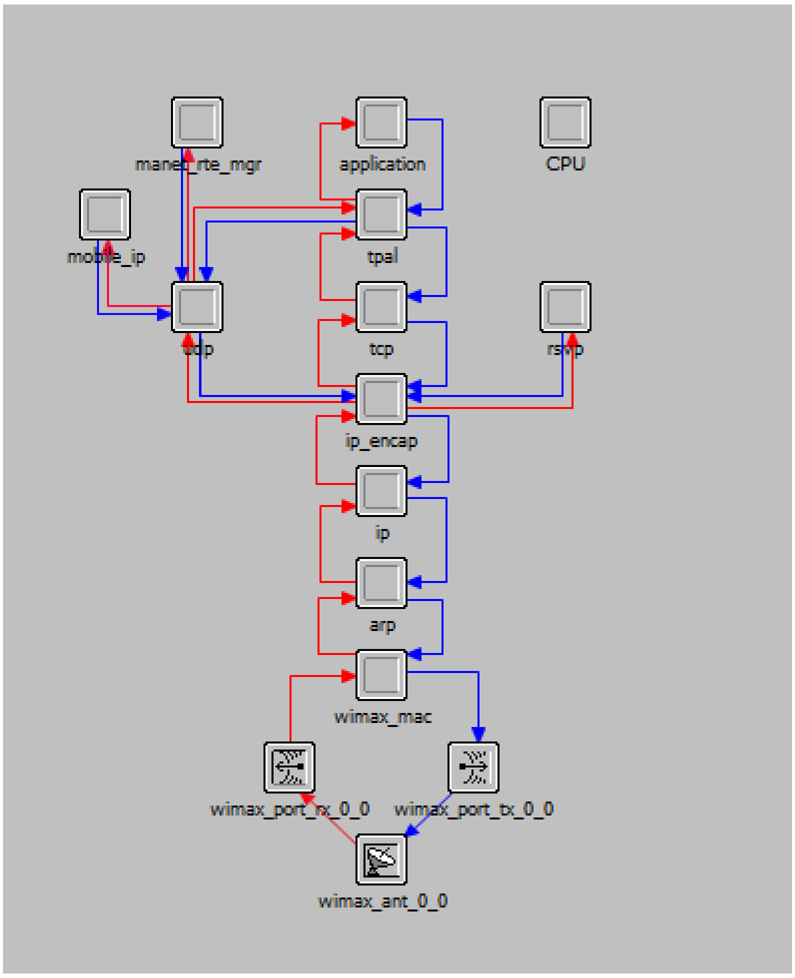


Figure 3-2: OPNET Node level of a WiMax user [89]

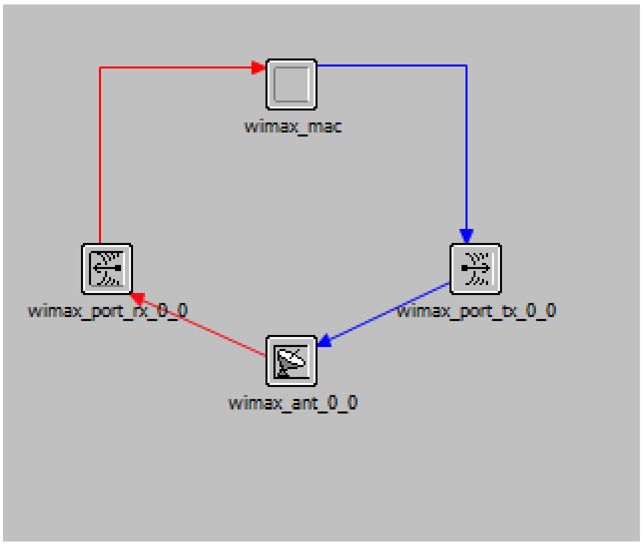


Figure 3-3: OPNET Node level of WiMax Base Station [89]

Figure 3-4 shows typical node model level of layer 3 router where it consists of nodes for different protocol stack for example OSPF, RSVP, IGRP and RIP protocols, also it consist of special interface for ATM interconnectivity. Table 3-1 shows samples of nodes used to build the node model and related description.

In the node model each node consists of a state model similar to a network flowchart, which describes the network packet flow and the related process and flow conditions. Figure 3-5 shows the state model level in OPNET modeler where several process are integrated to guarantee successful packet flow. The state model includes two object types, the state and the transition. The state represents process that executed by this state, while the transition is moving from one state to another one. The state is representing the lowest level of programing in OPNET where it consists of C++ code that executes the related process.

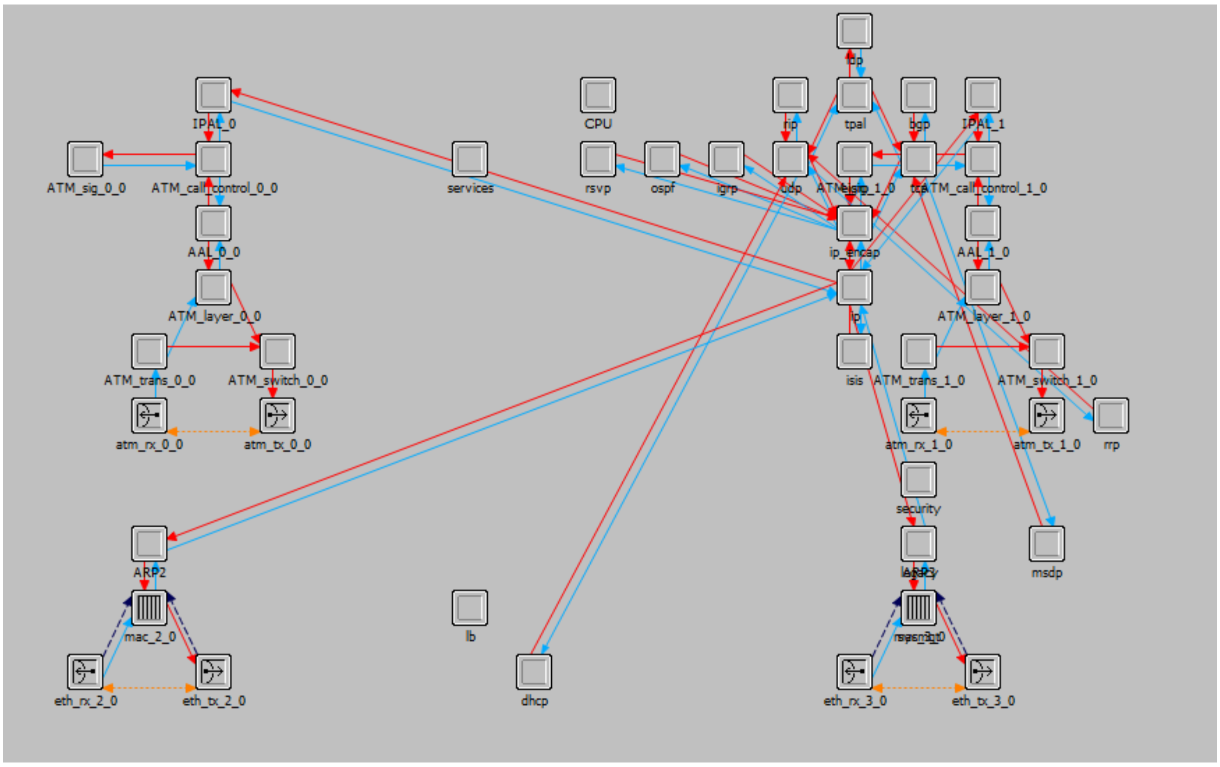


Figure 3-4: OPNET Node level of Layer 3 Router consists of nodes for different protocol stack [89]

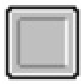
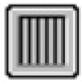
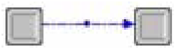
Object type	**Description**	**Symbol**
Processor	An object which can be programmed to execute certain system behaviour.	
Queue	An object that can provide internal packet queuing facility.	
Transmitter	Sending point which send packets to other modules.	
Receiver	Receiving point which receives the packets sent by other modules.	
Packet Stream	Communicates/buffers stream of packets from the source module to the destination module.	
Statistic Wire	Connects source and destination points to transfer data streams.	

Table 3-1: OPNET different node types [89]

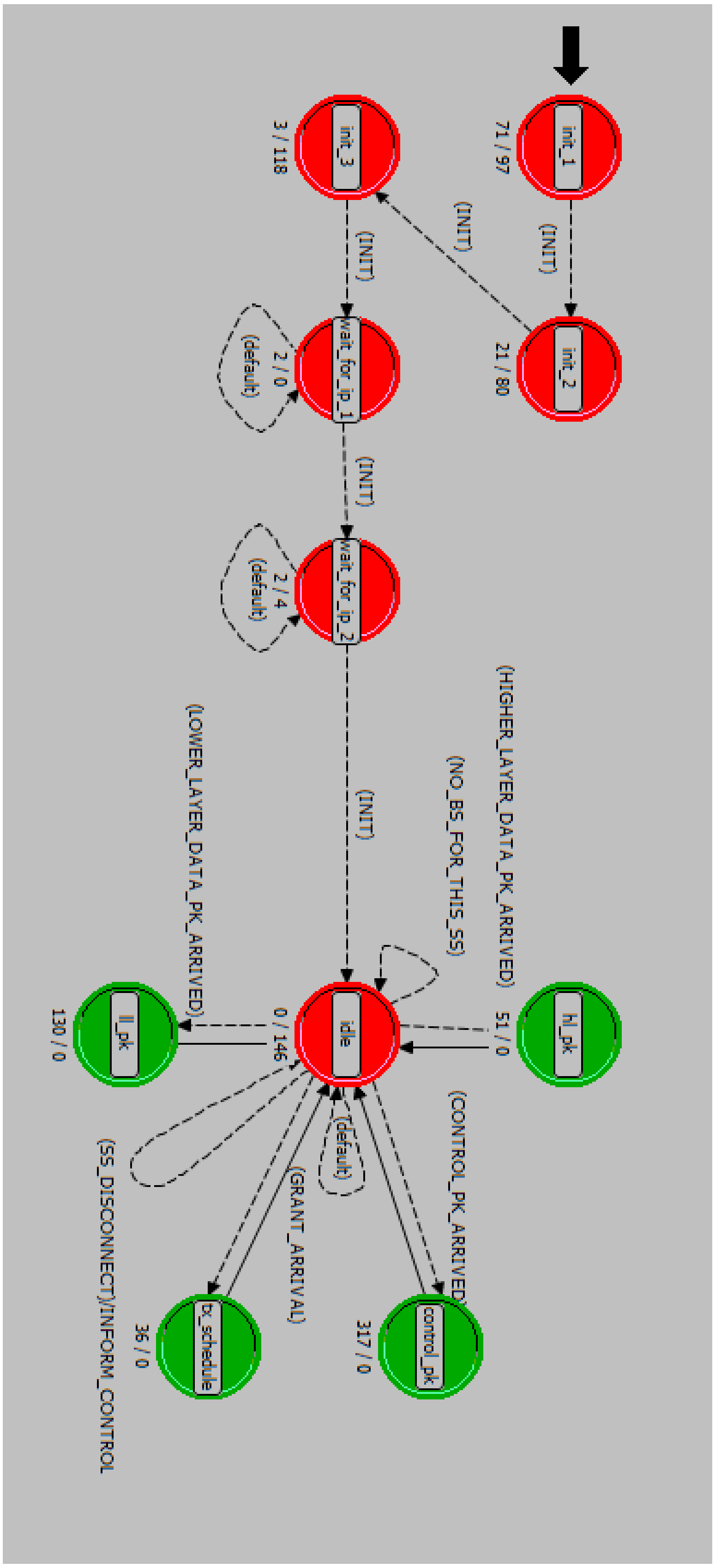


Figure 3-5: OPNET State Model of WiMax Mac Node [89]

The process can only have one state, and it moves from state to another one based on the events that occur. When the process enters in a forced state (the green state) it executes the process related actions and then exits upon finishing all actions. While in the unforced state (the red state) the process waits until a new invocation occurs. When the process moves from state to another one, this is called state transition, and this transition takes the process to a new state with new conditions or, alternatively, it may return it back to its old state, taking in consideration that all these transitions introduce no time delay.

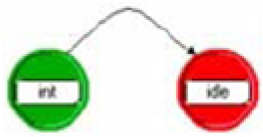
Object type	Definition	Graphical representation
State	The state contains the C++ code which expresses the process done by this state. This processing may occur while entering or leaving the state. There are two state types: forced (green) and unforced (Red).	
Transition	State transition is moving from state to another state. The state may be a source or destination point.	

Table 3-2: OPNET different state types [89]

Each point in the State Model is responsible of specific state and conditions and linked to one or more States to provide integrated function of the related Node. The State is the simplest level of graphical presentation in the OPNET modeler simulator, and it is represented by C++ code as shown in figure 3-6 as sample code.

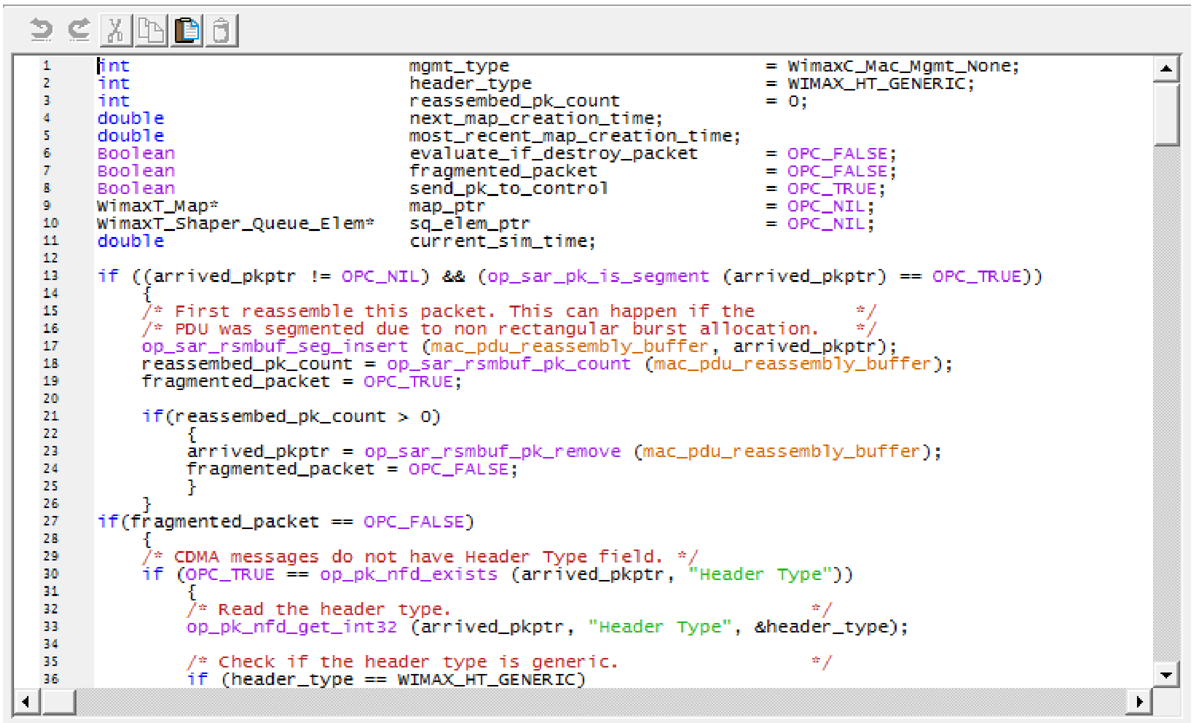


Figure 3-6: OPNET Sample C++ code of state model [89]

3.1.2 Cognitive Radio environment implementation

As highlighted earlier the OPNET default features do not include simulating cognitive radio or multiple dimension cognitive radio, so in order to implement the Cognitive radio environment in OPNET the following functionalities are implemented:

i-Spectrum sensing

ii-Implementation of the OC-MAC Protocol on MAC Layer

iii-Performance monitoring during SU activity and QoS assessment.

The above functionalities were implemented in OPNET by writing the related C++ code and verifying it as explained later in this chapter.

i- Spectrum Sensing

In Spectrum sensing, the Tx node updates the channel state table which contains the status of each channel. The Tx node keeps sending control packet regularly to check the PU activity/inactivity and this usually done by using configurable timers. During this process the channel selection may be occur through coordination between Tx and Rx nodes when they mutually agree about the available channel. After these arrangements the data is sent between Tx and Rx. Also there is a collision avoidance technique implemented in the MAC protocol used to avoid collision. The following two diagrams illustrate the control packets flow of the Tx and Rx nodes during the spectrum sensing phase [90, 91].

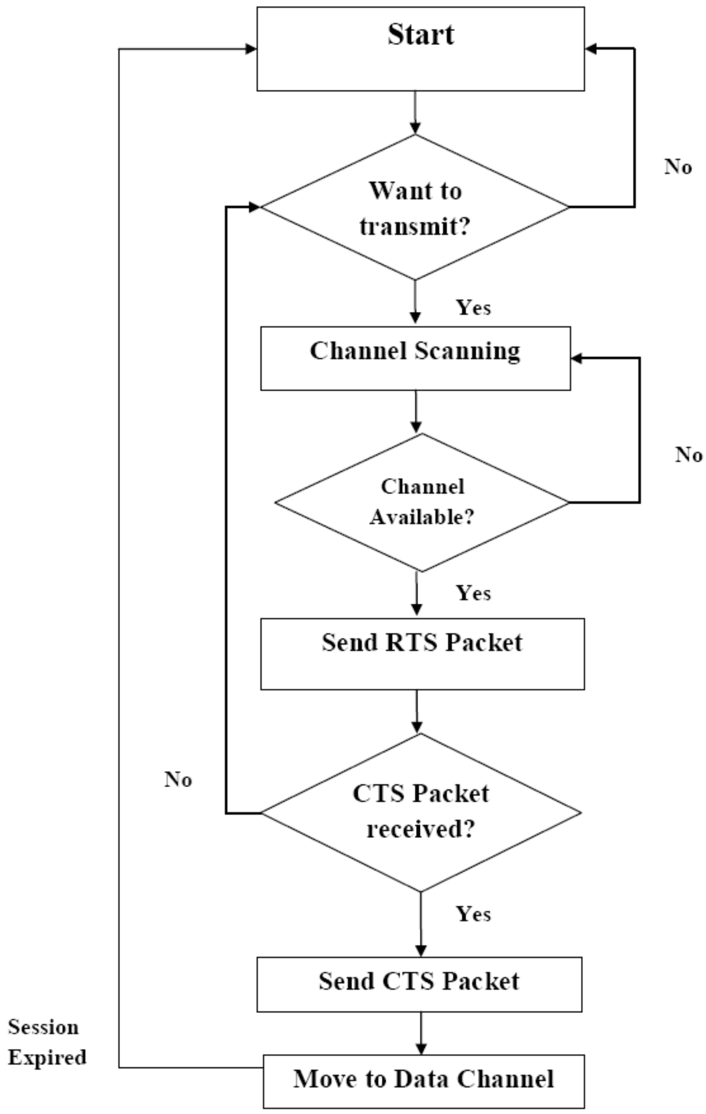


Figure 3-7: Control packets flow in Spectrum sensing at Tx side [90]

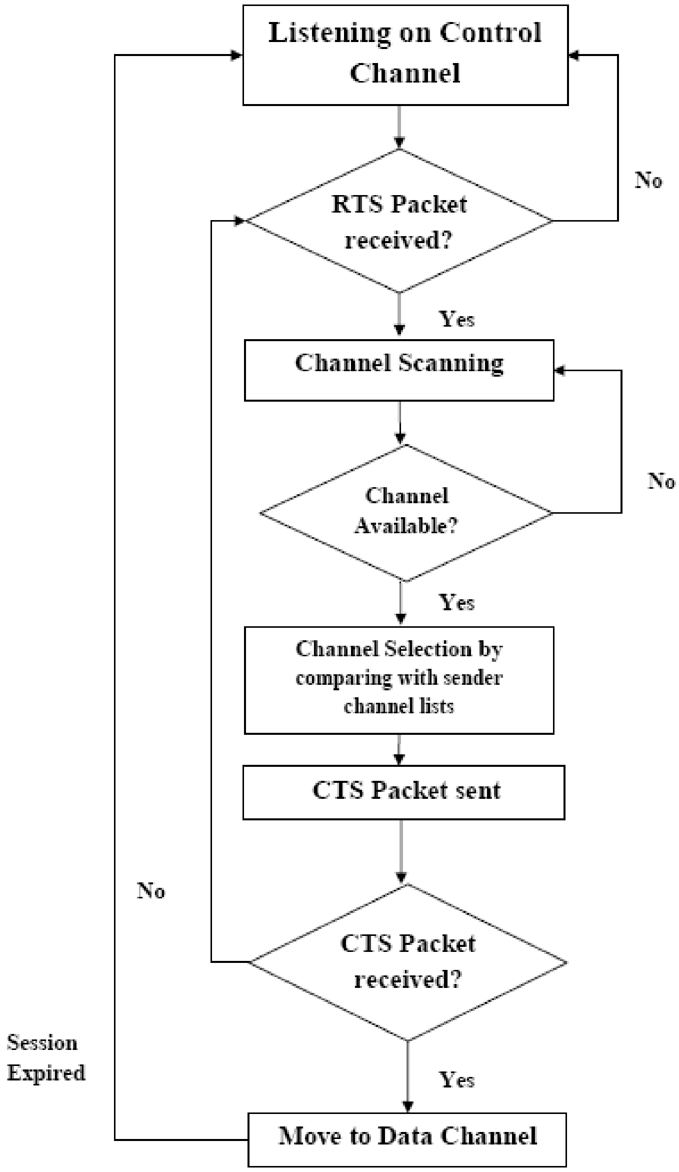


Figure 3-8: Control packets flow in Spectrum sensing at Rx side [90]

ii- Implementation of the OC-MAC Protocol on MAC Layer

The OC-MAC (Opportunistic Cognitive MAC) is special protocol that works at the MAC layer to support the opportunistic behaviour of the Cognitive Radio technique. The structure of the MAC protocol consists of five data channels and one control channel where this control channel is used in the competition between the CR nodes to reserve data channels. In this protocol the transmitter sends an RTS packet which contains the details of the available channels and may contain also the time frame allowed for this transmission (in case the CR network supports such information). Then the receiver uses this information to check the possibility of establishing a connection with the transmitter and, hence, sends a CTS packet back to transmitter which confirms/rejects the channel selection, also this CTS packet is transmitted to all neighbours to update the channel allocation table [92].

During the multiple dimension CR operation the simulator keeps observing and monitoring the network performance and assesses the Quality of Service. TCP throughput is selected as performance metric to measure the spectrum efficiency as it indicates whether the used technique facilitates reliable channel for the user or not, also date loss metrics are used to measure channel reliability [93, 94, 95]. Here are the metrics set used to evaluate the cognitive radio network performance:

- TCP Throughput
- Packet loss ratio
- Bit Error rate

iii- Performance monitoring during SU activity and QoS assessment

The main metrics which are used to judge the Quality of CR network services are the Packet Loss Ratio and the Bit Error Rate, which ae considered an indication for the channel health. Also the TC throughput is used to judge the ability of the channel to deliver the data at an acceptable rate, where adding new SU connections may result in degradation in other channel throughputs due to the Multiple Dimensions environment which allow SU to use different dimensions to get a spectrum hole. Each one of the three metrics mentioned (TCP throughput, Packet Loss and Bit Error Rate) has a pre-defined threshold value in the OPNET code which is used to alert the simulator when the related metric exceeds the pre-defined threshold value [96]. The observation process is running regularly using a timer to set its frequency, also it runs exceptionally in certain events for example when SU requests a channel or when existing user terminates its connection. Both Packet Loss ratio (dropped packets to total sent packets) and Bit Error rate are used as indication for the interference and SNR [97].

3.1.3 implementation of MD-CR and C-MD-CR code in OPNET

The functions explained in previous section were implemented in OPNET as C++ code allowing multiple dimensions (time and frequency) operation to enhance the performance of the conventional CR. The wireless module of OPNET modeler was modified by inserting additional code to perform CR related functions like: calculating the SCP (Successful Communication probability), checking the link quality of service and interference, checking the link throughput, classifying the user based on their priorities/applications, and terminating low priority links. Figures 3-9 and 3-10 illustrate flowchart for the logic used to simulate the MD-CR and C-MD-CR in OPNET software. In the flowchart of the MDCR model when a new request is received the network controller checks the channel state tables which include updated status for channel and Timeslots status to find out if the new connection is possible or not. In case the new connection is possible the controller checks the gap at all available dimensions (frequency and time in this scenario) then decides which dimensions will be used to allow the operation of the new connection. In MDCR the network controller calculates the successful communication probability in order to ensure that the MDCR improved the performance over the conventional CR. The code is shown in appendix B where the OPNET wireless module was amended to perform the following functions:

spct_sns this function simulates the spectrum sensing functionality where it checks the updated state table looking for spectrum gap at any dimension (frequency/time).

clsf_client this function classifies the clients based on the user type (primary / secondary) where primary users are assigned priority 1, and the application type where certain applications are assigned priority 2 and remaining applications are assigned priority 3.

scp_calc this function calculates the successful communication probability as explained in section 2.2

wqos_chk this function checks the QoS status based on predefined threshold values for QoS metrics (Throughput, Packet loss, BER, SNR). In case one of the QoS metrics exceed the predefined threshold the related link is terminated.

trmnt_link this function terminates the links starting with lowest priority clients. (user that violates the predefined QoS threshold values) *chk_actv* this function checks the link activity, and in case not active it returns the channel or timeslot id.

upd_s_table this function updates the state table which consists of information related to channel and timeslot occupation.

chk_freq_div this function recognizes the gap at the frequency dimension.

chk_time_div this function recognizes the gap at the time dimension.

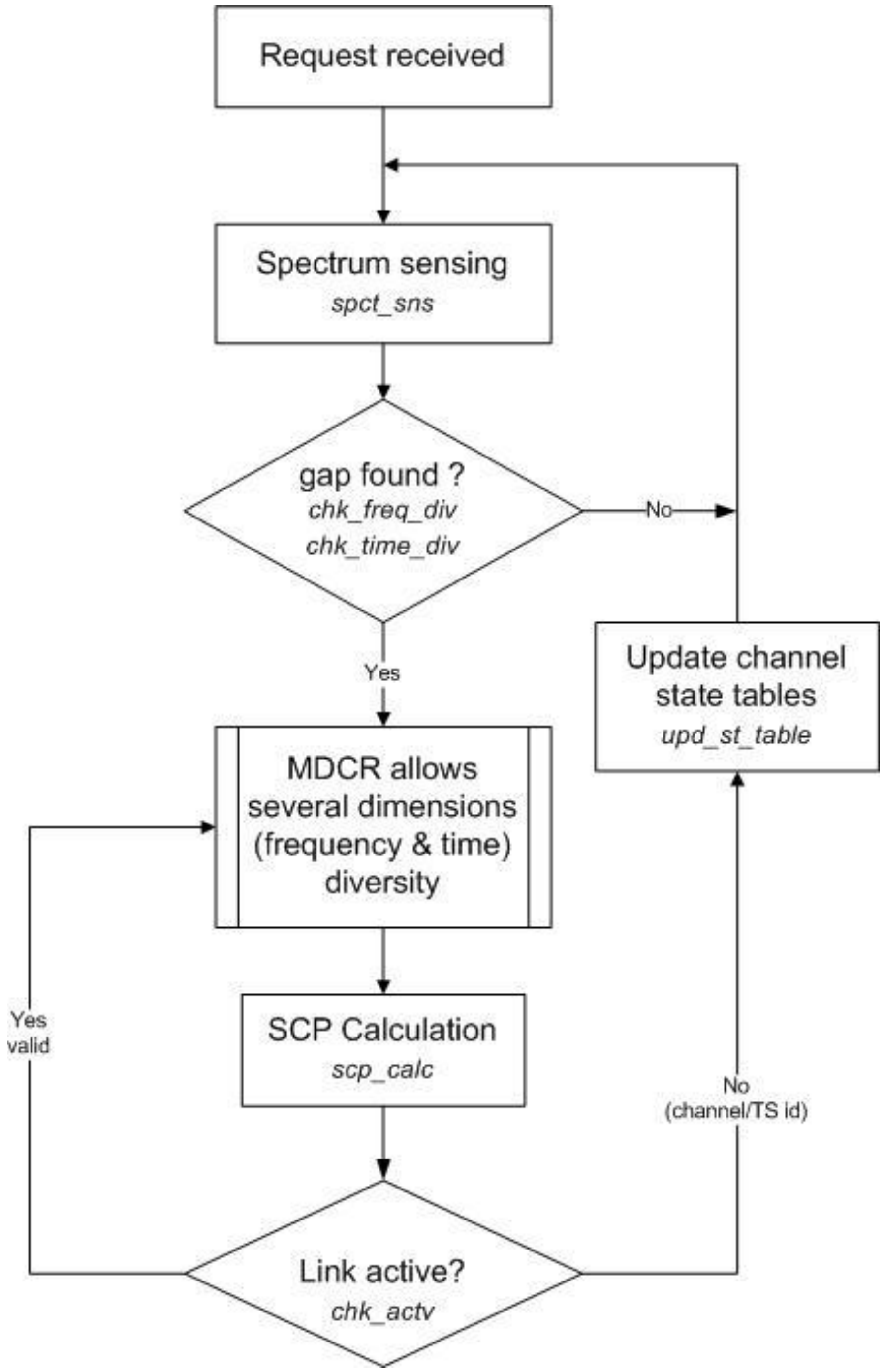


Figure 3-9: MD-CR simulation model

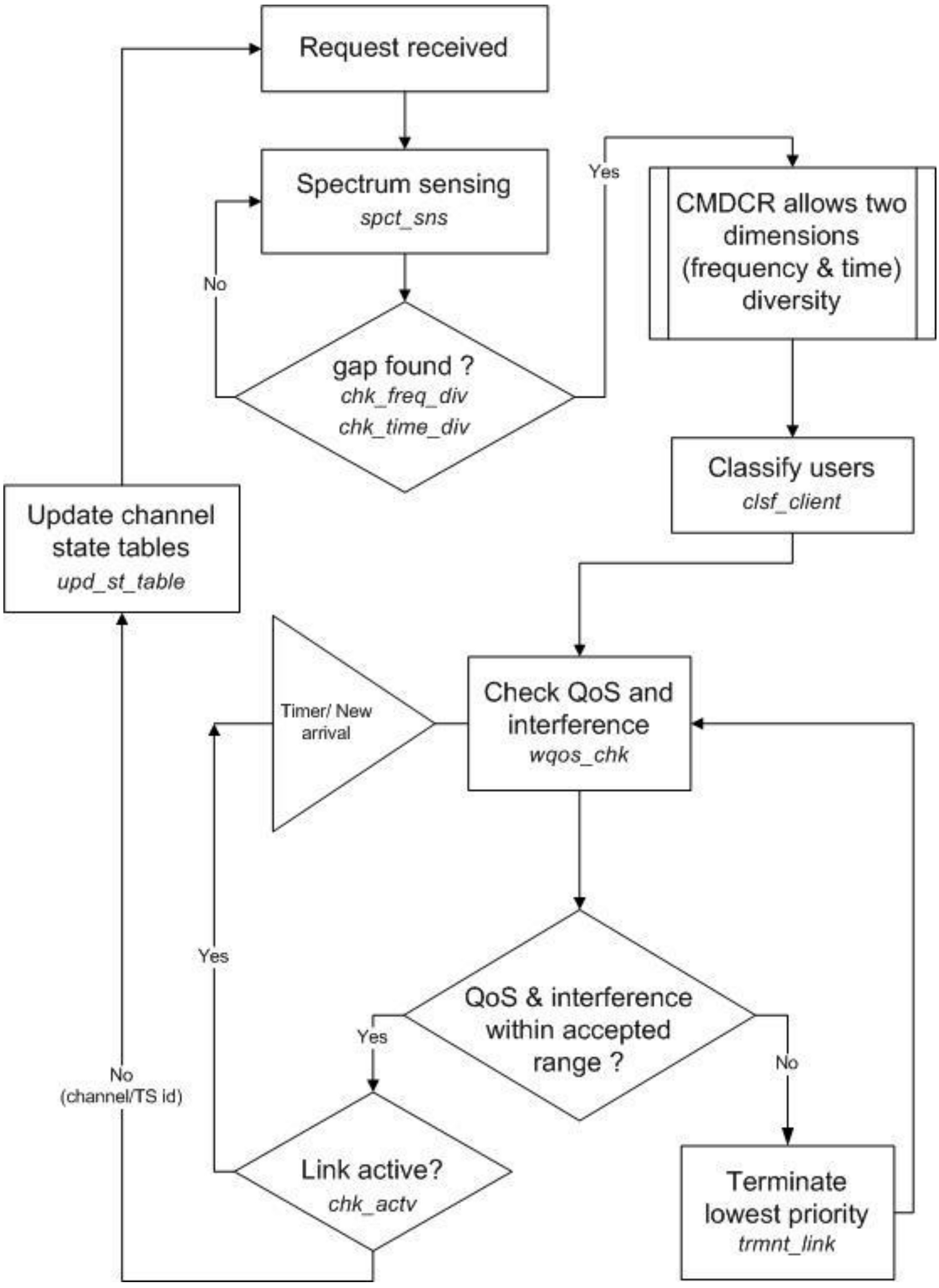


Figure 3-10: C-MD-CR simulation model

3.1.4 Verifying OPNET simulation performance after code modification

After modifying the C++ code of the OPNET simulator to employ the Cognitive Radio features, the following OPNET scenario was executed to verify the OPNET had correct functionality. The scenario introduces a WiMax network that applies CRN operation conditions to allow higher utilization [98, 99], and it consists of a single base station and 16 WiMax mobile users distributed between PU and SU users as shown in figure 3-11.

The scenario assumes a random pattern of operation periods for the PU users, where some PUs operates continuously along the simulation period (30 minutes) while other PUs get active and inactive along this period and give opportunity to some SUs to find free channel to use. This scenario was designed to test the capabilities of OPNET to simulate the conventional CR (during P2) and the multiple dimensions CR (during P3) where each time period applies different status and conditions to allow or prevent the secondary users using the spectrum as explained below. The simulation results are demonstrating three different operation modes of the network:

P1 Only PUs are active while no SUs are allowed to operate, as CR is not employed in this period from time 0 till time 4 minutes.

P2 50% of PUs are inactive and SUs can operate during the inactivity of the PUs. This is the conventional CR concept applied in the period from time 4 till 25 minutes.

P3 More SUs can operate simultaneously with the active PUs using the multiple dimensions CR concept in the period from time 25 till 60 minutes.

The above operation modes were designed in order to test the simulator capability of simulating conventional CR and multiple dimension CR, where P2 allows the SUs to search for spectrum gaps and use them, while P3 allows the SUs to employ multiple dimension CR technique to find channel.

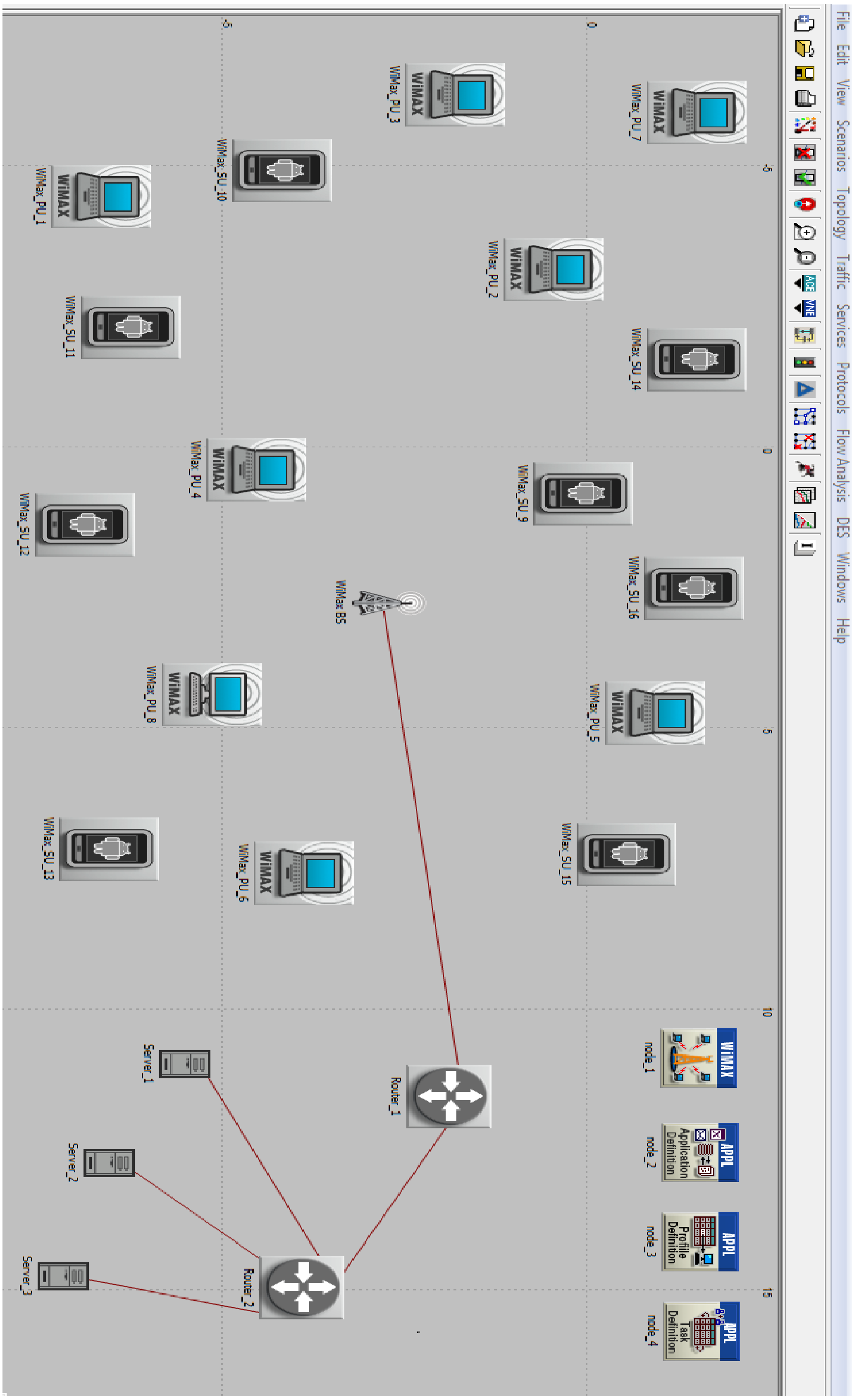


Figure 3-11: WiMax simulation scenario used to verify OPNET cognitive radio functionality

Simulation observation

Figure 3-12 shows the simulation outcome along 60 minutes, where the data throughput of the simulated WiMax network (measured at the WiMax BS) varies according to the time slot. the graph shows that the network throughput was minimal during P1, then increased during P2 (conventional CR mode) and then P3 (multi dimension CR mode) where the network was able to use the CR concept to increase the spectrum utilization. On the other hand the network suffered from high levels of noise in P2 and P3 due to the interference between PU and SU radio signal as shown in figure 3-13.

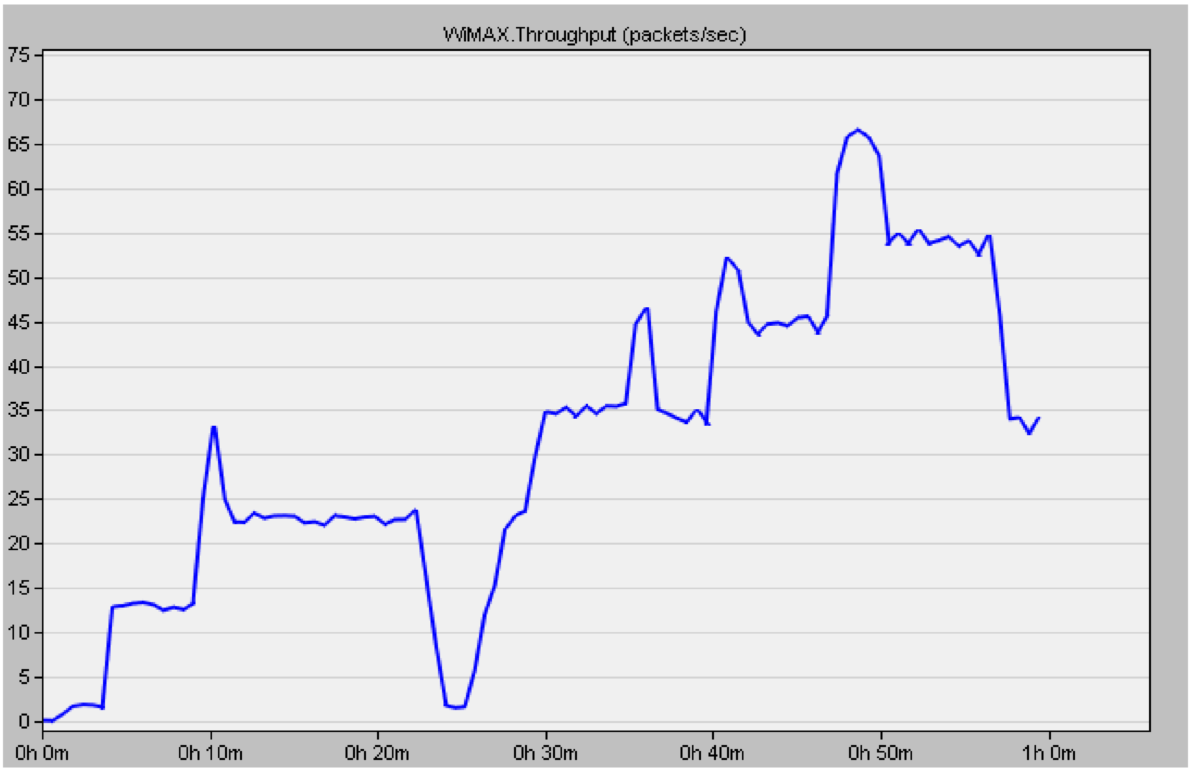


Figure 3-12: Network Throughput graph of OPNET Simulation model: Low throughput from time 0 to 4 minutes, average throughput from time 4 to 25 minutes in CR mode, and increasing throughput after time 25 minutes in multiple dimension CR mode

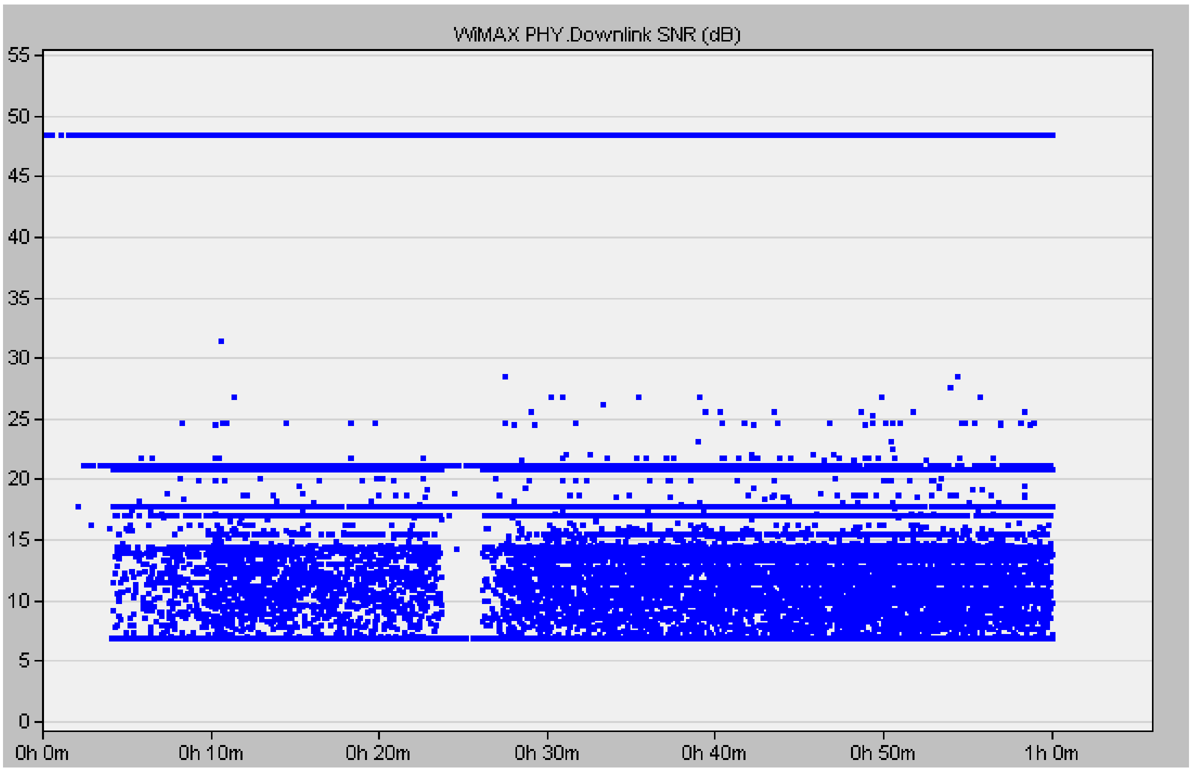


Figure 3-13: SNR graph of OPNET Simulation model

The above SNR results illustrated in figure 3-13 were captured at the WiMax base station BS , where the base station is acting as receiving point for all senders in the network. The result shows that the noise density was increased in the third period (starting time 25 minutes) as the SNR values tend to be less than 15 dB in the time period that multiple dimensions CR mode allows simultaneous operation of the SUs and PUs.

3.2 Simulating Cognitive Radio environment using NetSim Simulator

NetSim™ is network simulation software that used in protocol modelling and simulation. It was developed by TETCOS [100] and used by many universities and research organizations all over the world in network R&D purposes. It is written in C/C++ code and it uses XML configuration files to simulate different scenarios. The key advantage of NetSim is its capability to simulate Cognitive Radio Networks as an embedded feature with no need to write/customize special code, hence it was used in this project to verify and confirm the simulation results obtained from OPNET, as it has different basis of implementation that is completely independent of OPNET. NetSim also can simulate wide range of technologies like legacy networks, Wireless Sensor network, Wireless LAN, BGP and MPLS networks, cellular networks.

3.2.1 Cognitive Radio Simulation by NetSim

NetSim simulates Cognitive Radio Networks based on IEEE Wireless Regional Area Network WRAN standard 802.22 [69]. The PHY and MAC layer models cover Super frame, DS-MAP, US-MAP, BW request, Quiet period, and several operation modes including OFDMA, spectrum sensing function, spectrum management and connection establishment techniques, it also present customizable environment to simulate different technologies with possibility to modify and enhance its setup [101, 102]. The features of the NetSim Academic version cover performance reporting, packet tracking, packet animation and metrics monitoring beside simulating conventional technology standards and customizing simulation code to simulate special network scenarios.

Figure 3.14 shows the simulation environment of NetSim where Primary users are represented as Incumbent nodes, Secondary users as CR CPE nodes, and wireless base station as BS node.

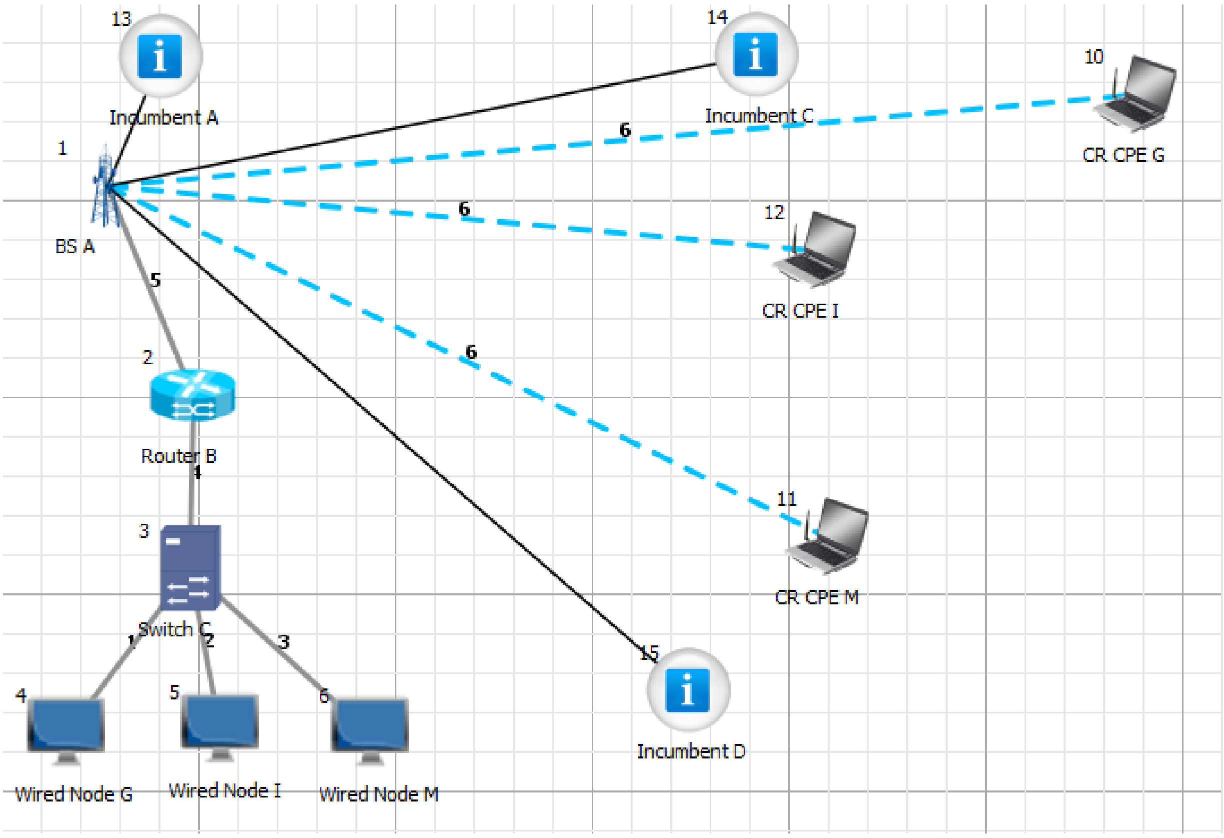


Figure 3-14: NetSim Network Simulation model

NetSim introduces a variety of parameters to configure the Cognitive Radio simulation environment, for example Transmitted power, operational frequency and time, modulation type, channels boundaries and other configuration parameters as shown in Figure 3-15.

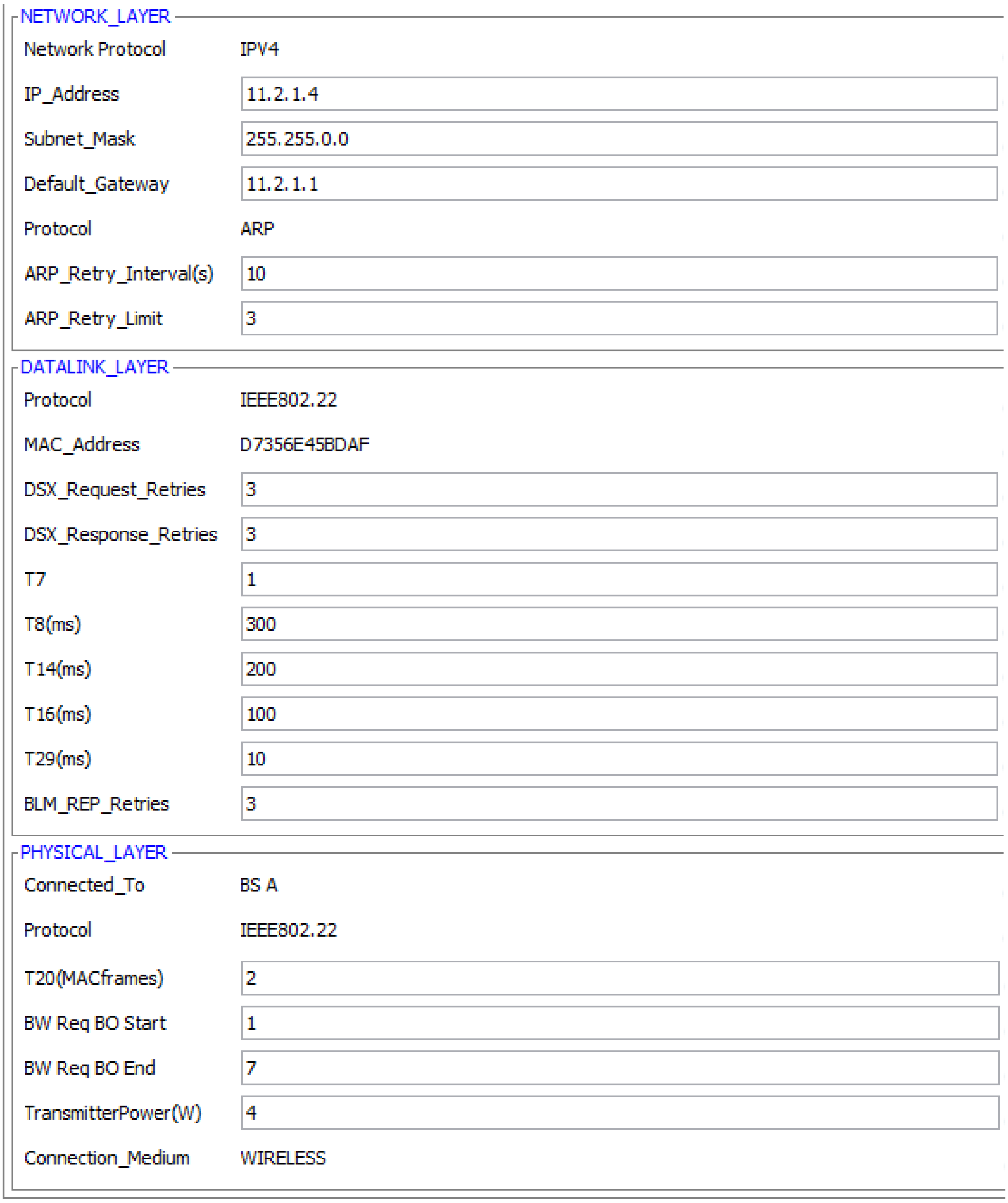


Figure 3-15: NetSim CR CPE configuration [100]

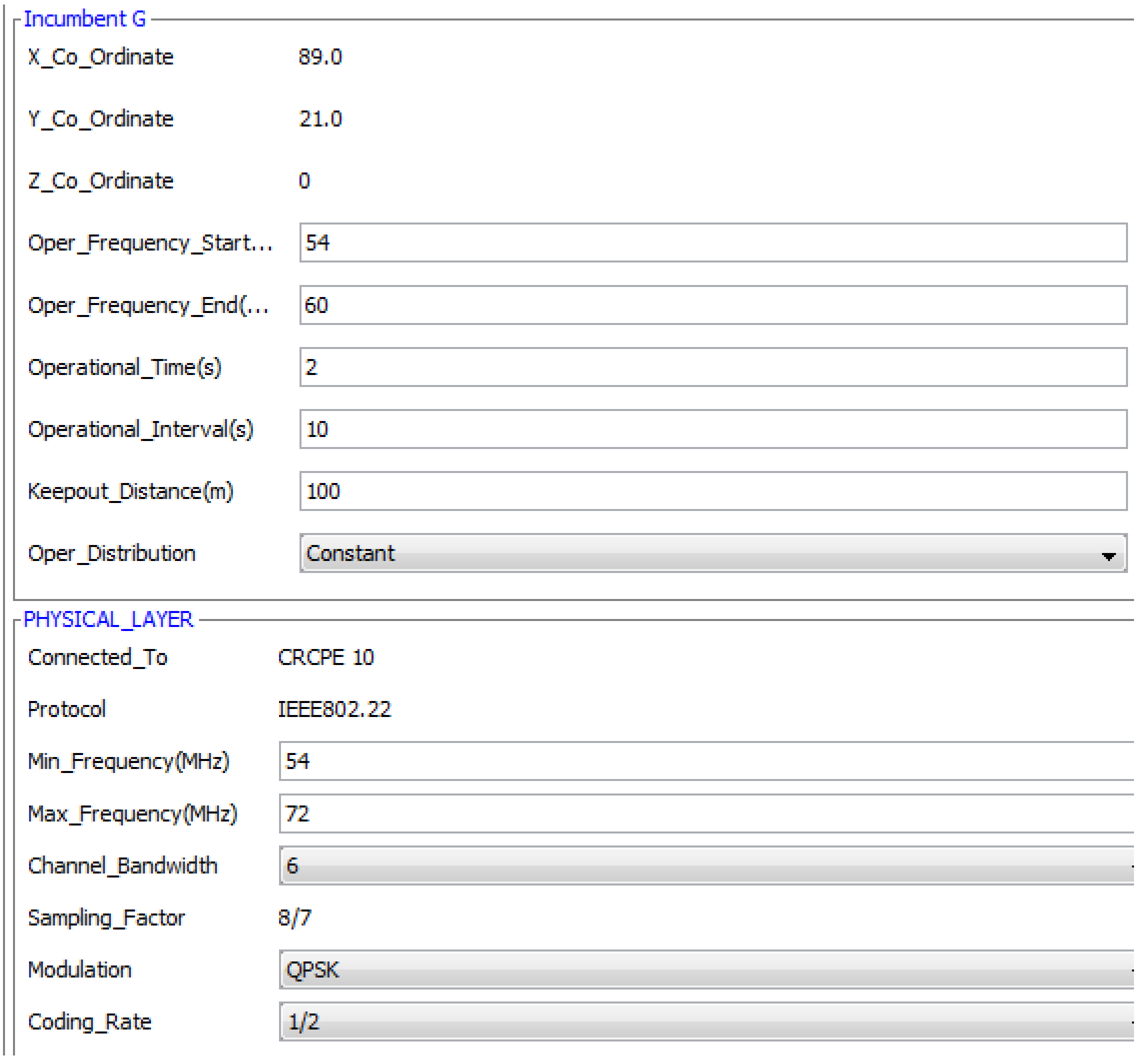


Figure 3-16: NetSim CR BS configuration [100]

Figure 3-16 shows the Base station configuration window where configuration parameters can be set for both incumbent and Cognitive Radio CPE users. Frequency and Time diversity may be used to separate the channels and prevent interference between different users.

3.3 Conclusion

Two network simulators were reviewed, where the OPNET modeler's structure was studied and explained how its C++ code was modified to include three functionalities related to cognitive radio and operation. The "spectrum sensing", "implementation of the OC-MAC protocol on MAC Layer" functions were added to support the operation of the CR and MD-CR techniques, while the "performance monitoring during SU activity and QoS assessment" function was added to the code to monitor the quality of service during the operation of the proposed C-MD-CR technique discussed in chapter five. The OPNET modeler modified code was verified by running test simulations as shown earlier.

CHAPTER FOUR

EVALUATING NETWORK PERFORMANCE IN MULTIPLE DIMENSION COGNITIVE RADIO NETWORKS

This chapter evaluates the performance of multiple dimension cognitive radio technique in terms of data throughput, packet loss and bit error rate based on network simulations. Finally it compares the simulation results of the multiple dimension cognitive radio to conventional cognitive radio technique.

Multiple Dimension Cognitive Radio technique was evaluated using the OPNET and NetSim Simulators, and its performance is compared to the conventional Cognitive radio approach. The following parameters and conditions were assumed to be used in the performance simulations as these parameters are common in data networks simulation:

- Channel bandwidth 6 Mbps
- Packet Arrival pattern based on Poisson distribution.
- Packet size = 1920 bits
- Modulation type used is QPSK as Phase Shift Keying Modulation technique minimizes the errors, and hence Signal to noise ratio is improved, hence better Quality of Service is achievable.
- In conventional CR, the average time interval of operation is 10 sec for both PU and SU, where each SU waits for its time slot to use the channel (time diversity).
- In MD-CR, no specific time interval is required as the SU uses multiple dimensions (time, frequency and modulation) to access the channel.
- Average arrival rate of users is 30 sec.

4.1 Network model

The simulation network as shown in Figure 4-1 consists of 3 primary users PU (Incumbent), 3 secondary users SU (CR CPE), connected to base-station (BS A) and transferring different traffic types like http, VoIP voice and CBR traffic. The base-station is connected to wired network which consists of layer 3 router and layer 2 switch to route the traffic to group of application servers (Wired Node) which serves the users applications. Layer 2 switch is forwarding the traffic to the right server inside the service provider's LAN, while the router is responsible of routing the packets within WAN consists of several service providers.

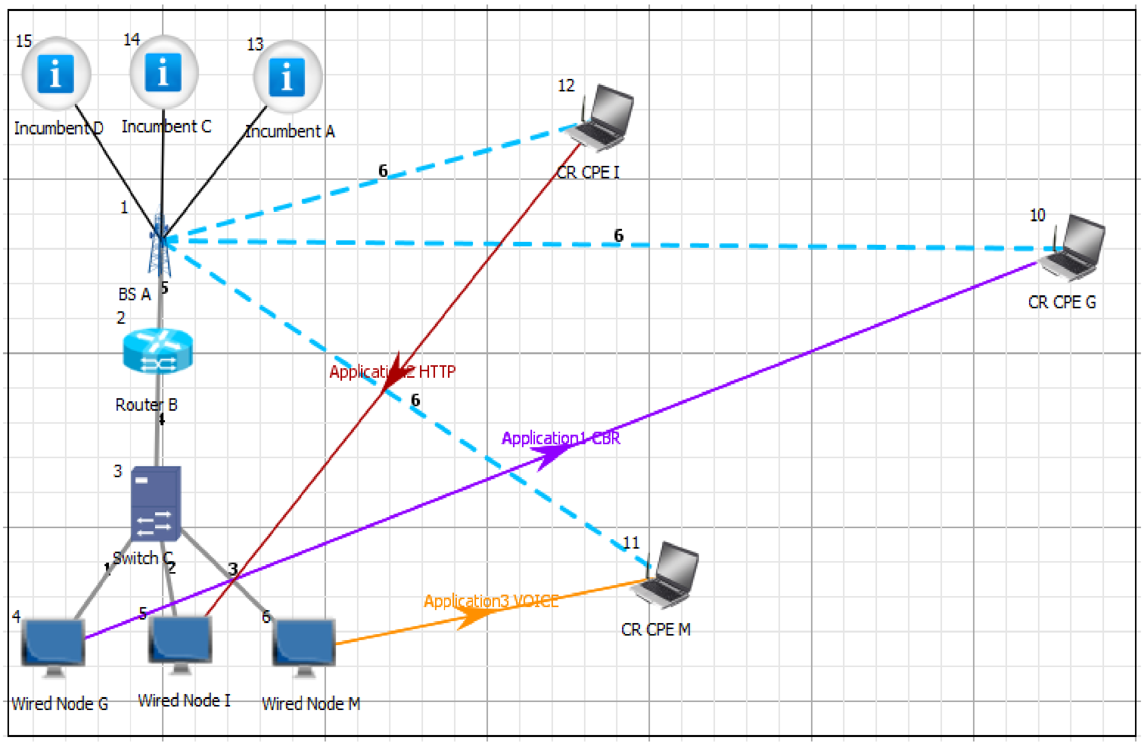


Figure 4-1: Performance evaluation Simulation network model

The service provider network consists of group of servers (Wired Node) that provide different services to the network users, for example http web browsing, voice over IP, and CBR service like file transfer. The application arrows in the figure shows the packets flow within each pair (Server – user) taking in consideration that the packets flow through the base station.

4.2 TCP throughput

Figure 4-2 shows TCP cumulative throughput for both conventional CR and MD-CR at the wired network. The graph shows that the TCP throughput of MD-CR technique is higher than the conventional CR, for example at time 600 seconds the throughput difference is more than 350 Megabits. The figure also indicates that the conventional CR suffers from "waiting" periods where the SU is waiting for the PU to finish its activity and releases the channel. While the MD-CR technique does not suffer from long waiting periods, its throughput varies according to the available resource (dimension) and its effect on channel quality of service.

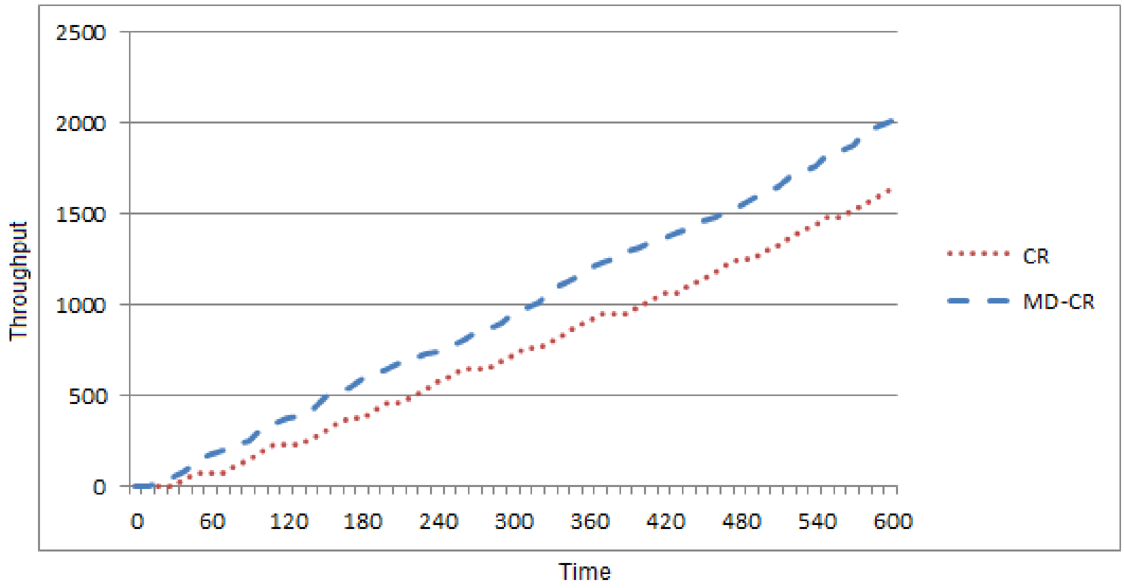


Figure 4-2: Accumulative Throughput of conventional CR, MD-CR Accumulative throughput in M bits, and Time in Seconds

4.3 Packet loss ratio

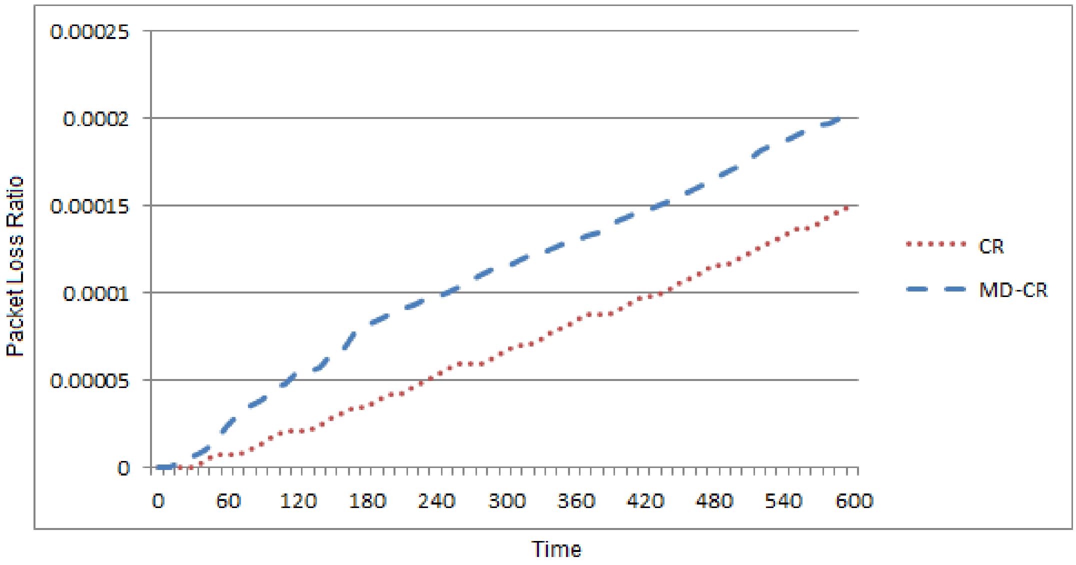


Figure 4-3: Packet Loss Ratio of conventional CR, MD-CR
Time in seconds

Figure 4-3 shows packet loss ratio for both conventional CR and MD-CR techniques. The packet loss ratio of the MD-CR technique is higher than the ratio of conventional CR due to the fact that it accepts higher number of SU connections making use of the multiple dimensions nature.

4.4 Bit error rate and SNR

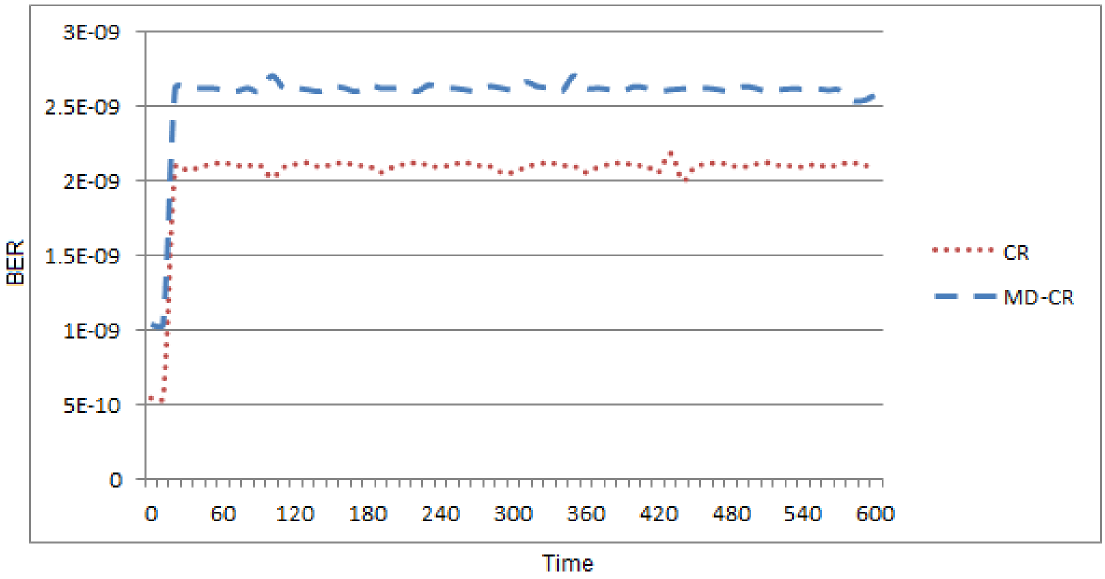


Figure 4-4: Bit Error Rate of conventional CR, MD-CR
Time in Seconds

Figure 4-4 shows Bit Error rate for both conventional CR and MD-CR techniques. The BER of the MD-CR technique is higher than the rate of conventional CR as a result of accepting higher number of SU connections making use of the multiple dimensions nature. However, this affects the Quality of Service. The related SNR values of the above BER can be deduced using the related BER/SNR curve of the modulation used in the simulation, where QPSK modulation was used in this simulation, the SNR/BER curve explained in appendix A shows that the related SNR values are arround 13 dB, and the SNR in case of conventional CR is higher than SNR in MD-CR case.

4.5 Simulation results analysis & findings

In the previous part the CR-TH in frequency diversity mode was simulated using OPNET simulator, and the following characteristics were tested and evaluated:

-TCP throughput

-Packet drop rates

-SNR

-BER

It worth mentioning that the simulation setup used in the previous section is a simple setup used to verify the operation and have preliminary evaluation for the above mentioned characteristics and from these results we can conclude the following:

1. The network's performance and quality of service (in terms of Throughput, BER, packet loss and SNR) are acceptable as long as the traffic load is not high.
2. The used simulator setup was designed to simulate few users, hence the low number of users supported the observation of point 1.
3. The network performance is still acceptable when both PUs and SUs operate simultaneously, also the SNR was accepted within the base station coverage area as the TCP throughput and packet drop rates are within the accepted range.
4. When traffic load increases, for example high rate file transfer (FTP) was used in simulation scenario instead of web browsing, the users located at longer distance are affected (TCP throughput degrades) despite of their type (primary or secondary). This complies with the basic rule in MD-CR analysis, that distance is the basic factor in its performance.

The above findings and conclusions show that the MD-CR technique is useful as a spectrum utilization technique. However in some cases it may have negative impact on the data performance (throughput, packet loss, ..).

4.6 Conclusion

The performance of the multiple dimension cognitive radio technique was evaluated using OPNET and NetSim simulations, also it was compared to the conventional cognitive radio technique. The simulation results show that the MD-CR technique has higher throughput due to its nature that tends to increase number of active channels and optimizing the successful communication probability SCP, on the other hand the MD-CR has higher packet loss ratio and bit error rate. Quality of service key performance indexes which were used to evaluate the performance of MD-CR and conventional CR at the base station are TCP throughput, packet loss ratio, bit error rate and signal to noise ratio. In next chapter a new technique is being proposed to balance between the high throughput of the MD-CR technique, and the quality of service.

CHAPTER FIVE

PROPOSED TECHNIQUE FOR CONTROLLING MD-CR QUALITY OF SERVICE

This chapter proposes the controlled multiple dimension cognitive radio technique C-MD-CR to overcome the quality of service issues related to multiple dimension cognitive radio technique. It simulates the proposed technique and evaluates its performance based on comparison with conventional and multiple dimension cognitive radio techniques.

5.1 Controlled MD-CR technique

As discussed earlier the effect of maximizing the successful communication probability in cognitive radio networks includes quality of service degradation when the network is required to deal with high demands. In many publications, for example [103,104], it is recommended to mitigate and limit the interference which results from network efforts to maximize the utilization, in order to improve the channel throughput.

In this research, a new model of MD-CR is proposed that includes a technique to optimise the relationship between maximizing the SCP and network throughput from one side and keeping the QoS within the accepted thresholds from the other side. This is important to avoid network performance deterioration which may result from a high user density in a single wireless domain as a result of maximizing the SCP. In this research, a full Cognitive Radio model has been implemented in the OPNET simulator by developing modified nodes with the appropriate software coding which include basic functionalities as explained in chapter 3. The purpose of this model is simulating the CR environment and study the network performance after applying the controlled multi dimension technique. The proposed technique observes the channel throughput on TCP level, also QoS parameters like Packet Loss and Bit Error rate, during the operation of the CR multi dimensions technique and alerts when the throughput degrades below certain level. The proposed technique has an “interactive cautious” nature which keeps monitoring the network performance and once it finds evidence of network performance deterioration it takes corrective action in order to keep the performance to an acceptable level.

The proposed technique executes the following sequence in order to compromise between maximizing the SCP and keeping the QoS in accepted level:

1. The network recognizes priority level of each user. The priority level is defined during user registration in the network based on predefined policy created by network operator, for example unlicensed users related to security and medical applications are assigned priority level 2, while commercial internet unlicensed users are assigned priority level 3. In this research project primary users are always having priority 1, while secondary users are assigned priority levels 2 and 3 according to their application.
2. When user asks for channel, the network checks its priority level. Users with priority level 1 are being allowed immediately with no more checks needed.
3. When priority 2 or 3 user asks for channel, the network checks the possibility of applying multiple dimension CR operation mode, optimizing the dimensions (frequency, time and modulation) based on operation conditions, calculates the estimated interference with the existing primary users, and then decides the validity of the new connection.
4. In case the new connection is valid, the network estimates related QoS parameters (Throughput, Bit Error Rate, and Packet loss) and decide the validity of this connection.

5. If the new connection satisfies the QoS requirements, the network allows the related user to establish the new connection.
6. During the network operation, the network keeps observing the QoS parameters and checks their values continuously.
7. If the network found QoS degradation in certain channel, the network terminates the involved priority 3 user.
8. After terminating priority 3 user the network re-check the QoS parameters based on short observation period.
9. In case the QoS parameters value still not satisfying the QoS pre-defined thresholds, the network terminates priority 2 user’s sessions.
10. After proper operation period the network may allow activating priority 2 and 3 user’s sessions again and goes through same cycle again starting step 4.

The flowchart shown in figure 5-1 illustrates the proposed process which employs MD-CR techniques (like Transmission hyperspace™) to solve the spectrum utilization problem, and in same time tries to gurantee certain level of QoS for the CR-TH system. Transmission dimensions used in this evaluations are frequency, time and modulation, where the network can change the modulation tecnique (predefined set of modulation techniques 32PSK, 16PSK, 16QAM and 64QAM) to reduce interferance and improve channel quality [105, 106, 107].

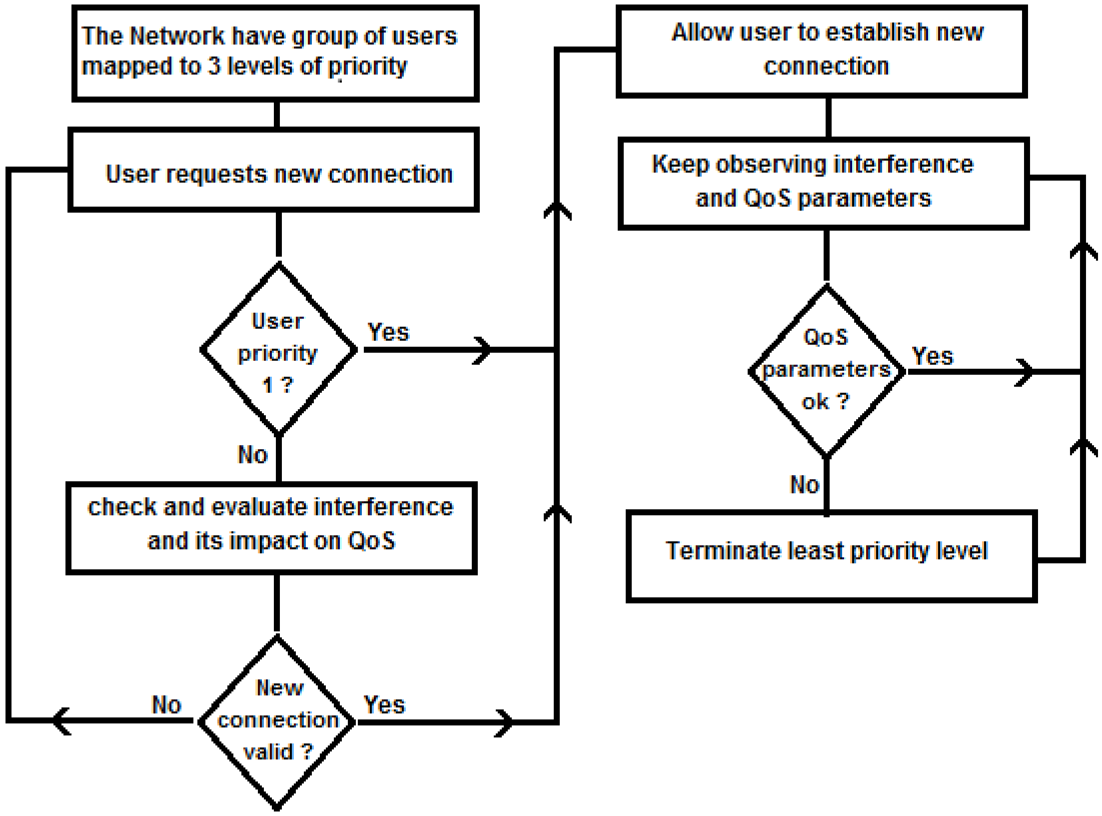


Figure 5-1: Proposed connection setup process to guarantee QoS level

In the following section the proposed technique is simulated to study and evaluate its behaviour against conventional and Multiple Dimension CR.

5.2 Simulating the Controlled MD-CR technique

The proposed technique was implemented in OPNET in order to simulate its capability to protect the Cognitive Radio network performance and keep the operation within certain range of accepted QoS. The Controlled Multiple Dimension Cognitive Radio (abbreviated as C-MD-CR) has a cautious nature as it keeps monitoring the network performance (in terms of Packet loss, Bit Error Rate, SNR and throughput) and makes use of its interactive nature to take corrective action, keeping the network performance within the accepted range. The accepted range is a set of threshold values which are defined in the simulator code to prevent QoS parameters (like packet loss and BER) to exceed certain levels. In a real world environment, such threshold values can be defined in the base station responsible for accepting/rejecting new connections and monitoring the active connections and deciding whether to allow its operation or terminates some of lowest priority connections, where group of predefined QoS KPI sets can be configured in the base station to introduce different levels of performance sensitivity according to the network applications (for example video, voice, texting, browsing). A typical threshold values used in simulations are:

Packet loss ratio: $1x10^{-5}$

BER: $1x10^{-3}$

SNR: 19 dB

The above values were defined based on theoretical recommendations [108] and they are not introducing typical QoS KPIs that used in real world, however the values were assumed to examine the efficiency of the proposed technique.

5.2.1 TCP Throughput

Figure 5-2 illustrates the cumulative throughput of the three models (conventional CR, MD-CR and controlled C-MD-CR), where the C-MD-CR throughput is higher than the throughput of the conventional CR and lower than the throughput of the MD-CR. This is due to the cautious nature of the C-MD-CR technique which may reject/terminate channels in order to keep the QoS within the accepted range. The C-MD-CR tends to keep the QoS above the predefined level hence it terminates lowest priority user connections regardless the spectrum utilization status.

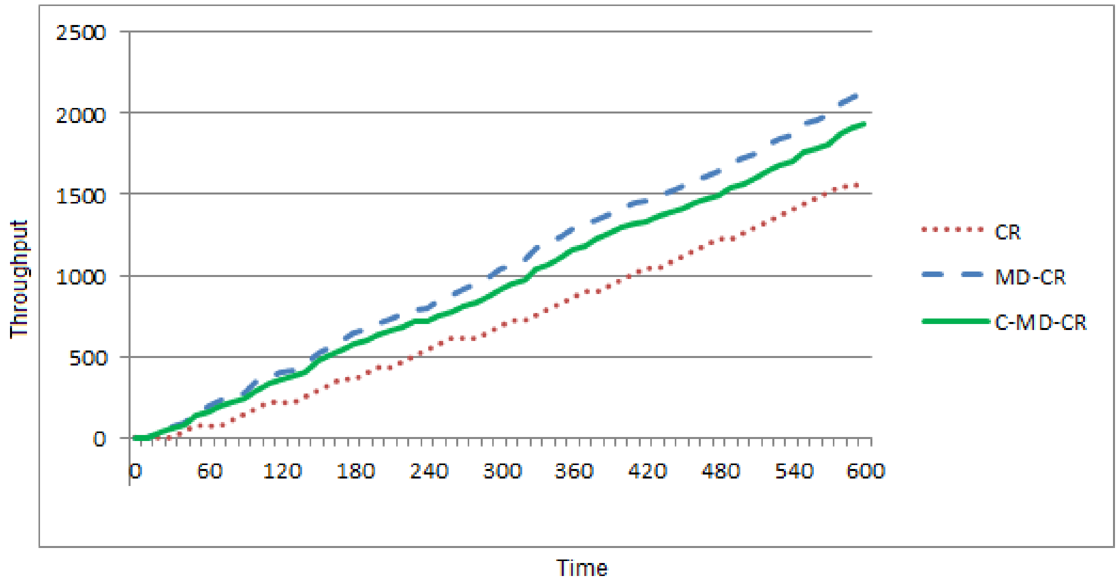


Figure 5-2: Throughput of Conventional CR, MD-CR and C-MD-CR Accumulative throughput in M bits, and Time in Seconds

The simulation indicates that the proposed C-MD-CR technique has higher utilization than the conventional CR, since the higher CP throughput at the base station end is an indication at higher network utilization, which confirms that the C-MD-CR technique can give relatively higher network utilization and TCP throughput, however it still cannot (due to its cautious nature) compete with the MD-CR in utilization.

5.2.2 Packet loss ratio

Figure 5-3 shows the commulative Packet loss ratio of the three techniques: conventional CR, Multiple dimensions CR (MD-CR), and Controlled MD-CR (C-MD-CR). The graph shows that the Packet Loss ratio of the Controlled MD-CR is relatively lower than the MD-CR technique, while it is higher than the conventional CR. This indicates that that the Controlled MD-CR technique may suffer from Packet Loss higher than the conventional CR due to its capability to increase the spectrum utilization, however it is much less than the MD-CR due to its conservative cautious nature.

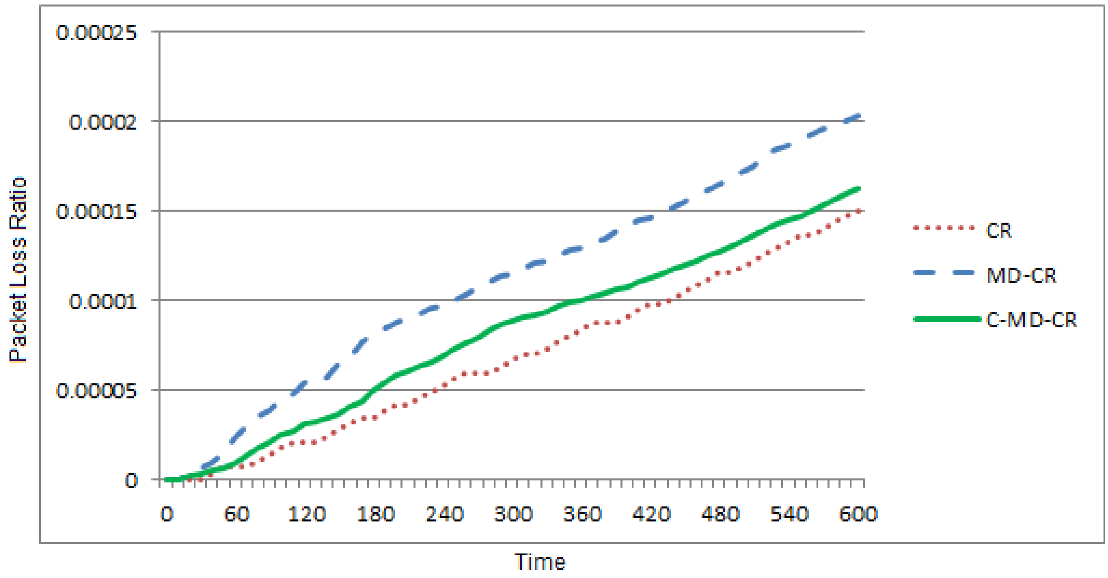


Figure 5-3: Packet Loss Ratio of Conventional CR, MD-CR and C-MD-CR Time in Seconds

The above graph shows that the Controlled MD-CR have Packet Loss values slightly higher than the conventional CR, which indicates that the C-MD-CR can achieve packet loss ration as low as the conventional CR technique although it improves the throughput and the elated spectrum utilization

5.2.3 Bit Error Rate and SNR

Figure 5-4 shows the Bit Error Rate of the conventional CR, MD-CR and Controlled MD-CR techniques. The related SNR can be calculated as explained in Appendix A, accordingly we can see that the SNR value of the Controlled MD-CR technique is around 26 dB which represents a good quality signal.

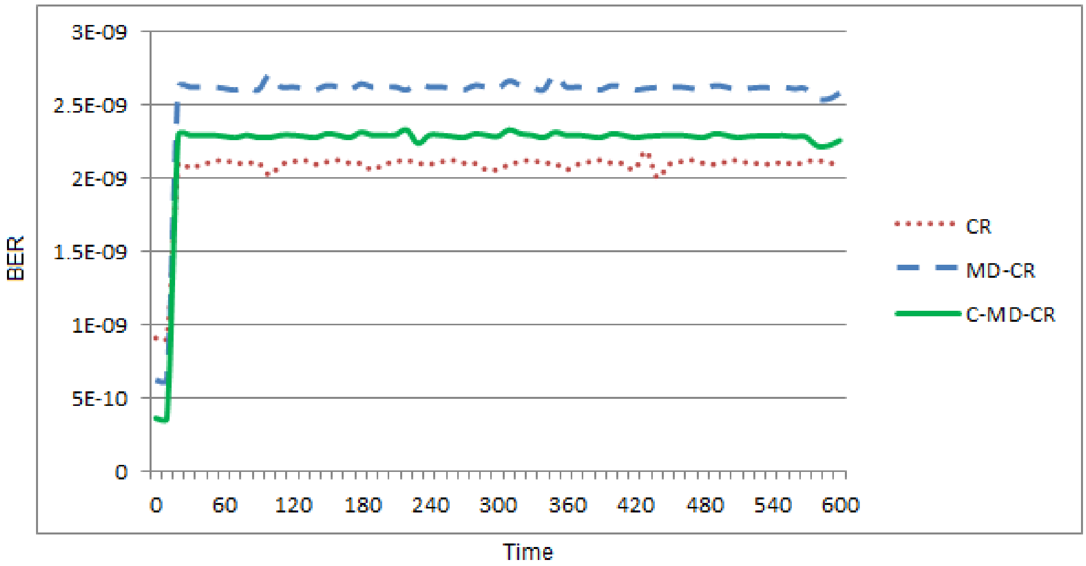


Figure 5-4: BER of Conventional CR, MD-CR and C-MD-CR
Time in Seconds

5.3 Changing the simulation parameters

In order to study the behaviour of the MD-CR and C-MD-CR techniques in different conditions the simulation environment was reconfigured to allow changing the arrival rate of the PUs also number of users in the simulated domain. The network model used in the simulations assumes default average PU arrival rate 30 second and operation period 10 seconds as explained in section 4.1, where each PU has the right to terminate the SU connection at arrival in conventional CR model.

Figure 5-5 illustrates the cumulative throughput of the three models (conventional CR, MD-CR and controlled C-MD-CR) where user arrival rate was changed to 15 sec to simulate higher channel demand and fast chages. The graph shows that the CR throughput was reduced; this is due to reducing the SU opportunity to find spectrum gap while the PUs are arriving more frequently. The C-MD-CR throughput was reduced as well however still higher than the conventional CR, while the MD-CR throughput is much higher than the conventional CR and C-MD-CR due to its nature which allows the SUU users to operate simultaneously with the PU users with no control interference.

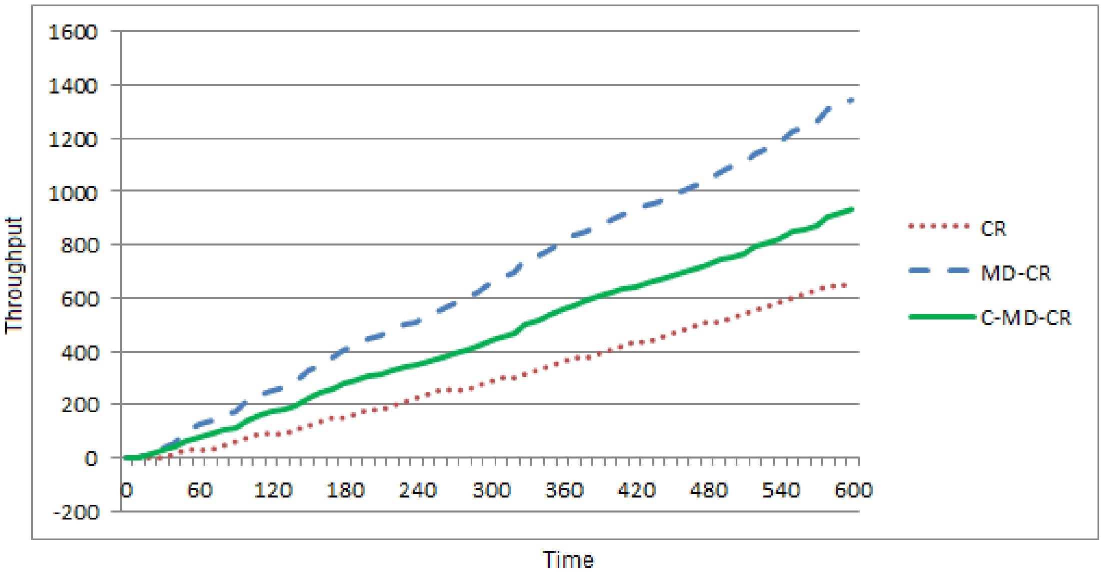


Figure 5-5: Throughput of Conventional CR, MD-CR and C-MD-CR
PU arrival rate every 15 sec
Accumulative throughput in M bits, and Time in Seconds

The packet loss ratio in this case is increased for the MD-CR technique due to the higher arrival rate of PUs, while the conventional CR and C-MD-CR have much less packet loss rates as shown in figure 5-6.

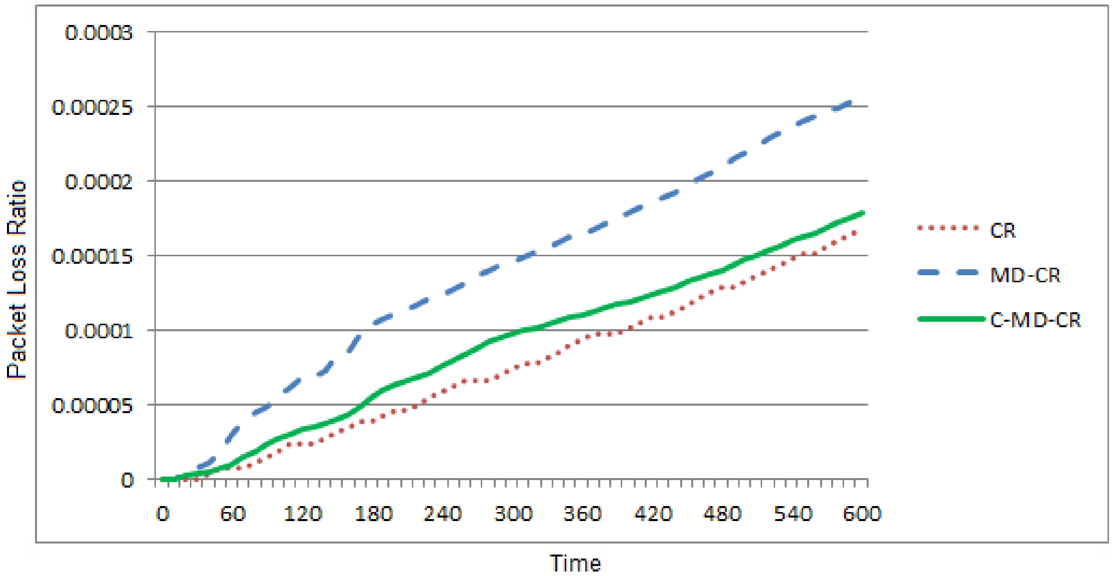


Figure 5-6: Packet Loss Ratio of Conventional CR, MD-CR and C-MD-CR PU arrival rate every 15 sec. Time in Seconds

On the other hand when reducing the arrival rate of the PUs, increasing the arrival period to 90 sec, it was noticed that the C-MD-CR throughput is improved close to the MD-CR throughput level, also the conventional CR throughput was improved however it is still much lower than the MD-CR and C-MD-CR as shown in figure 5-7. This indicates that the C-MD-CR may be more suitable for networks that encounter low PU arrival rates as this improves the SU opportunity to operate, yet with interference control.

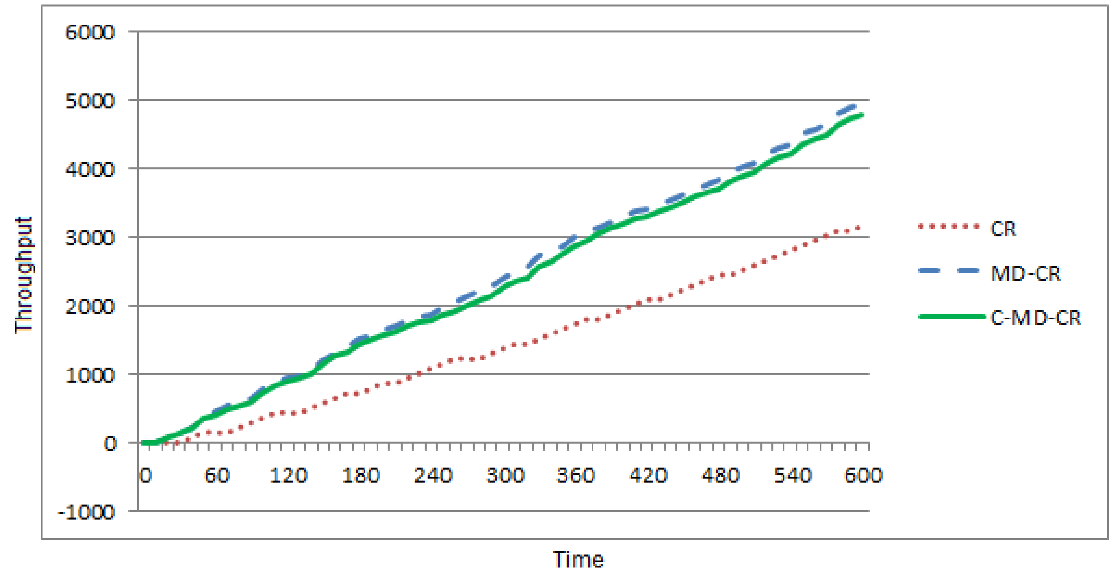


Figure 5-7: Throughput of Conventional CR, MD-CR and C-MD-CR
PU arrival rate every 90 sec
Accumulative throughput in M bits, and Time in Seconds

Figure 5-8 illustrates the effect of reducing the PU arrival rate on the packet loss ration of the three techniques, where the packet loss ratio of the three techniques were reduced keeping the same order level.

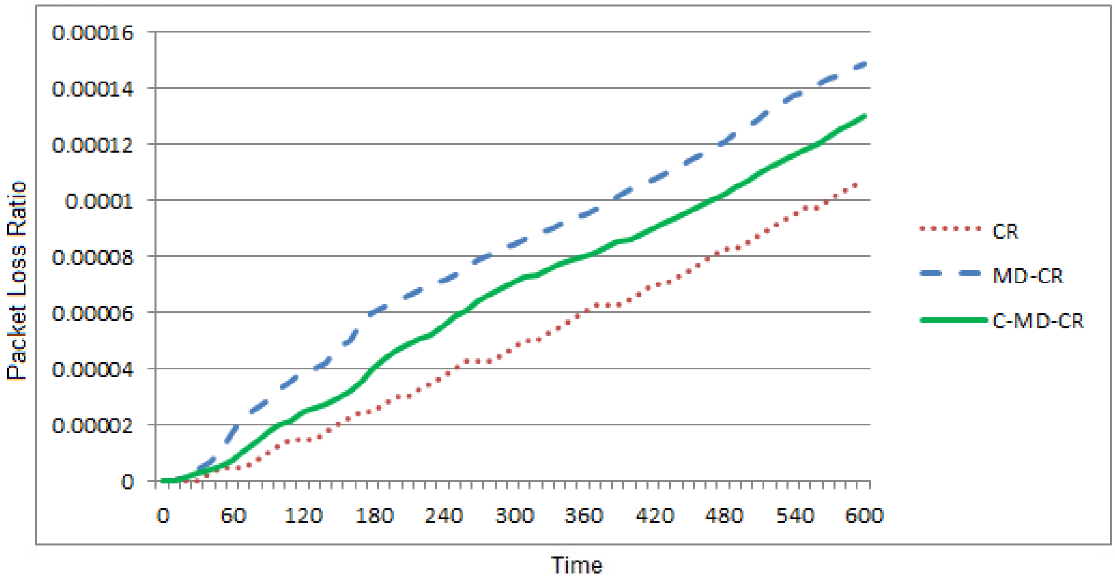


Figure 5-8: Packet Loss Ratio of Conventional CR, MD-CR and C-MD-CR
PU arrival rate every 90 sec. Time in Seconds

Figure 5-9 shows the Bit Error Rate of the conventional CR, MD-CR and Controlled MD-CR techniques in high density user environment, where number of users is increased to 14 instead of 6 to simulate higher traffic, also data traffic generation is increased by increasing the packet arrival rate and changing the packet arrival distribution to exponential instead of constant to increase the load dramatically. The result of the simulation shows that the BER level of the three techniques was increased compared to previous case in figure 5-4, however BER new level of the C-MD-CR technique is almost same as CR new level, and even lower due to its conservative and cautious nature toward the quality of service.

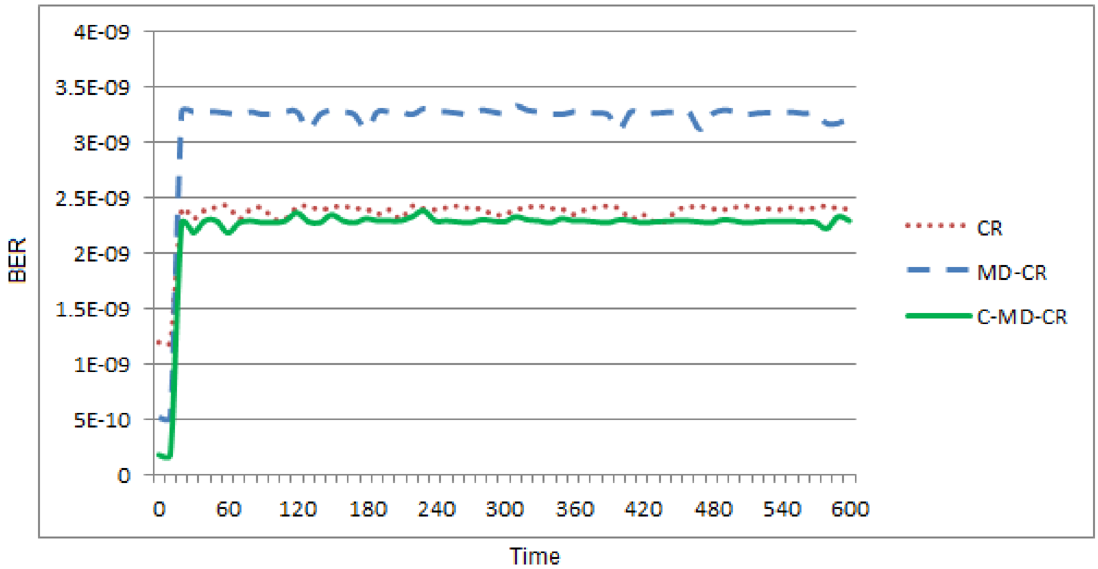


Figure 5-9: BER of Conventional CR, MD-CR and C-MD-CR high user density Time in Seconds

5.4 Simulation results findings

The simulation results show that the C-MD-CR technique has intermediate performance between the conventional CR and MD-CR in terms of throughput, packet loss, bit error rate and SNR, where it could achieve throughput level close to MD-CR technique while it keeps monitoring the quality of service KPIs, hence the proposed technique could balance between the need for high spectrum utilization and the less interference. The C-MD-CR with its cautious nature tends to increase the network throughput while keeping the packet loss and bit error rate at an acceptable level. This was verified also when changing the PU arrival rates where the C-MD-CR showed higher throughput (compared to the conventional CR) with packet loss rate less than the MD-CR technique.

5.5 Conclusion

Proposed technique to control the performance of the multiple dimension cognitive radio technique was studied in this chapter. The controlled multiple dimension cognitive radio technique C-MD-CR aims to keep the quality of service within predefined range by controlling number of active channels that serve both primary and secondary users taking in consideration that primary users always have higher priority in using the available channels. The C-MD-CR technique classifies the users in several priority levels and keep monitoring the network quality of service KPIs like throughput, packet loss, SNR and in case one of these KPIs crossed the predefined threshold it terminates the least priority channels in order to save the quality of service.

C-MD-CR was simulated using both OPNET and NetSim, where the simulation results show that the C-MD-CR technique has intermediate performance between the conventional CR and MD-CR in terms of throughput, packet loss, bit error rate and SNR, where it could achieve high throughput level close to MD-CR technique while the packet loss ratio and bit error rate were less, which means that C-MD-CR could satisfy the need for high spectrum utilization and had less interference due to its cautious nature it tends to increase the network throughput while keeping the packet loss and bit error rate at an acceptable level.

CHAPTER SIX

CONCLUSION AND FUTURE WORK

In this chapter, a summary of the thesis is presented, results are discussed and recommendations for further work are proposed. The conclusions are provided in Section 6.1, while the suggestions for future works are listed in Section 6.2.

6.1 Research conclusion

In this research, conventional spectrum utilization techniques like Cognitive Radio were studied, also, new trends in spectrum utilization by employing multiple dimensions (including frequency, time, antenna directionality, modulation) in the Cognitive Radio environment, and both techniques were compared to find which one is more capable of improving the spectrum utilization and to have accepted quality of service.

The main question addressed by this thesis is how to balance between the advantages of the conventional Cognitive radio technique, and the high utilization of the Multiple Dimensions Cognitive Radio technique, resulting from its capability to improve the successful communication probability based on the orthogonal transmission dimensions. This question was answered by:

- Studying and simulating the performance of both conventional Cognitive radio networks, and Multiple Dimensions Cognitive Radio Networks techniques.
- Observing the differences in performance between the two techniques in terms of Spectrum Utilization, Network Throughput, Packet Loss and Bit error rate.
- Proposing the controlled Multiple Dimensions Cognitive Radio (C-MD-CR) to optimize the utilization of the conventional cognitive radio technique while monitoring the Quality of Service in a cautious way. It classifies the users connected to the wireless domain according to user priority, and if QoS measures are not realized it terminates the lowest priority users.

The simulations show that the Multiple Dimension Cognitive Radio technique is useful in improving the spectrum utilization and maximize the Throughput based on making use of the multiple dimensions environment beside the opportunistic behaviour of the Cognitive Radio system. However it may affect the network performance due to its tendency to increase number of SU regardless the resulting interference. The Controlled Multiple Dimensions Cognitive Radio technique (C-MD-CR) is introduced in this research to keep the network performance within the accepted range, and based on its performance simulation the resulted Throughput is comparable to the MD-CR technique Throughput, while the interference is less. Table 6-1 summarizes and compares the performance of the three modes discussed in this research in terms of TCP Throughput, Spectrum Utilization and Interference.

	Conventional Cognitive Radio	**Multiple Dimensions Cognitive Radio**	**Controlled Multiple Dimensions Cognitive Radio**
Throughput	Avg.	High+	High
Spectrum utilization	Avg.	High+	High
Interference	Low	Avg.	Low

Table 6-1: comparison between the 3 CR models

The simulation results showed that the Multiple Dimensions CR has higher throughput and higher data error rate and this is due to its strong tendency to increase number of users in RF space, while the conventional Cognitive Radio has mild tendency to increase the utilization hence the data errors are less. When comparing the simulation results of the proposed technique (C-MD-CR) with the conventional and Multiple Dimensions CR techniques it was found that the C-MD-CR with its cautious nature and its tendency to improve the spectrum utilization it could achieve balance between the key advantages of the conventional and multiple dimension techniques from side, and the desired quality of service from another side.

The performance of the three techniques was verified through software simulations where network model consists of both PU and SU users was used to simulate the network behaviour, monitoring the performance metrics like throughput, packet loss and bit error rate while changing some operation parameters like number of users and user arrival rate. It was found that the conventional techniques are more capable of applying simple rules to keep the spectrum organized with average utilization and accepted Quality of Service, while the Multiple Dimensions CR techniques tend to improve the spectrum utilization in terms of “successful communication probability” regardless the Quality of Service.

6.2 Suggested future work

Future work could include involving other dimensions such as coding, power level and antenna directionality [109] to improve the PU and SU co-access performance [110], also to consider multiple hops instead of single hop, for example accessing primary base station through primary network access, and considering user mobility during simulation to measure the effect of distance in parallel to other dimensions.

Improving throughput is attractive research area where different techniques may be considered to balance between spectrum utilization and improved throughput, for example data link layer algorithms may be used to solve the trade-off problem between sensing and throughput [111], also special technique to perform spectrum sensing and data transmission at same time [112] may be combined with the C-MD-CR to improve the throughput significantly.

Another interesting research area is the spectrum sensing where it affect the overall performance of the cognitive radio network either by its impact on the network throughput or the efficiency of finding spectrum holes [113]. A novel techniques were developed to improve the spectrum sensing process for example using dynamic sensing strategies which can adaptively schedule the sensing frequency based on link conditions [114], or using the spatial technique which uses direct relays to find the spectrum holes, hence improve the CR opportunities [115, 116], or by using energy detection which help to improve the overall throughput of the CR network [117, 118, 119].

Several applications for Cognitive Radio were developed to allow user devices to sense the operation characteristics besides traditional spectrum sensing. This is useful in determining best operational settings for the device, for example best power level, modulation technique, mode of transmission, and suitable QoS mode, which opens several doors for more intelligent self-organized wireless networks in the future for example next generation LTE networks. Spectrum resource review and allocation can be added value for the cognitive radio networks where spectrum access efficiency can be improved [120].

Tailoring special transport protocol for cognitive radio networks as special edition of the Transmission Control Protocol (TCP) is important trend as well. Some research talked this issue like [53, 54, 121, 122] and could provide promising TCP model which can be utilized in future implementations [123, 124]. Also modulation techniques can contribute in enhancing the SNR and reducing interference and BER may be used to improve the communication quality by reducing the interference like 32PSK [68, 125, 126]. Some interesting research are focusing on employing spatial modulation techniques in order to have interference free spectrum in cognitive radio networks [107], while other research focus on developing special modulation techniques suitable for cognitive radio [127 - 133].

One of the interesting ideas which will be considered in the Controlled Multiple Dimensions CR is the adaptive packet size technique which is very useful to adapt the performance according to channel quality. An initial concept was introduced [134] to adapt energy efficiency of wireless networks to reduce the power consumption and improve the Quality of Service, and then developed in different approaches [138, 139] to introduce QoS aware techniques.

Internet of things (IoT) is very interesting research area in conjunction with Cognitive Radio networks [140], where spectrum utilization is required to support the IoT applications and cloud services to present reliable and smart IoT solutions [141, 142] which employs adaptive and reliable techniques that capable of controlling the network delay and energy efficiency in same time [143, 144] like adaptive packet size as discussed before.

Network security in cognitive radio networks becomes critical issue where threats like malicious and selfish attacks can take place [162, 163] where the attacker emulate fake low or high spectrum utilization to disturb the spectrum sensing process done by the secondary user [164]. Other security threats like spectrum sensing data falsification and control channel saturation attacks may occur in data link layer level where different techniques are required to be developed to prevent such threats. Network security is considered one of the QoS parameters today and the need for reliable security scheme is increasing where it should be combined with reliable spectrum utilization technique to keep the main opportunistic feature of the cognitive radio.

Also special implementations are proposed to make the cognitive radio networks suitable for the critical applications like medical and emergency applications [166, 167], moreover vehicular ad hoc networks (CRVs) is recent trend which employs the cognitive radio to serve the vehicle applications, however this kind of CRN application need to be aware of the nature of the user behaviour in terms of mobility, velocity and service geographical boundaries [168].

REFERENCES

[1] J. E. Brittain, “Electrical Engineering Hall of Fame”, Proceedings of the IEEE 2005, Volume 93, Issue 2, Year 2005.

[2] M. Zeng ; Victoria Univ., BC, Canada ; A. Annamalai ; V. K. Bhargava, "Recent advances in cellular wireless communications", IEEE Communications Magazine, Volume 37, Issue 9, 2002.

[3] Stallings, William; Stallings, William “Wireless communications and networks” text book, Upper Saddle River, N.J. : Prentice Hall2005

[4] Mustafa Kocakulak; Ismail Butun, "An overview of Wireless Sensor Networks towards internet of things", Computing and Communication Workshop and Conference (CCWC), March 2017 IEEE 7th Annual, Las Vegas, NV, USA

[5] Ahmed Qaddus; Abid Ali Minhas, "Wireless communication a sustainable solution for future smart grid networks", Open Source Systems & Technologies (ICOSST), 2016 International Conference on February 2017 Lahore, Pakistan

[6] ITU ICT Data and Statistics: National Telecommunication Agencies. http://www.itu.int/ITU-D/ict/links

[7] DAVID B. DAVIDSON “Computational Electromagnetics for RF and Microwave Engineering” text book, Cambridge University Press 2005

[8] David tsa, Pramod Viswanath “Fundamentals of Wireless Communication” text book, Cambridge University Press 2005

[9] Martin Cave, Sumit Majumdar, Ingo Vogelsang, "Handbook of Telecommunications Economics", Amsterdam: Elsevier Science B.V., 2002.

[10] "ITU releases 2014 ICT Figures", 5 May 2014, http://www.itu.int/net/pressoffice/press_releases/2014/23.aspx, and http://www.itu.int/en/ITU-D/Statistics/Pages/stat/default.aspx.

[11] Marja Matinmikko, Marko Höyhtyä, Miia Mustonen, Heli Sarvanko, Atso Hekkala, Marcos Katz, Aarne Mämmelä, Markku Kiviranta, Aino Kautio "Cognitive radio: An intelligent wireless communication system" VTT project: Channel state estimation and spectrum management for cognitive radios, 2008, http://www.vtt.fi

[12] W. Y. Lee and I. F. Akyildiz, "A Spectrum Decision Framework for Cognitive Radio Networks,"IEEE Transactions on Mobile Computing, vol. 10, no. 2, February 2011

[13] Hermsmeyer, Christian ; Chief Technical Office of the Optics Division at Alcatel-Lucent in Nuremberg, Germany ; Song, Haoyu ; Schlenk, Ralph ; Gemelli, Riccardo, "Towards 100G packet processing: Challenges and technologies" Bell Labs Technical Journal (Volume:14 ,issue:2) summer 2009.

[14] Sayed, Z. ; Bell Labs, Lucent Technol., Holmdel, NJ ; Ying Hu ; Tang, B. ; Mezhoudi, M. "Transport Technologies for Wireless Backbone Networks" Telecommunications Network Strategy and Planning Symposium, 2006. NETWORKS 2006. 12th International, November 2006.

[15] Q. Zhao and B. M. Sadler, "A survey of dynamic spectrum access" IEEE Signal Process. Mag., vol. 24, no. 3, pp. 79-89, May 2007.

[16] Piyush Gupta , P. R. Kumar, "The Capacity of Wireless Network" IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 46, NO. 2, MARCH 2000

[17] Qiang Zhang ; Fixed Network Res. Dept., Huawei Technol. Co., Ltd., Wuhan, China ; Quan Cao ; Shengmeng Fu ; Ruiqiang Ji "Compact silicon-based DML transmitter with extinction ratio improved for 40G TWDM PON OLT scenarios" Optical Fiber Communications Conference and Exhibition (OFC), 2015

[18] Aparicio-Pardo, R. ; Univ. Polit. de Cartagena (UPCT), Cartagena, Spain ; Pavon-Marino, P. ; Mukherjee, B. "Robust upgrade in optical networks under traffic uncertainty" Optical Network Design and Modeling (ONDM), 16th International Conference 2012.

[19] Mezhoudi, M. ; Bell Labs., Alcatel-Lucent Technol., Murray Hill, NJ, USA ; Ying Hu "Economical analysis of NG-optical backbone transport network" Wireless and Optical Communications Conference (WOCC), 2011 20th Annual April 2011.

[20] Kishor Patil; Ramjee Prasad; Knud Skouby, "A Survey of Worldwide Spectrum Occupancy Measurement Campaigns for Cognitive Radio", Devices and Communications (ICDeCom), 2011 International Conference on February 2011, Mesra, India.

[21] MacDonald, John T. "A survey of spectrum utilization in chicago." Illinois Institute of Technology, Tech. Rep (2007).

[22] Islam, Md Habibul, et al. "Spectrum survey in Singapore: Occupancy measurements and analyses." Cognitive Radio Oriented Wireless Networks and Communications, 2008. CrownCom 2008. 3rd International Conference on. IEEE, 2008.

[23] International Union of Radio Science, URSI Radio Science Bulletin No 354 issued September 2015.

[24] M. Mehdawi, N. Riley, K. Paulson, A. Fanan, and M. Ammar, "Spectrum Occupancy Survey in Hull-UK for Cognitive Radio Applications: Measurement & Analysis," International Journal of Scientifi c & Technology Research, 2, 4, April 2013, pp. 231-236.

[25] Federal Communications Commission Spectrum policy Task Force "Report of the Spectrum Efficiency Working Group" November 2002.

[26] Ericsson publications: the spectrum crunch, http://www.ericsson.com/openarticle/the-spectrum-crunch_1473414408_c

[27] Vaclav Valenta, Roman Marsalek," Survey on Spectrum Utilization in Europe: Measurements, Analyses and Observations," 5th International ICST Conference on Cognitive Radio Oriented Wireless Networks and Communications, Cannes, France, 2010.

[28] Wellens, Matthias, Jin Wu, and Petri Mahonen. "Evaluation of spectrum occupancy in indoor and outdoor scenario in the context of cognitive radio." Cognitive Radio Oriented Wireless Networks and Communications, 2007. CrownCom 2007. 2nd International Conference on. IEEE, 2007.

[29] J. M. Chapin and W. H. Lehr, "The Path to Market Success for Dynamic Spectrum Access Technology," IEEE Communications Magazine, May 2007, pp. 96-103.

[30] W. Lehr (chair), "Towards More Efficient Spectrum Management,"MIT Communications Futures Program White Paper, March 2014.

[31] Mitola, J.; Marshall, P.; Kwang-Cheng Chen; Mueck, M.; Zvonar, Z. "Software defined radio" Communications Magazine, IEEE 2016, Volume: 54, Issue: 1

[32] Mitola, J.; Maguire, G.Q., Jr. " Cognitive radio: making software radios more personal" Personal Communications, IEEE (Volume:6 , Issue: 4), August 1999.

[33] Haykin, Simon "Cognitive radio: brain-empowered wireless communications" Selected Areas in Communications, IEEE Journal on (Volume:23 , Issue: 2), Feb. 2005.

[34] F. Zirilli, "Review of the Book "Advanced Cognitive Radio Network"," International Journal of Communications, Network and System Sciences, Vol. 5 No. 4, 2012, pp. 207-207. doi: 10.4236/ijcns.2012.54025.

[35] N. Khambekar; C. Spooner; V. Chaudhary,"Characterization of the missed spectrum-access opportunities under dynamic spectrum sharing", 2016 8th International Conference on Communication Systems and Networks (COMSNETS)

[36] Tevfik Yiicek and Hiiseyin Arslan, "Spectrum Characterization for Opportunistic Cognitive Radio Systems" Military Communications Conference, 2006. MILCOM 2006. IEEE 2006

[37] R. Pinyi, W. Yichen, D. Qinghe and X. Jing, "A Survey on Dynamic Spectrum Access Protocols for Distributed Cognitive Wireless Networks," EURASIP Journal on Wireless Communications and Networking, 2012.

[38] Lagkas, Thomas D., ed. Evolution of Cognitive Networks and Self-Adaptive Communication Systems. IGI Global, 2013.

[39] Vieira de Lima, M. ; Telecommun. Metrol. Div., INMETRO Duque de Caxias, Caxias, Brazil ; da Silva Mello, L. "Cognitive radio simulation based on spectrum occupancy measurements at one site in Brazil", Microwave & Optoelectronics Conference (IMOC), 2013 SBMO/IEEE MTT-S International.

[40] Watanabe, K.; Ishibashi, K.; Kohno, R. "Performance of Cognitive Radio Technologies in the Presence of Primary Radio Systems" Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007. IEEE 18th International Symposium on year 2007.

[41] P. Phunchongharn, E. Hossain, D. Niyato an linga, “A Cognitive Radio System for E-Health Applications in a Hospital Environment,” IEEE Wireless Communications, Vol. 17, 2010.

[42] Passas, Virgilios, et al. "Towards the efficient performance of LTE-A systems: Implementing a cell planning framework based on cognitive sensing." Cognitive Cellular Systems (CCS), 2014 1st International Workshop on. IEEE, 2014.

[43] http://www.androcs.com/html/spectrum_management.html

[44] Andrew L. Drozd, “EMC Computer Modeling of Distributed Wireless Systems to Assure Efficient Utilization of the RF Transmission Hyperspace” ANDRO Computational Solutions, LLC, Rome, NY, USA 2006.

[45] A. L. Drozd, I. P. Kasperovich, C. E. Carroll, A. C. Blackburn, C. K. Mohan, P. K. Varshney, and D. D. Weiner, “Computational Electromagnetics applied to analyzing the efficient utilization of the RF Transmission Hyperspace” in Proc. IEEE/ACES Int. Conf. on Wireless Comm. and Applied Comp. Electromagnetics, Apr. 2005.

[46] A. L. Drozd, C. K. Mohan, P. K. Varshney, and D. D. Weiner, “Multiobjective joint optimization and frequency diversity for efficient utilization of the RF Transmission Hyperspace,” in Proc. 1st Int.Waveform Diversity and Design Conf., Edinburgh, UK, 2004.

[47] Onur Ozdemir, Andrew L.Drozd, Engin Masazade, P.K.Varshney, “Successful Communications in a Cognitive Radio Network with Transmission Hyperspace” in Global Telecommunications Conference (GLOBECOM 5-9 Dec.2011).

[48] Ankita Panwar, Piyush Bhardwaj, Onur Ozdemirt, Engin Masazade, Chilukuri K. Mohan, Pramod K. Varshney, and Andrew L. Drozd “On Optimization Algorithms for the Design of Multiband Cognitive Radio Networks” IEEE 2012.

[49] Kaniezhil, R., C. Chandrasekar, and S. NithyaRekha. "Performance evaluation of QoS parameters in dynamic spectrum sharing for heterogeneous wireless communication networks." arXiv preprint arXiv:1204.4550 (2012).

[50] Homayounzadeh, Alireza, and Mehdi Mahdavi. "Quality of service provisioning for real-time traffic in cognitive radio networks." IEEE Communications Letters 19.3 (2015): 467-470.

[51] Alok Kumar Jagadev, Binod Kumar Pattanayak, Ajit Kumar Nayak, Manojranjan Nayak, "EVALUATION OF QoS PARAMETERS ON TCP/IP IN WIRELESS AD HOC NETWORKS", Journal of Theoretical and Applied Information Technology, October 2011.

[52] Chakraborty, T.; Mukhopadhyay, A.; Bhunia, S.; Misra, I.S.; Sanyal, S.K. “Analysis and enhancement of QoS in cognitive radio network for efficient VoIP performance” Information and Communication Technologies (WICT), 2011 World Congress on 2011

[53] Jian Wang, Aiping Huang, Wei Wang, Zhaoyang Zhang and Vincent K N. Lau, “Analysis of TCP Throughput in Cognitive Radio Networks” GLOBECOM Workshops, page 930-935. IEEE, (2011).

[54] Y. Kondareddy and P. Agrawal, “Effect of dynamic spectrum access on transport control protocol performance,” in IEEE Global Telecommunications Conference, GLOBECOM, 2009.

[55] M. Dong, G. Sun, X. Wang, and Q. Zhang, “Combinatorial auction with time-frequency flexibility in cognitive radio networks,” in Proc. 2012 IEEE INFOCOM, pp. 2282–2290.

[56] Changle Li; Zhe Liu; Xiaoyan Geng; Mo Dong; Feng Yang; Xiaoying Gan; Xiaohua Tian; Xinbing Wang, “Two Dimension Spectrum Allocation for Cognitive Radio Networks”, IEEE Transactions on Wireless Communications, Volume 13, issue 3, year 2014.

[57] L. Gao, X. Wang, Y. Xu, and Q. Zhang, “Spectrum trading in cognitive radio networks: a contract-theoretic modeling approach,” IEEE J. Sel. Areas Communication, 2011.

[58] J. Huang, R. Berry, and M. Honig, “Auction-based spectrum sharing,” Mobile Network Applications, vol. 11, no. 3, pp. 405–418, 2006.

[59] S. Sodagari, A. Attar, and S. Bil´en, “On a truthful mechanism for expiring spectrum sharing in cognitive radio networks,” IEEE J. Sel. Areas Communications, 2011.

[60] Xiaojun Li ; Second Eng. Bur., China Telecom Constr. Co., Ltd., Xi'an, China ; Quanzhong Li ; Kai Liu, "Opportunistic Spectrum Sharing with Quality-of-Service Support in Cognitive Radio Networks", Computational Science and Engineering (CSE), 2014 IEEE 17th International Conference on December 2014

[61] Andrew S.Tanenbaum, “Computer Networks” textbook, 5th edition.

[62] ISO/IEC 7498-1 International Standard, Information technology – Open Systems Interconnection – Basic reference Model: The Basic Model ,second edition 1994-11-15, corrected 1996-06-15

[63] Miao, Guawang; Song, Guocong (2014). Energy and spectrum efficient wireless network design. Cambridge University Press. ISBN 1107039886

[64] Transmission Control Protocol, Protocol specifications, IETF RFC 793, September 1981.

[65] TCP Extensions For High Performance, IETF RFC 1323, May 1992.

[66] William Stallings, "Data and Computer Communications" textbook, 10th edition,

[67] Some Internet Architectural Guidelines and Philosophy, IETF RFC 3439, December 2002.

[68] Bernard Sklar "Digital Communications, Fundamentals and Applications", text book, Prentice Hall PTR2001.

[69] http://www.ieee802.org/22/, http://www.ieee802.org/11/

[70] Mahmoud H. , Yucek T. , Arslan H. , "OFDM FOR COGNITIVE RADIO: MERITS AND CHALLENGES" Wireless Communications, IEEE (Volume:16 , Issue: 2), April 2009

[71] Ahmad, K. ; Inst. Ind. IT, OWL Univ. of Appl. Sci., Lemgo, Germany ; Meier, U. ; Pape, A. ; Kwasnicka, H. "A Generic Cognitive Radio for Evaluating Coexistence Optimized Industrial Automation Systems" 6th Annual IEEE Communications Society Conference, June 2009.

[72] Andrea Goldsmith and Ivana Mari´c "Principles of Cognitive Radio " text book, Cambridge University Press 2013.

[73] Tragos, E.Z. ; Inst. of Comput. Sci., Found. for Res. & Technol. Hellas, Heraklion, Greece ; Zeadally, S. ; Fragkiadakis, A.G. ; Siris, V.A. "Spectrum Assignment in Cognitive Radio Networks" Communications Surveys & Tutorials, IEEE July 2013

[74] Kanke Gao, Onur Ozdemir, Dimitris A. Pados, Stella N. Batalama, Tommaso Melodia, Andrew L. Drozd, "Joint Admission Conrol and Resource Allocation in Cognitive Code-Division Networks" IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC) 2012.

[75] Guodong Zhao; Jun Ma; Ye Li; Tao Wu; Young H. Kwon; Anthony Soong; Chenyang Yang, "Spatial Spectrum Holes for Cognitive Radio with Directional Transmission" Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008.

[76] Z. Wu and B. Natarajan, "Interference Tolerant Agile Cognitive Radio: Maximize Channel Capacity of Cognitive Radio," Consumer Communications and Networking Conference, pp. 1027-1031, January 2007.

[77] Bassoo, Vandana, and Narvada Khedun. "Improving the quality of service for users in cognitive radio network using priority queueing analysis." IET Communications 10.9 (2016): 1063-1070.

[78] Pin-Chuan Liu, Da You Chen, Chh Lin Hu, Wei Cheng Sun, "Analyzing the TCP Performance on Mobile Ad-Hoc Networks" in ICACT 2011, South Korea, February 2011.

[79] T. Issariyakul, L. Pillutla, and V. Krishnamurthy, “Tuning radio resource in an overlay cognitive radio network for TCP: Greed isn’t good,” IEEE Communications Magazine, 2009.

[80] A. Slingerland, P. Pawelczak, R. Prasad, A. Lo, and R. Hekmat, “Performance of transport control protocol over dynamic spectrum access links” 2nd IEEE International Symposium - Dynamic Spectrum Access Networks. DySPAN, 2007.

[81] Marco Di Felice, Kaushik Roy, Luciano Bononi, "Modeling and performance evaluation of transmission control protocol over cognitive radio ad hoc networks", the 12th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems, 2009

[82] P. Gupta and P. R. Kumar, “The capacity of wireless networks,” IEEE Trans. Information Theory, vol. 46, no. 2, pp. 388–404, Mar. 2000.

[83] Ernesto J. Rivera-Lara, Rogelio Herrerías-Hernández, Jesús A. Pérez-Díaz and Carlos F. García-Hernández, "Analysis of the relationship between QoS and SNR for an 802.11g WLAN" 2008 IEEE

[84] Akin, S.; Gursoy, M.C. “Performance Analysis of Cognitive Radio Systems under QoS Constraintsand Channel Uncertainty” Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE

[85] A. F. Molisch, L. J. Greenstein and M. Shafi, “Propagation Issues for Cognitive Radio,” Proceedings of the IEEE, Vol. 97, No. 5, 2009.

[86] Yueran Gao, HaiboWang, Ning Weng and Lucas Vespa “Enhancing Sensor Network Data Quality via Collaborated Circuit and Network operations” Journal of Sensor and Actuator Networks, 2013, www.mdpi.com/journal/jsan

[87] Masahiro Miyoshi, Masashi Sugano, Masayuki Murata, "Performance Evaluation of TCP throughput on wireless Cellular Networks" in Vehicular Technology Conference, 2001. VTC 2001 Spring. IEEE VTS 53rd Vol.3.

[88] Jitendra Padhye, Victor Firoiu, Donald F. Towsley, and James F. Kurose, "Modeling TCP Reno Performance: A Simple Model and Its Empirical Validation" Networking, IEEE/ACM Transactions on volume:8, April 2000.

[89] www.opnet.com , www.riverbed.com

[90] Ghayoor Abbas Jafri, Ateeq Ur Rehman, Muhammad Tariq Sadiq, "Spectrum Sensing and Management in Cooperative Cognitive Radio" Blekinge Institute of Technology, May 2011

[91] Mahamuni, S. and Mishra, V. (2014) Performance Evaluation of Spectrum Detection in Cognitive Radio Network. International Journal of Communications, Network and System Sciences, 7, 485-496. doi: 10.4236/ijcns.2014.711049.

[92] Ateeq Ur Rehman, Muhammad Tariq Sadiq, Noman Shabbir and Ghayoor Abbas Jafri, "Opportunistic Cognitive MAC (OC-MAC) Protocol for Dynamic Spectrum Access in WLAN Environment" IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 6, No 1, November 2013

[93] M. Kartheek ; Department of ECE, Indian Institute of Science, Bangalore, India ; Rakesh Misra ; Vinod Sharma, "Performance analysis of data and voice connections in a cognitive radio network", Communications (NCC), 2011 National Conference on Jan. 2011 Bangalore.

[94] N. Faruk, M. Gumel, A. Oloyode and A. Ayeni, "Performance Analysis of Hybrid MAC Protocol for Cognitive Radio Networks," International Journal of

Communications, Network and System Sciences, Vol. 6 No. 1, 2013, pp. 18-28. doi: 10.4236/ijcns.2013.61003.

[95] Wang, Li-Chun, Anderson Chen, and David SL Wei. "A cognitive MAC protocol for QoS provisioning in ad hoc networks." Physical Communication 3.2 (2010): 105-118.

[96] Fazeli, M.; Vaziri, H. "Assessment of Throughput Performance Under OPNET Modeler Simulation Tools in Mobile Ad Hoc Networks (MANETs)" Computational Intelligence, Communication Systems and Networks (CICSyN), 2011 Third International Conference on 2011.

[97] Meghanathan, Natarajan, ed. Cognitive Radio Technology Applications for Wireless and Mobile Ad Hoc Networks. IGI Global, 2013.

[98] Saeid Ghahremani, Rashid H. Khokhar, Rafidah Md Noor, Ahmad Naebi, J. Kheyrihassankandi "On QoS Routing in Mobile WiMAX Cognitive Radio Networks" Computer and Communication Engineering (ICCCE), 2012 International Conference on July 2012

[99] Mostafa Fazeli , Hasan Vaziri, "Assessment Of Throughput Performance Under OPNET Modeler Simulation Tools In Mobile Ad Hoc Networks" Computational Intelligence, Communication Systems and Networks (CICSyN), 2011 Third International Conference on July 2011.

[100] http://tetcos.com/netsim_gen.html, http://tetcos.com/netsim_academic.html,

[101] Uysal, M. ; Dept. of Comput. Sci., Maryland Univ., College Park, MD, USA ; Acharya, A. ; Bennett, R. ; Saltz, J. "A customizable simulator for workstation networks" Parallel Processing Symposium, 1997. Proceedings., 11th International.

[102] Ciraci, S. ; Pacific Northwest Nat. Lab., Richland, WA, USA ; Akyol, B. “NetSim-Steer: A Runtime Steering Framework for Network Simulators” , Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), 2012 IEEE 20th International Symposium.

[103] He Gao ; Sch. of Instrum. Sci. & Opto-Electron. Eng., Beijing Univ. of Aeronaut. & Astronaut., Beijing, China ; Jun Li ; Fuqiang Zhou ; Rong Zou more authors, "Stability Throughput Analysis of Cognitive Radio Interference MIMO Channels", 2010 6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM), September 2010.

[104] Sara Sangtarash ; Faculty of Health, Engineering and Science, Victoria University, Melbourne 3011, Australia ; Hatef Sadeghi ; Walid A. Hassan ; Horace L. King more authors, "Using cognitive radio interference mitigation technique to enhance coexistence and sharing between DVB-T and LTE system", Future Network & Mobile Summit (FutureNetw), 2012

[105] Ritu Khullar, Sippy Kapoor, Naval Dhawan, “Modulation Technique for Cognitive Radio Applications”, International Journal of Engineering Research and Applications (IJERA), 2012.

[106] Byoung Jo Choi, Lajos Hanzo," Optimum ModeSwitching Assisted Adaptive Modulation", IEEE Transaction on Communication, 2001.

[107] Md Fazlul Kader, Soo Young Shin, “Interference Free Cooperative Spectrum Sharing in Cognitive Radio Networks using Spatial Modulation”, International Conference on Advances in Electronics Engineering ICAEE, Dhaka, Bangladesh, Dec 2015.

[108] ITU Recommendations for Data networks, open system communications and security. https://www.itu.int/rec/T-REC-X/en

[109] M. Babaei; U Aygolu, "Spectral efficient and interference free spectrum sharing in cognitive radio using signal space diversity", Communications and Networking, 2015 IEEE International Black Sea Conference.

[110] J. Backens; C. Xin; M. Song; C. Chen, "Dynamic Spectrum Co-Access Between the Primary Users and the Secondary Users", IEEE Transactions on Vehicular Technology, 2015, Volume: 64, Issue: 2

[111] Q. Wang; D. -W. Yue; F. C. M. Lau, "Optimisation of throughput in cognitive radio networks: an analysis at the data link layer", IET Communications, 2012

[112] Stotas, S.; Nallanathan, A. "On the Throughput and Spectrum Sensing Enhancement of Opportunistic Spectrum Access Cognitive Radio Networks" Wireless Communications, IEEE Transactions on 2012.

[113] A Ghasemi and E S Sousa, "Spectrum sensing in cognitive radio networks: requirements, challenges and design trade-offs," IEEE Communications Mag., vol. 46, Apr. 2008.

[114] W. Han ; Broadband Wireless Commun. Lab.& State Key Lab. (ISN), Xidian Univ., Xi"an, China ; J. Li ; Z. Tian ; Y. Zhang, "Dynamic Sensing Strategies for Efficient Spectrum Utilization in Cognitive Radio Networks", IEEE Transactions on Wireless Communications, September 2011

[115] Guodong Zhao; Jun Ma; Geoffrey Li; Tao Wu; Young Kwon; Anthony Soong; Chenyang Yang "Spatial spectrum holes for cognitive radio with relay-assisted directional transmission" IEEE Transactions on Wireless Communications, 2009.

[116] W.-Y. Lee and I. Akyildiz, "Optimal spectrum sensing framework for cognitive radio networks," IEEE Transactions on Wireless Communications, 2008.

[117] Cong Shi; Ying Wang; Tan Wang; Ping Zhang, “Joint optimization of detection threshold and throughput in multiband cognitive radio systems”, Consumer Communications and Networking Conference (CCNC), 2012 IEEE

[118] A. Lebda; E. Gemeay; S. Khamis; M. Khder, "Cognitive radio with fast sensing technique using autocorrelation approach", 2016 18th International Conference on Advanced Communication Technology (ICACT)

[119] Long Yuan; Feng Ren; Lu Xing; Tao Peng; Wenbo Wang "Wideband spectrum sensing algorithm based on frequency correlation", Communication Technology (ICCT), 2013 15th IEEE International Conference.

[120] G. I. Tsiropoulos; O. A. Dobre; M. H. Ahmed; K. E. Baddour, “Radio Resource Allocation Techniques for Efficient Spectrum Access in Cognitive Radio Networks”, IEEE Communications Surveys & Tutorials, Year: 2016, Volume: 18, Issue: 1

[121] K. Chowdhury, M. Di Felice, and I. Akyildiz, “TP-CRAHN: a transport protocol for cognitive radio ad-hoc networks,” in IEEE INFOCOM, 2009

[122] D. Sarkar and H. Narayan, “Transport layer protocols for cognitive networks”, IEEE Conference on Computer Communications Workshops,INFOCOM, 2010

[123] C. Luo, F. Yu, H. Ji, and V. Leung, “Cross-layer design for TCP performance improvement in cognitive radio networks,” IEEE Transactions on Vehicular Technology, 2010

[124] Y. Song, Y. Fang, and Y. Zhang, “Stochastic channel selection in cognitive radio networks,” in Proc. IEEE Global Telecommunications Conf. (GLOBECOM 2007), Nov. 2007.

[125] Daniel Warne,”Software Radio Architectures-Part 2”, 2004: 1-165

[126] Y. Linn, “New structures for modulation classification and SNR estimation with applications to Cognitive Radio and Software Defined Radio” Electrical and Computer Engineering (CCECE), 2014

[127] H. Shatila; M. Khedr; J. H. Reed, “Adaptive modulation and coding for WiMAX systems with vague channel state information using cognitive radio”, Performance Evaluation of Computer and Telecommunication Systems (SPECTS), 2010

[128] F. T. Foukalas; G. T. Karetsos, “A Study on the Performance of Adaptive Modulation and Cross-Layer Design in Cognitive Radio for Fading Channels” Informatics, 2009. PCI '09. 13th Panhellenic Conference.

[129] W. Peng; Ye Zhi Hui, “OFDM blending modulation systems in practical cognitive radio application” Image and Signal Processing (CISP), 2011 4th International Congress.

[130] G. Xu; Y. Lu, “Channel and Modulation Selection Based on Support Vector Machines for Cognitive Radio Wireless Communications”, Networking and Mobile Computing, 2006. WiCOM 2006.International Conference.

[131] T. Luo; F. Lin; T. Jiang; M. Guizani; W. Chen, “Multicarrier modulation and cooperative communication in multihopcognitive radio networks”, IEEE Wireless Communications 2011.

[132] I. Budiarjo; H. Nikookar; L. P. igthart,”Cognitive radio modulation techniques”, IEEE Signal Processing Magazine Volume: 25, Issue 6, 2008.

[133] M. Abdelbar; B. Tranter; T. Bose, "Cooperative Modulation Classification of multiple signals in Cognitive Radio Networks", Communications (ICC), 2014 IEEE International Conference.

[134] Zhang, Juan; Jiang, Hong; Huang, Zhenhua; Chen, Chunmei; Jiang, Hesong, "Study of Multi-Armed Bandits for Energy Conservation in Cognitive Radio Sensor Networks." Sensors 15, no. 4: 9360-9387, 2015.

[135] H. Li ; D. Grace ; P. D. Mitchell, "Multiple access with multi-dimensional learning for cognitive radio in open spectrum", Communications and Information Technologies (ISCIT), 2011 the 11th International Symposium.

[136] Zhihui Shu; Yi Qian; R. Q. Hu, "Delay based channel allocations in multi-hop cognitive radio networks", Wireless Communications and Mobile Computing Conference (IWCMC), 2015.

[137] ZHU HAN, K. J. RAY LIU, "Resource Allocation for Wireless Networks – Basic Techniques and Allocations" Cambridge University press 2008.

[138] Amna Jamal; Chen-Khong Tham; Wai-Choong Wong, "Dynamic Packet Size Optimization and Channel Selection for Cognitive Radio Sensor Networks", IEEE Transactions on Cognitive Communications and Networking (Volume: 1, Issue: 4, Dec. 2015)

[139] Akin, Sami, Marwan Hammouda, and Jürgen Peissig. "QoS Analysis of Cognitive Radios Employing HARQ." arXiv preprint arXiv:1702.01218 (2017).

[140] Shah, Munam Ali, Sijing Zhang, and Carsten Maple. "Cognitive radio networks for Internet of Things: Applications, challenges and future." Automation and Computing (ICAC), 2013 19th International Conference on. IEEE, 2013.

[141] Athar Ali Khan; Mubashir Husain Rehmani; Abderrezak Rachedi, "When Cognitive Radio meets the Internet of Things?", Wireless Communications and Mobile Computing Conference (IWCMC), IEEE September 2016 Paphos, Cyprus.

[142] D. B. Rawat, "ROAR: An architecture for Real-Time Opportunistic Spectrum Access in Cloud-assisted Cognitive Radio Networks", 2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC), Jan 2016.

[143] Majumdar, Chitradeep, et al. "Packet Size Optimization for Cognitive Radio Sensor Networks Aided Internet of Things." IEEE Access (2016).

[144] Majumdar, Chitradeep, et al. "Packet Size Optimization for Multiple Input Multiple Output Cognitive Radio Sensor Networks aided Internet of Things." IEEE Access (2017).

[145] N. Tadayon; S. Aïssa, "Modeling and Analysis Framework for Multi-Interface Multi-Channel Cognitive Radio Networks", IEEE Transactions on Wireless Communications, I EEE Journals & Magazines, 2015.

[146] M. López-Benítez; F. Casadevall, "Time-Dimension Models of Spectrum Usage for the Analysis, Design, and Simulation of Cognitive Radio Networks", IEEE Transactions on Vehicular Technology, 2013.

[147] C. Ghosh, S. Roy and D. Cavalcanti, "Coexistence Chalges for Heterogeneous Cognitive Wireless Networks in TV White Spaces," IEEE Wireless Communications, Vol. 18, No. 4, 2011, pp. 22-31.

[148] G. Auer, H. Haas and P. Omiyi, "Interference aware Medium Access for Dynamic Spectrum Sharing," 2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, Dublin, 17-20 April 2007.

[149] Y. Wang, P. Ren “Protocol with QoS Provision for Cognitive Ad Hoc Networks,” IEICE Transactions on Communications, Vol. E93- B, No. 6, pp. 1426-1429.

[150] ITU-R SM 1536. Frequency Channel Occupancy Measurements, 2001.

[151] I. Sobron; W. A. Martins; M. L. R. de Campos; M. Velez, "Incumbent and LSA Licensee Classification Through Distributed Cognitive Networks", IEEE Transactions on Communications, 2016, Volume: 64, Issue: 1

[152] S. Y. Lien; J. Lee; Y. C. Liang, "Random Access or Scheduling: Optimum LTE Licensed-Assisted Access to Unlicensed Spectrum”, IEEE Communications Letters, Year: 2016, Volume: 20, Issue: 3

[153] Jung-Hyuck Jo and Nikil Jayant “Measurement and Analysis of the Chanel Characteristics of an In-Building Wireless network” ARFTG Conference Digest-Spring, 54th (Volume:36), Atlanta, GA, USA Dec. 2000.

[154] Raghed El Bardan, Engin Masazade, Onur Ozdemir and Pramod K. Varshney, “Performance of Permutation Trellis Codes in Cognitive Radio Networks” Sarnoff Symposium (SARNOFF), 35th IEEE 2012.

[155] A. Goldsmith, S.A. Jafar, I. Maric, and S. Srinivasa, "Breaking spectrum gridlock with cognitive radios: An information theoretic perspective," Proc. IEEE, vol. 97, no. 5, pp. 894-914, May 2009.

[156] Pal, R.; Idris, D.; Pasari, K.; Prasad, N. "Characterizing reliability in cognitive radio networks" Applied Sciences on Biomedical and Communication Technologies, 2008.

[157] Shixian Wang; Heng Zhao; Botao Zhang; Hengzhu Liu; Lunguo Xie "An autonomic communication based conceptual and architecture model for cognitive radio

nodes" Wireless, Mobile and Multimedia Networks (ICWMNN 2010), IET 3rd International Conference on 2010.

[158] Shixian Wang; Hengzhu Liu; Lunguo Xie; Wenmin Hu, "Cognitive radio simulation environment realization based on autonomic communication" Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on 2011

[159] Federal Communications Commission, “Facilitating opportunities for flexible, efficient, and reliable spectrum use employing cognitive radio technologies”, notice of proposed rulemaking and order, FCC 03- 322, December 2003.

[160] Marojevic, V.; Vucevic, N.; Reves, X.; Gelonch, A. "Integrated Resource Management in Cognitive Radio" Mobile and Wireless Communications Summit, 2007.

[161] Rezaei, F. ; Tadaion, A. “Sum-Rate Improvement in Cognitive Radio Through Interference Alignment” Vehicular Technology, IEEE Transactions january 2015.

[162] Holcomb, Sterling, and Danda B. Rawat. "Recent security issues on cognitive radio networks: A survey." SoutheastCon, 2016. IEEE, 2016.

[163] Ahmed, Ismail K., and Abraham O. Fapojuwo. "Security threat assessment of simultaneous multiple Denial-of-Service attacks in IEEE 802.22 Cognitive Radio networks." World of Wireless, Mobile and Multimedia Networks, 2016 IEEE 17th International Symposium on A. IEEE, 2016.

[164] Sharma, Girraj, and Ritu Sharma. "A review on recent advances in spectrum sensing, energy efficiency and security threats in cognitive radio network." Microwave, Optical and Communication Engineering (ICMOCE), 2015 International Conference on. IEEE, 2015.

[165] Force, Spectrum Policy Task. "Spectrum policy task force report." Federal Communications Commission ET Docket 2 (2002): 135.

[166] Qiwei Zhang, Andre B. Kokkeler and Gerard J. Smit," A Reconfigurable Radio Architecture for Cognitive Radio in Emergency Networks" European Conference on Wireless Technology, Manchester, 2006.

[167] Chávez-Santiago, Raúl, et al. "Cognitive radio for medical body area networks using ultra wideband." IEEE Wireless Communications 19.4 (2012).

[168] Di Felice, Marco, Kaushik Roy Chowdhury, and Luciano Bononi. "Cognitive radio vehicular ad hoc networks: design, implementation, and future challenges." Mobile Ad Hoc Networking: Cutting Edge Directions, Second Edition (2013): 619-644.

[169] Tianlin Wang and Hazem H. Refai "Empirical Network Performance Analysis on IEEE 802.11g with Different Protocols and Signal to Noise Ratio Values" Wireless and Optical Communications Networks, WOCN 2005. Second IFIP International Conference on 6 March 2005

[170] Pin-Chuan Liu, Da-You Chen, Chih-Lin Hu, Wei-Cheng Sun, Jen-Hwa Lee, Chung-Kuang Chou, Wei- Kuan Shih "Analyzing the TCP Performance on Mobile Ad-Hoc Networks" Advanced Communication Technology (ICACT), 13th International Conference, Feb. 2011.

[171] Michael R. Souryal, Luke Klein-Berndt, Leonard E. Miller, and Nader Moayeri "Link Assessment in an Indoor 802.11 Network", 2006 IEEE Wireless Communications and Networking Conference (WCNC).

APPENDIX A

THE RELATION BETWEEN PACKET LOSS, BER AND SNR

When running network simulation tool we get an important simulation metrics which is Packet drop or loss ratio. This is the ratio between the lost packets to the overall transmitted packets, which usually resides in range less than 10^{-5}. Using the simulated Packet loss ratio we can conclude the Bit Error Rate from the following relation:

$$PLR = 1 - (1 - BER)^{PL}$$

Where:

PLR is the Packet loss ratio

BER is the Bit Error rate

PL is packet length in bits.

Each modulation scheme has its own BER vs SNR curve which shows the efficiency of the used modulation scheme in terms of BER and SNR values. The “Digital Communications, Fundamentals and Applications” text book [68] presented sample curves for the BER/SNR relation as illustrated in figures A-1 and A-2.

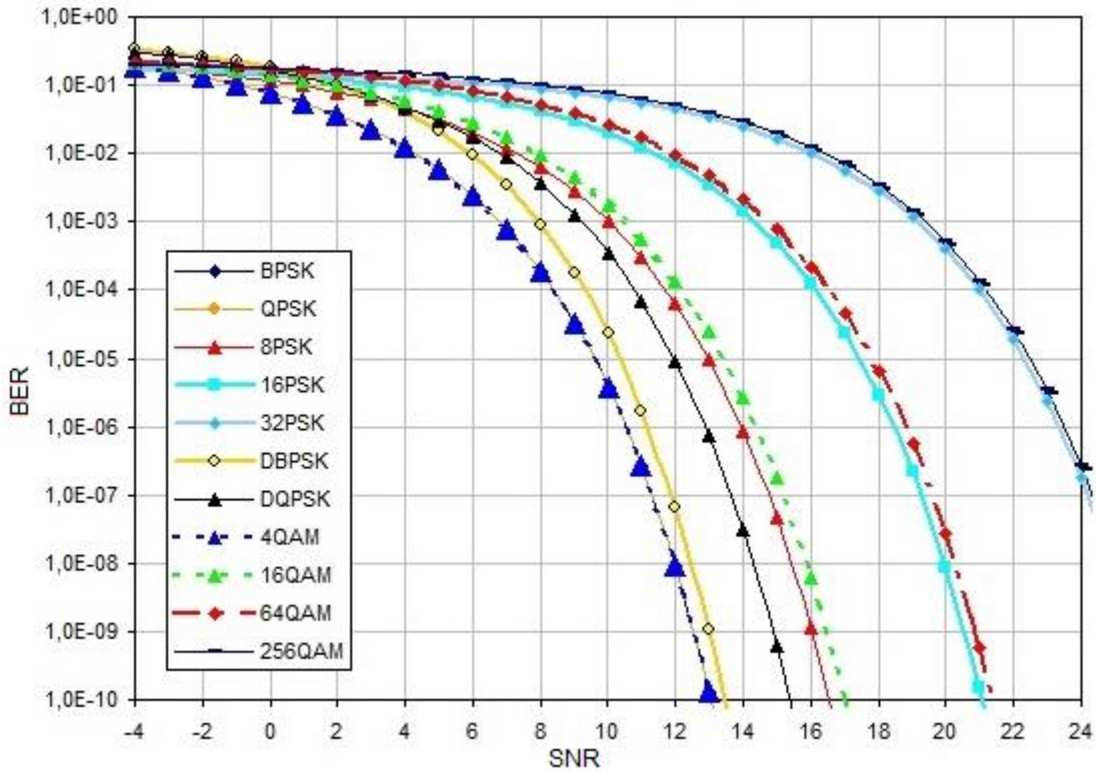


Figure A-1: Bit Error Rate vs SNR – various modulation types, Digital Communications text book [68]

As it is difficult to find direct mathematical relation represents the relation between the packet loss and the SNR, however several experimental measurements introduced relation between Packet loss and Signal to Noise Ratio (SNR), and from these measurements we can conclude the following:

- Most measurements indicates that the packets performance is excellent when the SNR value is above (30 – 25) dB.
- The packets performance is still accepted (less quality, but accepted performance) when the SNR is around 20 dB.
- The packets performance is unaccepted (very low quality) when SNR is much less than 20 dB.

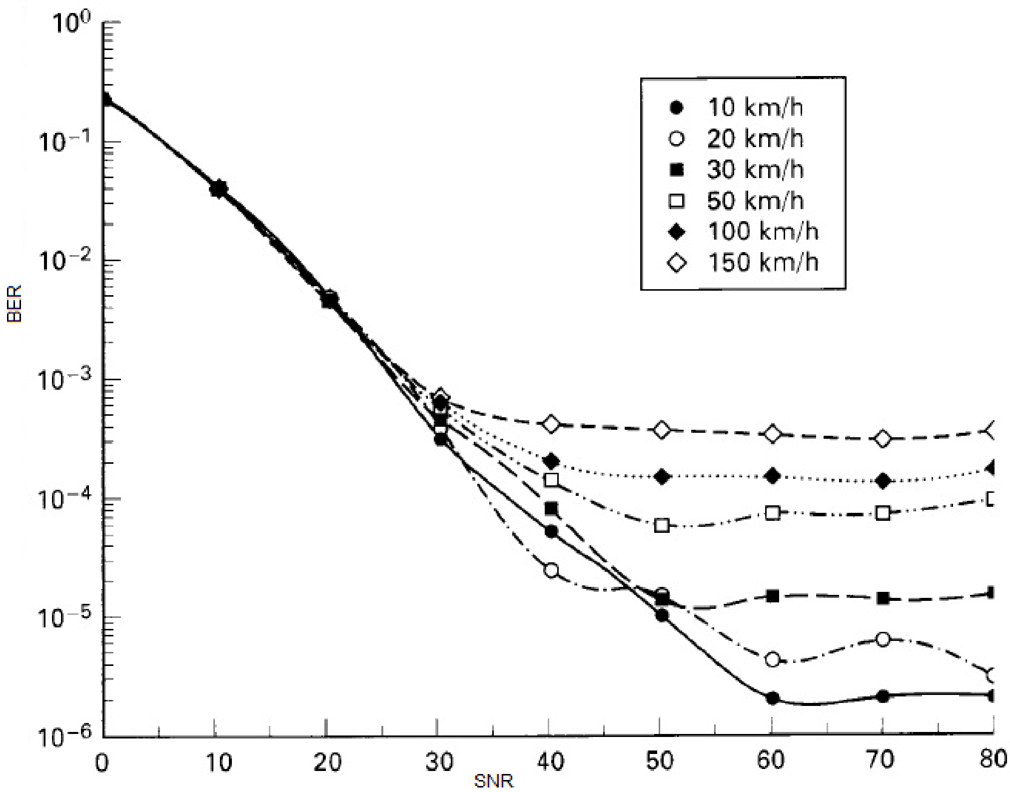


Figure A-2: BER vs SNR using QPSK modulation - various object speeds, Digital Communications text book [68]

Here are some experimental measurements were introduced in different researches which show the relation between Packet loss and SNR:

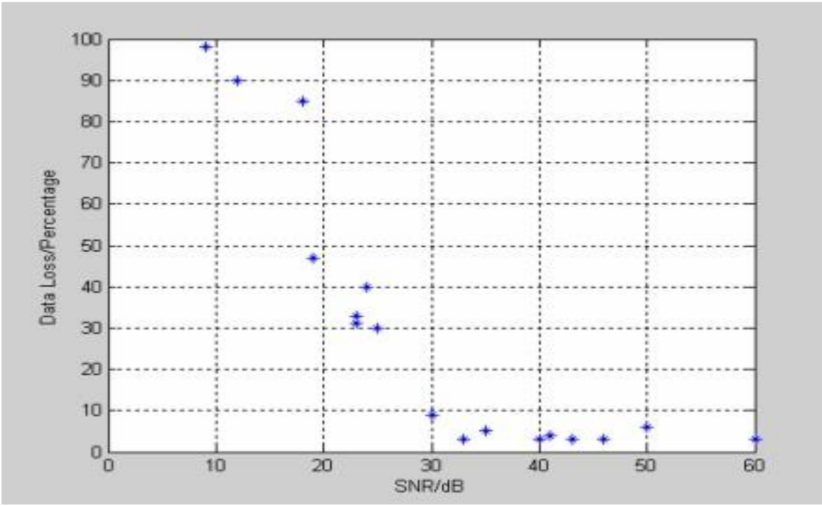


Figure A-3: relation between Data loss and SNR, University of Oklahoma [169]

The graph illustrated in Figure A-3 is related to analysis done by the University of Oklahoma- Tulsa, School of Electrical & Computer Engineering [169]. It shows the Data loss against SNR.

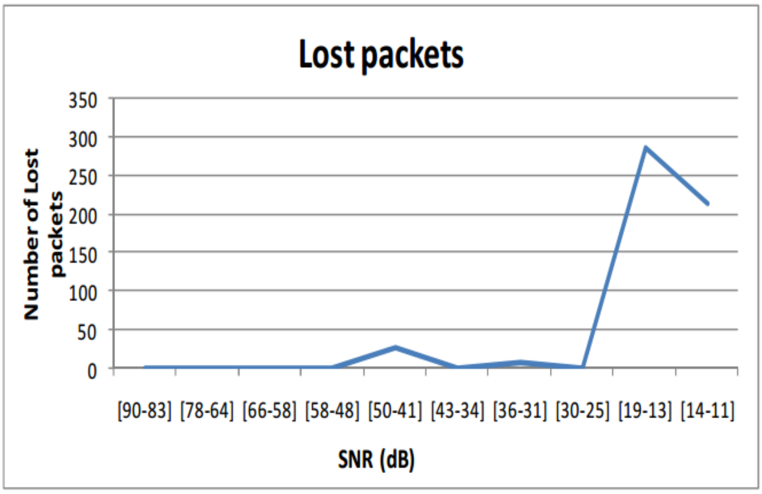


Figure A-4: relation between packet loss and SNR, ITRI Taiwan [170]

Analysing the TCP Performance on Mobile Ad-Hoc Networks [170] done by Industrial Technology Research Institute (ITRI), Taiwan. R.O.C. Department of Computer Science, National Tsing Hua University. The graph in Figure A-4 shows number of lost packets against different SNR ranges in mobile networks environment.

The following graphs shown in Figure A-5 are related to Link Assessment in an Indoor 802.11 Network done by the Wireless Communication Technologies Group, National Institute of Standards and Technology NIST, Maryland, USA [171]. The test environment included several levels of signal interference to evaluate the wireless network behaviour in each case.

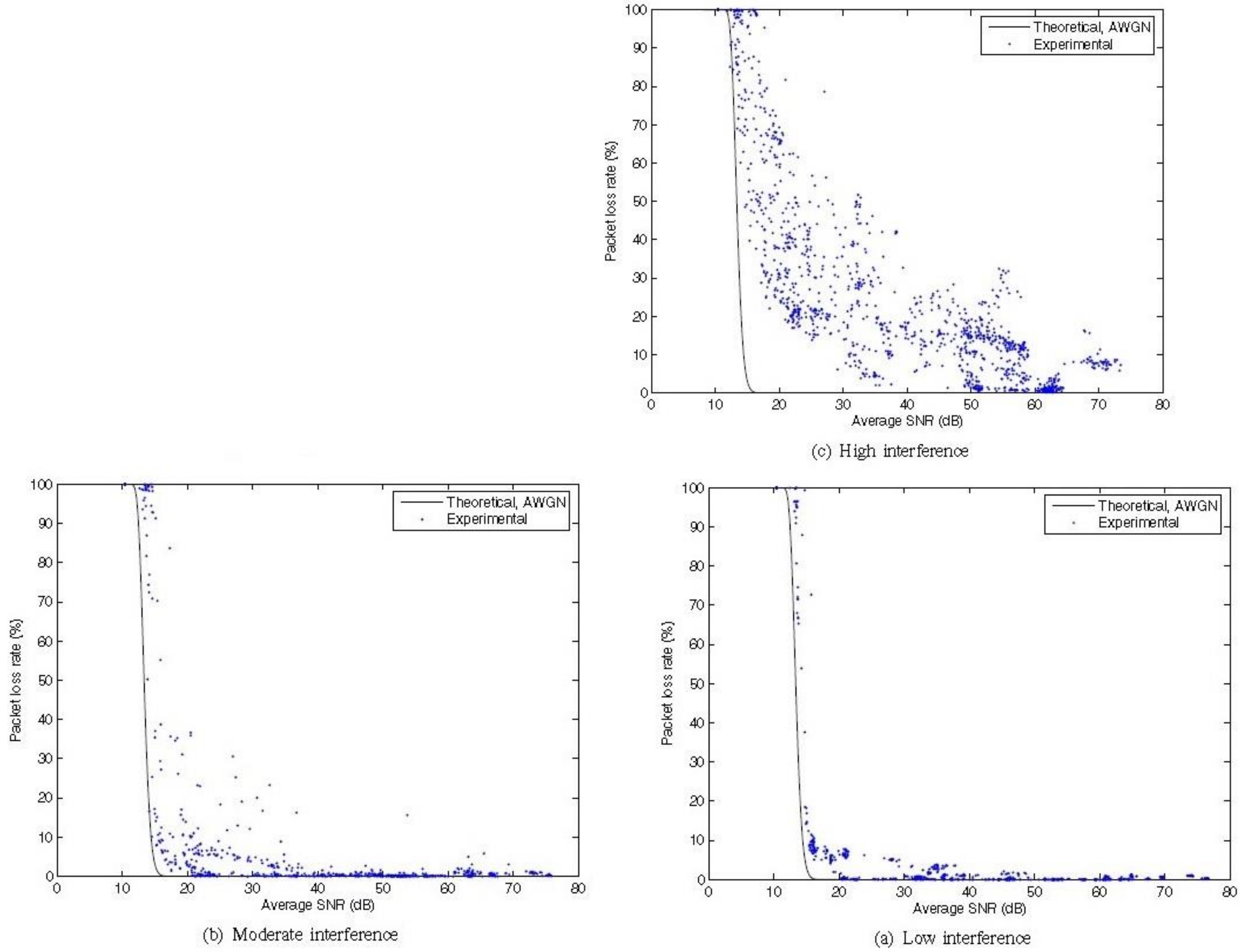


Figure A-5: Relation between packet loss rate and SNR with different levels of interference, NIST [171]

APPENDIX B

OPNET CODE

```
/*OPNET MDCR/CMDCR code            */
/*H.Fadel DMU 2016                 */

#include "wrls_phy_support.h"
#include "oms_wp_api.h"
#include "oms_data_def.h"
#include "string.h"
#include "oms_wireless_support.h"
#include "oms_string_support.h"
#include "ip_wwqos_internal.h"
#include "ip_wwqos_constants.h"
#include "ip_wwqos_forwarding_class.hpp"
#include "ip_wwqos_network.hpp"
#include "ip_wwqos_node.hpp"
#include "ip_wwqos_ifc.hpp"
#include "ip_wwqos_queue.hpp"
#include "ip_wwqos_policy.hpp"
#include "ip_wwqos_policy_statement.hpp"
#include "ip_wwqos_ct_bw.hpp"
#include "ip_wwqos_priority.hpp"
#include "ip_wwqos_set_info.hpp"
#include "ip_wwqos_traffic_class.hpp"
#include "ip_wwqos_queue_profile.hpp"
#include "prg_bin_hash.h"
#include "ets_api_obj.h"
#include "ets_api_model.h"
#include "ets_api_topo.h"
#include "ets_api_gui.h"
#include "prg_list_funcs.h"
#include "prg_string_funcs.h"
#include "prg_mem_funcs.h"
#include "oma_ot.h"
#include <opnet.h>
#include <ctype.h>
#include <oms_pr.h>
#include <math.h>
#include <string.h>
#include "main.h"
#include "802_22.h"
#include "SpectrumManager.h"
#include "Spectrumtester.h"
```

```
Cmohandle wrls_phy_catmem_handle   = OPC_NIL;

Boolean   wlrs_phy_global_init_done     = OPC_FALSE;
int       sw_cmdcr;
int       qosflg;


typedef struct WrlsT_Initial_Attach_Event_State
     {
     void*
          mac_profile_ptr;
     WrlsT_MAC_To_PHY_Convert_Proc_Ptr
     mac_to_phy_conv_proc_ptr;
     } WrlsT_Initial_Attach_Event_State;

typedef struct MCS_Scaling_Attrs
     {
     int  mcs_index;
     char      scaling_factors_str[512];
     } MCS_Scaling_Attrs;

Pmohandle      wrls_burst_alloc_info_field_pmh;
Boolean
     wrls_burst_alloc_info_field_pool_memory_ready_flag =
OPC_FALSE;

Void wrls_phy_mcarrier_pk_send (Packet* pkptr,
WrlsT_Transmitter_Info* tx_info_ptr, WrlsT_Phy_Chnl_Info*
phy_info_ptr, WrlsT_Phy_Mcarrier_Tx_Mgmt* mcarrier_tx_ptr,
WrlsT_Phy_Mcarrier_Burst_Info* burst_alloc_info_ptr, Objid
tx_module_objid);
int  wrls_phy_mcarrier_tx_conduit_index_get
(WrlsT_Phy_Mcarrier_Tx_Mgmt* mcarrier_mgmt_ptr, Objid
rxch_objid, double start_time, double end_time);

WrlsT_Phy_Mcarrier_Tx_Mgmt*
     wrls_phy_mcarrier_tx_mgmt_init (Objid mac_objid, int
next_cw_txch_start);
WrlsT_Rx_State_Info*              wrls_phy_rx_state_init
(Objid mac_objid, WrlsT_Phy_Chnl_Info*
wrls_phy_chnl_info_ptr);
void wrls_phy_support_burst_info_print
(WrlsT_Phy_Mcarrier_Burst_Info* burst_ptr);
double wrls_phy_power_pedestrian_pathloss_compute (double
distance, double freq);
double wrls_phy_power_vehicular_pathloss_compute (double
distance, double freq, double base_station_height_meters);
```

```
double wrls_phy_power_erceg_pathloss_compute (double
distance, double lambda, double freq, double
mobile_height_meters, double base_station_height_meters,
int terrain_type);

WrlsT_Power_Control_Info*
     wrls_phy_power_control_info_init (Objid mac_objid);

void wrls_phy_freq_bandwidth_all_channels_set (Objid
ch_comp_id, double base_frequency, double bandwidth,
OpT_Obj_Type obj_type);
WrlsT_Phy_Chnl_Info*
     wrls_phy_chnl_info_init_first_phase (Objid mac_objid,
WrlsT_Phy_Profile* wrls_phy_prof_ptr, WrlsT_Cell_Id bs_id,
WrlsT_Mac_Role mac_role);

void wrls_phy_chnl_info_init_second_phase
(WrlsT_Transmitter_Info* wrls_tx_info_ptr ,Objid mac_objid,
WrlsT_Phy_Profile* wrls_phy_prof_ptr, WrlsT_Cell_Id bs_id,
WrlsT_Mac_Role mac_role);

void wrls_phy_send_fragment(Packet *seg_pkptr,
WrlsT_Phy_Mcarrier_Burst_Info* burst_alloc_info_ptr,
WrlsT_Transmitter_Info *tx_info_ptr);
int  wrls_phy_fft_size_get_from_index(int fft_index);
void wrls_phy_mcs_info_init (WrlsT_Phy_Chnl_Info*
phy_info_ptr, const char *modulation_curve_names [], int
modulation_index_count );

void wrls_phy_mpath_channels_update (void* state_ptr, int
PRG_ARG_UNUSED(code));
int  wrls_phy_mpath_channel_instance_next_state_get (double
* probability_array , int markov_states_count);
void wrls_phy_mpath_channel_instance_update
(MultipathT_Channel_Model* mpath_channel_ptr,
MultipathT_Instance_Element*  instance_ptr, int
number_of_updates);
void wrls_phy_mpath_channel_evolve
(MultipathT_Channel_Model* mpath_channel_ptr,
MultipathT_Instance_Element*  instance_ptr, double
mpath_channel_state_change_time, int
max_channel_state_changes);
double wrls_phy_multipath_effective_snr_compute (Objid
tx_objid, Objid rx_objid, double avg_snr_db,
WrlsT_Phy_Chnl_Info* phy_chnl_info_ptr,
WrlsT_Phy_Mcarrier_Burst_Info*  burst_info_ptr,
WrlsT_Rx_State_Info* rx_state_ptr);
MultipathT_MCS_Scaling_Factors*
```

```
     wrls_phy_mcs_scaling_attributes_parse (Objid
mcs_scaling_attrs_objid);

Boolean wrls_phy_support_measurement_read_db
(WrlsT_Measurement_Entity* m_entity_ptr, double*
value_ptr);

EXTERN_C_BEGIN
void*     wrls_burst_info_copy_proc_ (void*
busrt_alloc_info_vptr, size_t size);
void*     wrls_phy_info_copy_proc_ (void* phy_info_vptr,
size_t size);
void wrls_phy_info_destroy_proc_ (void* phy_info_vptr);
int       wrls_phy_snr_function_elem_compare (const void
*snr_fn_elem1_vptr, const void *snr_fn_elem2_vptr);

EXTERN_C_END

void wrls_phy_power_consumption_from_burst_update
(WrlsT_Phy_Chnl_Info* phy_info_ptr,
WrlsT_Phy_Mcarrier_Burst_Info* burst_alloc_info_ptr);

void
wrls_phy_attr_module_phy_info_register (const char*
phy_module_name, WrlsT_MAC_To_PHY_Init_Info*
phy_init_info_ptr)
     {
     Objid
     phy_mod_objid;
     WrlsT_MAC_To_PHY_Measurement_Info*
     measure_info_ptr;
     WrlsT_MAC_To_PHY_Init_Info*
     copy_init_info_ptr;


     FIN (wrls_phy_attr_module_phy_info_register
(phy_module_name, phy_init_info_ptr));

     phy_mod_objid = op_id_from_name (op_topo_parent
(op_id_self ()), OPC_OBJTYPE_PROC, phy_module_name);



     copy_init_info_ptr = (WrlsT_MAC_To_PHY_Init_Info*)
op_prg_mem_alloc (sizeof (WrlsT_MAC_To_PHY_Init_Info));

     copy_init_info_ptr->client_info_ptr
               = phy_init_info_ptr->client_info_ptr    ;
```

```
     copy_init_info_ptr->cell_id
              = phy_init_info_ptr->cell_id;
     copy_init_info_ptr->mac_to_phy_profile_ptr
          = phy_init_info_ptr->mac_to_phy_profile_ptr;
     copy_init_info_ptr->mcarrier_subband_measures_proc_ptr
     = phy_init_info_ptr-
>mcarrier_subband_measures_proc_ptr;
     copy_init_info_ptr->phy_type
          = phy_init_info_ptr->phy_type;
     copy_init_info_ptr->phy_model_type
          = phy_init_info_ptr->phy_model_type;

set sw_cmdcr;
/* check channel state tables to find available frequency
channels and time slots */
spct_sns(chk_freq_div, chk_time_div);

if (sw_cmdcr == 1)
     {
     clsf_client(app_id, client_flg)

     }



     if (phy_init_info_ptr->measure_info_ptr != OPC_NIL)
          {
          measure_info_ptr =
(WrlsT_MAC_To_PHY_Measurement_Info*) op_prg_mem_alloc
(sizeof (WrlsT_MAC_To_PHY_Measurement_Info));

          measure_info_ptr->measure_keys_proc_ptr
          = phy_init_info_ptr->measure_info_ptr-
>measure_keys_proc_ptr;
          measure_info_ptr->measure_proc_ptr
          = phy_init_info_ptr->measure_info_ptr-
>measure_proc_ptr;

          measure_info_ptr->client_info_ptr
          = phy_init_info_ptr->measure_info_ptr-
>client_info_ptr;
          measure_info_ptr->measure_key_base
          = phy_init_info_ptr->measure_info_ptr-
>measure_key_base;
          measure_info_ptr->measurement_bucket
          = phy_init_info_ptr->measure_info_ptr-
>measurement_bucket;
          measure_info_ptr->measurement_window_size
```

```
          = phy_init_info_ptr->measure_info_ptr-
>measurement_window_size;
          measure_info_ptr->measurement_type
          = phy_init_info_ptr->measure_info_ptr-
>measurement_type;


          copy_init_info_ptr->measure_info_ptr
          = measure_info_ptr;
          }
     else
          copy_init_info_ptr->measure_info_ptr
          = OPC_NIL;


     op_ev_state_install (copy_init_info_ptr, OPC_NIL);
     op_intrpt_schedule_remote (op_sim_time (),
WRLSC_PHY_MAC_ACTIVATION_CODE, phy_mod_objid);
     op_ev_state_install (OPC_NIL, OPC_NIL);

     FOUT;
     }

void
wrls_phy_attr_module_measure_info_update (const char*
phy_module_name, WrlsT_MAC_To_PHY_Measurement_Info*
meas_info_ptr)
     {
     Objid
     phy_mod_objid;
     WrlsT_MAC_To_PHY_Measurement_Info*
     measure_info_ptr;

     /** register a measurement type.
                                  **/
     FIN (wrls_phy_attr_module_measure_info_update
(phy_module_name, meas_info_ptr));

     phy_mod_objid = op_id_from_name (op_topo_parent
(op_id_self ()), OPC_OBJTYPE_PROC, phy_module_name);

     measure_info_ptr =
(WrlsT_MAC_To_PHY_Measurement_Info*) op_prg_mem_alloc
(sizeof (WrlsT_MAC_To_PHY_Measurement_Info));

     measure_info_ptr->measure_keys_proc_ptr
     = meas_info_ptr->measure_keys_proc_ptr;
     measure_info_ptr->measure_proc_ptr
```

```
        = meas_info_ptr->measure_proc_ptr;

        measure_info_ptr->client_info_ptr
        = meas_info_ptr->client_info_ptr;
        measure_info_ptr->measure_key_base
        = meas_info_ptr->measure_key_base;
        measure_info_ptr->measurement_bucket
        = meas_info_ptr->measurement_bucket;
        measure_info_ptr->measurement_window_size
        = meas_info_ptr->measurement_window_size;
        measure_info_ptr->measurement_type
        = meas_info_ptr->measurement_type;

        /* process model              */

        op_ev_state_install (measure_info_ptr, OPC_NIL);
        op_intrpt_schedule_remote (op_sim_time (),
WRLSC_PHY_MEASURE_REGISTER_CODE, phy_mod_objid);
        op_ev_state_install (OPC_NIL, OPC_NIL);

        FOUT;
        }


void
wrls_phy_attr_module_measure_info_remove (const char*
phy_module_name, WrlsT_MAC_To_PHY_Measurement_Info*
meas_info_ptr)
        {
        Objid
        phy_mod_objid;
        WrlsT_MAC_To_PHY_Measurement_Info*
        measure_info_ptr;


        FIN (wrls_phy_attr_module_measure_info_remove
(phy_module_name, meas_info_ptr));

        phy_mod_objid = op_id_from_name (op_topo_parent
(op_id_self ()), OPC_OBJTYPE_PROC, phy_module_name);


        scp_calc();
        acflg == 1;
        actflg = chk_actv();
        if (actflg == 0) {upd_s_table(); }

        measure_info_ptr =
```

```
(WrlsT_MAC_To_PHY_Measurement_Info*) op_prg_mem_alloc
(sizeof (WrlsT_MAC_To_PHY_Measurement_Info));

     measure_info_ptr->measure_keys_proc_ptr
     = meas_info_ptr->measure_keys_proc_ptr;
     measure_info_ptr->client_info_ptr
     = meas_info_ptr->client_info_ptr;
     measure_info_ptr->measure_key_base
     = meas_info_ptr->measure_key_base;
     measure_info_ptr->measurement_type
     = meas_info_ptr->measurement_type;


     op_ev_state_install (measure_info_ptr, OPC_NIL);
     op_intrpt_schedule_remote (op_sim_time (),
WRLSC_PHY_MEASURE_DEREGISTER_CODE, phy_mod_objid);
     op_ev_state_install (OPC_NIL, OPC_NIL);

     FOUT;
     }

void
wrls_phy_notif_state_set (const char* phy_module_name,
WrlsT_Measurement_Module_Type mod_type,
WrlsT_Measurement_Type measure_type,
WrlsT_Measurement_Threshold_Direction threshold_dir,

                         double threshold, Objid
remote_mod_objid, int key, void* client_notify_vptr)
     {
     Objid
     phy_mod_objid;
     WrlsC_PHY_Notif_Update_Info*
     notif_update_info_ptr;

     /** PHY module.
                                           **/
     FIN (wrls_phy_notif_state_set (phy_module_name,
mod_type, measure_type, threshold_dir, threshold,
remote_mod_objid, key, client_notify_vptr));

     phy_mod_objid = op_id_from_name (op_topo_parent
(op_id_self ()), OPC_OBJTYPE_PROC, phy_module_name);

     /* Create an event state for the PHY process.
                                           */
     notif_update_info_ptr = (WrlsC_PHY_Notif_Update_Info*)
op_prg_mem_alloc (sizeof (WrlsC_PHY_Notif_Update_Info));
```

```
     notif_update_info_ptr->sub_client_id         =
(int)mod_type;
     notif_update_info_ptr->measure_type               =
measure_type;
     notif_update_info_ptr->threshold_dir         =
threshold_dir;
     notif_update_info_ptr->threshold             =
threshold;
     notif_update_info_ptr->remote_mod_objid      =
remote_mod_objid;
     notif_update_info_ptr->key                        =
key;
     notif_update_info_ptr->client_notify_vptr    =
client_notify_vptr;

     op_ev_state_install (notif_update_info_ptr, OPC_NIL);
     op_intrpt_schedule_remote (op_sim_time (),
WRLSC_PHY_NOTIF_UPDATE_CODE, phy_mod_objid);
     op_ev_state_install (OPC_NIL, OPC_NIL);

     FOUT;
     }

void
wrls_phy_attr_module_phy_scan_info_register (const char*
phy_module_name, WrlsT_Scan_Info* scan_info_ptr)
     {
     Objid phy_mod_objid;


     FIN (wrls_phy_attr_module_phy_scan_info_register ());

     phy_mod_objid = op_id_from_name (op_topo_parent
(op_id_self ()), OPC_OBJTYPE_PROC, phy_module_name);

     op_ev_state_install (scan_info_ptr, OPC_NIL);
     op_intrpt_schedule_remote (op_sim_time (),
WRLSC_PHY_SCAN_REGISTRATION_CODE, phy_mod_objid);
     op_ev_state_install (OPC_NIL, OPC_NIL);

     FOUT;
     }

void
wrls_phy_attr_module_phy_scan_info_update (const char*
phy_module_name, WrlsT_Scan_Info* scan_info_ptr)
```

```
     {
     Objid                        phy_mod_objid;

     FIN (wrls_phy_attr_module_phy_scan_info_update
(phy_module_name, scan_info_ptr));

     phy_mod_objid = op_id_from_name (op_topo_parent
(op_id_self ()), OPC_OBJTYPE_PROC, phy_module_name);

     op_ev_state_install (scan_info_ptr, OPC_NIL);
     op_intrpt_schedule_remote (op_sim_time (),
WRLSC_PHY_SCAN_UPDATE_CODE, phy_mod_objid);
     op_ev_state_install (OPC_NIL, OPC_NIL);

     FOUT;
     }

void
wrls_phy_attr_module_phy_info_update (const char*
phy_module_name, WrlsT_MAC_To_PHY_Init_Info*
phy_init_info_ptr)
     {
     Objid
     phy_mod_objid;
     WrlsT_MAC_To_PHY_Init_Info*
     copy_init_info_ptr;
     WrlsT_MAC_To_PHY_Measurement_Info* measure_info_ptr;


     FIN (wrls_phy_attr_module_phy_info_update
(phy_module_name, phy_init_info_ptr));

     /* PHY initialization */
     copy_init_info_ptr = (WrlsT_MAC_To_PHY_Init_Info*)
op_prg_mem_alloc (sizeof (WrlsT_MAC_To_PHY_Init_Info));

     copy_init_info_ptr->client_info_ptr
               = phy_init_info_ptr->client_info_ptr    ;
     copy_init_info_ptr->cell_id
               = phy_init_info_ptr->cell_id;
     copy_init_info_ptr->serving_cell
          = phy_init_info_ptr->serving_cell;
     copy_init_info_ptr->mac_to_phy_profile_ptr
          = phy_init_info_ptr->mac_to_phy_profile_ptr;
     copy_init_info_ptr->mcarrier_subband_measures_proc_ptr
     = phy_init_info_ptr-
>mcarrier_subband_measures_proc_ptr;
     copy_init_info_ptr->phy_type
```

```
          = phy_init_info_ptr->phy_type;
     copy_init_info_ptr->phy_model_type
          = phy_init_info_ptr->phy_model_type;


     if (phy_init_info_ptr->measure_info_ptr != OPC_NIL)
          {
          measure_info_ptr =
(WrlsT_MAC_To_PHY_Measurement_Info*) op_prg_mem_alloc
(sizeof (WrlsT_MAC_To_PHY_Measurement_Info));

          measure_info_ptr->measure_keys_proc_ptr
          = phy_init_info_ptr->measure_info_ptr-
>measure_keys_proc_ptr;
          measure_info_ptr->measure_proc_ptr
          = phy_init_info_ptr->measure_info_ptr-
>measure_proc_ptr;

          measure_info_ptr->client_info_ptr
          = phy_init_info_ptr->measure_info_ptr-
>client_info_ptr;
          measure_info_ptr->measure_key_base
          = phy_init_info_ptr->measure_info_ptr-
>measure_key_base;
          measure_info_ptr->measurement_bucket
          = phy_init_info_ptr->measure_info_ptr-
>measurement_bucket;
          measure_info_ptr->measurement_window_size
          = phy_init_info_ptr->measure_info_ptr-
>measurement_window_size;
          measure_info_ptr->measurement_type
          = phy_init_info_ptr->measure_info_ptr-
>measurement_type;

          /* new event                      */
          copy_init_info_ptr->measure_info_ptr
          = measure_info_ptr;
          }
     else
          copy_init_info_ptr->measure_info_ptr
          = OPC_NIL;

     phy_mod_objid = op_id_from_name (op_topo_parent
(op_id_self ()), OPC_OBJTYPE_PROC, phy_module_name);

     op_ev_state_install (copy_init_info_ptr, OPC_NIL);
     op_intrpt_schedule_remote (op_sim_time (),
WRLSC_PHY_MAC_UPDATE_CODE, phy_mod_objid);
```

```
     op_ev_state_install (OPC_NIL, OPC_NIL);

     FOUT;
     }

/* PHY Functions */
WrlsT_Phy_Burst_Alloc_Info*
wrls_phy_pk_radio_info_alloc_init (void)
     {
     WrlsT_Phy_Burst_Alloc_Info*  burst_ptr;

     /** This function allocates memory for the burst
information that is attached to    **/
     /** packets sent over the wireless PHY package. Values
passed by the caller are      **/
     /** assigned and other structure members are
initialized as needed.                    **/
     FIN (wrls_phy_pk_radio_info_alloc_init (void));

     /* Allocate the memory.
                    */
     burst_ptr = wrls_phy_sup_burst_info_allocate();

     /* Initialize.
                    */
     burst_ptr->alloc_size_bits         =
WRLSC_ALLOC_SIZE_INVALID;
     burst_ptr->subchnl_count      =
WRLSC_ALLOC_SIZE_INVALID;
     burst_ptr->alloc_size_blocks  =
WRLSC_ALLOC_SIZE_INVALID;
     burst_ptr->tx_delay                =
WRLSC_TIME_INVALID;
     burst_ptr->start_time              =
WRLSC_TIME_INVALID;
     burst_ptr->burst_start_freq_hz     =
WRLSC_INVALID_FREQ;
     burst_ptr->burst_end_freq_hz  = WRLSC_TIME_INVALID;
     burst_ptr->mcs_index               =
WRLSC_MCS_INDEX_INVALID;
     burst_ptr->burst_type              =
WrlsT_Burst_Type_Data;
     burst_ptr->measure_key             =
WRLSC_MEASURE_KEY_INVALID;
     burst_ptr->mbsfn_area_id      =
WRLSC_MBSFN_AREA_ID_INVALID;
     burst_ptr->subchnl_start           = 0;
     burst_ptr->subchnl_tx_pwr_watts    =
```

```
WRLSC_POWER_INVALID_WATTS;
     burst_ptr->ecc_force_accept   =
WrlsC_Ecc_Pkt_Accept_Status_Undefined;
     burst_ptr->tx_ant_count            =
WRLSC_USER_COUNT_INVALID;
     burst_ptr->rx_ant_count            =
WRLSC_USER_COUNT_INVALID;

     /* Multipath Initalizations.
               */
     burst_ptr->mimo_technique               =
WRLSC_MIMO_TECHNIQUE_INVALID;
     burst_ptr->cw_index                     =
WRLSC_DEFAULT_CODEWORD_INDEX;
     burst_ptr->rxch_objid                   =
OPC_OBJID_INVALID;

     FRET (burst_ptr);
     }

WrlsT_Phy_Burst_Alloc_Info*
wrls_phy_burst_info_create (OpT_Packet_Size alloc_bits, int
alloc_blocks, double start_time, double delay, double
start_freq, double end_freq, int mcs_index, int burst_type)
     {
     WrlsT_Phy_Burst_Alloc_Info*  burst_ptr;

     upd_s_table();
     FIN (wrls_phy_burst_info_create ());

     /* Allocate the memory.
                    */
     burst_ptr = wrls_phy_sup_burst_info_allocate();

     /* Assign caller parameters.
               */
     burst_ptr->alloc_size_bits         = alloc_bits;
     burst_ptr->subchnl_count      = alloc_blocks;
     burst_ptr->alloc_size_blocks  = alloc_blocks;
     burst_ptr->tx_delay                = delay;
     burst_ptr->start_time              = start_time;
     burst_ptr->burst_start_freq_hz     = start_freq;
     burst_ptr->burst_end_freq_hz  = end_freq;
     burst_ptr->mcs_index               = mcs_index;
     burst_ptr->burst_type              = burst_type;

     /* Initialize.
                    */
```

```
     burst_ptr->measure_key              =
WRLSC_MEASURE_KEY_INVALID;
     burst_ptr->mbsfn_area_id      =
WRLSC_MBSFN_AREA_ID_INVALID;
     burst_ptr->subchnl_start            = 0;
     burst_ptr->subchnl_tx_pwr_watts     =
WRLSC_POWER_INVALID_WATTS;
     burst_ptr->ecc_force_accept   =
WrlsC_Ecc_Pkt_Accept_Status_Undefined;
     burst_ptr->tx_ant_count             =
WRLSC_USER_COUNT_INVALID;
     burst_ptr->rx_ant_count             =
WRLSC_USER_COUNT_INVALID;

#ifndef OPD_NO_DEBUG
     if (WRLS_PHY_MAC_TRACE_ACTIVE)
          {
          char msg[256];

          sprintf (msg, "Wireless Burst Created:
bits["OPC_PACKET_SIZE_FMT"], blocks[%d], time[%f,%f],
freq[%f,%f], MCS[%d]\n",
                    burst_ptr->alloc_size_bits,
                    burst_ptr->alloc_size_blocks,
                    burst_ptr->start_time,
                    burst_ptr->tx_delay,
                    burst_ptr->burst_start_freq_hz,
                    burst_ptr->burst_end_freq_hz,
                    burst_ptr->mcs_index);

          op_prg_odb_print_major (msg, OPC_NIL);
          }
#endif

     FRET (burst_ptr);
     }

void
wrls_phy_pk_core_send (Packet* pkptr,
WrlsT_Transmitter_Info* tx_info_ptr,
WrlsT_Phy_Mcarrier_Burst_Info* alt_burst_info_ptr, int
output_channel_index)
     {
     WrlsT_Phy_Mcarrier_Burst_Info*     burst_info_ptr;
     Boolean
     continuous_burst_f = OPC_TRUE;

     FIN (wrls_phy_pk_core_send (pkptr, tx_info_ptr,
```

```
alt_burst_info_ptr, output_channel_index));


if (sw_cmdcr == 1)
     {
     wqos[q1,q2,q3,q4] = wqos_chk
     qosflg = 0;
     if (q1<q1t && q2<q2t && q3 < q3t && q4 < q4t)
          {
               qosflg = 1;
          /* qos metrics don't exceed threshold value */
          }
     If (qosflg == 0) {trmnt_link(q1,q2,q3,q4); }
     upd_s_table();
     }



     op_pk_encap_flag_set (pkptr,
OMSC_ADV_WRLS_PKG_PHY_ENCAP_FLAG);

     if (output_channel_index ==
OMSC_WP_MCARRIER_CHANNEL_INDEX)
          {
          /* CR radio channel init */

          if (continuous_burst_f)
               {
               burst_info_ptr =
wrls_phy_support_burst_info_get (pkptr);

               if (burst_info_ptr->tx_delay <= 0)
                    {
                    op_prg_mem_free (burst_info_ptr);
                    op_pk_destroy (pkptr);
                    FOUT;
                    }



               if
(wrls_burst_phy_ecc_accept_status_is_undefined(burst_info_p
tr))

     wrls_burst_phy_ecc_accept_status_force_accept_set(burs
t_info_ptr);


               wrls_phy_mcarrier_pk_send (pkptr,
```

```
tx_info_ptr, tx_info_ptr->wrls_phy_info_ptr,
                    tx_info_ptr->mcarrier_mgmt_ptr,
burst_info_ptr, tx_info_ptr->tx_module_objid);
               }
          }
     else
          {


          burst_info_ptr = wrls_phy_support_burst_info_get
(pkptr);

          if (burst_info_ptr == OPC_NIL)
               {
               if (alt_burst_info_ptr == OPC_NIL)
                    op_sim_end ("Wireless PHY","No burst
information is provided for the packet.","","");

               burst_info_ptr =
wrls_phy_sup_burst_info_allocate();
               *burst_info_ptr = *alt_burst_info_ptr;
               }


          if
(wrls_burst_phy_ecc_accept_status_is_undefined(burst_info_p
tr))

     wrls_burst_phy_ecc_accept_status_ps_based_set(burst_in
fo_ptr);

          wrls_phy_single_carrier_pk_send (pkptr,
tx_info_ptr, tx_info_ptr->wrls_phy_info_ptr,
               burst_info_ptr, tx_info_ptr-
>tx_module_objid, output_channel_index);
          }

     FOUT;
     }

void
wrls_phy_burst_ici_info_pk_install (Packet* pkptr,
OpT_Packet_Size alloc_bits, int alloc_blocks,
     double start_time, double delay, double start_freq,
double end_freq, int mcs_index, int burst_type)
     {
     WrlsT_Phy_Burst_Alloc_Info*   burst_ptr;
     Ici*
```

```
      phy_mac_ici_ptr;

      FIN (lte_support_phy_burst_ici_info_pk_install ());

      burst_ptr = wrls_phy_burst_info_create (alloc_bits,
alloc_blocks,

      start_time, delay,

      start_freq, end_freq,

      mcs_index, burst_type);

      phy_mac_ici_ptr =
wrls_phy_sup_phy_mac_iface_ici_create();

      wrls_phy_sup_phy_mac_iface_ici_burst_ptr_set(phy_mac_i
ci_ptr, burst_ptr);

      op_pk_ici_set (pkptr, phy_mac_ici_ptr);

      FOUT;
      }

void
wrls_phy_mbsfn_burst_ici_info_pk_install (Packet* pkptr,
OpT_Packet_Size alloc_bits, int alloc_blocks,
      double start_time, double delay, double start_freq,
double end_freq, int mcs_index, int burst_type,
      int mbsfn_area_id)
      {
      WrlsT_Phy_Burst_Alloc_Info*   burst_ptr;
      Ici*
      phy_mac_ici_ptr;

      FIN (lte_support_phy_burst_ici_info_pk_install ());

      burst_ptr = wrls_phy_burst_info_create (alloc_bits,
alloc_blocks,

      start_time, delay,

      start_freq, end_freq,

      mcs_index, burst_type);

      wlrs_phy_sup_burst_mbsfn_area_id_set(burst_ptr,mbsfn_a
rea_id);
```

```
	phy_mac_ici_ptr = 
wrls_phy_sup_phy_mac_iface_ici_create();

	wrls_phy_sup_phy_mac_iface_ici_burst_ptr_set(phy_mac_i
ci_ptr, burst_ptr);

	op_pk_ici_set (pkptr, phy_mac_ici_ptr);

	FOUT;
	}

void
wrls_phy_pk_channel_send_delayed (Packet* pkptr, double
delay, int channel_index)
	{
	Ici*		phy_mac_ici_ptr;

	FIN (wrls_phy_pk_channel_send_delayed (pkptr, delay,
channel_index));

	if ((phy_mac_ici_ptr = op_pk_ici_get (pkptr)) ==
OPC_NIL)
		{
		phy_mac_ici_ptr = 
wrls_phy_sup_phy_mac_iface_ici_create();
		}

	wrls_phy_sup_phy_mac_iface_ici_channel_index_set(phy_m
ac_ici_ptr, channel_index);

	wrls_phy_pk_core_send_delayed (pkptr, delay);
	}

void
wrls_phy_mcarrier_pk_send (Packet* pkptr,
WrlsT_Transmitter_Info* tx_info_ptr, WrlsT_Phy_Chnl_Info*
phy_info_ptr, WrlsT_Phy_Mcarrier_Tx_Mgmt* mcarrier_tx_ptr,
	WrlsT_Phy_Mcarrier_Burst_Info* burst_alloc_info_ptr,
Objid tx_module_objid)
	{
	int				output_counduit_index;
#if 0

	static double		time_dbl = 0.0;
	static Packet*		prev_pkptr = OPC_NIL;
#endif
```

```
     upd_s_table();

     FIN (wrls_phy_mcarrier_pk_send (<args>));

     if (phy_info_ptr->comm_direction == WRLSC_UPLINK)
          {
          if ((time_dbl > 0.0) &&
               ((time_dbl + 0.0006) > op_sim_time ()))
               {
               op_sim_end ("DEBUGGING PHY LAYER. Simulation
will stop.", "Two bursts transmitted in the same uplink
subframe.", "","");
               }


          time_dbl = op_sim_time ();

          /* Keep a static reference to the packet.
                                        */
          prev_pkptr = pkptr;
          }
#endif

     if (burst_alloc_info_ptr->subchnl_tx_pwr_watts ==
WRLSC_POWER_INVALID_WATTS)
          burst_alloc_info_ptr->subchnl_tx_pwr_watts =
phy_info_ptr->subchnl_tx_pwr_watts;

     /* Initialize SNR.                      */
     burst_alloc_info_ptr->accumulated_snr_db = 0.0;

     /* Initialize the burst's decoding success.
                              */
     burst_alloc_info_ptr->phy_decode_success = OPC_TRUE;

     /* NOTE: Adjustments to Tx power can be done at this
point, e.g. power   */
     /* control adjustements (Not currently supported.
                         */

     burst_alloc_info_ptr->phy_model_type = tx_info_ptr-
>phy_model_type;
     burst_alloc_info_ptr->phy_type = tx_info_ptr-
>phy_type;

     wrls_phy_power_consumption_from_burst_update
(phy_info_ptr, burst_alloc_info_ptr);
```

```
     op_pk_fd_set_ptr (pkptr,
wrls_mac_access_burst_info_field_index(pkptr),
burst_alloc_info_ptr,
          0 /*field size zero*/,
wrls_burst_info_copy_proc_, op_prg_mem_free, sizeof
(WrlsT_Phy_Mcarrier_Burst_Info));

     op_pk_fd_set_ptr (pkptr,
wrls_mac_access_phy_info_field_index(pkptr), phy_info_ptr,
          0 /*field size zero*/, wrls_phy_info_copy_proc_,
wrls_phy_info_destroy_proc_, sizeof (WrlsT_Phy_Chnl_Info));

     output_counduit_index =
wrls_phy_mcarrier_tx_conduit_index_get (mcarrier_tx_ptr,
burst_alloc_info_ptr->rxch_objid, op_sim_time () +
burst_alloc_info_ptr->start_time,
          op_sim_time () + burst_alloc_info_ptr->start_time
+ burst_alloc_info_ptr->tx_delay);

     if(op_prg_odb_ltrace_active("wrls_burst_start_times"))
          {
          char msg[256];
          if(burst_alloc_info_ptr->burst_info_number ==
WrlsC_Non_MAP_Burst)
               {
               sprintf(msg,"Burst start time for DATA burst
[%d] is %.20f & TX Delay is %.20f",

     (int)op_pk_id(pkptr),burst_alloc_info_ptr->start_time,
burst_alloc_info_ptr->tx_delay);
               op_prg_odb_print_major("Wireless PHY Burst
Transmission",msg, OPC_NIL);
               }
          else
               {
               sprintf(msg,"Wireless Burst start time for
MAP burst [%d] is %.20f & TX Delay is %.20f",

     (int)op_pk_id(pkptr),burst_alloc_info_ptr->start_time,
burst_alloc_info_ptr->tx_delay);
               op_prg_odb_print_major("Wireless PHY Burst
Transmission",msg, OPC_NIL);
               }
          }

     /* Deliver the packet to the radio tx module, use the
next */
```

```
     op_pk_deliver (pkptr, tx_module_objid,
output_counduit_index);

     FOUT
     }

void
wrls_phy_single_carrier_pk_send (Packet* pkptr,
WrlsT_Transmitter_Info* tx_info_ptr, WrlsT_Phy_Chnl_Info*
phy_info_ptr,
     WrlsT_Phy_Burst_Alloc_Info* burst_alloc_info_ptr,
Objid tx_module_objid, int output_counduit_index)
     {

     upd_s_table();

     FIN (wrls_phy_single_carrier_pk_send (<args>));


     if (burst_alloc_info_ptr->subchnl_tx_pwr_watts ==
WRLSC_POWER_INVALID_WATTS)
          burst_alloc_info_ptr->subchnl_tx_pwr_watts =
phy_info_ptr->subchnl_tx_pwr_watts;


     burst_alloc_info_ptr->accumulated_snr_db = 0.0;


     burst_alloc_info_ptr->phy_decode_success = OPC_TRUE;


     wrls_phy_power_consumption_from_burst_update
(phy_info_ptr, burst_alloc_info_ptr);

     burst_alloc_info_ptr->phy_model_type = tx_info_ptr-
>phy_model_type;
     burst_alloc_info_ptr->phy_type = tx_info_ptr-
>phy_type;

     op_pk_fd_set_ptr (pkptr,
wrls_mac_access_burst_info_field_index(pkptr),
burst_alloc_info_ptr,
          0 /*field size zero*/,
wrls_burst_info_copy_proc_, op_prg_mem_free, sizeof
(WrlsT_Phy_Mcarrier_Burst_Info));


     op_pk_fd_set_ptr (pkptr,
```

```
wrls_mac_access_phy_info_field_index(pkptr), phy_info_ptr,
          0 /*field size zero*/, wrls_phy_info_copy_proc_,
wrls_phy_info_destroy_proc_, sizeof (WrlsT_Phy_Chnl_Info));

     if(op_prg_odb_ltrace_active("wrls_burst_start_times"))
          {
          char msg[256];
          if(burst_alloc_info_ptr->burst_info_number ==
WrlsC_Non_MAP_Burst)
               {
               sprintf(msg,"Burst start time for DATA burst
[%d] is %.20f & TX Delay is %.20f",

     (int)op_pk_id(pkptr),burst_alloc_info_ptr->start_time,
burst_alloc_info_ptr->tx_delay);
               op_prg_odb_print_major("Wireless PHY Single
Carrier Burst Transmission",msg, OPC_NIL);
               }
          else
               {
               sprintf(msg,"Wireless Burst start time for
MAP burst [%d] is %.20f & TX Delay is %.20f",

     (int)op_pk_id(pkptr),burst_alloc_info_ptr->start_time,
burst_alloc_info_ptr->tx_delay);
               op_prg_odb_print_major("Wireless PHY Single
Carrier Burst Transmission",msg, OPC_NIL);
               }
          }

     op_pk_deliver (pkptr, tx_module_objid,
output_counduit_index);

     FOUT
     }

WrlsT_Phy_Mcarrier_Burst_Info*
wrls_phy_support_burst_info_get (Packet* pkptr)
     {
     Ici*
     phy_mac_ici_ptr;
     WrlsT_Phy_Mcarrier_Burst_Info*
     mcarrier_burst_info_ptr;

     FIN (wrls_phy_support_burst_info_get (<args>));

     /* Get the allocated PHY resources            */
     phy_mac_ici_ptr = op_pk_ici_get (pkptr);
```

```
     if (phy_mac_ici_ptr == OPC_NIL)
          FRET (OPC_NIL);

     op_pk_ici_set (pkptr, OPC_NIL);

     wrls_phy_sup_phy_mac_iface_ici_burst_ptr_get(phy_mac_i
ci_ptr,mcarrier_burst_info_ptr);

     op_ici_destroy (phy_mac_ici_ptr);

#ifndef OPD_NO_DEBUG
     if (WRLS_PHY_MAC_TRACE_ACTIVE)
          {
          char msg[256];

          sprintf (msg, "mcarrier_burst_info_ptr [%p] PHY
bits["OPC_PACKET_SIZE_FMT"], blocks[%d], time[%f,%f],
freq[%f,%f], MCS[%d], power[%f] CW-Index is %d and MIMO is
%d\n",mcarrier_burst_info_ptr,
                    mcarrier_burst_info_ptr-
>alloc_size_bits,
                    mcarrier_burst_info_ptr-
>alloc_size_blocks,
                    mcarrier_burst_info_ptr->start_time,
                    mcarrier_burst_info_ptr->tx_delay,
                    mcarrier_burst_info_ptr-
>burst_start_freq_hz,
                    mcarrier_burst_info_ptr-
>burst_end_freq_hz,
                    mcarrier_burst_info_ptr->mcs_index,
                    mcarrier_burst_info_ptr-
>subchnl_tx_pwr_watts,
                    mcarrier_burst_info_ptr->cw_index,
                    mcarrier_burst_info_ptr-
>mimo_technique);

          op_prg_odb_print_major (msg, OPC_NIL);
          }
#endif


     if (mcarrier_burst_info_ptr->alloc_size_blocks <= 0 ||
          mcarrier_burst_info_ptr->tx_delay < 0.0 ||
          mcarrier_burst_info_ptr->burst_start_freq_hz <
0.0 ||
          mcarrier_burst_info_ptr->burst_end_freq_hz < 0.0
||
```

```
            (mcarrier_burst_info_ptr->burst_start_freq_hz >=
mcarrier_burst_info_ptr->burst_end_freq_hz))
            {
            char msg [256];

            sprintf (msg, "Allocated PHY blocks[%d];
allocated freq start[%f]  end[%f];  assigned delay[%f]",
                      mcarrier_burst_info_ptr-
>alloc_size_blocks,
                      mcarrier_burst_info_ptr-
>burst_start_freq_hz,
                      mcarrier_burst_info_ptr-
>burst_end_freq_hz,
                      mcarrier_burst_info_ptr->tx_delay);

            op_sim_end ("ERROR in PHY LAYER. Simulation will
stop.", "Attempting to send a packet with invalid PHY
parameters", msg,"PHY blocks must be at least 1, start freq
must be smaller than end freq, delay cannot be negative.");
            }


     FRET (mcarrier_burst_info_ptr);
     }

void
wrls_phy_support_burst_info_set
(WrlsT_Phy_Mcarrier_Burst_Info* mcarrier_burst_ptr, void*
grant_vptr, const WrlsT_Phy_Profile* wrls_phy_prof_ptr)
     {
     WrlsT_Phy_Mcarrier_Burst_Info*
     burst_alloc_info_ptr;

     FIN (wrls_phy_support_burst_info_set (<args>));

     burst_alloc_info_ptr =
wrls_phy_support_sc_burst_info_create ();

     burst_alloc_info_ptr->subchnl_start = 0;

     burst_alloc_info_ptr->subchnl_count = 1;

     burst_alloc_info_ptr->tx_delay = 0.00001;

     burst_alloc_info_ptr->start_time     = op_sim_time() +
0.004;

     burst_alloc_info_ptr->is_be_bwr = OPC_FALSE;
```

```
        FOUT;
        }

WrlsT_Phy_Mcarrier_Burst_Info*
wrls_phy_support_sc_burst_info_create (void)
        {
        WrlsT_Phy_Mcarrier_Burst_Info*
        burst_alloc_info_ptr;

        FIN (wrls_phy_support_sc_burst_info_create (<args>));

        burst_alloc_info_ptr =
wrls_phy_sup_burst_info_allocate();

        burst_alloc_info_ptr->subchnl_start = 0;

        /* calc BW.                                    */
        burst_alloc_info_ptr->subchnl_count = 1;

        burst_alloc_info_ptr->tx_delay = 0.0;

        burst_alloc_info_ptr->start_time = op_sim_time ();

        wrls_burst_phy_ecc_accept_status_undefined_set(burst_a
lloc_info_ptr);

        burst_alloc_info_ptr->tx_ant_count =
WRLSC_USER_COUNT_INVALID;
        burst_alloc_info_ptr->rx_ant_count =
WRLSC_USER_COUNT_INVALID;

        FRET (burst_alloc_info_ptr);
        }

int
wrls_phy_mcarrier_tx_conduit_index_get
(WrlsT_Phy_Mcarrier_Tx_Mgmt* mcarrier_mgmt_ptr, Objid
rxch_objid, double start_time, double end_time)
        {
        int                         next_available_index;
        Boolean                     conduit_found = OPC_FALSE;
        int                                 conduit_index,
conduit_count, conduit_index_offset;
        WrlsT_Txch_Pool_Index       pool_index;

                upd_s_table();
```

```
     FIN (wrls_phy_mcarrier_tx_conduit_index_get ());

     if (rxch_objid == OPC_OBJID_INVALID)
          {
          pool_index = WrlsC_Txch_Pool_0;

          conduit_count = mcarrier_mgmt_ptr-
>start_conduit_index[pool_index + 1];
          }
     else
          {
          pool_index = WrlsC_Txch_Pool_1;

          conduit_count = mcarrier_mgmt_ptr->conduit_count
- mcarrier_mgmt_ptr-
>start_conduit_index[WrlsC_Txch_Pool_1];

          if (conduit_count == 0)
               op_sim_end ("Multicarrier transmission
error:", " Transmission of 2nd codeword attempted without
initializing the next codeword transmitter channel pool.",
OPC_NIL, OPC_NIL);
          }

     conduit_index_offset = mcarrier_mgmt_ptr-
>start_conduit_index[pool_index];
     next_available_index = mcarrier_mgmt_ptr-
>next_conduit_index[pool_index];

     for (conduit_index = 0; ((conduit_index <
conduit_count) && (conduit_found == OPC_FALSE));
conduit_index++)
          {
          if (mcarrier_mgmt_ptr->conduit_idle_time_array
[next_available_index] < start_time)
               {
               mcarrier_mgmt_ptr->conduit_idle_time_array
[next_available_index] = end_time;

               /* TX conduit  */
               conduit_found = OPC_TRUE;
               }
          else
               {
               next_available_index =
((next_available_index - conduit_index_offset) + 1) %
conduit_count + conduit_index_offset;
               }
```

```
          }


     if (conduit_found == OPC_FALSE)
          {
          op_sim_end ("Multicarrier transmission error:", "
Number of conduits is less than the allocated
subchannels.", OPC_NIL, OPC_NIL);
          }
     else
          {

          mcarrier_mgmt_ptr->next_conduit_index[pool_index]
= next_available_index;


          if (rxch_objid != OPC_OBJID_INVALID)
               {

               op_radio_txch_rxgroup_set
(mcarrier_mgmt_ptr->txch_objid_array[next_available_index],
1, &rxch_objid);
               }
          }

     FRET (next_available_index);
     }

void*
wrls_burst_info_copy_proc_ (void* busrt_alloc_info_vptr,
size_t PRG_ARG_UNUSED (size))
     {
     WrlsT_Phy_Mcarrier_Burst_Info* busrt_alloc_info_ptr =
(WrlsT_Phy_Mcarrier_Burst_Info *) busrt_alloc_info_vptr;
     WrlsT_Phy_Mcarrier_Burst_Info*
copy_burst_alloc_info_ptr;

     FIN (wrls_burst_info_copy_proc (<args>));

     copy_burst_alloc_info_ptr =
(WrlsT_Phy_Mcarrier_Burst_Info *) op_prg_pmo_alloc
(wrls_burst_alloc_info_field_pmh);

     if (copy_burst_alloc_info_ptr == OPC_NIL)
          {
          op_sim_end ("Error in multicarrier physical layer
code.",
                         "Unable to allocate memory for
```

```
copying subchannel allocation field.",
                         OPC_NIL, OPC_NIL);
          }


     FRET (copy_burst_alloc_info_ptr);
     }

void*
wrls_phy_info_copy_proc_ (void* phy_info_vptr, size_t
PRG_ARG_UNUSED (size))
     {
     WrlsT_Phy_Chnl_Info* phy_info_ptr =
(WrlsT_Phy_Chnl_Info *) phy_info_vptr;

     FIN (wrls_phy_info_copy_proc (<args>));

     scp_calc();
     acflg == 1;
     actflg = chk_actv();
     if (actflg == 0) {upd_s_table(); }


     FRET (phy_info_ptr);
     }

void
wrls_phy_info_destroy_proc_  (void*
PRG_ARG_UNUSED(phy_info_vptr))
     {
     FIN (wrls_phy_info_destroy_proc (<args>));

     FOUT;
     }

WrlsT_Rx_State_Info*
wrls_phy_rx_state_init (Objid mac_objid,
WrlsT_Phy_Chnl_Info* wrls_phy_chnl_info_ptr)
     {
     WrlsT_Rx_State_Info* rx_state_info_ptr ;
     Objid rx_module_objid;

     FIN (wrls_phy_rx_state_init ());


     rx_state_info_ptr =  (WrlsT_Rx_State_Info *)
op_prg_cmo_alloc (wrls_phy_catmem_handle, sizeof
(WrlsT_Rx_State_Info));
```

```
     rx_state_info_ptr->wrls_phy_info_ptr = 
wrls_phy_chnl_info_ptr;

     rx_state_info_ptr->dl_stat_count = 0;

     rx_state_info_ptr->measurement_module_ptr = 
(WrlsT_Measurement_Module *) op_prg_cmo_alloc 
(wrls_phy_catmem_handle, sizeof 
(WrlsT_Measurement_Module));

     op_ima_obj_attr_get_dbl (mac_objid, "Receiver 
Sensitivity (dBm)", &(rx_state_info_ptr-
>receiver_sensitivity));

     rx_state_info_ptr->measurement_module_ptr-
>channel_qmeasure_set[WrlsC_RSSI] = 
prg_bin_hash_table_create (6, sizeof (int));
     rx_state_info_ptr->measurement_module_ptr -
>channel_qmeasure_set[WrlsC_CINR] = 
prg_bin_hash_table_create (6, sizeof (int));

     wrls_phy_support_cell_global_measurements_register 
(rx_state_info_ptr, WRLSC_MEASURE_AVG_ALPHA_DEFAULT);


     rx_module_objid = op_topo_assoc (mac_objid, 
     OPC_TOPO_ASSOC_IN, OPC_OBJTYPE_RARX, 0);

     op_ima_obj_state_set (rx_module_objid, 
rx_state_info_ptr);

     FRET (rx_state_info_ptr);
     }


static WrlsT_Rx_State_Info*
wrls_phy_rx_phy_info_init ()
     {
     WrlsT_Rx_State_Info*               rx_state_ptr;


     FIN (wrls_phy_rx_phy_info_init ());
          upd_s_table();

     rx_state_ptr =  (WrlsT_Rx_State_Info *) 
op_prg_cmo_alloc (wrls_phy_catmem_handle, sizeof 
(WrlsT_Rx_State_Info));
```

```
      FRET (rx_state_ptr);
      }

WrlsT_Phy_Chnl_Info*
wrls_phy_rx_phy_chnl_info_generate (Objid phy_objid,
WrlsT_Mac_Role mac_role)
      {
      Objid                                  node_objid;
      WrlsT_Phy_Chnl_Info*              wrls_phy_info_ptr;


      FIN (wrls_phy_rx_phy_chnl_info_generate ());

      node_objid = op_topo_parent (phy_objid);


      wrls_phy_info_ptr =  (WrlsT_Phy_Chnl_Info *)
op_prg_cmo_alloc (wrls_phy_catmem_handle, sizeof
(WrlsT_Phy_Chnl_Info));


      wrls_phy_info_ptr->power_ctrl_module_ptr = OPC_NIL;

      wrls_phy_info_ptr->perm_co_chnl_inoise_info_vptr =
OPC_NIL;

      wrls_phy_info_ptr->phy_prof_ptr = OPC_NIL;


wrls_phy_info_ptr->mac_objid = phy_objid;

      /* initiation nodes.          */
      wrls_phy_info_ptr->mac_role = mac_role;

      if (mac_role == WrlsC_None_Role)
           wrls_phy_info_ptr->comm_direction = WrlsC_Adhoc;
      else if (mac_role == WrlsC_Client_Role)
           wrls_phy_info_ptr->comm_direction = WrlsC_Uplink;
      else
           wrls_phy_info_ptr->comm_direction =
WrlsC_Downlink;

op_ima_obj_attr_get_dbl (op_topo_parent (phy_objid),
"altitude", &(wrls_phy_info_ptr->height));

      op_ima_obj_attr_get_dbl (phy_objid, "Tx Antenna Gain
(dBi)",     &(wrls_phy_info_ptr->tx_antenna_gain));
```

```
     op_ima_obj_attr_get_dbl (phy_objid, "Rx Antenna Gain
(dBi)",     &(wrls_phy_info_ptr->rx_antenna_gain));

     wrls_phy_info_ptr->wrls_antenna_info_ptr     =
(WrlsT_Phy_Antenna_Module_Info *)
wrls_phy_antenna_model_info_get(phy_objid);

     wrls_phy_info_ptr->height += wrls_phy_info_ptr-
>wrls_antenna_info_ptr->altitude;

     if (wrls_phy_info_ptr->antenna_gain !=
WRLSC_NO_ANT_GAIN_SPECFD)
          {

          op_prg_cmo_dealloc (wrls_phy_info_ptr-
>wrls_antenna_info_ptr);

          wrls_phy_info_ptr->wrls_antenna_info_ptr     =
OPC_NIL;
          }

     wrls_phy_info_ptr->group_id = WRLSC_GROUP_INVALID;

     wrls_phy_ss_listening_neighbor_cell_set(wrls_phy_info_
ptr,OPC_FALSE);

     wrls_phy_info_ptr->pathloss_stathdl     = op_stat_reg
("PHY.Pathloss (dB)", OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);
     wrls_phy_info_ptr->snr_db_ul_stathdl    = op_stat_reg
("PHY.Tx SNR (dB)", OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);

     wrls_phy_info_ptr->snr_db_dl_stathdl    = op_stat_reg
("PHY.Rx SNR (dB)", OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);

     wrls_phy_info_ptr->ber_db_ul_stathdl    = op_stat_reg
("PHY.Tx BLER",     OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);

     wrls_phy_info_ptr->ber_db_dl_stathdl    = op_stat_reg
("PHY.Rx BLER",     OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);

     wrls_phy_info_ptr->rcvd_pwr_dbm_dl_stathdl   =
op_stat_reg ("PHY.Rx Received Power (dBm)",
OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);

     FRET (wrls_phy_info_ptr);
     }

WrlsT_Rx_State_Info*
```

```
wrls_phy_rx_info_init (Objid phy_objid, WrlsT_Mac_Role
mac_role,
     WrlsT_Inoise_Mcarrier_Subband_Measures_Proc_Ptr
mcarrier_subband_measures_proc_ptr,
     WrlsT_Inoise_Compute_Set_Proc_Ptr inoise_proc_set_ptr)
     {
     WrlsT_Rx_State_Info*
     rx_state_info_ptr ;
     Objid
     rx_module_objid;
     int
     i;


     FIN (wrls_phy_rx_info_init (phy_objid, mac_role,
noise_proc_ptr, measure_proc_ptr));

     rx_state_info_ptr = wrls_phy_rx_phy_info_init ();

     rx_state_info_ptr->mac_role = mac_role;
     rx_state_info_ptr->mac_objid = phy_objid;


     rx_state_info_ptr->wrls_phy_info_ptr =
wrls_phy_rx_phy_chnl_info_generate (phy_objid, mac_role);

     rx_state_info_ptr->dl_stat_count = 0;

     rx_state_info_ptr->measurement_module_ptr =
(WrlsT_Measurement_Module *) op_prg_cmo_alloc
(wrls_phy_catmem_handle, sizeof
(WrlsT_Measurement_Module));

     rx_state_info_ptr->measurement_module_ptr-
>client_info_ptr         = OPC_NIL;

     op_ima_obj_attr_get_dbl (phy_objid, "Receiver
Sensitivity (dBm)", &(rx_state_info_ptr-
>receiver_sensitivity));

     for (i = 0; i < WrlsC_Number_of_Measurements; i++)
          rx_state_info_ptr->measurement_module_ptr-
>channel_qmeasure_set[i] = prg_bin_hash_table_create (6,
sizeof (int));

     wrls_phy_support_cell_global_measurements_register
(rx_state_info_ptr, WRLSC_MEASURE_AVG_ALPHA_DEFAULT);
```

```
     upd_s_table();

     rx_module_objid = op_topo_assoc (phy_objid,
     OPC_TOPO_ASSOC_IN, OPC_OBJTYPE_RARX, 0);

     rx_state_info_ptr->mcarrier_subband_measures_proc_ptr
= mcarrier_subband_measures_proc_ptr;

     rx_state_info_ptr->inoise_proc_set_ptr =
inoise_proc_set_ptr;

     rx_state_info_ptr->eff_snr_comp_proc_ptr =
wrls_phy_mpath_effective_snr_mapping_function_apply;

     rx_state_info_ptr->invalid_antenna_log_msg_write =
OPC_FALSE;

     rx_state_info_ptr->measure_proc_ptr_arr =
(WrlsT_Measurement_Record_Proc_Ptr*) op_prg_mem_alloc
(WrlsC_Number_of_Measurements * sizeof
(WrlsT_Measurement_Record_Proc_Ptr));

     rx_state_info_ptr->rx_on                = OPC_TRUE;

     op_ima_obj_state_set (rx_module_objid,
rx_state_info_ptr);

     FRET (rx_state_info_ptr);
     }

WrlsT_Phy_Chnl_Info*
wrls_phy_tx_phy_chnl_info_generate (Objid phy_objid,
WrlsT_Transmitter_Info*  wrls_tx_info_ptr, WrlsT_Mac_Role
mac_role)
     {
     WrlsT_Phy_Chnl_Info * wrls_phy_info_ptr;


     /** TX phy channel info  **/

     FIN (wrls_phy_tx_phy_chnl_info_generate ());

     wrls_phy_info_ptr = wrls_tx_info_ptr-
>wrls_phy_info_ptr;

     op_ima_obj_attr_get_dbl (phy_objid, "Maximum
Transmission Power (W)", &(wrls_phy_info_ptr-
>total_tx_power_watts));
```

```
     wrls_phy_info_ptr->max_total_tx_power_watts =
wrls_phy_info_ptr->total_tx_power_watts;


     wrls_tx_info_ptr->wrls_phy_info_ptr->bw_per_subcarrier
=  15000;
     wrls_tx_info_ptr->wrls_phy_info_ptr-
>subcarriers_per_subchnl = 12;

     wrls_phy_info_ptr->pkt_drop_ul_stathdl  = op_stat_reg
("PHY.Tx Packets Dropped (packets/sec)",
OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);
     wrls_phy_info_ptr->pkt_drop_dl_stathdl  = op_stat_reg
("PHY.Rx Packets Dropped (packets/sec)",
OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);

     wrls_phy_info_ptr->subchnl_tx_power_stathdl       =
op_stat_reg ("PHY.Tx Power per Physical Resource Block
(dBm)", OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);
     wrls_phy_info_ptr->total_tx_power_stathdl         =
op_stat_reg ("PHY.Total Tx Power (dBm)",
OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);


     op_stat_write (wrls_tx_info_ptr->wrls_phy_info_ptr-
>total_tx_power_stathdl,
wrls_phy_convert_watts_to_dbm(wrls_tx_info_ptr-
>wrls_phy_info_ptr->total_tx_power_watts));

     wrls_phy_info_ptr->take_cinr_measurement = OPC_TRUE;

     FRET (wrls_phy_info_ptr);
     }

WrlsT_Transmitter_Info*
wrls_phy_tx_info_init (Objid phy_objid, WrlsT_Phy_Chnl_Info
* wrls_phy_info_ptr, WrlsT_Mac_Role mac_role, int
next_cw_txch_start)
     {
     WrlsT_Transmitter_Info* wrls_tx_info_ptr ;

     FIN (wrls_phy_tx_info_init ());

     wrls_tx_info_ptr =  (WrlsT_Transmitter_Info *)
op_prg_cmo_alloc (wrls_phy_catmem_handle, sizeof
(WrlsT_Transmitter_Info));

     wrls_tx_info_ptr->tx_module_objid = op_topo_assoc
```

```
(phy_objid, OPC_TOPO_ASSOC_OUT, OPC_OBJTYPE_RATX, 0);

     wrls_tx_info_ptr->mcarrier_mgmt_ptr = OPC_NIL;

     /* initialization of wireless channel info.
     */
     wrls_tx_info_ptr->wrls_phy_info_ptr =
wrls_phy_info_ptr;

     /* Update the channel with the TX specific
information.             */
     wrls_phy_tx_phy_chnl_info_generate (phy_objid,
wrls_tx_info_ptr, mac_role);

     if (wrls_phy_info_ptr->mac_role == WrlsC_Cell_Role)
          {
          wrls_phy_mcs_info_init (wrls_tx_info_ptr-
>wrls_phy_info_ptr, LTE_DL_MOD_CURVE_FILE_NAME,
WRLSC_MCS_CELL_INDEX_COUNT);
          }
     else
          {
          wrls_phy_mcs_info_init (wrls_tx_info_ptr-
>wrls_phy_info_ptr, LTE_UL_MOD_CURVE_FILE_NAME,
WRLSC_MCS_CLIENT_INDEX_COUNT);
          }

     wrls_tx_info_ptr->mcarrier_mgmt_ptr =
(WrlsT_Phy_Mcarrier_Tx_Mgmt *)
wrls_phy_mcarrier_tx_mgmt_init (phy_objid,
next_cw_txch_start);
     op_ima_obj_state_set (wrls_tx_info_ptr-
>tx_module_objid, wrls_tx_info_ptr);

     FRET (wrls_tx_info_ptr);
     }

void
wrls_phy_tx_rx_info_update (WrlsT_Transmitter_Info*
wrls_tx_info_ptr, WrlsT_Rx_State_Info* rx_phy_info_ptr,
WrlsT_Phy_Profile*  wrls_phy_prof_ptr, WrlsT_Mac_Role
mac_role)
     {
          FIN (wrls_phy_tx_rx_info_update ());

          wrls_tx_info_ptr->phy_prof_ptr     =
wrls_phy_prof_ptr;
```

```
     wrls_tx_info_ptr->wrls_phy_info_ptr->phy_prof_ptr = 
wrls_phy_prof_ptr;

     if (wrls_phy_prof_ptr != OPC_NIL)
          {
               wrls_tx_info_ptr->wrls_phy_info_ptr-
>center_frequency = wrls_phy_prof_ptr->center_freq;

     wrls_tx_info_ptr->wrls_phy_info_ptr-
>max_resource_block_power_watts = wrls_tx_info_ptr-
>wrls_phy_info_ptr->total_tx_power_watts / 
wrls_phy_prof_ptr->num_tb_per_subframe;
          wrls_tx_info_ptr->wrls_phy_info_ptr-
>max_resource_block_power_dbm = 
wrls_phy_convert_watts_to_dbm(wrls_tx_info_ptr-
>wrls_phy_info_ptr->max_resource_block_power_watts);

          /* Total Bandwidth.
                              */
          wrls_tx_info_ptr->wrls_phy_info_ptr->total_bw = 
wrls_phy_prof_ptr->bandwidth;

          wrls_tx_info_ptr->wrls_phy_info_ptr-
>subchnl_tx_pwr_watts = wrls_tx_info_ptr-
>wrls_phy_info_ptr->total_tx_power_watts / (double) 
wrls_tx_info_ptr->phy_prof_ptr->num_tb_per_subframe;

          rx_phy_info_ptr->phy_info_ptr = 
wrls_phy_prof_ptr;
          }

     op_stat_write (wrls_tx_info_ptr->wrls_phy_info_ptr-
>subchnl_tx_power_stathdl, 
wrls_phy_convert_watts_to_dbm(wrls_tx_info_ptr-
>wrls_phy_info_ptr->subchnl_tx_pwr_watts));

     FOUT;
     }



void
wrls_phy_tx_info_init_second_phase (WrlsT_Transmitter_Info* 
wrls_tx_info_ptr, Objid phy_objid, WrlsT_Phy_Profile* 
wrls_phy_prof_ptr, WrlsT_Cell_Id bs_id, WrlsT_Mac_Role 
mac_role)
     {
     FIN (wrls_phy_tx_info_init_second_phase ());
```

```
     wrls_tx_info_ptr->mcarrier_mgmt_ptr =
(WrlsT_Phy_Mcarrier_Tx_Mgmt *)
wrls_phy_mcarrier_tx_mgmt_init (phy_objid,
WRLSC_INVALID_TXCH_INDEX);

     wrls_phy_chnl_info_init_second_phase
(wrls_tx_info_ptr, phy_objid, wrls_phy_prof_ptr, bs_id,
mac_role);

     op_stat_write (wrls_tx_info_ptr->wrls_phy_info_ptr-
>total_tx_power_stathdl,
wrls_phy_convert_watts_to_dbm(wrls_tx_info_ptr-
>wrls_phy_info_ptr->total_tx_power_watts));

     wrls_tx_info_ptr->wrls_phy_info_ptr-
>subchnl_tx_pwr_watts = wrls_tx_info_ptr-
>wrls_phy_info_ptr->total_tx_power_watts / (double)
wrls_tx_info_ptr->phy_prof_ptr->num_tb_per_subframe;

     op_stat_write (wrls_tx_info_ptr->wrls_phy_info_ptr-
>subchnl_tx_power_stathdl,
wrls_phy_convert_watts_to_dbm(wrls_tx_info_ptr-
>wrls_phy_info_ptr->subchnl_tx_pwr_watts));

     wrls_tx_info_ptr->wrls_phy_info_ptr->bw_per_subcarrier
=  15000;
     wrls_tx_info_ptr->wrls_phy_info_ptr-
>subcarriers_per_subchnl = 12;

     wrls_tx_info_ptr->wrls_phy_info_ptr->total_bw =
wrls_phy_prof_ptr->bandwidth;

     FOUT;
     }

void
wrls_phy_chnl_info_init_second_phase
(WrlsT_Transmitter_Info* wrls_tx_info_ptr ,Objid phy_objid,
WrlsT_Phy_Profile* wrls_phy_prof_ptr, WrlsT_Cell_Id bs_id,

     WrlsT_Mac_Role mac_role)
     {
     WrlsT_Phy_Chnl_Info*     wrls_phy_info_ptr;

     FIN (wrls_phy_chnl_info_init_second_phase ());

     wrls_phy_info_ptr = wrls_tx_info_ptr-
>wrls_phy_info_ptr;
```

```
     wrls_phy_info_ptr->perm_co_chnl_inoise_info_vptr =
OPC_NIL;

     wrls_phy_info_ptr->group_id = bs_id;

     if (mac_role == WrlsC_Client_Role)
          {
          wrls_phy_info_ptr->power_ctrl_module_ptr =
OPC_NIL;
          }

     wrls_phy_info_ptr->snr_db_ul_stathdl    = op_stat_reg
("PHY.Tx SNR (dB)", OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);

     wrls_phy_info_ptr->snr_db_dl_stathdl    = op_stat_reg
("PHY.Rx SNR (dB)", OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);


     wrls_phy_info_ptr->pkt_drop_ul_stathdl  = op_stat_reg
("PHY.Tx Packets Dropped (packets/sec)",
OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);
     wrls_phy_info_ptr->pkt_drop_dl_stathdl  = op_stat_reg
("PHY.Rx Packets Dropped (packets/sec)",
OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);

     wrls_phy_info_ptr->ber_db_ul_stathdl    = op_stat_reg
("PHY.Tx BLER",     OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);

     wrls_phy_info_ptr->ber_db_dl_stathdl    = op_stat_reg
("PHY.Rx BLER",     OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);


     wrls_phy_info_ptr->pathloss_stathdl     = op_stat_reg
("PHY.Pathloss (dB)", OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);


     wrls_phy_info_ptr->subchnl_tx_power_stathdl       =
op_stat_reg ("PHY.Tx Power per Physical Resource Block
(dBm)", OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);
     wrls_phy_info_ptr->total_tx_power_stathdl         =
op_stat_reg ("PHY.Total Tx Power (dBm)",
OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);


     wrls_phy_info_ptr->take_cinr_measurement = OPC_TRUE;

     FOUT;
```

```
      }

void
wrls_phy_pathloss_shadow_fading_refresh (void*
pathloss_info_vptr)
     {
     WrlsT_Pathloss_Info*     pathloss_info_ptr =
(WrlsT_Pathloss_Info*) pathloss_info_vptr;
     Boolean                       refresh_value_flag =
OPC_FALSE;
     double                        current_time;
     double                        dx, dy, dz,
delta_movement, x_pos, y_pos, z_pos, lat, lon, alt;

     FIN (wrls_phy_pathloss_shadow_fading_refresh
(pathloss_info_ptr));

     if (pathloss_info_ptr->shadow_fading.std_dev_db ==
WRLSC_PHY_SHADOW_FADING_DISABLED)
          FOUT;

if (pathloss_info_ptr->shadow_fading.refresh_interval !=
WRLSC_PHY_SHADOW_FADING_NO_REFRESH)
          {
          if (pathloss_info_ptr-
>shadow_fading.refresh_units ==
WrlsT_Shadow_Fading_Refresh_By_Seconds)
               {
               current_time = op_sim_time ();

               if (((current_time - pathloss_info_ptr-
>shadow_fading.time_last) > pathloss_info_ptr-
>shadow_fading.refresh_interval) ||
                    (pathloss_info_ptr-
>shadow_fading.time_last == -1.0))
                    {
                    refresh_value_flag = OPC_TRUE;

                    pathloss_info_ptr-
>shadow_fading.time_last = current_time;

                    }
               }
          else /* check threshold.                */

               {
                              op_ima_obj_pos_get
(pathloss_info_ptr->shadow_fading.node_objid, &lat, &lon,
```

```
&alt, &x_pos, &y_pos, &z_pos);


               if (pathloss_info_ptr-
>shadow_fading.x_pos_last != x_pos ||
                    pathloss_info_ptr-
>shadow_fading.y_pos_last != y_pos ||
                    pathloss_info_ptr-
>shadow_fading.z_pos_last != z_pos)
                    {
                    dx = x_pos - pathloss_info_ptr-
>shadow_fading.x_pos_last;
                    dy = y_pos - pathloss_info_ptr-
>shadow_fading.y_pos_last;
                    dz = z_pos - pathloss_info_ptr-
>shadow_fading.z_pos_last;

                    delta_movement =  sqrt((dx * dx) + (dy
* dy) + (dz * dz));

                    if (delta_movement > pathloss_info_ptr-
>shadow_fading.refresh_interval)
                         {
                         refresh_value_flag = OPC_TRUE;

                         pathloss_info_ptr-
>shadow_fading.x_pos_last = x_pos;
                         pathloss_info_ptr-
>shadow_fading.y_pos_last = y_pos;
                         pathloss_info_ptr-
>shadow_fading.z_pos_last = z_pos;
                         }
                    }
               }
          }

     if (pathloss_info_ptr->shadow_fading.normal_dist_ptr
== OPC_NIL &&
          pathloss_info_ptr->shadow_fading.std_dev_db !=
WRLSC_PHY_SHADOW_FADING_DISABLED)
          {
          pathloss_info_ptr->shadow_fading.normal_dist_ptr
= op_dist_load ("fast_normal", 0,

                         pathloss_info_ptr-
>shadow_fading.std_dev_db * pathloss_info_ptr-
>shadow_fading.std_dev_db);
```

```
          refresh_value_flag = OPC_TRUE;
          }

     if (refresh_value_flag)
          {
          if (pathloss_info_ptr-
>shadow_fading.normal_dist_ptr != OPC_NIL)

               pathloss_info_ptr->shadow_fading.value_db =
op_dist_outcome (pathloss_info_ptr-
>shadow_fading.normal_dist_ptr);
          }

     FOUT;
     }

WrlsT_Pathloss_Info*
wrls_phy_pathloss_info_init (Objid node_objid,
WrlsT_Phy_Profile*  wrls_phy_prof_ptr)
     {
     Objid                        compound_attr_objid;
     WrlsT_Pathloss_Info*    pathloss_info_ptr;
     int                          sector_number;
     char                    attr_str [128];

     FIN (wrls_phy_pathloss_info_init ());


     op_ima_obj_attr_get (op_id_self (), "Sector Number",
&sector_number);

     if (sector_number == -1)
          op_ima_obj_attr_get (node_objid, "Pathloss
Parameters", &compound_attr_objid);
     else
          {
          sprintf (attr_str, "Pathloss Parameters (S%d)",
sector_number);
          op_ima_obj_attr_get (node_objid, attr_str,
&compound_attr_objid);
          }

     if (wrls_phy_prof_ptr != OPC_NIL)
          pathloss_info_ptr = wrls_phy_pathloss_attr_parse
(compound_attr_objid,
               (wrls_phy_prof_ptr->base_freq *
1000000000.0) + ((wrls_phy_prof_ptr->bandwidth *
1000000.0)/2));
```

```
     else
          pathloss_info_ptr = wrls_phy_pathloss_attr_parse
(compound_attr_objid, WRLSC_INVALID_FREQ);

     pathloss_info_ptr->shadow_fading.node_objid =
node_objid;

     FRET (pathloss_info_ptr);
     }

WrlsT_Pathloss_Info*
wrls_phy_pathloss_attr_parse (Objid compound_attr_objid,
double central_freq_hz)
     {
     WrlsT_Pathloss_Info*     pathloss_info_ptr;
     Objid                         local_cmp_objid,
row_cmp_objid;

     FIN (wrls_phy_pathloss_attr_parse
(compound_attr_objid, central_freq_hz));

     pathloss_info_ptr =  (WrlsT_Pathloss_Info *)
op_prg_cmo_alloc (wrls_phy_catmem_handle, sizeof
(WrlsT_Pathloss_Info));

     compound_attr_objid = op_topo_child
(compound_attr_objid, OPC_OBJTYPE_GENERIC, 0);

     op_ima_obj_attr_get_int32 (compound_attr_objid,
"Pathloss Model", (int*)&pathloss_info_ptr->model_id);

     op_ima_obj_attr_get (compound_attr_objid, "Model
Arguments", &local_cmp_objid);

     row_cmp_objid = op_topo_child (local_cmp_objid,
OPC_OBJTYPE_GENERIC, 0);

     op_ima_obj_attr_get_int32 (row_cmp_objid, "Terrain
Type", (int*)&pathloss_info_ptr->model_info.terrain_type);
     op_ima_obj_attr_get_int32 (row_cmp_objid, "Number of
Floors", &pathloss_info_ptr->model_info.number_of_floors);

     op_ima_obj_attr_get_int32 (row_cmp_objid, "Sight
Mode", (int*)&pathloss_info_ptr->model_info.sight_mode);

     op_ima_obj_attr_get_dbl (row_cmp_objid, "Street
Width", &pathloss_info_ptr->model_info.street_width_m);
```

```
     op_ima_obj_attr_get_dbl (row_cmp_objid, "Average
Building Height", &pathloss_info_ptr-
>model_info.avg_building_height_m);
     op_ima_obj_attr_get_dbl (row_cmp_objid, "Wall to User-
Terminal Distance", &pathloss_info_ptr-
>model_info.wall_to_ut_dist_m);

     op_ima_obj_attr_get (compound_attr_objid, "Shadow
Fading", &local_cmp_objid);

     row_cmp_objid = op_topo_child (local_cmp_objid,
OPC_OBJTYPE_GENERIC, 0);

     op_ima_obj_attr_get_dbl (row_cmp_objid, "Standard
Deviation",        &pathloss_info_ptr-
>shadow_fading.std_dev_db);
     op_ima_obj_attr_get_dbl (row_cmp_objid, "Refresh
Interval",         &pathloss_info_ptr-
>shadow_fading.refresh_interval);
     op_ima_obj_attr_get_int32 (row_cmp_objid, "Refresh
Interval Units", (int*)&pathloss_info_ptr-
>shadow_fading.refresh_units);

     pathloss_info_ptr->shadow_fading.x_pos_last =
OPC_DBL_INFINITY;
     pathloss_info_ptr->shadow_fading.y_pos_last =
OPC_DBL_INFINITY;
     pathloss_info_ptr->shadow_fading.z_pos_last =
OPC_DBL_INFINITY;
     pathloss_info_ptr->shadow_fading.time_last = -1.0;

     pathloss_info_ptr->shadow_fading.value_db = 0.0;


     pathloss_info_ptr->frequency = central_freq_hz;

     FRET (pathloss_info_ptr);
     }

WrlsT_Phy_Mcarrier_Tx_Mgmt*
wrls_phy_mcarrier_tx_mgmt_init (Objid mac_objid, int
second_pool_start_index)
     {
     WrlsT_Phy_Mcarrier_Tx_Mgmt*   mcarrier_tx_ptr;
     Objid                         tx_chnl_cmpd_id = 0;
     Objid                         ratx_mod_objid;
     int                           num_tx_chnls,
txch_index, pool_index;
```

```
     FIN (wrls_phy_mcarrier_tx_init (<args>));

     mcarrier_tx_ptr =  (WrlsT_Phy_Mcarrier_Tx_Mgmt *)
op_prg_cmo_alloc (wrls_phy_catmem_handle, sizeof
(WrlsT_Phy_Mcarrier_Tx_Mgmt));

     ratx_mod_objid = op_topo_assoc (mac_objid,
     OPC_TOPO_ASSOC_OUT, OPC_OBJTYPE_RATX, 0);

     op_ima_obj_attr_get (ratx_mod_objid, "channel",
&tx_chnl_cmpd_id);

     num_tx_chnls = op_topo_child_count (tx_chnl_cmpd_id,
OPC_OBJTYPE_RATXCH);

     mcarrier_tx_ptr->conduit_count = num_tx_chnls;

     mcarrier_tx_ptr-
>start_conduit_index[WrlsC_Txch_Pool_0] = 0;

     if (second_pool_start_index > 0 &&
second_pool_start_index < num_tx_chnls)
          {
          mcarrier_tx_ptr-
>start_conduit_index[WrlsC_Txch_Pool_1] =
second_pool_start_index;
          }
     else
          {
          mcarrier_tx_ptr-
>start_conduit_index[WrlsC_Txch_Pool_1] = num_tx_chnls;
          }

     for (pool_index = WrlsC_Txch_Pool_0; pool_index <
WrlsC_Txch_Pools_Count; pool_index++)
          mcarrier_tx_ptr->next_conduit_index [pool_index]=
mcarrier_tx_ptr->start_conduit_index[pool_index];

     mcarrier_tx_ptr->conduit_idle_time_array = (double *)
op_prg_mem_alloc (num_tx_chnls * sizeof (double));

     mcarrier_tx_ptr->txch_objid_array = (Objid *)
op_prg_mem_alloc (num_tx_chnls * sizeof (Objid));

     for (txch_index = 0; txch_index < num_tx_chnls;
txch_index++)
          {
```

```
          mcarrier_tx_ptr->conduit_idle_time_array
[txch_index] = 0.0;
          mcarrier_tx_ptr->txch_objid_array[txch_index] =
op_topo_child (tx_chnl_cmpd_id, OPC_OBJTYPE_RATXCH,
txch_index);

          if (mcarrier_tx_ptr-
>start_conduit_index[WrlsC_Txch_Pool_1] !=
WRLSC_INVALID_TXCH_INDEX &&
               txch_index >= mcarrier_tx_ptr-
>start_conduit_index[WrlsC_Txch_Pool_1])
               op_radio_txch_rxgroup_set(mcarrier_tx_ptr-
>txch_objid_array[txch_index], 0, OPC_NIL);
          }

     if (op_prg_odb_ltrace_active ("wrls_phy_init"))
          {
          char msg [256], msg1[256];
          sprintf (msg, "Initializing Mcarrier Mgmt Object
for MAC["OPC_OBJ_ID_FMT"] with #counduit[%d]", mac_objid,
mcarrier_tx_ptr->conduit_count);
          sprintf (msg1, "Pool 0: start_index[%d], Next
Index[%d]; Pool 1: startindex [%d], Next Index[%d]",
               mcarrier_tx_ptr-
>start_conduit_index[WrlsC_Txch_Pool_0], mcarrier_tx_ptr-
>next_conduit_index[WrlsC_Txch_Pool_0],
               mcarrier_tx_ptr-
>start_conduit_index[WrlsC_Txch_Pool_1], mcarrier_tx_ptr-
>next_conduit_index[WrlsC_Txch_Pool_1]);
          op_prg_odb_print_major (msg, msg1, OPC_NIL);
          }

     if
(!wrls_burst_alloc_info_field_pool_memory_ready_flag)
          {
          wrls_burst_alloc_info_field_pmh =
op_prg_pmo_define ("Multicarrier burst allocation
information", sizeof (WrlsT_Phy_Mcarrier_Burst_Info),
1024);


     wrls_burst_alloc_info_field_pool_memory_ready_flag =
OPC_TRUE;
          }

     FRET (mcarrier_tx_ptr);
     }
```

```
void
wrls_phy_tx_power_stat_write (WrlsT_Transmitter_Info*
tx_info_ptr)
     {
     FIN (wrls_phy_tx_power_stat_write ());

     op_stat_write (tx_info_ptr->wrls_phy_info_ptr-
>subchnl_tx_power_stathdl,
wrls_phy_convert_watts_to_dbm(tx_info_ptr-
>wrls_phy_info_ptr->subchnl_tx_pwr_watts));
     op_stat_write (tx_info_ptr->wrls_phy_info_ptr-
>total_tx_power_stathdl,
wrls_phy_convert_watts_to_dbm(tx_info_ptr-
>wrls_phy_info_ptr->total_tx_power_watts));

     FOUT;
     }

void
wrls_phy_power_consumption_from_burst_update
(WrlsT_Phy_Chnl_Info* phy_info_ptr,
WrlsT_Phy_Mcarrier_Burst_Info* burst_alloc_info_ptr)
     {

     FIN (wrls_phy_power_consumption_from_burst_update ());

     phy_info_ptr-
>tx_report_info.accumulated_power_consumption_watts_sec +=
          (burst_alloc_info_ptr->subchnl_tx_pwr_watts *
burst_alloc_info_ptr->subchnl_count) *
burst_alloc_info_ptr->tx_delay;

     phy_info_ptr-
>tx_report_info.accumulated_power_consumption_sec +=
burst_alloc_info_ptr->tx_delay;

     FOUT;
     }

void
wrls_phy_support_burst_info_print
(WrlsT_Phy_Mcarrier_Burst_Info* burst_ptr)
     {
     char msg1[256], msg2[256];
     FIN (wrls_phy_support_burst_info_print (burst_ptr));

     sprintf (msg1, "Subchannels: Start[%d] Count[%d]
Pwr[%f]W -- Time: Start[%f] Delay[%f] ",
```

```
          burst_ptr->subchnl_start, burst_ptr-
>subchnl_count,
          burst_ptr->subchnl_tx_pwr_watts, burst_ptr-
>start_time,
          burst_ptr->tx_delay);

     sprintf (msg2, "Phy TYpe[%d], Mod[%d], Cod[%d], Data
Size[%f]bytes",
          burst_ptr->phy_type, burst_ptr->modulation_index,
          burst_ptr->coding_index, burst_ptr-
>data_size_bytes);

     op_prg_odb_print_minor (msg1, msg2, OPC_NIL);

     FOUT;
     }


double
wrls_phy_power_erceg_pathloss_compute (double distance,
double lambda, double freq_hz, double mobile_height_meters,
double base_station_height_meters, int terrain_type)
     {
     double pathloss;
     double a_term;
     double gamma;
     double delta_pl_f, delta_pl_h;
     double a, b, c;
     static double constant_a_values[3] = {4.6, 4.0, 3.6};
     static double constant_b_values[3] = {0.0075, 0.0065,
0.005};
     static double constant_c_values[3] = {12.6, 17.1,
20.0};

     FIN (wrls_phy_power_erceg_pathloss_compute (distance,
lambda, freq_hz, mobile_height_meters, base_height_meters,
terrain_type));

     a = constant_a_values [terrain_type];
     b = constant_b_values [terrain_type];
     c = constant_c_values [terrain_type];

     if (base_station_height_meters < 10.0)
          base_station_height_meters = 10.0;
     else if (base_station_height_meters > 80.0)
          base_station_height_meters = 80.0;

     if (mobile_height_meters < 2.0)
```

```
          mobile_height_meters = 2.0;
     else if (mobile_height_meters > 10.0)
          mobile_height_meters = 10.0;

     a_term = 20.0 * log10 ((WRLSC_FOUR_PI *
100.0)/lambda);

     gamma = (a - (b * base_station_height_meters) + (c /
base_station_height_meters));

     pathloss = a_term + (10.0 * gamma * log10
(distance/100.0));

     delta_pl_f = 6.0 * log10 (freq_hz/(2000.0*1e6));

     if (terrain_type == WrlsC_Pathloss_Terrain_C)
          {
          delta_pl_h = -20.0 * log10
(mobile_height_meters/2.0);
          }
     else
          {
          delta_pl_h = -10.8 * log10
(mobile_height_meters/2.0);
          }

     pathloss = pathloss + delta_pl_f + delta_pl_h;

     FRET (pathloss);
     }

double
wrls_phy_power_pedestrian_pathloss_compute (double
distance, double freq_hz)
     {
     double pathloss;
     double free_space_pathloss;

     FIN (wrls_phy_power_pedestrian_pathloss_compute
(distance, freq_hz));

     pathloss = 40.0 * log10 (distance/1000.0) + 30.0 *
log10 (wrls_phy_convert_hz_to_mhz(freq_hz)) + 49.0;

     free_space_pathloss =
wrls_phy_free_space_pl_compute(wrls_phy_convert_hz_to_mhz(f
req_hz),distance/1000.0);
```

```
     if (pathloss < free_space_pathloss)
          pathloss = free_space_pathloss;

     FRET (pathloss);
     }

double
wrls_phy_power_vehicular_pathloss_compute (double distance,
double freq_hz, double base_station_height_meters)
     {
     double pathloss;

     FIN (wrls_phy_power_vehicular_pathloss_compute
(distance, freq_hz, base_station_height_meters));

     if (base_station_height_meters <= 0.0)
          base_station_height_meters = 1.0;
     else if (base_station_height_meters > 50.0)
          base_station_height_meters = 50.0;

     pathloss = 40.0 * (1.0 - (0.004 *
base_station_height_meters)) * log10 (distance/1000.0)
                         - 18.0 * log10
(base_station_height_meters)
                         + 21.0 * log10
(wrls_phy_convert_hz_to_mhz(freq_hz)) + 80.0;

     FRET (pathloss);
     }

double
wrls_phy_indoor_office_pathloss_compute (double distance,
double freq_hz, int number_of_floors)
     {
     double pathloss;

     FIN (wrls_phy_power_pedestrian_pathloss_compute
(distance, freq));

     pathloss = 37.0 + 30.0 * log10 (distance/1000.0) +
18.3 * pow(number_of_floors, (((number_of_floors + 2) /
(number_of_floors + 1)) - 0.46));

     FRET (pathloss);
     }

double
wrls_phy_power_urban_macrocell_pathloss_compute (double
```

```
distance, double freq, double base_station_height_meters,
double mobile_station_height_meters)
     {
     double pathloss;
     double tmp;

     FIN (wrls_phy_power_urban_macrocell_pathloss_compute
(distance, freq, base_station_height_meters,
mobile_station_height_meters));

     if (base_station_height_meters <
mobile_station_height_meters)
          {
          tmp = base_station_height_meters;
          base_station_height_meters =
mobile_station_height_meters;
          mobile_station_height_meters = tmp;
          }

     if (base_station_height_meters < 30.0)
          base_station_height_meters = 30.0;
     else if (base_station_height_meters > 200.0)
          base_station_height_meters = 200.0;

     if (mobile_station_height_meters < 1.0)
          mobile_station_height_meters = 1.0;
     else if (mobile_station_height_meters > 10.0)
          mobile_station_height_meters = 10.0;

     freq = freq / 1000000.0;

     pathloss = (44.9 - 6.55 * log10
(base_station_height_meters)) * log10 (distance / 1000.0) +
45.5 +
                    ((35.46 - (1.1 *
mobile_station_height_meters)) * log10 (freq)) -
                    (13.82 *
log10(base_station_height_meters)) +
                    (0.7 * mobile_station_height_meters) +
3.0;

     FRET (pathloss);
     }

double
wrls_phy_power_suburban_macrocell_pathloss_compute (double
distance, double freq, double base_station_height_meters,
double mobile_station_height_meters)
```

```
     {
     double pathloss;
     double tmp;

     FIN
(wrls_phy_power_suburban_macrocell_pathloss_compute
(distance, freq, base_station_height_meters,
mobile_station_height_meters));

     if (base_station_height_meters <
mobile_station_height_meters)
          {
          tmp = base_station_height_meters;
          base_station_height_meters =
mobile_station_height_meters;
          mobile_station_height_meters = tmp;
          }

     if (base_station_height_meters < 30.0)
          base_station_height_meters = 30.0;
     else if (base_station_height_meters > 200.0)
          base_station_height_meters = 200.0;

     if (mobile_station_height_meters < 1.0)
          mobile_station_height_meters = 1.0;
     else if (mobile_station_height_meters > 10.0)
          mobile_station_height_meters = 10.0;

     freq = freq / 1000000.0;

     pathloss = (44.9 - 6.55 * log10
(base_station_height_meters)) * log10 (distance / 1000.0) +
45.5 +
                    ((35.46 - (1.1 *
mobile_station_height_meters)) * log10 (freq)) -
                    (13.82 *
log10(base_station_height_meters)) +
                    (0.7 * mobile_station_height_meters);

     FRET (pathloss);
     }

double
wrls_phy_power_urban_microcell_pathloss_compute (double
distance, double freq)
     {
     double pathloss;
```

```
     FIN (wrls_phy_power_urban_microcell_pathloss_compute
(distance, freq));


     FRET (pathloss);
     }


double
wrls_phy_pathloss_compute (Packet* pkptr, double
prop_distance_meters, WrlsT_Rx_State_Info* rx_state_ptr,
WrlsT_Phy_Chnl_Info*     tx_wrls_chn_ptr, Boolean
is_jammer)
     {
     double                        pathloss;
     double                        tx_center_freq_hz;
     double                        min_freq, bandwidth;
     double                        lambda;
     WrlsT_Pathloss_Info*     pl_info_ptr;
     WrlsT_Phy_Chnl_Info*     phy_chnl_info_ptr;
     WrlsT_Phy_Chnl_Info*     ss_phy_chnl_info_ptr;
     double                        ut_height_meters,
bs_height_meters;
     int                           pathloss_type,
terrain_type;
     Stathandle*                   pathloss_stathdl_ptr;
     double                        free_space_pathloss;


     FIN (wrls_phy_pathloss_compute (pkptr,
prop_distance_meters, rx_state_ptr, tx_wrls_chn_ptr));


     if(is_jammer == OPC_FALSE)
          {

          if(pkptr !=OPC_NIL)
               op_pk_fd_access_read_only_ptr (pkptr,
phy_info_field_index(pkptr), (const
void**)&phy_chnl_info_ptr);
          else
               phy_chnl_info_ptr = tx_wrls_chn_ptr;


          if (phy_chnl_info_ptr->comm_direction ==
WrlsC_Downlink ||
               phy_chnl_info_ptr->comm_direction ==
WrlsC_Adhoc)
```

```
               {
               ut_height_meters = rx_state_ptr-
>wrls_phy_info_ptr->height;
               ss_phy_chnl_info_ptr = rx_state_ptr-
>wrls_phy_info_ptr;
               pl_info_ptr = rx_state_ptr-
>wrls_phy_info_ptr->pathloss_info_ptr;
               bs_height_meters = phy_chnl_info_ptr-
>height;
               pathloss_stathdl_ptr = &(rx_state_ptr-
>wrls_phy_info_ptr->pathloss_stathdl);
               }
          else
               {
               bs_height_meters = rx_state_ptr-
>wrls_phy_info_ptr->height;
               pl_info_ptr = phy_chnl_info_ptr-
>pathloss_info_ptr;
               ut_height_meters = phy_chnl_info_ptr-
>height;
               ss_phy_chnl_info_ptr = phy_chnl_info_ptr;

               pathloss_stathdl_ptr = &(phy_chnl_info_ptr-
>pathloss_stathdl);
               }

          if (phy_chnl_info_ptr->phy_prof_ptr != OPC_NIL)
               tx_center_freq_hz = phy_chnl_info_ptr-
>phy_prof_ptr->base_freq;
          else if (pkptr != OPC_NIL)
               {
               tx_center_freq_hz   = op_td_get_dbl (pkptr,
OPC_TDA_RA_TX_FREQ) + (op_td_get_dbl (pkptr,
OPC_TDA_RA_TX_BW) / 2.0);
               }
          else
               op_sim_end ("Wireless PHY Error.",
"wrls_phy_pathloss_compute: No center frequency is provided
to compute pathloss.", "Simulation will terminate.", "");

          pathloss_type = pl_info_ptr->model_id;

          terrain_type  = pl_info_ptr-
>model_info.terrain_type;
          }
     else
          {
          pl_info_ptr    = rx_state_ptr->wrls_phy_info_ptr-
```

```
>pathloss_info_ptr;

          if (pl_info_ptr == OPC_NIL)
               op_sim_end ("Pathloss Information not 
available.",
                    "Make sure the intended receiver has 
pathloss related information available to accurately 
compute the received power in wrls_power pipeline stage", 
OPC_NIL, OPC_NIL);

          if (rx_state_ptr->mac_role == WrlsC_Client_Role)
               {
               ut_height_meters              = 
rx_state_ptr->wrls_phy_info_ptr->height;
               bs_height_meters           = op_td_get_dbl 
(pkptr, OPC_TDA_RA_TX_ALT);
               }
          else
               {
               bs_height_meters              = 
rx_state_ptr->wrls_phy_info_ptr->height;
               ut_height_meters           = op_td_get_dbl 
(pkptr, OPC_TDA_RA_TX_ALT);
               }

          min_freq            = op_td_get_dbl (pkptr, 
OPC_TDA_RA_TX_FREQ);
          bandwidth           = op_td_get_dbl (pkptr, 
OPC_TDA_RA_TX_BW);

          tx_center_freq_hz = min_freq + bandwidth/2.0;

          }
     switch (pathloss_type)
          {
          case WrlsC_Pathloss_Free_Space:
               pathloss = 
wrls_phy_free_space_pl_compute(wrls_phy_convert_hz_to_mhz(t
x_center_freq_hz),prop_distance_meters/1000.0);
               break;
          case WrlsC_Pathloss_Erceg:

               lambda = WRLSC_C / tx_center_freq_hz;

               pathloss = 
wrls_phy_power_erceg_pathloss_compute 
(prop_distance_meters, lambda, tx_center_freq_hz,
                              ut_height_meters,
```

```
bs_height_meters, terrain_type);
               break;

          case WrlsC_Pathloss_Pedestrian:

               pathloss =
wrls_phy_power_pedestrian_pathloss_compute
(prop_distance_meters, tx_center_freq_hz);
               break;

          case WrlsC_Pathloss_Vehicular:

               pathloss =
wrls_phy_power_vehicular_pathloss_compute
(prop_distance_meters, tx_center_freq_hz,
bs_height_meters);
               break;

          case WrlsC_Suburban_Macrocell:
               pathloss =
wrls_phy_power_suburban_macrocell_pathloss_compute
(prop_distance_meters, tx_center_freq_hz,
                              bs_height_meters,
ut_height_meters);
               break;

          case WrlsC_Urban_Macrocell:
               pathloss =
wrls_phy_power_urban_macrocell_pathloss_compute
(prop_distance_meters, tx_center_freq_hz,
                              bs_height_meters,
ut_height_meters);
               break;

          case WrlsC_Urban_Microcell:
               pathloss =
wrls_phy_power_urban_microcell_pathloss_compute
(prop_distance_meters, tx_center_freq_hz);
               break;

          case WrlsC_Pathloss_Indoor_Office:
               pathloss =
wrls_phy_indoor_office_pathloss_compute
(prop_distance_meters, tx_center_freq_hz,
                                   pl_info_ptr-
>model_info.number_of_floors);
               break;
          case WrlsC_InH_ITU_M2335:
```

```
                pathloss = wrls_phy_InH_pathloss_compute
(prop_distance_meters, tx_center_freq_hz,
                                    pl_info_ptr-
>model_info.sight_mode);
                break;

            case WrlsC_UMi_ITU_M2335:
                pathloss =
wrls_phy_UMi_outdoor_pathloss_compute
(prop_distance_meters, tx_center_freq_hz,
                                    bs_height_meters,
ut_height_meters, pl_info_ptr->model_info.sight_mode);
                break;

            case WrlsC_UMi_OtoI_ITU_M2335:
                pathloss =
wrls_phy_UMi_OtoI_pathloss_compute (prop_distance_meters,
tx_center_freq_hz,
                                    bs_height_meters,
ut_height_meters, pl_info_ptr->model_info.sight_mode,
                                    pl_info_ptr-
>model_info.wall_to_ut_dist_m);
                break;

            case WrlsC_UMa_ITU_M2335:
                pathloss = wrls_phy_UMa_pathloss_compute
(prop_distance_meters, tx_center_freq_hz,
                                    bs_height_meters,
ut_height_meters,   pl_info_ptr-
>model_info.avg_building_height_m,
                                    pl_info_ptr-
>model_info.street_width_m, pl_info_ptr-
>model_info.sight_mode);
                break;

            case WrlsC_SMa_ITU_M2335:
                pathloss = wrls_phy_SMa_pathloss_compute
(prop_distance_meters, tx_center_freq_hz,
                                    bs_height_meters,
ut_height_meters,   pl_info_ptr-
>model_info.avg_building_height_m,
                                    pl_info_ptr-
>model_info.street_width_m, pl_info_ptr-
>model_info.sight_mode);
                break;

            case WrlsC_RMa_ITU_M2335:
                pathloss = wrls_phy_RMa_pathloss_compute
```

```
(prop_distance_meters, tx_center_freq_hz,
                                   bs_height_meters,
ut_height_meters,   pl_info_ptr-
>model_info.avg_building_height_m,
                                   pl_info_ptr-
>model_info.street_width_m, pl_info_ptr-
>model_info.sight_mode);
               break;

          case WrlsC_Urban_Hata_Ext:
               pathloss =
wrls_phy_hata_ext_urban_pathloss_compute
(prop_distance_meters, tx_center_freq_hz,
                                   bs_height_meters,
ut_height_meters);

               break;

          case WrlsC_Suburban_Rural_Hata_Ext:
               pathloss =
wrls_phy_hata_ext_suburban_rural_pathloss_compute
(prop_distance_meters, tx_center_freq_hz,
                                   bs_height_meters,
ut_height_meters);
               break;
          default:
               op_sim_end ("Wireless PHY Package
Error:","Unknown pathloss model specified by the
developer.", "Review any customization of the pathloss
models.","");
          }

     if (pathloss_type != WrlsC_Pathloss_Free_Space)
          {

          free_space_pathloss =
wrls_phy_free_space_pl_compute(wrls_phy_convert_hz_to_mhz(t
x_center_freq_hz),prop_distance_meters/1000.0);

          if (pathloss < free_space_pathloss)
               pathloss = free_space_pathloss;
          }


     wrls_phy_pathloss_shadow_fading_refresh
((void*)pl_info_ptr);
```

```
     pathloss +=
wrls_phy_pathloss_shadow_fading_get(pl_info_ptr);

     if((pkptr !=OPC_NIL) && (is_jammer == OPC_FALSE))
          if (op_td_get_int (pkptr,
OPC_TDA_RA_MATCH_STATUS) == OPC_TDA_RA_MATCH_VALID)
               /* Write the pathloss statistic measure.
          */
               op_stat_write ((*pathloss_stathdl_ptr),
(double) pathloss);

     FRET (pathloss);
     }

void wrls_phy_cochnl_permutation_data_init(void)
     {


     FIN (wrls_phy_cochnl_permutation_data_init ());


     FOUT;
     }

Boolean
wrls_phy_support_measurement_by_type_read
(WrlsT_Measurement_Type type, int key, WrlsT_Rx_State_Info*
rx_info_ptr, double* value_ptr)
     {
     WrlsT_Measurement_Entity*     m_entity_ptr = OPC_NIL;
     PrgT_Bin_Hash_Table*          hash_ptr   = OPC_NIL;
     Boolean                            is_read         =
OPC_FALSE;

     FIN (wrls_phy_support_measurement_by_type_read (type,
key, rx_info_ptr, value_ptr));

     hash_ptr = rx_info_ptr->measurement_module_ptr-
>channel_qmeasure_set [type];
     m_entity_ptr = (WrlsT_Measurement_Entity*)
prg_bin_hash_table_item_get (hash_ptr, (int*) &key);

     is_read = wrls_phy_support_measurement_read_db
(m_entity_ptr, value_ptr);

     FRET (is_read);
    }
```

```
WrlsT_Notification_State*
wrls_phy_support_notification_state_get
(WrlsT_Measurement_Entity * m_entity_ptr, int
sub_client_id, WrlsT_Measurement_Threshold_Direction
threshold_direction)
     {
     List * notify_state_lptr = OPC_NIL;
     WrlsT_Notification_State*          notify_state_ptr =
OPC_NIL;
     int index;
     int num_entries;

     FIN (wrls_phy_support_notification_state_get
(m_entity_ptr, sub_client_id, threshold_direction));

     notify_state_lptr = m_entity_ptr->notify_state_lptr;

     if (notify_state_lptr == OPC_NIL)
          FRET (OPC_NIL);

     num_entries = op_prg_list_size (notify_state_lptr);

     for (index = 0; index < num_entries; index++)
          {
          notify_state_ptr = (WrlsT_Notification_State*)
op_prg_list_access (notify_state_lptr, index);
          if (notify_state_ptr->sub_client_id ==
sub_client_id && notify_state_ptr->threshold_direction ==
threshold_direction)
               FRET (notify_state_ptr);
          }

     FRET (OPC_NIL);
     }
          upd_s_table();


void
wrls_phy_support_measurement_by_type_notification_set
(WrlsT_Measurement_Type type, int key, WrlsT_Rx_State_Info*
rx_info_ptr, Boolean is_enabled, int sub_client_id,
WrlsT_Measurement_Threshold_Direction threshold_direction)
     {
     WrlsT_Measurement_Entity*          m_entity_ptr   =
OPC_NIL;
     PrgT_Bin_Hash_Table*               hash_ptr       =
OPC_NIL;
     WrlsT_Notification_State*          n_state_ptr    =
```

```
OPC_NIL;



     FIN
(wrls_phy_support_measurement_by_type_notification_set
(type, key, rx_info_ptr, is_enabled, sub_client_id,
threshold_direction));

     hash_ptr = rx_info_ptr->measurement_module_ptr-
>channel_qmeasure_set [type];
     m_entity_ptr = (WrlsT_Measurement_Entity*)
prg_bin_hash_table_item_get (hash_ptr, (int*) &key);

     if (OPC_NIL == m_entity_ptr)
          {
          op_sim_error (OPC_SIM_ERROR_WARNING, "Warning
from the function
wrls_phy_support_measurement_by_type_notification_set",
               "Measurement entity not found for the
specified key...make sure a measurement entity exists
before attempting to enable/disable notifications");
          FOUT;
          }

     n_state_ptr = wrls_phy_support_notification_state_get
(m_entity_ptr, sub_client_id, threshold_direction);

     if (n_state_ptr == OPC_NIL)
          {
          op_sim_error (OPC_SIM_ERROR_WARNING, "Warning
from the function
wrls_phy_support_measurement_by_type_notification_set",
               "Notification entity not found for the
specified key...make sure a notification entity exists
before attempting to enable/disable notifications");
          FOUT;
          }

     if (OPC_FALSE == is_enabled)
          op_ev_cancel_if_valid (n_state_ptr->notify_evh);

     n_state_ptr->is_enabled = is_enabled;

     FOUT;
     }

void
```

```
wrls_phy_support_measurement_state_set (int key,
WrlsT_Rx_State_Info* rx_info_ptr, Boolean action)
     {
     WrlsT_Measurement_Entity*     m_entity_ptr = OPC_NIL;
     PrgT_Bin_Hash_Table*          hash_ptr   = OPC_NIL;
     int
     measurement_type_count;

     FIN (wrls_phy_support_measurement_state_set (key,
rx_info_ptr, action));

     for (measurement_type_count = 0;
measurement_type_count < WrlsC_Number_of_Measurements;
measurement_type_count++)
          {
          hash_ptr = rx_info_ptr->measurement_module_ptr-
>channel_qmeasure_set [measurement_type_count];
          m_entity_ptr = (WrlsT_Measurement_Entity*)
prg_bin_hash_table_item_get (hash_ptr, (int*) &key);
          m_entity_ptr->measure_state_ptr->active = action;
          }

     FOUT;
     }

WrlsT_Measurement_State*
wrls_phy_support_measurement_state_create_alpha (double
alpha)
     {
     WrlsT_Measurement_State* m_state_ptr = OPC_NIL;


     FIN (wrls_phy_support_measurement_state_create_alpha
(alpha));

     m_state_ptr = (WrlsT_Measurement_State*)
op_prg_cmo_alloc (wrls_phy_catmem_handle, sizeof
(WrlsT_Measurement_State));

     m_state_ptr->avg         = (double) 0.0;
     m_state_ptr->sq_avg      = (double) 0.0;
     m_state_ptr->stdev       = (double) 0.0;
     m_state_ptr->alpha      = alpha;

     m_state_ptr->last_timestamp = WRLSC_TIMESTAMP_INVALID;

     m_state_ptr->active = ACTIVATE_STATE;
```

```
	FRET (m_state_ptr);
	}

void
wrls_phy_support_cell_global_measurements_register
(WrlsT_Rx_State_Info* rx_state_info_ptr, double alpha)
	{
	WrlsT_Measurement_State*		measure_state_ptr
	= OPC_NIL;
	int						i;

	FIN
(wrls_phy_support_cell_global_measurements_register
(rx_state_info_ptr, alpha));

	for (i = WrlsC_NI; i <
WrlsC_Number_of_Global_Measurements; i++)
		{
		measure_state_ptr = (WrlsT_Measurement_State*)
wrls_phy_support_measurement_state_create_alpha (alpha);

		rx_state_info_ptr->measurement_module_ptr-
>global_measurement_entity[i] = (WrlsT_Measurement_Entity*)
wrls_support_measurement_entity_create (measure_state_ptr);

		if (op_prg_odb_ltrace_active ("wrls_measure"))
			{
			char msg [256];

			sprintf (msg, "Initializing Global Measure
[%d]",i);
			op_prg_odb_print_major ("*** NEW Global
measurement element.", msg, OPC_NIL);
			}
		}

	FOUT;
	}

Boolean
wrls_phy_support_global_measurement_by_type_read
(WrlsT_Global_Measurement_Type type, WrlsT_Rx_State_Info*
rx_info_ptr, double* value_ptr)
	{
	Boolean				is_read		=
OPC_FALSE;

	FIN (wrls_phy_support_measurement_by_type_read (type,
```

```
rx_info_ptr, value_ptr));

     is_read = wrls_phy_support_measurement_read_db
(rx_info_ptr->measurement_module_ptr-
>global_measurement_entity[type], value_ptr);

     FRET (is_read);
    }

void
wrls_phy_support_measurement_deregister
(WrlsT_Rx_State_Info* rx_state_info_ptr, int key,
WrlsT_Measurement_Type type)
     {
     PrgT_Bin_Hash_Table*               hash_table_ptr;
     WrlsT_Measurement_Entity*          m_entity_ptr
     = OPC_NIL;


     FIN (wrls_phy_support_measurement_deregister ());

     hash_table_ptr = rx_state_info_ptr-
>measurement_module_ptr->channel_qmeasure_set[type];

     m_entity_ptr = (WrlsT_Measurement_Entity*)
prg_bin_hash_table_item_remove (hash_table_ptr, &key);

     if(m_entity_ptr == OPC_NIL)
          {
          FOUT;
          }

     if (m_entity_ptr->measure_state_ptr)
          {
          if (m_entity_ptr->measure_state_ptr-
>measured_values_lptr)
               {
               op_prg_list_free (m_entity_ptr-
>measure_state_ptr->measured_values_lptr);
               op_prg_mem_free (m_entity_ptr-
>measure_state_ptr->measured_values_lptr);
               }

          op_prg_mem_free (m_entity_ptr-
>measure_state_ptr);
          }

     if (m_entity_ptr->notify_state_lptr)
```

```
          {
          op_prg_list_free (m_entity_ptr-
>notify_state_lptr);
          op_prg_mem_free (m_entity_ptr-
>notify_state_lptr);
          }

     op_prg_mem_free (m_entity_ptr);


     FOUT;
     }
          upd_s_table();

void
wrls_phy_support_measurement_register (WrlsT_Rx_State_Info*
rx_state_info_ptr, int key, double measure_window_size,
double measure_bucket,
     WrlsT_Measurement_Type type,
WrlsT_Measurement_Record_Proc_Ptr measure_proc_ptr)
     {
     WrlsT_Measurement_Entity*          m_entity_ptr
     = OPC_NIL;
     WrlsT_Measurement_Entity*          tmp_entity_ptr
          = OPC_NIL;
     WrlsT_Measurement_State*           m_state_ptr
     = OPC_NIL;
     PrgT_Bin_Hash_Table*               hash_table_ptr;

     FIN (wrls_phy_support_measurement_register
(rx_state_info_ptr, key, measure_window_size,
measure_bucket, type, measure_proc_ptr));

     hash_table_ptr = rx_state_info_ptr-
>measurement_module_ptr->channel_qmeasure_set[type];

     if (OPC_NIL == hash_table_ptr)
          {
          if (op_prg_odb_ltrace_active ("wrls_measure"))
               {
               char           msg [256];
               char           type_str
[WrlsC_Number_of_Measurements][16] = {"RSSI", "CINR"};

               sprintf (msg, "For type %s, a hash table for
storing the measurement module was not found...no
measuremente will be recorded for %s",
                    type_str [type], type_str [type]);
```

```
                op_prg_odb_print_major (msg, OPC_NIL);

                }

           FOUT;
           }

      rx_state_info_ptr->measure_proc_ptr_arr [type] =
measure_proc_ptr;

      m_entity_ptr = (WrlsT_Measurement_Entity*)
prg_bin_hash_table_item_get (hash_table_ptr, &key);

      if (PRGC_NIL == m_entity_ptr)
           {
           m_state_ptr = (WrlsT_Measurement_State*)
wrls_support_measurement_state_create (key,
measure_window_size, measure_bucket);

           m_entity_ptr =
wrls_support_measurement_entity_create (m_state_ptr);

           m_entity_ptr->measure_type = type;

           if (type == WrlsC_RSRQ)
                {
                m_entity_ptr->measured_rsrq_stathdl  =
op_stat_reg ("PHY.Measured RSRQ (dB)",
OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);
                }
           if (op_prg_odb_ltrace_active ("wrls_measure"))
                {
                char                         msg [256];

                sprintf (msg, "Inserting measurement entity
[%p] in hash[%p] by key [%d]", m_entity_ptr,
hash_table_ptr, key);
                op_prg_odb_print_major ("NEW measurement
element.", msg, OPC_NIL);
                }

           prg_bin_hash_table_item_insert (hash_table_ptr,
&key, m_entity_ptr, (void**) &tmp_entity_ptr);
           }

      FOUT;
      }
```

```
WrlsT_Measurement_State*
wrls_support_measurement_state_create (int key, double
measure_window_size, double measure_bucket)
     {
     WrlsT_Measurement_State*           m_state_ptr;

     FIN (wrls_support_measurement_state_create (key,
measure_window_size, measure_bucket));

     m_state_ptr = (WrlsT_Measurement_State*)
op_prg_cmo_alloc (wrls_phy_catmem_handle, sizeof
(WrlsT_Measurement_State));

     m_state_ptr->key                            = key;

     m_state_ptr->avg                            =
(double) 0.0;
     m_state_ptr->sq_avg                         =
(double) 0.0;
     m_state_ptr->stdev                          =
(double) 0.0;
     m_state_ptr->bucket                         =
measure_bucket;
     m_state_ptr->alpha                          =
WRLSC_MEASURE_AVG_ALPHA_DEFAULT;

     m_state_ptr->window_size               =
measure_window_size;

     if (m_state_ptr->window_size !=
WRLSC_MEASURE_WNDW_SIZE_NONE)
          m_state_ptr->measured_values_lptr  =
op_prg_list_create ();
     else
          m_state_ptr->measured_values_lptr = OPC_NIL;

     /* At present, measurement module is active.
                    */
     m_state_ptr->active                         =
ACTIVATE_STATE;

     FRET (m_state_ptr);
     }

WrlsT_Measurement_Entity*
wrls_support_measurement_entity_create
(WrlsT_Measurement_State* m_state_ptr)
```

```
     {
     WrlsT_Measurement_Entity*              m_entity_ptr;

     FIN (wrls_support_measurement_entity_create
(m_state_ptr));

     m_entity_ptr = (WrlsT_Measurement_Entity*)
op_prg_cmo_alloc (wrls_phy_catmem_handle, sizeof
(WrlsT_Measurement_Entity));

     /* Intialize SNR             */
     m_entity_ptr->measured_snr_dim_stathdl = OPC_NIL;
     m_entity_ptr->measured_snr_dim_stat_attempted =
OPC_FALSE;

     m_entity_ptr->notify_state_lptr = op_prg_list_create
();

     m_entity_ptr->measure_state_ptr = m_state_ptr;

     FRET (m_entity_ptr);
     }

void
wrls_phy_support_measurement_dim_stat_reg
(WrlsT_Rx_State_Info* rx_state_ptr, int key, char*
stat_name, char* stat_annot_str, int max_limit)
     {
     PrgT_Bin_Hash_Table*               hash_ptr
     = OPC_NIL;
     WrlsT_Measurement_Entity*          m_entity_ptr
     = OPC_NIL;


FIN (wrls_phy_support_measurement_dim_stat_reg
(rx_state_ptr, key, stat_name, stat_key_str, max_limit))

     if (rx_state_ptr->dl_stat_count >= max_limit)
          FOUT;

     hash_ptr = rx_state_ptr->measurement_module_ptr-
>channel_qmeasure_set [WrlsC_SINR];

     m_entity_ptr = (WrlsT_Measurement_Entity*)
prg_bin_hash_table_item_get (hash_ptr, (int*) &(key));

     if (m_entity_ptr->measured_snr_dim_stat_attempted ==
OPC_FALSE)
```

```
          {
          m_entity_ptr->measured_snr_dim_stathdl =
Oms_Dim_Stat_Reg (rx_state_ptr->wrls_phy_info_ptr-
>mac_objid, "PHY",

                                   stat_name,
stat_annot_str, OPC_STAT_LOCAL);

          Oms_Dim_Stat_Annotate (m_entity_ptr-
>measured_snr_dim_stathdl, stat_annot_str);
          m_entity_ptr->measured_snr_dim_stat_attempted =
OPC_TRUE;
          rx_state_ptr->dl_stat_count++;
          }

     FOUT;
     }

void
wrls_phy_support_measurement_by_type_notification_threshold
_set (void* client_notify_vptr, WrlsT_Measurement_Type
type, int key, WrlsT_Rx_State_Info* rx_info_ptr, double
value,

                              int sub_client_id, const
char* sub_client_name,
WrlsT_Measurement_Threshold_Direction threshold_direction,
Objid remote_module_objid)
     {
     WrlsT_Measurement_Entity*     m_entity_ptr        =
OPC_NIL;
     WrlsT_Notification_State*     n_state_ptr;
     PrgT_Bin_Hash_Table*          hash_ptr            =
OPC_NIL;
     int                                list_size, i;
     Boolean                            n_state_found
     = OPC_FALSE;

          upd_s_table();

     FIN
(wrls_phy_support_measurement_by_type_notification_threshol
d_set (client_notify_vptr, type, key, rx_info_ptr, value,
sub_client_id, sub_client_name, threshold_direction,
remote_module_objid));


     hash_ptr = rx_info_ptr->measurement_module_ptr-
```

```
>channel_qmeasure_set [type];
     m_entity_ptr = (WrlsT_Measurement_Entity*)
prg_bin_hash_table_item_get (hash_ptr, (int*) &key);


     if (m_entity_ptr == OPC_NIL)
          {
          op_sim_error (OPC_SIM_ERROR_WARNING, "Warning
from the function
wrls_phy_support_measurement_by_type_notification_threshold
_set",
               "Measurement entity not found for the
specified key...make sure a measurement entity exists
before attempting to register notifications");

          FOUT;
          }

     if (op_prg_odb_ltrace_active ("wrls_measure"))
          {
          char                         msg [256];

          sprintf (msg, "M Type [%d]: Found measurement
entity [%p] in hash[%p] by key [%d]", type, m_entity_ptr,
hash_ptr, key);
          op_prg_odb_print_major (msg, OPC_NIL);
          }


     if ((list_size = op_prg_list_size (m_entity_ptr-
>notify_state_lptr)) > 0)
          {
          for (i = 0; i < list_size; i++)
               {
               n_state_ptr = (WrlsT_Notification_State*)
op_prg_list_access (m_entity_ptr->notify_state_lptr, i);

               if ((n_state_ptr->sub_client_id ==
sub_client_id) && (n_state_ptr->threshold_direction ==
threshold_direction))
                    {
                    n_state_found = OPC_TRUE;
                    break;
                    }
               }


     if (n_state_found == OPC_TRUE)
```

```
               {
               n_state_ptr->threshold
          = value;
               n_state_ptr->client_mod_objid
     = remote_module_objid;

               if (op_prg_odb_ltrace_active
("wrls_measure"))
                    {
                    char           msg [512];
                    char           threshold_str [2][8]
                                        = {"DOWN", "UP"};

                    sprintf (msg, "Notif state [%p] found
for type [%s] and threshold [%s], threshold set to [%lf]",
                         n_state_ptr, n_state_ptr-
>sub_client_name, threshold_str [threshold_direction],
value);
                    op_prg_odb_print_minor (msg, OPC_NIL);
                    }

               if (client_notify_vptr != OPC_NIL)
                    {


                    n_state_ptr->client_state_ptr
     = client_notify_vptr;
                    }

                    }
          }


if (n_state_found == OPC_FALSE)
          {
          n_state_ptr =
wrls_support_notification_state_create (client_notify_vptr,
               value, remote_module_objid, sub_client_id,
sub_client_name, threshold_direction, type, key);

          if (op_prg_odb_ltrace_active ("wrls_measure"))
               {
               char           msg [512];
               char           threshold_str [2][8]
                                   = {"DOWN", "UP"};

               sprintf (msg, "New notif state [%p] created
for sub client [%s] and threshold [%s] with threshold
```

```
[%lf]",
                    n_state_ptr, n_state_ptr-
>sub_client_name, threshold_str [threshold_direction],
value);
               op_prg_odb_print_minor (msg, OPC_NIL,
value);
               }

          op_prg_list_insert (m_entity_ptr-
>notify_state_lptr, n_state_ptr, OPC_LISTPOS_TAIL);
          }

     FOUT;
     }

WrlsT_Notification_State*
wrls_support_notification_state_create (void*
client_notify_vptr, double threshold, Objid
client_mod_objid,
                                                      int
sub_client_id, const char*    sub_client_name,

     WrlsT_Measurement_Threshold_Direction
threshold_direction,

     WrlsT_Measurement_Type measure_type, int key)
     {
     WrlsT_Notification_State*
     notify_state_ptr = OPC_NIL;


     FIN (wrls_support_notification_state_create
(client_notify_vptr, threshold, client_mod_objid,
sub_client_id, threshold_direction, measure_type, key));


notify_state_ptr = (WrlsT_Notification_State*)
op_prg_cmo_alloc (wrls_phy_catmem_handle, sizeof
(WrlsT_Notification_State));

          notify_state_ptr->client_state_ptr          =
client_notify_vptr;
     notify_state_ptr->threshold            = threshold;
     notify_state_ptr->is_enabled           = OPC_TRUE;
     notify_state_ptr->client_mod_objid     =
client_mod_objid;
     notify_state_ptr->threshold_direction   =
threshold_direction;
```

```
     notify_state_ptr->sub_client_id         =
sub_client_id;
     notify_state_ptr->sub_client_name       =
prg_string_copy (sub_client_name);
     notify_state_ptr->measure_type               =
measure_type;
     notify_state_ptr->key                        = key;

     FRET (notify_state_ptr);
     }

Boolean
wrls_phy_support_measurement_read_db
(WrlsT_Measurement_Entity* m_entity_ptr, double* value_ptr)
     {
     WrlsT_Measurement_State*                m_state_ptr =
OPC_NIL;


     FIN (wrls_phy_support_measurement_read_db
(m_entity_ptr));

     if (OPC_NIL == m_entity_ptr) FRET (OPC_FALSE);
     m_state_ptr = m_entity_ptr->measure_state_ptr;
     if (OPC_NIL == m_state_ptr) FRET (OPC_FALSE);
     if (OPC_DBL_NEG_INFINITY == m_state_ptr->avg_db) FRET
(OPC_FALSE);

     (*value_ptr) = (double)(m_state_ptr->avg_db);
     FRET (OPC_TRUE);
     }

void
wrls_phy_support_measurement_write
(WrlsT_Measurement_Entity* m_entity_ptr, double value_db)
     {
     WrlsT_Measurement_State*                m_state_ptr
               = OPC_NIL;
     WrlsT_Notification_State*
     n_state_ptr             = OPC_NIL;
     double
     value;
     List *
notify_state_lptr        = OPC_NIL;
     int
notify_state_index       = 0;
     int
num_notify_states        = 0;
```

```
     WrlsT_Measurement_Value*                head_entry;
     WrlsT_Measurement_Value*                tail_entry
                = OPC_NIL;
     int
     old_list_size, list_size;
     double
     stale_value                   = 0;
     int
     notif_code;
     double
     current_time;
     Boolean
     continue_looping;
     Boolean
     notif_ok                 = OPC_FALSE;
     int
     notif_intrpt_base;

     FIN (wrls_phy_support_measurement_write (m_entity_ptr,
value_db));

     value = (double) pow (10.0, value_db/10.0);

     m_state_ptr = m_entity_ptr->measure_state_ptr;

     current_time = op_sim_time ();

     if (m_state_ptr->measured_values_lptr != OPC_NIL)
          {
          list_size = old_list_size = op_prg_list_size
(m_state_ptr->measured_values_lptr);

          continue_looping = (op_prg_list_size
(m_state_ptr->measured_values_lptr) > 0);

          while (continue_looping)
               {
               tail_entry = (WrlsT_Measurement_Value*)
op_prg_list_access (m_state_ptr->measured_values_lptr,
OPC_LISTPOS_TAIL);


               if ((current_time - tail_entry->time) >
m_state_ptr->window_size)
                    {
                    op_prg_list_remove (m_state_ptr-
>measured_values_lptr, OPC_LISTPOS_TAIL);
                    stale_value += tail_entry->value;
```

```
                    op_prg_mem_free (tail_entry);
                    list_size -= 1;
                    continue_looping = (list_size > 0);

                    notif_ok = OPC_TRUE;
                    }
               else
                    continue_looping = OPC_FALSE;
               }
          }


     if (m_state_ptr->measured_values_lptr == OPC_NIL)
          {
          m_state_ptr->avg              = (double) value;
          m_state_ptr->sq_avg           = (double) value *
value;
          m_state_ptr->stdev            = (double)
(m_state_ptr->sq_avg - m_state_ptr->avg * m_state_ptr-
>avg);
          m_state_ptr->avg_db           = (double) (10.0 *
log10 (m_state_ptr->avg));


          notif_ok = OPC_TRUE;
          }

     else
          {
          head_entry                              =
(WrlsT_Measurement_Value*) op_prg_mem_alloc (sizeof
(WrlsT_Measurement_Value));
          head_entry->value             = value;
          head_entry->time              = current_time;
          op_prg_list_insert (m_state_ptr-
>measured_values_lptr, head_entry, OPC_LISTPOS_HEAD);

          list_size = op_prg_list_size (m_state_ptr-
>measured_values_lptr);

          m_state_ptr->avg              = (double)
(old_list_size * m_state_ptr->avg - stale_value +
value)/(double) (list_size);
          m_state_ptr->sq_avg           = (double)
(old_list_size * m_state_ptr->sq_avg - stale_value *
stale_value + value * value)/(double) (list_size);
          m_state_ptr->stdev            = (double)
(m_state_ptr->sq_avg - m_state_ptr->avg * m_state_ptr-
```

```
>avg);
          m_state_ptr->avg_db          = (double) (10.0 *
log10 (m_state_ptr->avg));
          }

     if (op_prg_odb_ltrace_active ("wrls_measure"))
          {
          char msg [512];

          sprintf (msg, "Measurement entity key [%d]
Latest value [%.2lf]dB  [%E]mW  [%.2lf] dBm; AVG [%.2lf]dB
[%E]mW",
               m_state_ptr->key, value_db, value,
m_state_ptr->avg_db ,
wrls_phy_convert_watts_to_dbm(m_state_ptr->avg),
m_state_ptr->avg);
          op_prg_odb_print_major (msg, OPC_NIL);
          }

     if (m_entity_ptr->measured_snr_dim_stathdl != OPC_NIL)
           Oms_Dim_Stat_Write (m_entity_ptr-
>measured_snr_dim_stathdl , m_state_ptr->avg_db);

     notify_state_lptr                  = m_entity_ptr-
>notify_state_lptr;
     num_notify_states                  = op_prg_list_size
(notify_state_lptr);

     for (notify_state_index = 0; notify_state_index <
num_notify_states; notify_state_index++)
          {
          n_state_ptr =  (WrlsT_Notification_State*)
op_prg_list_access (notify_state_lptr, notify_state_index);

          if (op_prg_odb_ltrace_active ("wrls_measure"))
               {
               char               msg [512];

               sprintf (msg, "Evaluating the statistic
against the threshold [%E] in direction [%s|%d]
Notification enabled [%s]",
                    n_state_ptr->threshold,
                    n_state_ptr->threshold_direction ==
WrlsC_Measurement_Threshold_Direction_Down ? "DOWN" : "UP",
                    n_state_ptr->threshold_direction,
                    n_state_ptr->is_enabled ? "Yes":"No");

               op_prg_odb_print_minor (msg, OPC_NIL);
```

```
			}

		if ((n_state_ptr->is_enabled) && (notif_ok) &&
			(((m_state_ptr->avg_db < n_state_ptr-
>threshold) && (n_state_ptr->threshold_direction ==
WrlsC_Measurement_Threshold_Direction_Down))
				|| ((m_state_ptr->avg_db > n_state_ptr-
>threshold) && (n_state_ptr->threshold_direction ==
WrlsC_Measurement_Threshold_Direction_Up))
				|| ((n_state_ptr->threshold_direction
== WrlsC_Alert_Threshold_Direction_Up) && (m_state_ptr->avg
> n_state_ptr->threshold))
				|| ((n_state_ptr->threshold_direction
== WrlsC_Alert_Threshold_Direction_Down) && (m_state_ptr-
>avg < n_state_ptr->threshold))
				))
			{

			if (m_entity_ptr->measure_type ==
WrlsC_RSRQ)
				{
				notif_intrpt_base =
WRLSC_MEASURE_RSRQ_THRESHOLD_CROSSED;
				}
			else
				{
				notif_intrpt_base =
WRLSC_MEASURE_NOTIF_THRESHOLD_CROSSED;
				}

			n_state_ptr->m_state_ptr = m_state_ptr;

			notif_code = (((m_state_ptr->key << 1) +
n_state_ptr->threshold_direction) <<
WRLSC_REMOTE_INTRPT_PURPOSE_BITS) + notif_intrpt_base;
			op_ev_state_install (n_state_ptr, OPC_NIL);
			n_state_ptr->notify_evh =
op_intrpt_schedule_remote (op_sim_time(), notif_code,
n_state_ptr->client_mod_objid);
			op_ev_state_install (OPC_NIL, OPC_NIL);

			if (op_prg_odb_ltrace_active
("wrls_measure"))
				{
				char msg [256];

				sprintf (msg, "\tNotification threshold
[%d] with interrupt code [%d]",
```

```
                         n_state_ptr->client_mod_objid,
notif_code);
                    op_prg_odb_print_minor (msg, OPC_NIL);
                    }
               }
          }

     FOUT;
     }

double
wrls_phy_support_measurement_read
(WrlsT_Measurement_Entity* m_entity_ptr, double start_time,
double measure_time, Boolean return_db)
     {
     WrlsT_Measurement_State*                m_state_ptr
          = m_entity_ptr->measure_state_ptr;
     double
     window_avg              = 0.0;
     int
     num_data_points     = 0;
     WrlsT_Measurement_Value*
     measure_value_ptr;
     PrgT_List_Cell*
     list_cell_ptr;

     FIN (wrls_phy_support_measurement_read (m_entity_ptr,
start_time, measure_time));

     if (m_state_ptr->measured_values_lptr == OPC_NIL)
          {
          if (return_db)
               {
               FRET (m_state_ptr->avg_db);
               }
          else
               {
               FRET (m_state_ptr->avg);
               }
          }
     else
          {
          list_cell_ptr = prg_list_head_cell_get
(m_state_ptr->measured_values_lptr);

          while (list_cell_ptr != PRGC_NIL)
               {
               measure_value_ptr = (WrlsT_Measurement_Value
```

```
*) prg_list_cell_data_get (list_cell_ptr);

               if ((measure_value_ptr->time <= start_time)
&& (measure_value_ptr->time >= (start_time -
measure_time)))
                    {
                    window_avg += measure_value_ptr->value;
                    num_data_points ++;

                    list_cell_ptr = prg_list_cell_next_get
(list_cell_ptr);
                    }
               else
                    list_cell_ptr = PRGC_NIL;
               }

          if (num_data_points == 0)
               FRET (0.0);

          window_avg = window_avg/num_data_points;

          if (return_db)
               {
               FRET ((double) (10.0 * log10 (window_avg)));
               }
          else
               {
               FRET (window_avg);
               }
          }
     }

void
wrls_phy_support_measurement_update
(WrlsT_Measurement_Entity* m_entity_ptr, double value)
     {
     WrlsT_Measurement_State*               m_state_ptr
          = m_entity_ptr->measure_state_ptr;
     WrlsT_Measurement_Value*
     measure_value_ptr;
     int
     list_size;

     FIN (wrls_phy_support_measurement_update
(wrls_phy_support_measurement_update (m_entity_ptr,
value)));

     if ((m_state_ptr->measured_values_lptr != OPC_NIL) &&
```

```
((list_size = op_prg_list_size (m_state_ptr-
>measured_values_lptr)) > 0))
          {
          measure_value_ptr = (WrlsT_Measurement_Value*)
op_prg_list_access (m_state_ptr->measured_values_lptr,
OPC_LISTPOS_HEAD);

          measure_value_ptr->value += value;

          m_state_ptr->avg              = (double)
(list_size * m_state_ptr->avg + value)/(double)
(list_size);
          m_state_ptr->sq_avg           = (double)
(list_size * m_state_ptr->sq_avg + value * value)/(double)
(list_size);
          m_state_ptr->stdev            = (double)
(m_state_ptr->sq_avg - m_state_ptr->avg * m_state_ptr-
>avg);
          m_state_ptr->avg_db           = (double) (10.0 *
log10 (m_state_ptr->avg));
          }

     FOUT;
     }

void
wrls_phy_support_bs_id_set    (WrlsT_Phy_Chnl_Info*
wrls_chnl_info_ptr, WrlsT_Cell_Id bs_id)
     {
     FIN (wrls_phy_support_bs_id_set (wrls_chnl_info_ptr,
bs_id));

     wrls_chnl_info_ptr->group_id = bs_id;

     FOUT;
     }

double
wrls_phy_subchnl_bkgnoise_compute_db (WrlsT_Phy_Chnl_Info*
wrls_chnl_info_ptr)
     {
     double         rx_noisefig, rx_temp, rx_bw;
     double         bkg_temp, bkg_noise, amb_noise;
     double         total_noise;
     double         noise_dbm;

     FIN (wrls_phy_subchnl_bkgnoise_compute_db
(wrls_chnl_info_ptr));
```

```
     rx_noisefig = pow (10.0, (WRLSC_NOISE_FIGURE/ 10.0));

     rx_temp = (rx_noisefig - 1.0) * WRLSC_BKG_TEMP;

     bkg_temp = WRLSC_BKG_TEMP;

     if (wrls_chnl_info_ptr->phy_type == WrlsC_OFDMA)
          {
          rx_bw = (wrls_chnl_info_ptr->bw_per_subcarrier);
          }
     else
          rx_bw = (wrls_chnl_info_ptr->total_bw);


     bkg_noise = (rx_temp + bkg_temp) * rx_bw *
WRLSC_BOLTZMANN;

     amb_noise = rx_bw * WRLSC_AMB_NOISE_LEVEL;

     total_noise = amb_noise + bkg_noise;
     noise_dbm =
wrls_phy_convert_watts_to_dbm(total_noise);

     FRET (noise_dbm);
     }

void
wrls_phy_rx_freq_bandwidth_set (Objid rx_objid, double
input_base_freq_hz, double input_bandwidth_hz)
     {
     Objid                         chnl_comp_objid;
     double                  base_frequency;
     double                  bandwidth;

          upd_s_table();
     FIN (wrls_phy_rx_freq_bandwidth_set (rx_objid,
input_base_freq_hz, input_bandwidth_hz));


     op_ima_obj_attr_get (rx_objid, "channel",
&chnl_comp_objid);

     wrls_phy_freq_bandwidth_all_channels_set
(chnl_comp_objid, base_frequency, bandwidth,
OPC_OBJTYPE_RARXCH);
```

```
      FOUT;
      }

void
wrls_phy_freq_bandwidth_all_channels_set (Objid
ch_comp_objid, double base_frequency, double bandwidth,
OpT_Obj_Type obj_type)
      {
      int num_channels;
      int ch_index;
      Objid ith_child_objid;

      for (ch_index = 0; ch_index < num_channels;
ch_index++)
           {
           ith_child_objid = op_topo_child (ch_comp_objid,
obj_type, ch_index);

           op_ima_obj_attr_set_dbl (ith_child_objid, "min
frequency", base_frequency);
           op_ima_obj_attr_set_dbl (ith_child_objid,
"bandwidth", bandwidth);
           }

      FOUT;
      }
```