Universidade de Aveiro Departamento de Electrónica, Telecomunicações e Informática
2008

David Guerra Vieira

Qualidade de Serviço em Redes IEEE 802.16 com Topologia em Malha

Universidade de Aveiro Departamento de Electrónica, Telecomunicações e Informática
2008

David Guerra Vieira

Qualidade de Serviço em Redes IEEE 802.16 com Topologia em Malha

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e de Telecomunicações, realizada sob a orientação científica da Drª. Susana Sargento, Professora Auxiliar, e do Dr. Francisco Fontes, Professor Auxiliar Convidado, ambos do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

o júri

presidente

Dr. Rui Valadas
Professor Associado do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

orientador

Dra. Susana Sargento
Professora Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

co-orientador

Dr. Francisco Fontes
Professor Auxiliar convidado do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

arguente

Dr. Paulo Mendes
Lider do Grupo *Internet Architectures and Networking* (IAN) do Instituto de Engenharia de Sistemas e Computadores (INESC) do Porto

agradecimentos

Aos orientadores do Mestrado Integrado, Professora Dra. Susana Sargento e Professor Dr. Francisco Fontes, pela sábia orientação, e por toda a disponibilidade e incentivo.

Ao colaborador, Mestre Pedro Neves por todos os conhecimentos e motivação transmitidos.

E à minha família pela compreensão, encorajamento e incondicional apoio.

Os meus sinceros agradecimentos.

palavras-chave

Mesh, QoS, WiMAX, IEEE 802.16, Redes sem fios, Redes de Acesso Metroplitanas

resumo

O IEEE 802.16 apresenta-se actualmente como a tecnologia mais avançada e aliciante para o acesso de banda larga metropolitano. A sua topologia *ponto-multiponto* (PMP) foi desenvolvida desde o início com suporte para qualidade de serviço (QoS) gerida pelo controlador ou operador da rede, podendo assim complementar, as actuais soluções móveis de terceira geração. Por outro lado, a topologia opcional “em malha” (*Mesh*) permite a criação de redes auto-configuráveis e com encaminhamento de tráfego através de vários pontos da rede. No entanto, as especificações e mecanismos de QoS apresentados na norma não são consistentes para estes dois modos de operação. Com a presente dissertação pretende-se estudar e avaliar uma arquitectura de QoS para o modo *Mesh*, baseada nos mecanismos delineados para a topologia PMP, permitindo a coexistência dos dois modos de operação. A arquitectura apresentada foca-se numa gestão eficiente da largura de banda utilizando mensagens de controlo ao nível MAC introduzidas pelo standard IEEE 802.16. Os resultados obtidos mostram a eficiência das classes de serviço implementadas, convergindo com os requisitos de QoS do modo PMP.

keywords Mesh, QoS, WiMAX, IEEE 802.16, Wireless Networks, Metroplitan Networks

abstract

The IEEE 802.16 standard is by now the most advanced and attractive technology for the metropolitan broadband access. The *point-to-multipoint* (PMP) topology was developed from the beginning with quality of service (QoS) support, managed by the network operator, thus complementing the existing third-generation mobile solutions. On the other hand, the alternative *Mesh* topology allows the creation of self-configuring networks with traffic routing through various nodes. However, the QoS specifications and mechanisms presented in the standard are not consistent for these two operation modes. The present work aims to study and evaluate a QoS architecture for the Mesh mode, based on mechanisms designed to PMP and thus allowing the coexistence of the two operation modes. The proposed architecture focuses on an efficient network bandwidth management, using control messages at the MAC level as suggested in the IEEE 802.16 standard. The results show the efficiency of the implemented service classes, coming to a convergence with the quality requirements announced by PMP mode.

CONTENTS

Index of Figures

Index of Tables

Abbreviations and acronyms

ADSL	Asynchronous Digital Subscriber Line
ANACOM	National Communications Authority (Portugal)
ARQ	Automatic Repeat Request
ATM	Asynchronous Transfer Mode
BE	Best Effort Service
BPSK	Binary Phase Shift Keying
BS	Base Station
BWA	Broadband Wireless Access
CBR	Constant Bit Rate
CEPT	Conference of European Postal & Telecommunications
CPE	Customer Premise Equipment
DFS	Dynamic Frequency Selection
DSA	Dynamic Service Addition
DSC	Dynamic Service Change
DSD	Dynamic Service Deletion
DSL	Digital Subscriber Line
DSSS	Direct-Sequence Spread Spectrum
DSx	Dynamic Service Messages
FDD	Frequency Division Duplexing
FEC	Forward Error Correction
FEBA	Fair End-To-End Bandwidth Access
FHSS	Frequency-Hopping Spread Spectrum
FWA	Fixed Wireless Access
GUI	Graphical User Interface
IE	Information Element
IEEE	Institute of Electrical and Electronics Engineers
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
LMDS	Local Multipoint Distribution System
LOS	Line Of Sight
MAC	Media Access Control
MCS	Media Convergence Sublayer
MSH-CSCH	Mesh Centralized Schedule

MSH-CSCF	Mesh Centralized Schedule Configuration
MSH-DSCH	Mesh Distributed Schedule
MSH-NCFG	Mesh Network Configuration
MSH-NENT	Mesh Network Entry
NENT	Network Entry
NLOS	No Line Of Sight
nrtPS	Non Real-Time Polling Service
OFDMA	Orthogonal Frequency Division Multiple Access
OFDM	Orthogonal Frequency Division Multiplexing
PDU	Protocol Data Unit
PICS	Protocol Implementation Conformance Statement
PMP	Point-to-Multipoint
PHY	Physical Layer
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
rtPS	Real-Time Polling Service
SNR	Signal to Noise Ration
SOHO	Small Offices and Home Offices
SS	Subscriber Station
TCL	Tool Command Language
TDD	Time Division Duplex
TOS	Type of Service
UGS	Unsolicited Grant Service
VoIP	Voice over IP
WI-FI	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access
WirelessMAN	Wireless Metropolitan Area Networks
WirelessHUMAN	Wireless High-speed Unlicensed Metropolitan Area Networks
WLL	Wireless Local Loop

Chapter 1

1 Introduction

In the past few years, the IEEE 802.11 Standard has been widely adopted in SOHO (small and home offices), coffees, airports, etc. However, this standard has been handicapped in transmission distance, bandwidth, Quality of Service (QoS) and transmission security. The advent of IEEE 802.16 standard is emerging as a promising broadband wireless technology to finally resolve the “last mile” problem of Internet access in interoperation with IEEE 802.11. IEEE 802.16 is able to provide high-speed broadband up to 75 Mbps with the coverage of metropolitan area with Medium Access Control (MAC) layer QoS supporting, and will be widely deployed in the upcoming years.

IEEE 802.16 MAC protocol is mainly designed for point-to-multipoint (PMP) access in wireless broadband application. To accommodate the more demanding physical environment and different service requirements of the frequencies between 2 and 11 GHz, the 802.16k project enhanced the function on MAC to provide automatic repeat request (ARQ) and Mesh support. The Mesh mode is the extension to the PMP mode that allows for organic growth in coverage of the network, with low initial investment in infrastructure. In addition, a mesh inherently provides a robust network due to the possibility of multiple paths for communication between nodes. Thereby, a mesh can help to route data around obstacles or provide coverage to areas which may not be covered using the PMP setup with a similar position for the BS. A mesh also enables the support of local community networks as well as enterprise wide wireless backbone networks.

The above scenarios make the Mesh mode very attractive to network providers, companies, and user communities.

1.1 Motivation

The MAC protocol of 802.16 PMP is connection-oriented. It provides different levels of QoS to meet all kind of transmission services, including data, video and voice over IP (VoIP). Over the last years many researchers had proposed and implemented QoS architectures for the 802.16 PMP mode, but algorithms for achieving QoS for 802.16 Mesh network are still missing. The method for the QoS problem remains an open issue for further exploration.

1.2 Objectives

The aim of this thesis is to study the quality of service in IEEE 802.16 networks, particularly on Mesh topology, where the standard has lack of algorithms to achieve QoS levels similar to those defined for PMP mode. Along this thesis we outline the data transmission process in the Mesh IEEE 802.16 networks, addressing the standard guidelines as well as challenges and gaps in areas such as service class support.

The main focus of this work consists on the implementation and performance evaluation of one QoS architecture designed to reach service flow parameters for UGS, rtPS, nrtPS and BE service classes as they were introduced by 802.16 standard for PMP mode. This implementation was carried out on the popular open source Network Simulator v2 (NS-2).

1.3 Contributes of the Thesis

The main contribution of this thesis is the extension of the existent QoS models to address the lack of control and support for differentiated services offered by the provider of the next generation Mesh Networks.

1.4 Document Outline

This document includes more five chapters. In the second chapter we briefly introduce the IEEE 802.16 standard and WiMAX brand and the main characteristics that differentiate them from another wireless standards. In Chapter 3 we introduce the basic methods of Mesh networks operation. Chapter 4 presents an overview of QoS support and point out the missed packages for the IEEE 802.16 Mesh mode. Chapter 5 describes our QoS implementation and the performed evaluation and results. Chapter 6 concludes this thesis and provides some guidelines for further work.

Chapter 2

2 The IEEE 802.16 Standard

2.1 Introduction

The IEEE 802.16 standard defines the air interface for wireless metropolitan networks. It was originally designed to provide *last-mile* broadband access in metropolitan areas, with data rates comparable to DSL, Cable or T1.

This standard uses technologies such as WLL *(Wireless Local Loop)* and LMDS *(Local Multipoint Distribution System)* [5] to establish distribution systems of voice, data, internet and video on broadband networks using a network architecture similar to cellular networks. It also works as an extension of access technologies to broadband internet as ADSL (*Asynchronous Digital Subscriber Line*) and Cable.

Comparing the 802.16 standard to the 802.11 *(Wi-Fi)* standard, the 802.16 standard offers more advantages especially in the coverage area, which can reach 50 Km in open field instead of the typical 100 to 400 meters reached with the IEEE 802.11 standard. In QoS it offers support for VoIP use (*voice over IP*) and streaming (*audio and video transmission*). And finally provide support for a larger number of users.

The physical layer of 802.16 standard supports TDD (*Time Division Multiplexing*) and FDD (*Frequency Division Duplexing*) and bandwidth per channel between 1.25 to 20 MHz. The carrier operates at virtually on any frequency, allowing support for frequency ranges from 2 up to 66 GHz in either licensed and unlicensed bands. The currently available equipments operates in the bands 2.4, 2.5, 3.5, 5 and 5.8 GHz

In an environment with no line of sight (*NLOS*) one part of the radio signal is reflected by buildings and walls which causes degradations in some frequency ranges. So, its necessary integrate a protocol that can be able to cope with the loss caused by these mitigations. That protocol is the OFDM (*Orthogonal Frequency Division Multiplexing*). Unlike the FHSS or DSSS[1], it allows hundreds of carriers at the same time, which minimize the path loss with obstacles.

The evolution of this technology is based on IEEE (*Institute of Electrical and Electronics Engineers, Inc.)* workgroups, which is a nonprofit organization, world leader in technological advances.

2.2 The IEEE 802.16 Standards

In January 2003 it was published the first 802.16 standard that covers the 10-66GHz frequency range. The difficulty of propagating waves at frequencies above 6 GHz, mainly in hard metrological conditions (rain, snow), and the need for line of sight (*LOS*) between the transmitter and receiver meant that in January 2003 the Committee approved the 802.16a as an extension of the frequency ranges of the previous version to ranges below 11GHz.

The 802.16a standard, with a range of frequencies in the 2-11GHz licensed and unlicensed band, makes possible the reaching of transmission peak speeds in the order of 70Mbps (with only one subscriber station and short distances) and ranges up to 40 km (with line of sight and highly directional antennas). It also includes the new specification for the Mesh topology use.

The IEEE 802.16d standard, commonly called 802.16-2004, was published on March 24, 2004 and was set to the amendment to the IEEE 802.16 versions published so far. It uses OFDM as a technique to access the channel and only supports fixed or nomadic access, which means that it does not allow access to mobile Subscriber Stations. It supports

[1] Frequency-hopping Spread Spectrum (FHSS) and Direct-Sequence Spread Spectrum (DSSS) are methods of transmitting radio signals by rapidly switching a carrier among many frequency channels, using a pseudorandom sequence known to both transmitter and receiver.

environments LOS (*Line of Sight*) in 11-66GHz band and NLOS (*Non Line Of Sight*) in the band <11GHz. QoS and safety were also improved.

The IEEE802.16e standard, published on February 28, 2006, has allowed total mobility (speed of displacement up to 150 km/h), handoff and roaming at high speed to the Subscriber Station. The mobile services operate in the lower band (2 to 6GHz) and use a shared channel of 15Mbps that supports data-rates around 512kbps. It uses scalable OFDMA and the cell size is typically 5 Km. New media services, as well as new specifications for QoS and Security, were also implemented for outdoor environments. Equipments based on this protocol are not compatible with 802.16-2004.

But these standards are yet in constant development and upgrading news amendments. Currently there are five active versions [2]:

1. **IEEE Standard 802.16-2004** {Revision of IEEE Std 802.16 (including IEEE Std 802.16-2001, IEEE Std 802.16c-2002, and IEEE Std 802.16a-2003) developed under the temporary draft designation "P802.16-REVd"} IEEE Standard for Local and Metropolitan Area Networks - *Part 16: Air Interface for Fixed Broadband Wireless Access Systems;*

   This standard was amended by:

   - **IEEE 802.16g-2007** *Part 16: Air Interface for Fixed Broadband Wireless Access Systems - Management Plane Procedures and Services;*
   - **IEEE 802.16f-2005** *Part 16: Air Interface for Fixed Broadband Wireless Access Systems - Management Information Base;*
   - **IEEE 802.16e-2005** *Part 16: Air Interface for Fixed Broadband Wireless Access Systems- Physical and Medium Access Control Layers for Combined Fixed and Mobile* Operation in Licensed Bands;

2. **IEEE Standard 802.16.2-2004** {Revision of IEEE Std 802.16.2-2001} IEEE Recommended Practice for Local and Metropolitan Area Networks - Coexistence of Fixed Broadband Wireless Access Systems;

3. **IEEE Standard 802.16/Conformance03-2004** Standard for Conformance to IEEE 802.16 - *Part 3: Radio Conformance Tests (RCT) for 10-66 GHz WirelessMAN-SC^{TM} Air Interface;*

4. **IEEE Standard 802.16/Conformance04** Standard for Conformance to IEEE Standard 802.16 - *Part 4: Protocol Implementation Conformance Statement (PICS) Proforma for Frequencies below 11 GHz*
5. **IEEE Standard 802.16k** (amendment of IEEE Std 802.1D {as previously amended by IEEE Std 802.17a} Standard for Local and Metropolitan Area Networks: Media Access Control (MAC) Bridges - Bridging of 802.16

The draft standards under development are:

1. **IEEE Draft P802.16h** - Improved Coexistence Mechanisms for License-Exempt Operation;
2. **IEEE Draft P802 .16i** - Mobile Management Information Base
3. **IEEE Draft P802.16j –** Multi-hop Relay Specification
4. **IEEE Draft P802.16Rev2** - Consolidate 802.16-2004, 802.16e, 802.16f, 802.16g and possibly 802.16i into a new document.

2.3 WiMAX Forum

WiMAX (*World Wide Interoperability for Microwave Access*) founded in June 2001 [4], is a nonprofit organization, formed by telecommunications operators (British Telecom, France Telecom) and several manufacturers (INTEL, NOKIA, Siemens. Its aim is to accelerate the introduction of BWA (*Broadband Wireless Access*) technologies through the certification of equipment based on 802.16 standards, making it possible levels of price/performance that are impossible to achieve with proprietary technologies (2G, 3G,...). It provides specifications for fixed communication LOS in the range of 10-66GHz (Std 802.16/Conformance03-2004), for fixed or nomadic communications NLOS in the range of 2-11GHz (Std 802.16-2004, 802.16.2-2004, 802.16/Conformance04 and 802.16k) and also sets specifications for mobile stations to 150 km/h in the range of 2-6GHz (Std 802.16e2005).

An operator that chooses interoperability and equipment based on standards, benefits from a growing mass market and reduces the risk of implementation, not getting limited to a single manufacturer. Its Base Station is compatible with any Subscriber Station provided it is certified by the WiMAX Forum.

A product manufacturer only receives the WiMAX forum certification if it meets the standards and ensures interoperability with other certified equipment. WiMAX Forum is similar to Wi-Fi Alliance in promoting the standard IEEE 802.11.

The first certification lab was opened in July 2005 for the IEEE 802.16-2004 standard, in the 3.5GHz band, and began immediately receiving equipment for testing.

On January 16, 2006, the first 6 products certified by the WiMAX Forum were announced: 3 Base Stations (*Grip Size Networks*, *Redline Communications* and *Sequans Communications*) and 3 Subscriber Station (*Redline Communications*, *Sequans Communications* and *Wavesat Wireless Inc*.). This number has been exponentially increasing in the last two years. Actually there are over than 980 licensed products.

2.4 Frequency bands of WiMAX products

The WiMAX standard, due to its wide range of frequencies of operation, make it virtually compatible in any spectrum world, unlike Wi-Fi that only defines the 2.4GHz and 5 GHz free frequencies as valid ones. WiMAX forum determined that initially it will focus on the procedures for submission and interoperability testing in equipments that support the physical layer OFDM 256 and operate in licensed bands of 2.5 GHz and 3.5 GHz and the unlicensed band of 5.8 GHz.

Fig. 1 shows the world distribution of WiMAX licensed and unlicensed bands.

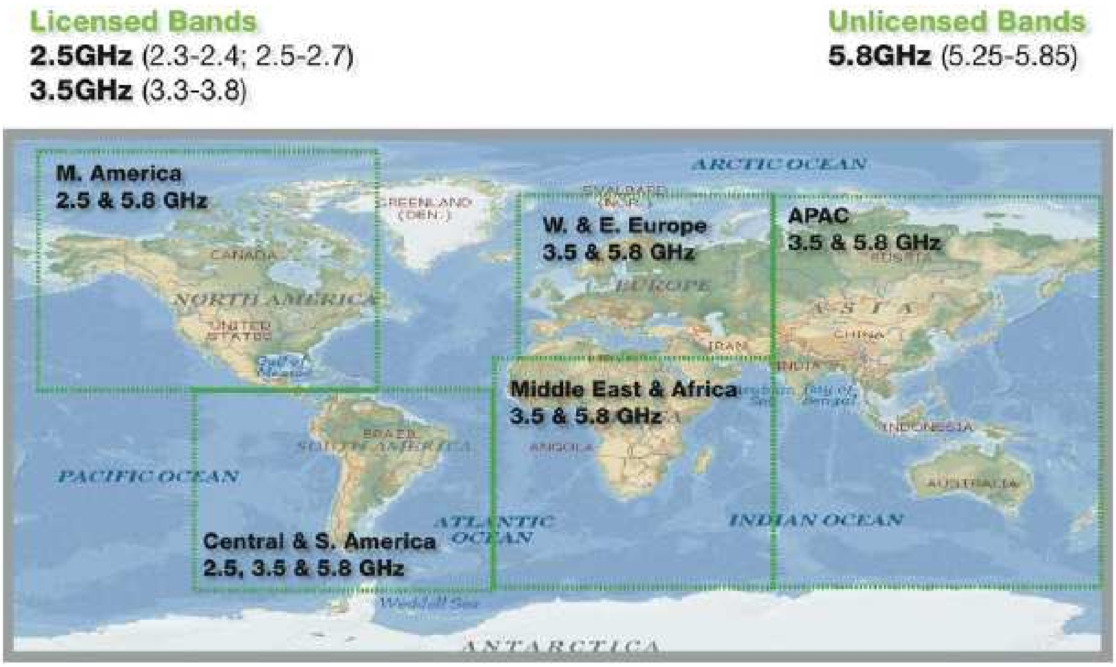


Fig. 1 – Representation of licensed and unlicensed bands

2.5 WiMAX in Portugal

Fig. 2 shows the current frequencies that do not need license from the spectrum regulator in Portugal (ANACOM).

The 3.5GHz and 3.6GHz band is a special band for WiMAX due to its spectral characteristics. Instituto de Telecomunicações and Portugal Telecom have license for 3.5GHz band (under the European Project *DAIDALOS - PT Inovação*). *Novis* and *Oni* have licenses for 3.6GHz band.

Some WiMAX deployments are in course in Portugal, some of which stand out: private networks for interconnection of buildings - University of Covilhã-Hospital of Covilhã and plans to interconnect Covilhã-Fundão-Castelo Branco.

There are not any WiMAX licenses allocated in Portugal. The National Telecommunications Authority is attending the debate on this technology at European level, in particular at CEPT (European Conference of Postal and Telecommunications

Administrations). At this stage of discussion, technical and regulatory aspects related to the possible introduction of WiMAX in the frequency ranges of 3.5 GHz (3400-3800 MHz) and 5.8 GHz (5725-5875 MHz) are under analysis.

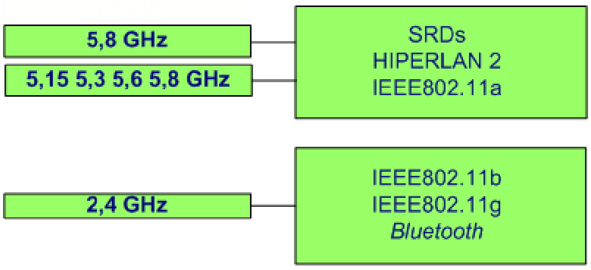


Fig. 2 – Free license frequencies in Portugal

Given the current and planned uses of these frequency bands, studies are on going to assess whether these bands can actually be shared with WiMAX. It should be noted that it is highly important and convenient the harmonization at European level of the solutions to be implemented under this new technology.

Given the interest that the matter is awakening to the national level, the PCI-ANACOM will, as soon as possible, provide information on the regulatory and technical framework of WiMAX in Portugal.

2.6 WiMAX Technology

The term WiMAX has been used generically to describe wireless licenses and systems based on the 802.16-2004 standard in the range of 2-11GHz. The standard specifies only the layer 1 and 2 (see

Fig. 3) but it is compatible with different technologies of layer 3 and above.

The main technical properties of the first two layers of the 802.16-2004 are briefly introduced in the next points.

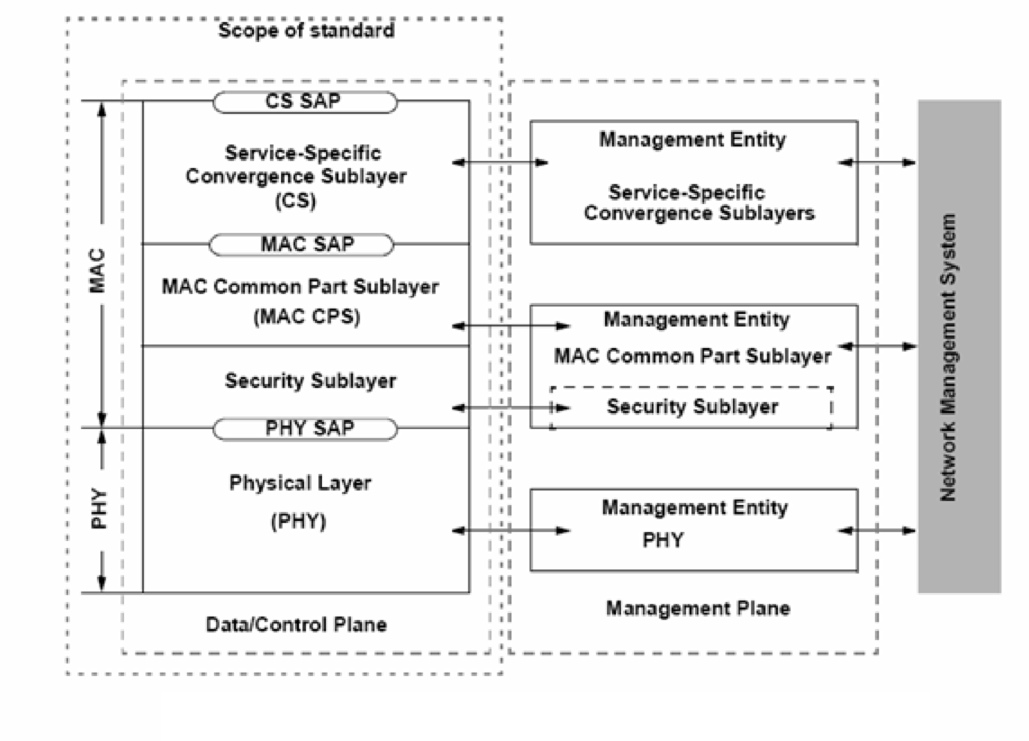


Fig. 3 - IEEE Std 802.16 protocol layering [1]

2.6.1 Main Features

Below are summarized the main features of the first two layers of the 802.16-2004 standard:

- Bandwidth of up to 70Mbps in a 20MHz channel.
- Channel varies from 1.25MHz up to 20MHz.
- Support for LOS and NLOS environments.
- Radius of 8 km NLOS, radius of 16 Km LOS, range of 50 km in Point-to-point LOS for fixed access.
- Full-Duplex or Half-Duplex with TDD and FDD.
- Operation in licensed or free spectrum.
- Carrier based on multiple frequencies with OFDM and OFDMA (2-11GHz).

- Technology for 1 and 2 network layers. Only the PHY and MAC layers are specified by the standard. Compatible with layer 3 communication protocols (IPv4, IPv6, ATM ...).
- Soft Handoff not specified in the standard, optional for each manufacturer.
- Roaming can be implemented but it is considered a higher level capacity that goes beyond the scope of the WiMAX Forum certification program, which cares about the PHY and MAC layers.
- Differentiated QoS levels.
- Adaptive modulation (BPSK, QPSK, 16QAM e 64QAM).
- Point-to-point and Point-to-Multipoint topologies and optionally *logical Mesh Networks*.
- New and advanced security algorithms.
- Support for the use of Adaptive Antenna Systems (Smart antennas) and MIMO (Multiple Input Multiple Output).

The wireless metropolitan access networks (WMAN) defines two types of stations:

- **BASE STATION (BS) -** controls and manages the connections. Send downlink data in different channels for each subscriber. The base station can cover multiple sectors with the help of sectorized antennas. Each BS is identified with a single MAC address of 48bits.
- **SUBSCRIBER STATION (SS) -** The subscriber station is a terminal that communicates with the base station. The *uplink* is point-to-point in a point-to-multipoint network configuration but in a mesh configuration can either be point-to-point and point-to-multipoint. Each SS in the same sector and at the same frequency channel receives the same information. Each SS is identified by a single MAC address of 48bits.

2.7 Physical Layer

The standard 802.16d defines four technologies for interfacing with the environment:

- **WirelessMAN-SC™ 10-66GHz -** modulation based on a single carrier. Each channel has a width of 25-28 MHz, raw data up to 120Mbit/s, used in LOS needs.
- **WirelessMAN-SCa™ 2-11GHz -** modulation based on a single carrier. It is similar to the previous one but with lower output due to the decline in the spectrum area and with support for NLOS environments. Example: backhaul links.
- **WirelessMAN-OFDM™ 2-11GHz -** 256-carrier Orthogonal-Frequency Division Multiplexing. It is designed to NLOS environments. Example: fixed access.
- **WirelessMAN-OFDMA™ 2-11GHz -** 2048-carrier Orthogonal-Frequency Division Multiple Access. It is designed to NLOS environments. Example: mobile access.

The last two technologies are the most frequently used for NLOS. Initially, the manufacturers have preferred using WirelessMAN-OFDM™ 256 in its equipments due to its lower complexity and ease of synchronization with respect to OFDMA 2048. Bandwidth is variable and may take values between 1.25-20MHz depending on each manufacturer and the bandwidth available. The technology can be extended for lower frequencies such as 700MHz which will be used in the U.S.

Following points present the relevant characteristics of 802.16 physical layer.

Dynamic adaptive modulation - this property allows the base station to change its modulation scheme depending of transmission conditions. For example, if a base station cannot establish a robust connection with a subscriber using the scheme of higher order modulation, 64QAM (Quadrature Amplitude Modulation), it can reduce to 16QAM or QPSK (Quadrature Phase Shift Keying) that reduces the data-rate but increases the effective range. The 802.16-2004 standard defines up to 7 combinations of modulations, that are used depending on the SNR (Signal to Noise Ratio) condition. The characteristics of these modulations are shown on Table 1.

Rate ID	Modulation rate	Coding	Information bits/symbol	Information bits/ OFDM symbol	Peak data rate in 5 MHz (Mb/s)
0	BPSK	1/2	0.5	88	1.89
1	QPSK	1/2	1	184	3.95
2	QPSK	3/4	1.5	280	6.00
3	16QAM	1/2	2	376	8.06
4	16QAM	3/4	3	568	12.18
5	64QAM	2/3	4	760	16.30
6	64QAM	3/4	4.5	856	18.36

Table 1 - Characteristics of modulations used by IEEE 802.16 standard

Duplexing TDD/FDD - the options contained in WiMAX allow compatibility with the requirements imposed on carriers of each country. The WiMAX systems can be configured in FDD (Frequency Division Duplex) or TDD (Time Division Duplex) mode. In FDD mode the full-duplex communication is carried out in 2 channels at different frequencies, one for upload and another for download. Normally the mobile station has the lowest frequency because it implies less power from the source. In TDD mode the channel is divided into slots of time for upload and download. It is also a full-duplex communication. As it uses only one channel, the transmission rate is reduced by half.

Scalability - the great flexibility of WiMAX allows the use of multiple frequencies (licensed or free) and channel bandwidth, which are required by the application or also by the restrictions imposed by the regulatory authority for allocation of spectrum. Today the equipment allows the frequencies: 2.3-2.4GHz, 3.3-3.8GHz, 4.9-5.0GHz, 5.8GHz and channels with 1.75, 3.5, 5, 7, 10, 14 MHz.

Coverage - it is provided support for technologies that increase the NLOS coverage (no line of sight) as the Mesh topologies, Smart antennas and MIMO multiple antennas.

Dynamic Frequency Selection - in license exempt bands several carriers may have to live in the same spectrum area. WiMAX incorporates the dynamic selection frequency technology where the radio automatically searches an available channel.

Error Correction Techniques - Uses Forward Error Correction (FEC) which adds redundancy into the transmission by repeating some of the information bits. Bits that are missing or are in error can be corrected at the receiving end. The frames that can not be corrected are relayed through the use of ARQ methodology (Automatic Repeat Request).

Power control - Uses control power algorithms to increase system performance and mobile stations autonomy. The base station sends control power information to all SSs so that they radiate just the needed power for the contacted service.

Multiple topologies - Specifications for two modes of operation: a point-to-multipoint (PMP) mode and a Mesh operation mode (see Fig. 4). The PMP mode supports networks where all subscriber stations (SS) are within one hop from the base station (BS). The traffic may take place only between a BS and its SSs. Direct communication between two SSs is not supported in this mode.

On the other hand the Mesh mode allows the network to function even when all subscriber stations are not within direct range of the base station. Thus, essentially the Mesh operation mode permits the routing of data between two subscriber stations as well as between the base station and subscriber stations over a multi-hop route.

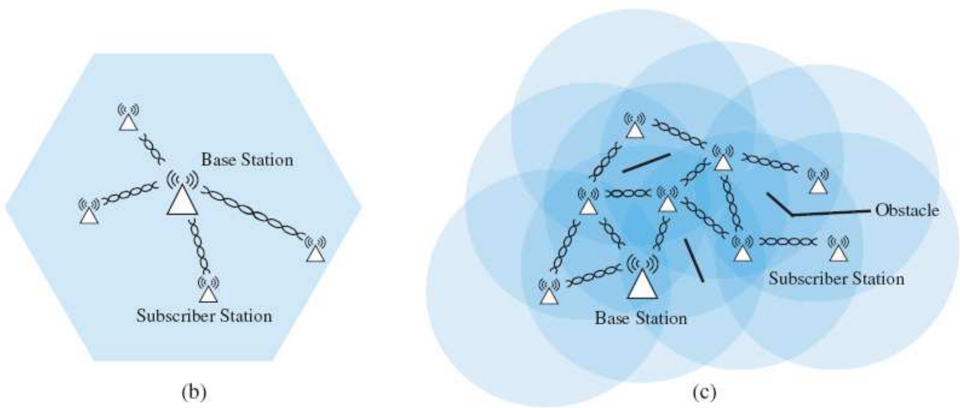


Fig. 4 – (b) Point-to-multipoint (PMP) and. (c) Mesh operation modes in 802.16 standard

2.8 Summary

In this chapter we introduced the WiMAX project group and the state of WiMAX evolution in Portugal. We still presented the main physical and MAC features of IEEE 802.16 standard.

In the next chapter we will introduce the basic methods of Mesh networks operation, with special attention for the methods of bandwidth resources sharing for the multiple nodes (SS) communications.

Chapter 3

3 Mesh Networks through IEEE 802.16 standard

The optional Mesh mode is designed to operate in the below 11 GHz frequency band. The IEEE 802.16-2004 specifies both WirelessMAN-OFDM^{TM} [2] and WirelessHUMAN^{TM}(-OFDM) [3] air interfaces to operate in the Mesh mode. The WirelessMAN-OFDM^{TM} is meant for operation in licensed bands. The WirelessHUMAN^{TM} is specified for operation in license-exempt frequency bands. The operation in the license-exempt frequency band requires the implementation of additional dynamic frequency selection (DFS) mechanisms to avoid interference with other networks operating in the same frequency band. The standard allows only time division duplex (TDD) operation in the Mesh mode. Fig. 5shows the logical frame structure for the Mesh mode of operation. A frame consists of two parts, the control subframe and the data subframe. The control subframe is dedicated to the transmission of control and management messages. The data subframe is mainly used for transmission of data messages; however it may be also used to transmit some control messages. To enable multiple nodes to share access to the wireless medium, the control subframe is divided into a number of transmission opportunities. The data subframe is similarly divided into a number of minislots.

3.1 Mesh frame structure

The frame duration depends on the configuration used in the Mesh network and can be fixed by the Mesh base station. The frame duration is fixed; on a change in the frame duration all nodes in the network need to resynchronize themselves to the BS. The selected frame duration can be identified by the frame duration code specified in the "Network descriptor" (data structure which is propagated throughout the network via network configuration messages). The standard specifies frame duration codes 0 - 6, corresponding to frame duration ranging from 2.5 ms to 20 ms [1]. The frame duration codes 7 - 255 are reserved for future use.

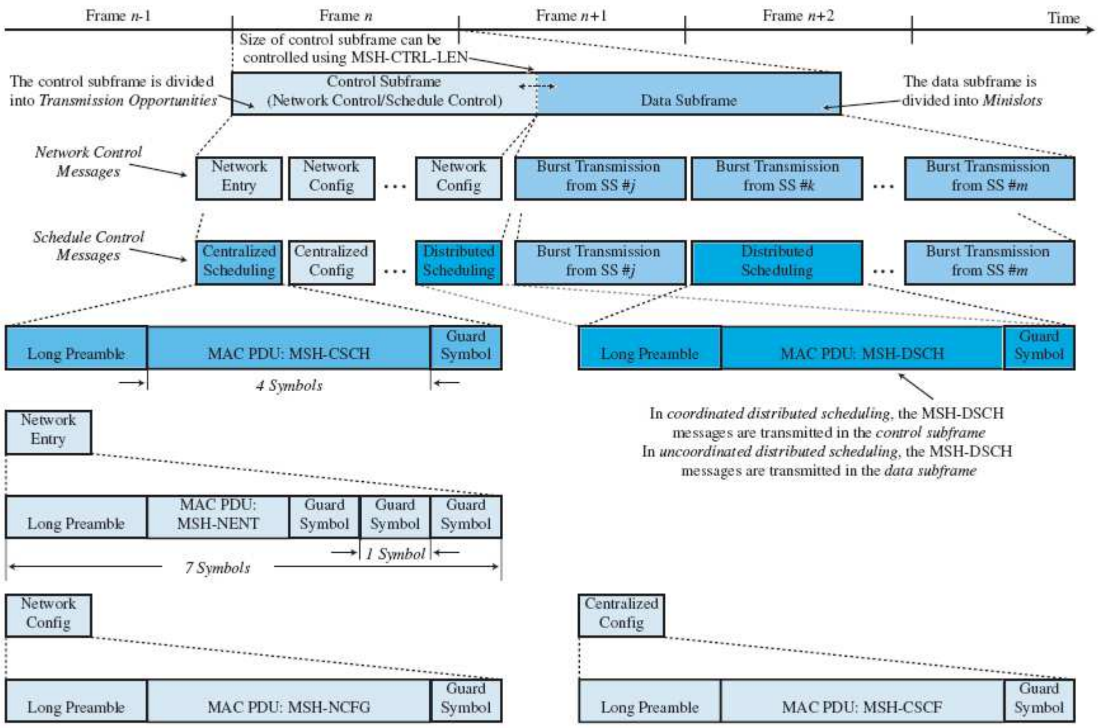


Fig. 5 - Frame Structure for Mesh mode and corresponding management messages [6]

The number of OFDM symbols per frame depends on the channelization parameters and the channel bandwidth. The amount of data per OFDM symbol depends on the modulation used. All transmissions in the control subframe are sent using QPSK-1/2 with the mandatory coding scheme.

To enable multiple nodes to share access to the medium in the control subframe, these are divided into a set of transmission opportunities. Fig. 5 shows the division of control subframe in a set of transmission opportunities. A transmission opportunity is composed of seven consecutive OFDM (orthogonal frequency division multiplexing) symbols.

The number of transmission opportunities in a control subframe can be controlled by using the variable *MSH-CTRL-LEN* in the Network descriptor. Let *OFDM_SYM_PER_FRAME* represent the total number of OFDM symbols for the entire frame. Given that the control subframe has *MSH-CTRL-LEN* transmission opportunities, with each opportunity being composed of seven OFDM symbols, the number of OFDM symbols for the control subframe is *OFDM_SYM_PER_CTRL_SUBFRAME* which is given by the Equation 1.

$$OFDM\_SYM\_PER\_CTRL\_SUBFRAME = MSH\text{-}CTRL\text{-}LEN \times 7 \quad (1)$$

The remaining OFDM symbols are used for the data subframe. Thus, the number of OFDM symbols for the data subframe, *OFDM_SYM_PER_DATA_SUBFRAME*, is given by Equation 2.

$$OFDM\_SYMPER\_DATA\_SUBFRAME = OFDM\_SYMPER\_FRAME - OFDM\_SYM\_PER\_CTRL\_SUBFRAME \quad (2)$$

There are two types of **control** subframes depending on their function as listed below:

- Network Control Subframe

- Schedule Control Subframe

Network control subframes are used to transmit management messages related to network control activities. Network control implies the functions needed to maintain the synchronization in the network and cohesion throughout the Mesh network. Network control messages help to distribute the network configuration parameters to neighboring nodes and also allow new nodes to synchronize themselves with the network, join an existing Mesh network and establish logical links to neighboring nodes.

The other type of control subframe is the **schedule control subframe**. The schedule control subframe, similar to the network control subframe, has *MSH-CTRL-LEN* transmission opportunities. The transmission opportunities in the schedule control subframe are used by the nodes for transmitting MAC management messages which help to set up transmission schedules. Unlike most other contemporary MAC standards the 802.16 standard requires the nodes to explicitly reserve bandwidth for transmission on the logical links to neighboring nodes prior to the transmission. To enable nodes to reserve bandwidth for the transmission of data the standard specifies a set of control messages. This enables the nodes in the network to synchronize their data transmissions in a collision free manner. These scheduling messages are transmitted by the nodes in the transmission opportunities in the schedule control subframe. The standard specifies both **centralized** as well as completely **distributed** mechanisms to schedule data transmissions. These scheduling messages help to schedule transmission of data in the data subframe. The process according to which nodes access the medium during the control subframe (both the **network control** as well as **schedule control** subframes) will be described in detail in the Section 3.2.

The OFDM symbols in a frame not used by the control subframe compose the data subframe. To enable multiple nodes to share access to the medium in the data subframe, the data subframe is divided into units called minislots. A minislot is the smallest unit of bandwidth allocation. The maximum number of minislots possible in a data subframe is specified to be 256 in the standard. The exact number of minislots in the data subframe depends on the selected frame duration. The number of OFDM symbols per minislot (with the exception of the last minislot), *OFDM_SYM_PER_MINI_SLOT*, is given by Equation3.

$$OFDM\_SYM\_PER\_MINI\_SLOT = [OFDM\_SYM\_PER\_DATA\_SUBFRAME / 256] \qquad (3)$$

The mechanism for reservation of minislots for accessing the medium during the data subframe will be explained in detail in Section 3.3.

3.2 Medium access control for the Control Subframe

We first look at the mechanisms applicable for the **network control** subframe followed by the corresponding mechanisms for the **schedule control** subframe.

Fig. 6 shows the network topology and notation that we will refer for the following examples in this Chapter.

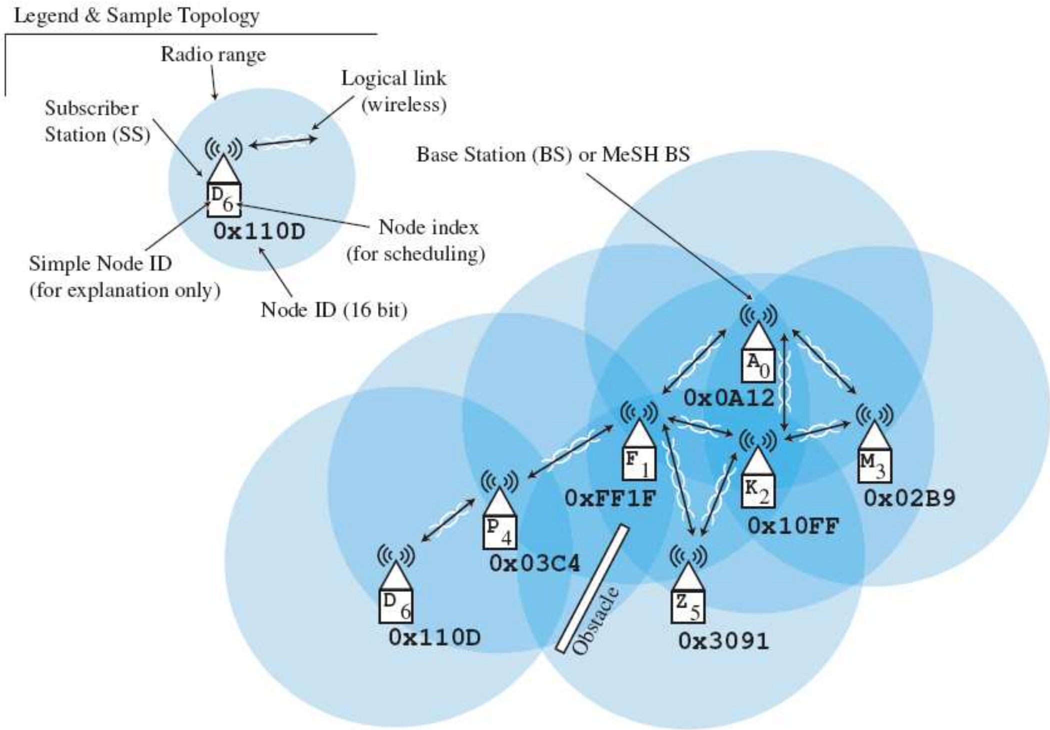


Fig. 6 – Network topology and used notation [6]

3.2.1 Network Control Subframe

MSH-NCFG (Mesh network configuration) and MSH-NENT (Mesh network entry) messages are transmitted in the network control subframe. The first transmission opportunity in the network control subframe is reserved for the network entry and is called the NetEntry slot. The NetEntry slot is used for the transmission of MSH-NENT messages. The remaining transmission opportunities (*MSH-CTRL-LEN* − 1) are used for the transmission of MSH-NCFG (NCFG) messages. We first look at the process for accessing the NetEntry slot followed by the procedure for accessing the other network control transmission opportunities (slots).

The NetEntry slots are used by "new nodes"[1] to transmit MSH-NENT (NENT) messages. To access the NetEntry slot the new nodes use a two staged process. The initial NENT message is sent in a random, contention based fashion in a free NetEntry slot immediately following a transmission of an NCFG (network configuration message) by the

[1] "new nodes" are nodes which have not yet been fully registered and are not yet a part of the existing mesh network.

targeted sponsor. The NCFG transmitted by the targeted sponsor should have a sponsored MAC address 0x000000000000 indicating the willingness of the target sponsor to enable new nodes to join the network by functioning as a sponsor. The initial NENT contains a request information element identifying the targeted sponsor and specifying the new node's MAC address. If the targeted sponsor accepts the NENT request it transmits a MSH-NCFG message with the sponsored MAC address field containing the new node's MAC address. After the sponsor advertises the new node's MAC address in the NCFG message, the new node may send a NENT message in the immediately following NetEntry opportunity. To access the NENT slots, new nodes use the algorithm specified by the pseudocode in APPENDIX I, procedures *RecIncomingMSH-NCFG_Msg()* (line 16-35) and *NetworkControlSubframeStart()* (line 37-75) to decide if a new node should transmit a NENT message in the corresponding NetEntry slot. A new node has to receive at least two NCFG messages containing the Network Descriptor from the potential (target) sponsor before it can start with the network entry process (with the targeted node as a potential sponsor). The network entry process is outlined in APPENDIX II.

We have now seen how new nodes access the NetEntry slot in a contention based manner. The remaining (MSH-CTRL-LEN − 1) slots in the network control subframe (reserved for transmission of network configuration (NCFG) messages) are accessed by the nodes in a contention free manner. To enable contention free access to these slots the nodes use a distributed election algorithm to decide which node transmits the NCFG message in a given transmission opportunity. Thus, the nodes coordinate their transmissions in a two-hop neighborhood to ensure collision free transmission[2] of the NCFG messages.

The pseudocode outlining the algorithm used by the nodes to access the non-NetEntry slots (MSH-NCFG transmit opportunities) in the network control subframe can be found in the standard ([1] pp. 159 - 160). Fig. 7 illustrates the distributed election process. We first introduce some of the terms used in the Fig. 7.

[2] i.e., no collision occurs at the intended receivers of the NCFG message transmitted by a node. The intended receivers of the NCFG message are all the neighbors of the node transmitting the message.

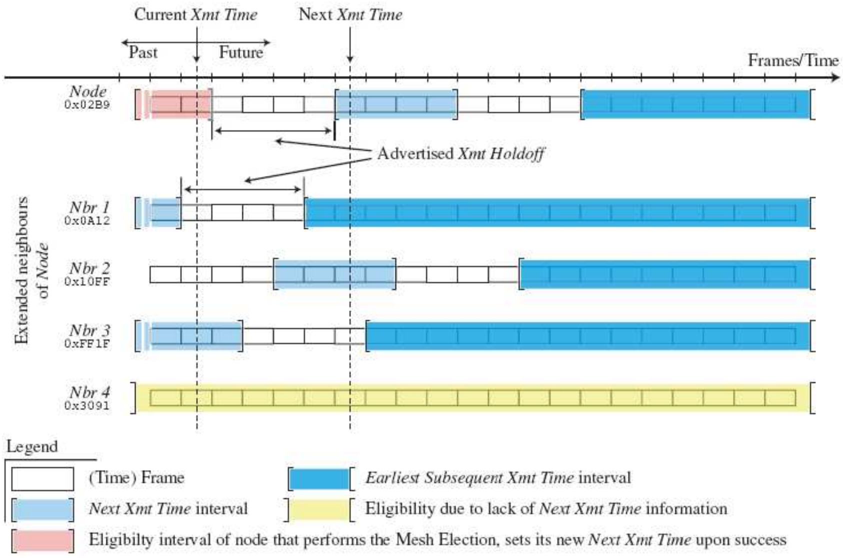


Fig. 7 - Medium access in the network control subframe (Non-NetEntry slots) [6]

On the time axis we have temporally ordered subsequent network control transmission opportunities (excluding the NetEntry slots). The *Xmt Holdoff Time* (advertised holdoff time) is the number of MSH-NCFG transmit opportunities after *Next Xmt Time* that the node is not eligible to transmit MSH-NCFG messages. The *Xmt Holdoff Time* is given in Equation 4.

The *Next Xmt Time* is the next MSH-NCFG eligibility interval for a node. The eligibility interval comprises of a set of consequent MSH-NCFG transmission opportunities in which the node is permitted to transmit a NCFG message provided it wins the distributed Mesh election algorithm specified in the standard. The parameter *Next Xmt Mx* alongwith the *Xmt Holdoff Exponent* help to determine the *Next Xmt Time* interval. As explained in the standard ([1] pg. 83), the Next *Xmt Time* is computed as the range given by Equation 5.

$$\text{Xmt Holdoff Time} = 2^{\text{Xmt Holdoff Exponent} + 4} \tag{4}$$

$$2^{\text{Xmt Holdoff Exponent}} \times \text{Next Xmt Mx} < \text{Next Xmt Time} \leq 2^{\text{Xmt Holdoff Exponent}} \times (\text{Next Xmt Mx} + 1) \tag{5}$$

For example, when the *Xmt Holdoff Exponent* = 4 and the *Next Xmt Mx* = 3, then the node is considered eligible for its next MSH-NCFG transmission between 49 and 64 transmission opportunities away and ineligible before that time. The values for the *Next Xmt Mx* and *Xmt Holdoff Exponent* are advertised by the nodes in their MSH-NCFG messages (as a part of the *NetConfig Schedule* info). If the *Next Xmt Mx* field has the value 0x1F, the node transmitting this message should be considered to be eligible to transmit from the time indicated by this value and every MSH-NCFG opportunity thereafter (i.e. the *Xmt Holdoff Time* is considered to be 0). All neighbors for whom the up to date values for the *Next Xmt Mx* and *Xmt Holdoff Exponent* are not known are assumed to be eligible to transmit MSH-NCFG messages in every subsequent MSH-NCFG transmission opportunity. The value for the variable *Earliest Subsequent Xmt Time* for a node is obtained by adding the node's *Next Xmt Time* to the node's *Xmt Holdoff Time*.

The standard refers: during the current Xmt Time of a node (i.e., the time slot when a node transmits its MSH-NCFG packet), the node uses the following procedure to determine its *Next Xmt Time*. Here, the current *Xmt Time* is a transmission opportunity which lies within the node's MSH-NCFG eligibility interval (defined previously). For nodes which do not manage to win the Mesh election in their eligibility interval, the eligibility interval is the *Earliest Subsequent Xmt Time* interval as shown in Fig. 7. For nodes which haven't yet transmitted a MSH-NCFG message (i.e. new nodes), the eligibility interval is all subsequent MSF-NCFG opportunities until the node transmits the MSH-NCFG by winning the distributed Mesh election and thereby calculating its *Next Xmt Time*.

Having explained the meaning of the current *Xmt Time*, we next explain the mechanism used by nodes to access MSH-NCFG transmission opportunities within their eligibility interval. As shown in Fig. 7, consider a node which has a MSH-NCFG message for transmission and which has the current transmit time (i.e. current time/current transmission opportunity) within its eligibility interval. The node then looks up its extended neighborhood[3] information to find a set of eligible competing nodes (i.e. nodes in the extended neighborhood which compete for the current transmission opportunity). To determine the set of competing nodes, the node computes its *TempXmtTime* as the current *Xmt Time* plus the node's advertised *Xmt Holdoff Time*. The set of competing nodes then contains those nodes for which their *Next Xmt Time* interval includes *TempXmtTime* or their *Earliest Subsequent Xmt Time* is equal to or smaller than *TempXmtTime*. Let the set of node IDs of the competing nodes be stored by the array *CompetingNodeIDList*[]. The

[3] The extended neighborhood of a node contains all the nodes within two-hops or three-hops from the node as specified in the network descriptor.

node then calls the MeshElection(*TempXmtTime*, *OwnNodeID*, *CompetingNodeIDList*) as shown in the standard to determine if it wins the election, transmits the NCFG message in the current transmission opportunity thereby setting its *Next Xmt Time* interval to *TempXmtTime*. In case the node does not win the election, it sets the *TempXmtTime* to the next MSH-NCFG opportunity and repeats the above process. The Mesh election carries out a pseudorandom mixing of the arguments in a fair manner to determine if a node wins the current transmission opportunity or not. The pseudocode for the MeshElection algorithm is specified in the standard [1], pg. 160. All nodes independently carry out the above process when accessing the MSH-NCFG slots. The Mesh election algorithm ensures that only one of the concurrent competing nodes wins a given transmission opportunity. Thus, the Mesh election ensures contention free access to the MSH-NCFG slots in a fair manner.

3.2.2 Schedule Control Subframe

The schedule control subframe, like the network control subframe, has a total of *MSH-CTRL-LEN* transmission opportunities. The parameter *MSH-DSCH-NUM* in the Network Descriptor specifies the maximum number of distributed scheduling (MSH-DSCH) messages that may occur in a schedule control subframe. Thus, in a schedule control subframe the last *MSH-DSCH-NUM* transmission opportunities are reserved for the transmission of the MSH-DSCH messages. The first (*MSH-CTRL-LEN − MSH-DSCH-NUM*) transmission opportunities are reserved for the transmission of centralized scheduling messages (MSH-CSCH, MSH-CSCF). We denote the transmission opportunities reserved for the transmission of distributed scheduling messages as MSH-DSCH transmission opportunities. The transmission opportunities set aside for the transmission of centralized scheduling messages are denoted as MSH-CSCH transmission opportunities. To access the MSH-DSCH slots, the nodes use the same procedure as is used for accessing the MSH-NCFG slots. The corresponding values for the parameters *Next Xmt Mx*, *Xmt Holdoff Exponent* for distributed scheduling are transmitted by the nodes in the MSH-DSCH messages (as information within the MSH-DSCH scheduling information element). Everything else is similar to the description for the MSH-NCFG messages, except that the nodes are now referring to the MSH-DSCH transmission opportunities instead of the MSH-NCFG transmission opportunities.

The scheduling of the centralized scheduling messages (MSH-CSCH messages and the MSH-CSCF messages) on the other hand uses a centralized mechanism and does not use the distributed MeshElection algorithm outlined above. The MSH-CSCF messages play a central role in the functioning of the coordinated centralized scheduling mechanism. The message structure of the MSH-CSCF message can be seen in APPENDIX III.

The MSH-CSCF message is generated by the BS. The BS first transmits the message to all its neighbors. The nodes receiving the MSH-CSCF message rebroadcast the message in their order specified in the scheduling tree in the MSH-CSCF message. The scheduling tree or routing tree specifies a tree consisting of subset of nodes in the Mesh network. The nodes present in the scheduling tree are identified by their Node IDs. The tree also specifies implicitly an index for each node in the scheduling tree. The position of the node (Node ID) in the list in the MSH-CSCF message corresponds to the index for the particular node (Node ID). The field *NumberOfNodes* in the MSH-CSCF message specifies the number of nodes included in the scheduling tree (including the BS). As shown in APPENDIX III, the MSH-CSCF message consists of a list of Node IDs: the position of a Node ID in this list corresponds to the index for that node (0 to (*NumberOfNodes* − 1)). A scheduling tree entry specifies a Node ID, *NumberOfChildren* (number of children for node with Node ID), and for each child the index for the child node (index based on position of the child’s Node ID in the MSH-CSCF message) along with the uplink and downlink burst profile for transmissions from/to the child node respectively.

Fig. 8 shows an example of a scheduling tree specified in a MSH-CSCF message. As shown in the table in Fig. 8, the MSH-CSCF message specifies a list of nodes to be included in the scheduling tree. The nodes are identified by specifying their Node IDs. The position of the node in this list gives the index for the node. Thus, as can be seen from the figure, node with ID 0x0A12 is specified first thus has index 0, the node 0xFF1F is specified next and has index 1 and so on. Corresponding to each node in the list is a scheduling tree entry. This entry specifies, for each node, information about its children (again identified by the indexes for the children within the list). Thus, in the above example, for node 0x0A12 information is specified about its children (nodes with index 1, index 2, and index 3). From the indexes for the children one can map the children to nodes with IDs 0xFF1F, 0x10FF, and 0x02B9 respectively. For each child node the scheduling tree entry contains the uplink/downlink burst profile for transmissions from and to the corresponding child node. The network topology in Fig. 8 shows the scheduling tree corresponding to the list specified in the table.

Node	Node ID	Node Index	Child Indices, corresponding Burst Profiles
A	0x0A12	0	{1:[bp_{u1},bp_{d1}], 2:[bp_{u2},bp_{d2}], 3[bp_{u3},bp_{d3}]}
F	0xFF1F	1	{4:[bp_{u4},bp_{d4}], 5:[bp_{u5},bp_{d5}]}
K	0x10FF	2	{}
M	0x02B9	3	{}
P	0x03C4	4	{6:[bp_{u6},bp_{d6}]}
Z	0x3091	5	{}
D	0x110D	6	{}

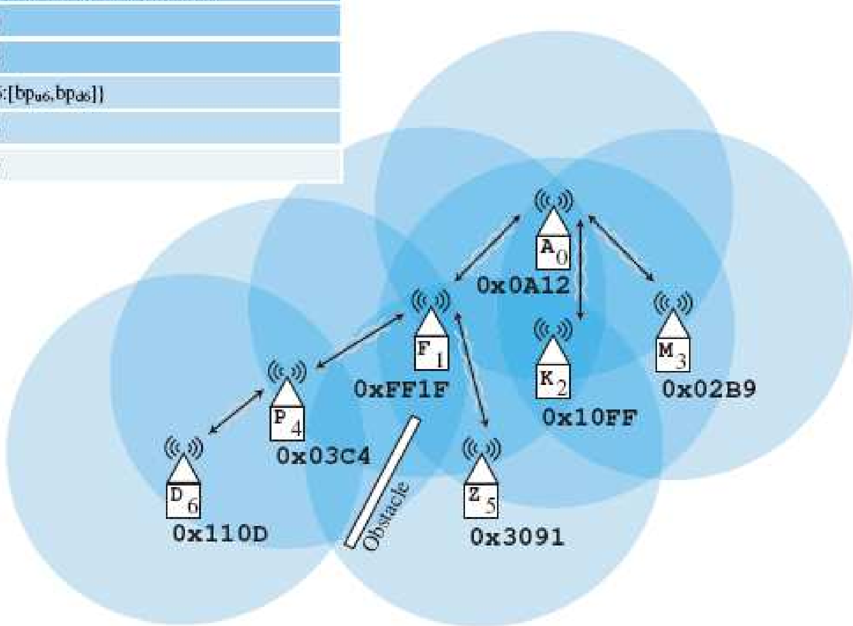


Fig. 8 - MSH-CSCF schedule example [6]

We have thus seen how the scheduling tree is specified. We next look at the scheduling of the centralized scheduling messages (MSH-CSCF and MSH-CSCH) in detail. As already mentioned the first (*MSH-CTRL-LEN* − *MSH-DSCH-NUM*) transmission opportunities in a schedule control subframe are reserved for the transmission of the centralized scheduling messages. For the following discussion we consider only these transmission opportunities reserved for transmission of centralized scheduling messages. The 802.16 standard uses the following procedure for the transmission of the MSH-CSCF and MSH-CSCH messages down the scheduling tree. The BS first transmits the message (MSH-CSCF/MSH-CSCH) followed by transmission (rebroadcast) of the message by its children, in the order in which they appear in the scheduling tree. This is then followed by a retransmission of the message by the children of the nodes which transmitted the message in the previous round, again in the order of appearance in the scheduling tree. The above proceeds until all the nodes in the scheduling tree have transmitted in the appropriate order. If a node needs to transmit a message immediately after receiving it, a delay of *MinCSForwardingDelay* is inserted. The value *MinCSForwardingDelay* specifies the minimum delay in OFDM symbols that is inserted between the end of a reception and the start of transmission of a centralized scheduling message at a node. The value for the *MinCSForwardingDelay* is fixed and is specified in the Network Descriptor.

For transmission of the MSH-CSCH messages up the scheduling tree the following procedure is used. The MSH-CSCH transmissions up the scheduling tree starts after all the nodes in the scheduling tree have received the downward transmissions. The transmission order for the upward transmissions is determined as follows. The nodes with the higher hop count from the BS transmit the message before those with a lower hop count. For nodes with the same hop count, the transmissions are ordered as per the order in the scheduling tree specified in the MSH-CSCF message.

Thus, for the sample network shown in Fig. 8, for the downtree transmissions the nodes transmit in the following order 0x0A12, 0xFF1F, 0x10FF, 0x02B9, 0x03C4, 0x3091, 0x110D. For the uptree transmissions the transmission order is as follows: 0x110D, 0x03C4, 0x3091, 0xFF1F, 0x10FF, 0x02B9.

3.3 Medium access control for the Data Subframe

The data subframe is used by the nodes in the Mesh network to transmit data messages. The nodes schedule the transmissions in the data subframe using either centralized scheduling mechanisms or distributed scheduling mechanisms. To schedule these transmissions the nodes use MAC management messages which are transmitted in the schedule control subframes. In this section we look at the message exchange and procedure used to schedule data transmissions. We will only detail distributed scheduling mechanism because in our simulations we consider Mesh networks auto-configured and independent of Base Station control. The nodes are all subscriber stations (SS). There is no need to centralize scheduling in one single node (BS) because the entire network is responsible for scheduling the internal data transmissions.

In Subsection 3.3.1 we discuss the coordinated distributed scheduling mechanism and in Subsection 3.3.2 we elaborate on the uncoordinated distributed scheduling mechanism.

3.3.1 Coordinated Distributed Scheduling

Coordinated distributed scheduling is used by nodes in the Mesh network to determine their transmission schedules in a decentralized manner. Coordinated distributed scheduling enables the nodes to schedule their transmissions such that they do not collide with the data transmissions scheduled by other nodes in the Mesh network. The IEEE 802.16 standard defines the MSH-DSCH (Mesh distributed scheduling messages) and other information elements which are transmitted with the MSH-DSCH message. These messages and data structures allow nodes to propagate information about scheduled transmissions (requests and grants), slots available for scheduling further transmission (available resources), to other nodes in the neighborhood. MSH-DSCH messages are used for both coordinated distributed scheduling as well as for uncoordinated distributed scheduling. Fig. 9 outlines the structure of the MSH-DSCH message.

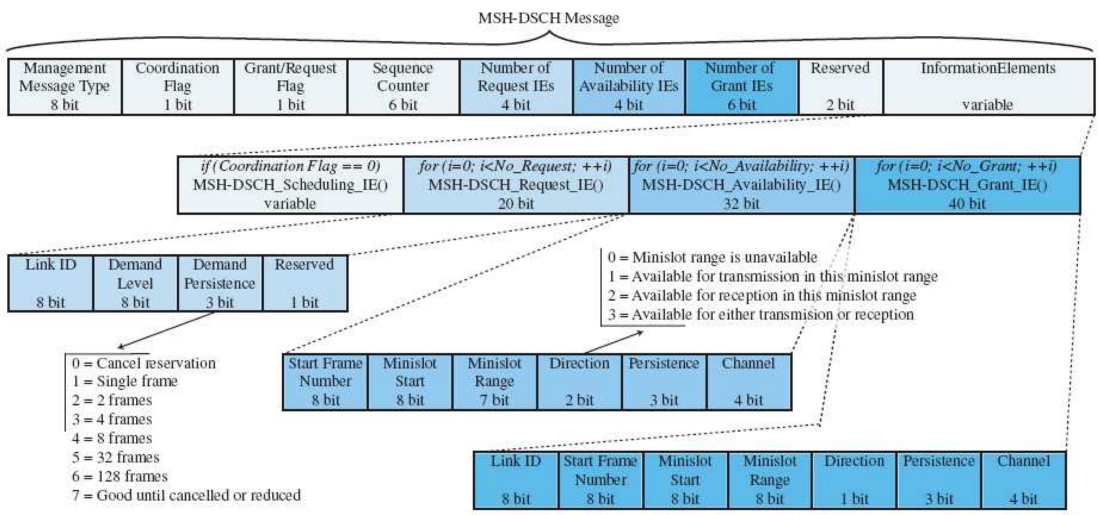


Fig. 9 - MSH-DSCH message structure [6]

The MSH-DSCH message contains a set of information elements (also denoted in short as *IEs*), which are data structures that store a particular types of information. These information elements help a node to schedule its own transmissions and notify the neighboring nodes of its own schedule. These information elements play a crucial role in distributed scheduling (both coordinated as well as uncoordinated). Considering the crucial role of these information elements in distributed scheduling, the next sub-subsections, address the individual information elements in detail.

3.3.1.1 MSH-DSCH Information Elements (IEs)

Fig. 9 shows the significant information elements which may be included in the MSH-DSCH message.These are as follows:

- MSH-DSCH_Request_IE: The MSH-DSCH_Request_IE (*R_IE*) or request is used by the node to specify its bandwidth demand for a particular link. As seen from Fig. 9, the request specifies the link (via the *Link ID*) for which bandwidth is required. The value for the field *Demand Level* and *Demand Persistence* are used to quantify the bandwidth required. The value for the field *Demand Level* specifies the number of minislots required in a frame to satisfy the bandwidth demand (assuming the current burst profile). The value of the field *Demand Persistence* helps to specify the number of consecutive frames for which the demanded minislots are required. The nodes use a three-way handshake for scheduling transmissions (reserving bandwidth for transmissions) on a link. The request is the first message exchanged in the three-way handshake. We look into the details of the three-way handshake later.

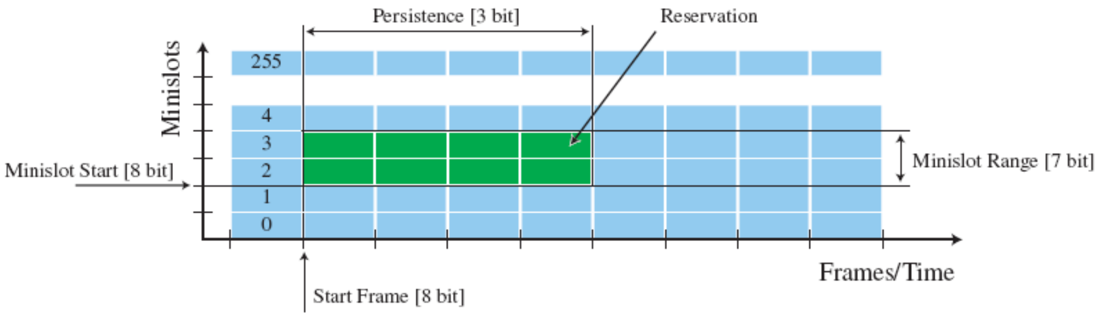


Fig. 10 - State map of the nodes that can be represented in A_ IEs or in G_IEs [6]

- MSH-DSCH_Availability_IE: The MSH-DSCH_Availability_IE (*A_IE*), or availability, helps the node to convey to its neighbors the status of individual minislots (over a number of consecutive frames). An availability information element specifies the status of a two-dimensional (frames, minislots) block of minislots (see Fig. 10) The starting frame of the block is identified by the field *Start Frame Number*, whereas the number of consecutive frames covered in the block is specified by the field *Persistence*. The values for the field *Persistence* and their meaning is similar to that for the *Persistence* field in the request information element. The second dimension of the block is specified by a range of consecutive minislots. In order to specify this range, the availability information element uses the fields *Minislot Start* and *Minislot Range*. The value for *Minislot Start* specifies the start position of the minislot range specified by the availability within a frame. The *Minislot Range* specifies the number of consecutive minislots specified in the current availability starting from the specified start minislot position. The *Direction* field helps to indicate the status of the minislot with respect to data transmissions which may be scheduled in the minislots specified by the availability. As shown in Fig. 9, the *Direction* (status) of a minislot can be either *Available*, *Receive Available*, *Transmit Available* or *Unavailable*. Slots (minislots) with status *Available* may be used for scheduling both transmission of data to neighboring nodes as well as reception of data from neighboring nodes. Slots having status *Receive Available* may only be used for scheduling reception of data from neighboring nodes. Slots having status *Transmit Available* may be used only for scheduling transmissions to neighboring nodes. Slots with status *Unavailable* may neither be used for scheduling transmissions to neighbors, nor for scheduling reception of data from neighbors. Table 2 summarizes the information about the slot status and its interpretation. The *Channel* field represents the logical channel (maps to a physical frequency channel) specified by the availability information element.

  Each node maintains internally the state of all the minislots via a set of availability information elements. The *Availability IEs* are normally sent at the end of the bandwidth request-grant procedure to inform the neighborhood about the current status of the granted slots.

Minislot *status* (represented using *Direction* field in an A_IE)	**Scheduling possible in slot**	**Possible reason slot status**
Available	both transmission as well as reception can be scheduled	default initial status of a slot, implies that none of the neighbors of the node have scheduled transmissions or reception of data in the slot
Receive Available	only reception of data can be scheduled	implies that at least one neighbor has scheduled reception of data in the minislot
Transmit Available	only data transmission can be scheduled	implies that at least one neighbor has scheduled data transmission in the slot
Unavailable	slot cannot be used to schedule further transmission or reception of data	implies that the node itself has either scheduled data transmission or reception in this slot, or at least one neighbor has scheduled data transmission and at least one neighbor has scheduled data reception in this slot

Table 2 – Minislot *status* interpretation

- MSH-DSCH_Grant_IE: The MSH-DSCH_Grant information elements (*Grnt_IEs*) are used for sending grants in response to a bandwidth request as well as for sending a confirmation (grant confirmation) for a received bandwidth grant. The field *Direction* in the grant information element helps to distinguish between a bandwidth grant and a grant confirmation. Please note the *Direction* field in the grant information elements (grant as well as confirm) does not have the same interpretation as the *Direction* field in the MSH-DSCH_Availability_IE.

  As shown in Fig. 9, the grant information element, in addition to the *Direction* field consists of the fields: *Link ID*, *Start Frame Number*, *Minislot start*, *Minislot range*, *Persistence*, and *Channel*. To enable the neighbors to know to whom the grant information element is addressed the transmitting node includes the transmit *Link ID* in the field *Link ID*[4] The field *Start Frame Number* specifies the starting frame number for the validity of the grant. The fields *Minislot start* and *Minislot range* together specify a consecutive set of minislots granted for the transmissions. The *Persistence* field is to be interpreted similar to the *Persistence* field for the availability information element and the request information element. The value 0 for the *Persistence* field implies a cancel grant information element. This is used by requester or granter to

[4] A node chooses a unique *Link ID* per neighbor, also called transmit link identifier (*Xmt Link ID*). The tuple (*Xmt Node ID, Link ID*) then uniquely address a neighboring node. The neighbors are informed about the chosen link identifier for transmissions to them via the link establishment protocol outlined in the standard.

cancel/reduce bandwidth reservations with *Persistence*7 (good until cancelled). No cancellation of bandwidth reserved with a *Persistence* less than good until cancel is possible.

The interpretation for the *Channel* field is similar to that for the *Channel* field in the availability information element.

3.3.1.2 Three-way handshake process for Reserving Bandwidth

As shown in Fig. 11 the bandwidth reservation mechanism relies on a three-way handshake (bandwidth request - bandwidth grant - bandwidth grant confirmation).

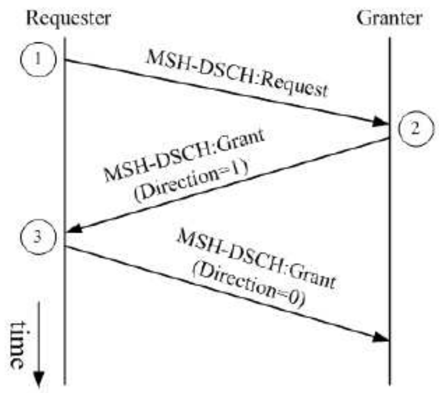


Fig. 11 – Three-way handshake process [5]

The steps in the three way handshake are:

(1) Request: The request is specified via the request information element (*R_IE*) described in Subsection 3.3.1-1). The transmitting node uses the *Link ID* to uniquely identify the link for which the node needs bandwidth. The number of minislots per frame and their *Persistence* is additionally specified. The *R_IE* is transmitted as part of the MSH-DSCH message.

The MSH-DSCH message also contains a set of availability information elements (*A_IE*) when a *R_IE* is present in the MSH-DSCH message. The *A_IEs* sent with the *R_IE*s indicate a two-dimensional block of bandwidth (frame-range, minislot-range) which may be used by the node transmitting the *R_IE* for the transmission for which bandwidth is requested. Let the set of *A_IEs* transmitted with the *R_IE* be denoted by *Xmt_AIE*. The node transmitting the bandwidth request is termed *requester* in the following.

(2) Grant: The MSH-DSCH message containing a *R_IE* is received by all the neighbors of the node which transmitted the request. The nodes then process the *R_IE* to determine the *Link ID* specified in the *R_IE*. The tuple (transmit node identifier (*Xmt Node ID*), *Link ID*) allows the neighbor to identify if the request is for bandwidth on a link directed to itself. The node to which the request is directed is termed granter in the following. The granter looks up its own set of availability information elements to select a subset of availabilities (range of slots and frames) where it is allowed to schedule reception of data transmissions from its neighbors.

Let us denote the latter set of availabilities by *Rcpt_AIE*. The granter must now choose a range of minislots from the set *Xmt_AIE* ∩ *Rcpt AIE* for the grant which is sent to the requester. The number of minislots per frame and their *Persistence* for the grant is chosen so as to satisfy the request. This is a range of minislots which can be reserved for the transmission in question without disturbing other already scheduled data transmissions.

The grant is received by all the neighbors of the granter. These then update their availability status to reflect the scheduled reception of data indicated by the grant. The grant is specified using a *Grnt_IE* with the direction bit set to 1. We denote grant by *G_IE*. To allow a neighboring node to decide if the grant is directed to it the granter sets the *Link ID* field in the *Grnt_IE* to the *Xmt Link ID* corresponding to the neighbor being addressed, i.e., the *Link ID* for the link (*granter*, *requester*).

(3) Grant Confirmation: The requester transmits a MSH-DSCH message containing a grant confirmation (i.e. a grant information element *(Grnt_IE)* with the direction bit set to 0). We denote a grant confirmation by *GC_IE*. The *GC_IE* and the corresponding *G_IE* have all fields similar except for the direction and the *Link ID*. The *Link ID* in the *GC_IE* corresponds to the *Link ID* for the link (*requester,granter*). The grant confirmation informs all the neighbors of the requester of the scheduled transmission. The neighbors then update their availabilities to reflect the newly scheduled transmission. Transmission of data in the reserved slots is allowed only after the transmission of the grant confirmation.

The three-way handshake is also used for canceling bandwidth reserved with *Persistence* 7 (good until cancelled). The request for a cancel can be identified that it specifies a number of minislots to be cancelled where the *Persistence* field in the *R_IE* has a value 0. The grant and the grant confirmation also have value 0 for the *Persistence* field when canceling bandwidth reserved with *Persistence* 7. Bandwidth reserved with a *Persistence* less than 7 cannot be cancelled. We denote a grant cancel and a grant cancel confirmation by *G_IE* and *GC_IE* respectively (to distinguish them from regular grants and grant confirmations).

3.3.1.3 Updating the Availabilities

The set of *A_IEs* maintained by the nodes represents the state of the resources (minislots) available at the nodes. If the availabilities are improperly updated then they can lead to an inconsistent view of the resources available at neighboring nodes, which in turn can lead to conflicting schedules. Therefore it is vital that the availabilities are consistent at the nodes. Initially the nodes start with either a single *A_IE* or a couple of *A_IEs* in their set of availabilities. These initial *A_IEs* cover the entire range of minislots available (for distributed scheduling) in the data subframe and have a *Persistence* 7. The status of the minislots is represented by the direction field of the *A_IEs*, which is set to *Available* (since initially all the slots may be used for scheduling both data transmissions as well as reception of data). If the node itself schedules a data transmission (i.e. the requester) or a data reception (i.e. the granter), then the corresponding minislots are marked as *Unavailable*. Nodes which are themselves not involved in the scheduled transmission also need to update the status of their availabilities to avoid scheduling conflicting data transmissions or receptions. We denote the latter nodes as passive nodes with respect to the transmission being scheduled (i.e. the nodes in the neighborhood of the requester and those in the neighborhood of the granter). A passive node which receives a MSH-DSCH with a grant knows that a node in its neighborhood has scheduled reception of data in the minislots specified by the corresponding *G_IE*. The passive node may then not schedule transmissions in the latter minislots as this would lead to collisions.

Thus based on the current status of these minislots at the passive node, the new status is decided such that transmission is not allowed by the new status of the minislots. This implies one of the following transitions in the status of the minislots on processing the *G_IE*: (current status of minislot(s) → new status of minislot(s))

- Available → Receive Available

- Receive Available→ Receive Available
- Transmit Available→ Unavailable
- Unavailable → Unavailable

Similarly, passive nodes which receive a MSH-DSCH message with a grant confirmation (*GC_IE*) know that a node in their neighborhood has scheduled transmission of data in the minislots specified be the corresponding *G_IE* . This implies that they may not schedule reception of data in the latter minislots as this would lead to collisions.

Thus, the passive nodes receiving the grant confirmation (*GC_IE*) will update the status of the corresponding minislots based on the current status of the minislots. The new status is such that reception is not possible. This leads to the following minislot status transitions on processing the *GC_IE*: current status of minislot(s) → new status of minislot(s))

- Available → Transmit Available
- Receive Available→ Unavailable
- Transmit Available→ Transmit Available
- Unavailable → Unavailable

The status of the slots needs to be changed also when minislots reserved for a scheduled transmission are freed (via cancel request — grant cancel — grant cancel confirmation, three-way handshake). The nodes involved in the handshake as well as the passive nodes need to update their availabilities in order to maintain a consistent picture of the resources available at the nodes. The slot state transitions on reception of a grant cancel or grant cancel confirmation are the inverse of the transitions on reception of a grant or grant confirmation, with certain restrictions. The slot state transitions on reception of a grant cancel or grant cancel confirmation may not lead to a new state for the slot which allows scheduling conflicting transmissions.

3.3.2 Uncoordinated Distributed Scheduling

The mechanisms involved in uncoordinated distributed scheduling are similar to those for coordinated distributed scheduling. The only difference between the two is that all the control messages (MSH-DSCH) required for the three-way handshake in uncoordinated distributed scheduling are exchanged in the data subframe. Thus, the request and the grant confirmation are sent by the requester in minislots reserved for the link (requester, granter). The grant is transmitted by the granter in minislots reserved for the link (granter, requester). This, naturally, implies that not all the passive neighbors of the requester and granter are able to overhear the message exchange (none of the other passive neighbors will be able to interpret the handshake if the links are encrypted). This in turn means that the passive neighbors may schedule conflicting transmissions. Hence, it is recommended that uncoordinated distributed scheduling be used only for scheduling short term transmissions. The transmissions scheduled via uncoordinated scheduling should not lead to a conflict with the schedules established via coordinated distributed scheduling.

3.4 Summary

In this chapter we outlined the main characteristics of 802.16 Mesh process such as: network initial configuration; distribution election procedure to access transmission opportunities by control messages as well as schedule messages; scheduling mechanisms for access data subframe and three-way handshake for reserving bandwidth to transmit data over network nodes.

Chapter 4

4 QoS support in the IEEE 802.16 Mesh mode

In this chapter we first provide an overview of the QoS support mechanisms specified in the standard for the PMP mode and next an overview of those for the Mesh mode.

4.1 QoS support in the 802.16 PMP mode

Quality of service is provisioned in the PMP mode on a per-connection basis. All data, either from the SS to the BS or vice versa is transmitted within the context of a connection, identified by the connection identifier (CID) specified in the MAC protocol data unit (PDU). The CID is a 16-bit value that identifies a connection to equivalent peers in the MAC at both the BSs as well as the SSs. It also provides a mapping to a service flow identifier (SFID). The SFID defines the QoS parameters which are associated with a given connection (CID). The SFID is a 32-bit value and is one of the core concepts of the MAC protocol. It provides a mapping to the QoS parameters for a particular data entity.

Fig. 12 shows the core objects involved in the QoS architecture as specified in the standard for the PMP mode. Each MAC PDU is transmitted using a particular CID, which is in turn associated with a single service flow identified by a SFID. Thus, many PDUs may be transmitted within the context of the same service flow but a single MAC PDU is associated with exactly one service flow. Fig. 12 also shows that there are different sets of QoS parameters associated with a given service flow. These are the *ProvisionedQoSParamSet*, *AdmittedQoSParamSet*, and *ActiveQoSParamSet*. The *provisioned parameter set* is a set of parameters provisioned using means outside the scope of the 802.16 standard, such as with the help of a network management system. The *admitted parameter set* is a set of QoS parameters for which resources (bandwidth, memory, etc.) are being reserved by the BS (SS). The *active parameter set* is the set of QoS parameters defining the service actually being provided to the active flow. For example, the BS transmits uplink and downlink maps specifying bandwidth allocation for the service flow's active parameter set. Only an active service flow is allowed to transmit packets. To enable the dynamic setup and configuration of service flows, the standard specifies a set of MAC management messages called dynamic service messages (DSx *messages*). These are the dynamic service addition (DSA), dynamic service change (DSC), and the dynamic service deletion (DSD) messages. The various QoS parameters associated with a service flow are negotiated using these messages.

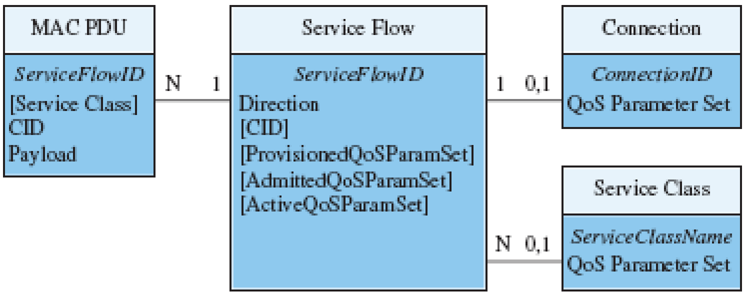


Fig. 12 - QoS object model for IEEE 802.16-204 PMP mode [7]

Typical service parameters associated with a service flow are *traffic priority*, *minimum reserved rate*, *tolerated jitter*, *maximum sustained rate*, *maximum traffic burst*, *maximum latency*, and *scheduling service*. The BS may optionally create a service class as shown in figure. A service class is a name given to a particular set of QoS parameters, and can be considered as a macro for specifying a set of QoS parameters typically used. The value for the scheduling service parameter in the QoS parameter set specifies the data scheduling service associated with a service flow. The 802.16 standard currently defines the following data scheduling services: *unsolicited grant service* (UGS), *real-time polling service* (rtPS), *non-real time polling service* (nrtPS), and *best effort* (BE). The UGS is meant to support real-time data streams consisting of fixed size data packets issued periodically. The rtPS is meant to support data streams having variable sized data packets issued at periodic intervals. The nrtPS is designed to support delay tolerant streams of variable sized data packets for which a minimum data rate is expected. The BE traffic is serviced on a space available basis. For service flow associated with the scheduling service UGS, the BS allocates a static amount of bandwidth to the SS in every frame. The amount of bandwidth granted by the BS for this type of scheduling service depends on the maximum sustained traffic rate of the service flow. For rtPS service flows, the BS offers real-time, periodic, unicast request opportunities meeting the flow's requirements and allowing the SS to request a grant of the desired size. For nrtPS the BS, similar to the case of a rtPS service flow, offers periodic request opportunities. However, these request opportunities are not real-time, and the SS can also use contention based request opportunities in addition to the unicast request opportunities for a nrtPS service flow as well as the unsolicited data grant types. For a BE service flow no periodic polling opportunities are granted. The SS uses contention request opportunities, unicast request opportunities and unsolicited data grant burst types. Table 3 shows the QoS specifications for 802.16 PMP service classes and corresponding applications.

QoS Category	Applications	QoS Specifications
UGS Unsolicited Grant Service	VoIP	• Maximum Sustained Rate • Maximum Latency Tolerance • Jitter Tolerance
rtPS Real-Time Packet Service	Streaming Audio or Video	• Minimum Reserved Rate • Maximum Sustained Rate • Maximum Latency Tolerance • Traffic Priority
nrtPS Non-Real-Time Packet Service	File Transfer Protocol (FTP)	• Minimum Reserved Rate • Maximum Sustained Rate • Traffic Priority
BE Best-Effort Service	Data Transfer, Web Browsing, etc.	• Maximum Sustained Rate • Traffic Priority

Table 3 - QoS specifications for PMP mode as referenced in the standard [4]

To summarize, the PMP mode provides the BS with efficient means to manage the bandwidth optimally and at the same time satisfy the requirements of the individual admitted service flows.

4.2 QoS support in the 802.16 Mesh mode

In contrast to the PMP mode, the QoS in Mesh mode is provisioned on a packet-by-packet basis. Thus, the per-connection QoS provisioning using the DSx messages as introduced previously is not applicable. This design decision helps to reduce the complexity of implementing the Mesh mode.

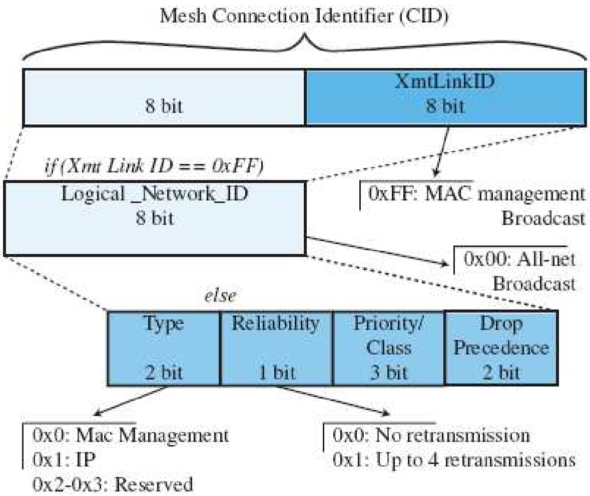


Fig. 13 - Mesh Connection Identifier (CID) [7]

The connection identifier (CID) in the Mesh mode is shown in Fig. 13. The Mesh CID is used to differentiate the forwarding service a PDU should get at each individual node. As can be seen from Fig. 13 it is possible to assign a priority to each MAC PDU. Based on the priority, the transmission scheduler at a node can decide if a particular PDU should be transmitted before another. The field reliability specifies the number of retransmissions for the particular MAC PDU (if needed). The drop precedence specifies the dropping likelihood for a PDU during congestion. Messages with higher drop precedence are more likely to be dropped. In effect, QoS specification for the Mesh mode is limited to specifying the priority of a MAC PDU, the reliability and its drop precedence. Given the same reliability and drop precedence and MAC PDU type, the MAC will attempt to provide a lower delay to PDUs with higher priority. The above QoS mechanism, however, does not allow the node to estimate the optimal bandwidth requirement for transmissions on a particular link. This is because (just based on the above interpretation as presented in the 802.16 standard), the node is not able to identify the expected arrival characteristics of the traffic and classify it into the different categories as traffic requiring UGS, rtPS, nrtPS or BE service.

Resuming, QoS mechanisms in the Mesh mode are not consistent with those provided for the PMP mode. In addition, the per-packet QoS specification for the Mesh mode does not allow a node to optimally estimate the amount of bandwidth required for transmission on a link, as no information about the data scheduling service required for the traffic is included explicitly in the QoS specification in the Mesh CID.

4.3 QoS architecture for the 802.16 Mesh mode

Fig. 14 shows the QoS architecture for efficient management of bandwidth in Mesh mode as introduced by [7]. This QoS architecture was adapted in our implemented simulation module[1].

The module *Packet Classifier* shown in the figure provides the functionality of the service-specific convergence sublayer. Table 4 shows the mapping used to classify traffic from the network layer using the IP TOS field and the corresponding values assigned to fields of the Mesh CID by *Packet Classifier*. Based on the values for the fields priority, drop precedence, and reliability is used the mapping shown in table to identify the scheduling service (UGS, rtPS, nrtPS or BE) for the data packets.

Network Layer Priority (e.g., IP Type of Service)	802.16 Service Class	802.16 MSH CID Priority/Class	802.16 MSH CID Drop Precedence	802.16 MSH CID Reliability
0 1	Best Effort (BE) BE	0 1	3 3	0 1
2 3	non-realtime Polling Service (nrtPS) nrtPS	2 3	2 2	0 1
4 5	real-time Polling Service (rtPS) rtPS	4 5	1 1	0 1
6 7	Unsolicited Grant Service (UGS) UGS	6 7	0 0	0 1

Table 4 - Mapping the IP type of service (TOS) to mesh CID and Service Class [7]

[1] In APENDICES IV, V and VI we show the headers of the main functions of the three QoS MAC control modules, Coordinator Management, Bandwidth Management and Data Management respectively.

A similar mapping function may be implemented for other network protocols. After classification of data received from the upper layers, the packets are sent to the *Data Management Module* as shown in Fig. 14.

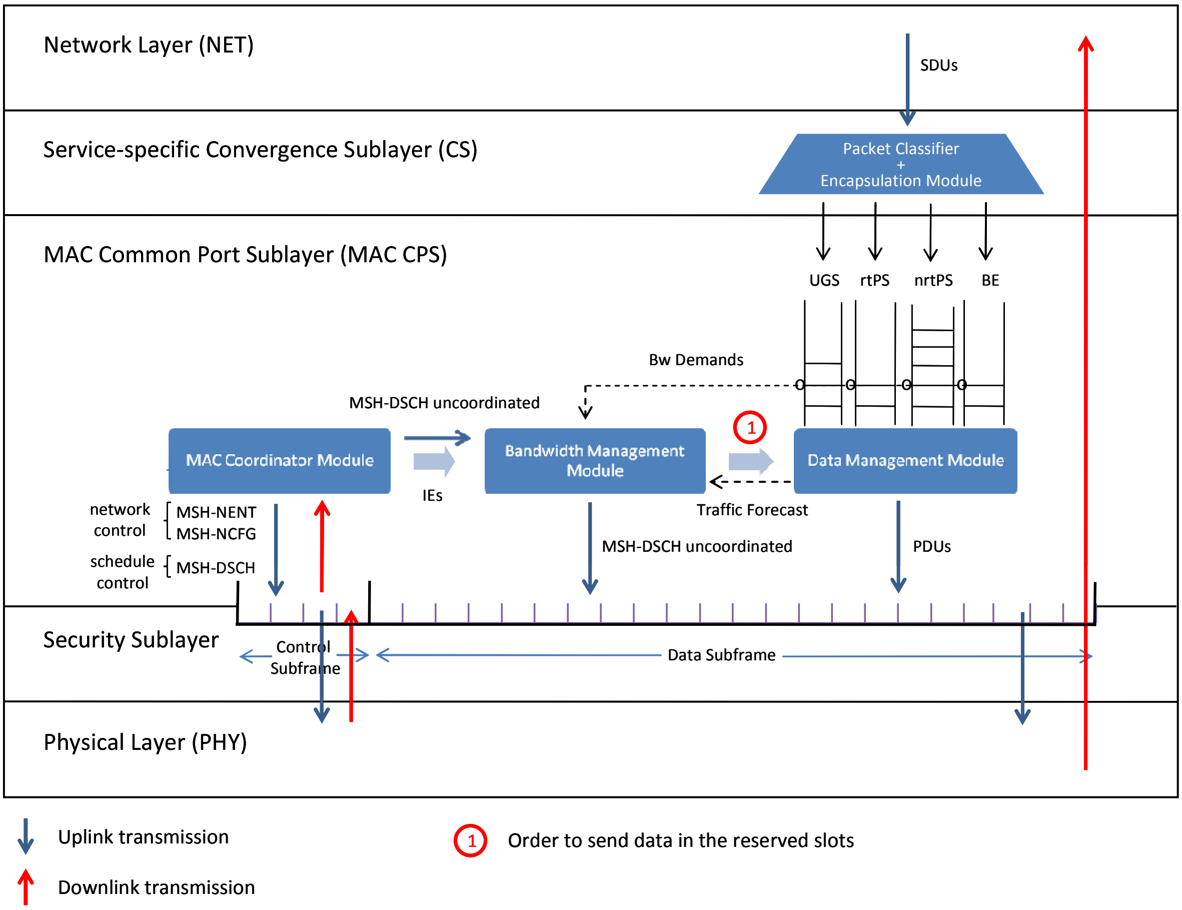


Fig. 14 – The adapted QoS architecture

The *Data Management Module* enqueues the arriving packets in the corresponding queue. Based on the congestion situation, it can also decide which packets may be dropped. The *Data Management Module* keeps an account of the minislots reserved for transmission for each link to a neighbor at a node. It then sends the appropriate data packet from its queues for transmission on the wireless medium to the lower layer in a minislot reserved for transmission. In addition this module keeps a running estimate of the incoming data rate in each queue and, based on the policy to be implemented, notifies the *Bandwidth Management Module* of the current bandwidth requirements for each class of traffic.

The *MAC Coordinator Module* shown in Fig. 14 is responsible for handling all kinds of MAC management messages. It handles MAC management messages received from the lower layer. If the MAC management message corresponds to a bandwidth request or a grant or grant-confirmation, this module updates the respective internal tables and extracts the relevant parameters (information elements, *IEs*, contained in the message). These parameters are then sent to the *Bandwidth Management Module* for further processing when required. In addition, it is also responsible for processing MAC management messages received during the network control subframe. This module maintains information about the schedules of the neighbors, the node identifiers of the neighbors, details about the physical two-hop neighborhood, the *Link IDs* assigned for transmission to and reception from each neighboring node. The *MAC Coordinator Module* is responsible for executing the Mesh election algorithm specified in the standard to decide if management messages may be transmitted in a given transmission opportunity in the control subframe.

The *Bandwidth Management Module* is responsible for generating bandwidth requests when more bandwidth is required, or generating cancel requests to free bandwidth when it is no longer required. It is also responsible for processing bandwidth requests received from the neighboring nodes and taking appropriate action when a grant or grant-confirmation is received. All the above request, grants and grant-confirmations are sent as information elements within a MSH-DSCH. The *Bandwidth Management Module* receives information about instantaneous bandwidth demand from the *Data Management Module*. The *Bandwidth Management Module* maintains internally a set of MSH-DSCH *Availability IEs*. The complete set of MSH-DSCH *Availability IEs* describes the local status of individual minislots over all frames in the future. When generating a MSH-DSCH message to request bandwidth for transmission, the *Bandwidth Management Module* creates a MSH-DSCH Request IE describing the amount of minislots required (specified by the demand level field in the MSH-DSCH Request IE) and the number of frames over the bandwidth is required (denoted by the demand persistence field in the MSH-DSCH Request IE). Besides handling the requests, availabilities and grants, this module also manages MSH-DSCH to be transmitted on data subframe (uncoordinated distributed scheduling).

Finally, the *Bandwidth Management Module* send transmissions orders to the *Data Manage Module* to scheduling output PDUs in reserved slots (in data subframe). In our adapted architecture the *Bandwidth Management Module* is the core module of the MAC layer control (consult in APPENDIX V the main header functions of this module).

Due to the classification of traffic into the different scheduling services, the *Bandwidth Management Module* is able to estimate the arrival characteristics of traffic and make an intelligent choice for the persistence value to be sent with the request. As an example, the *Bandwidth Management Module* requests minislots with persistence 7 (good until cancelled or reduced, see Fig. 9 – Chapter 3) only when the data scheduling service associated with the traffic is UGS. These maps the UGS service provided in the PMP mode where a node receives a constant amount of bandwidth for the lifetime of the connection.

In the PMP mode the rtPS scheduling service is meant to support real-time data streams. To support such service in the Mesh mode one requires opportunities for requesting bandwidth in real-time. Using coordinated distributed scheduling a node, however, has to compete with other nodes in its two-hop neighborhood for transmission opportunities in which a bandwidth request can be sent. Nodes using distributed scheduling need to complete the three-way request/grant/grant-confirm handshake procedure before data can be transmitted using the reserved bandwidth. It is thus not possible to complete the handshake in real-time if it is used only coordinated distributed scheduling and the topology is highly connected.

Hence, as can be seen from Fig. 14 the *Bandwidth Management Module* sends messages for the rtPS (MSH-DSCH uncoordinated) in the data subframe using uncoordinated scheduling.. In addition, internally, to ensure a minimum delay, the traffic from the rtPS class can borrow (be transmitted in) bandwidth reserved for UGS traffic. UGS traffic can then borrow bandwidth back from the reserved bandwidth for the rtPS class as soon as the uncoordinated scheduling handshake is over. A characteristic of rtPS is that it has a variable bit rate. Thus, it is highly inefficient to request a fixed amount of slots for transmission for rtPS with persistence 7. This may lead to many of these slots being unused in many frames. As a solution, it is used an estimation of the number of slots required per frame to send the arriving rtPS data, and request those slots with a persistence

5 (reservation is valid for 32 frames). Using uncoordinated scheduling to reserve bandwidth for a long term is not recommended as it may lead to collisions.

For the nrtPS class we require periodic request opportunities, which need not be in real-time. nrtPS traffic is moreover delay tolerant. Thus we can use an estimator to find out the amount of minislots required per frame and send requests with a persistence smaller than 7. As a result, we can periodically (using transmission opportunities in the schedule control subframe) reserve the exact amount of bandwidth required for transmitting nrtPS data. The BE service is very similar to the nrtPS service with the difference that it is served on a space-available basis. Thus, for BE the estimated number of minislots is reserved with a persistence less than 7. The difference to nrtPS is that traffic belonging to UGS and rtPS are allowed to borrow bandwidth reserved for BE traffic.

On receiving a request, the *Bandwidth Management Module* is also responsible for processing the request to find a mutually suitable set of slots for a grant which is able to satisfy the request. A poor choice for the grant would be for example a grant starting at a frame before the three-way handshake can be completed, this means that the slots in that range will remain unused (data transmission using the granted slots may not start till the three-way handshake is complete as required by the standard). On the other hand, if the grant starts from a frame much in the future after completion of the three-way handshake it leads to additional delay before transmission can start.

The *Bandwidth Management Module* is also responsible for maintaining an up to date status of the MSH-DSCH *Availability IEs* stored locally at a node. This involves updating the status when receiving or transmitting either a grant or grant confirmation.

4.4 Summary

This chapter introduced the standard specification for achieving QoS in PMP mode and also the open packages for the QoS Mesh mode. We still described an architecture for achieving quality of service in this operation mode. This architecture, originally presented in [7], was adapted for our implemented Mesh QoS module.

Chapter 5

5 Implementation of Mesh QoS

In this chapter we will present the developed QoS architecture. The main objective of this development was to achieve a differentiation of traffic classes and QoS parameters as those provided by standard PMP mode.

The starting point was the QoS architecture outlined in [7]. In our work we have proposed an adaptation of that architecture to the simulator project (ns2mesh80216) provided by University of Pisa and Georgia Institute of Technology [13].

In the following sections we will introduce the operation mode of the ns2mesh8016 simulator, where we carried out the implementation of the QoS architecture, followed by the details of this implementation in section 5.3. Finally in section 5.4, we show the results an performance tests obtained in our simulation.

5.1 Network Simulators with Support for IEEE802.16 Mesh Mode

In the beginning of this work there were only two simulators that provided support for Mesh topology of the IEEE802.16 standard (only for fixed access). Those are NCTUns [11] and ns2mesh80216 [13].

We have worked with ns2mesh80216 because it runs on the popular Network Simulator NS-2 [12] and provides a simpler language, based on c++/.tcl. Additionally it is used by an extensive community of telecommunications researchers and developers that provide a wide support and feedback.

The NS-2 is UNIX based and operates on multiple systems: Linux, Mac OS X, Cygwin (Windows). On the other hand the NCTUns runs only on Linux Fedora/Red Hat v9.

5.2 The ns2mesh80216 Project

5.2.1 Introduction

The ns2mesh80216 consists on a patch to the NS-2 (v2.31) that allows IEEE 802.16 wireless Mesh networks simulation. This extension does not support Point-to-Multipoint simulations.

The functions for enabling data transmission at MAC layer are fully implemented. Access to the data sub-frame is negotiated by means of the three-way handshake specified by the standard, while scheduling is implemented according to the Fair End-to-end Bandwidth Access (FEBA) algorithm [8]. Access to the control sub-frame is also implemented according to the standard distributed election procedure.

The medium access controller sublayer (MAC) is not interoperable with NS-2 routing algorithms and physical interference modules. Instead, base classes for routing and wireless channel modeling are specified by means of the Shortest-Path-First and Protocol-Model reference implementations. For instance, there is no way to specify the (x, y) position of the nodes (unlike 802.11 simulations in NS-2), because links between nodes are logical.

For the simulations scenarios, there are a few pre-concepted mesh topologies provided on patch. Some examples are shown in Fig. 15.

To run simulations with, e.g., shadowing, and path-loss, it is necessary to extend the Channel class provided by simulator in order to support the desired model.

This extension does not include any QoS algorithm to achieve efficient bandwidth management for data reservation/transmission process. It only provides priorities schemes for handling exchanged traffic flows (i.e. WFQ, FIFO).

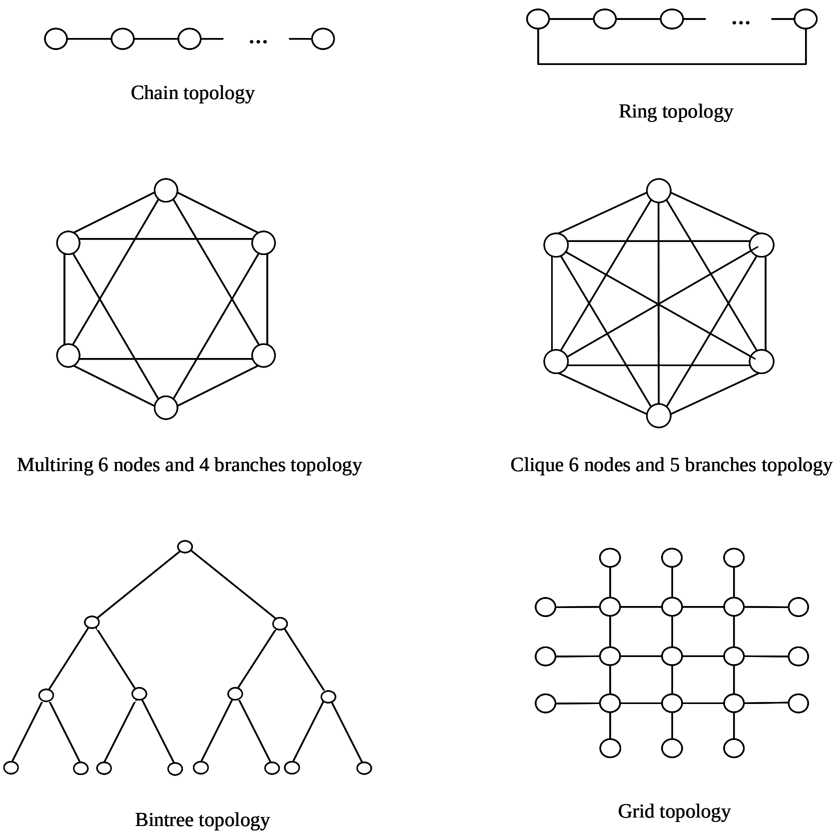


Fig. 15 – Examples of topologies implemented on ns2mesh80216 simulator

5.2.2 Ns2mesh80216 Bandwidth Reservation Process

The ns2mesh80216 defines a continuous model for reservation of the amount of requested slots. The node which makes the request of bandwidth sets the *demand level* (Request_ISs) corresponding to the number of packets waiting for transmission in its internal buffer.

In the ns2mesh8016 approach, the requester node always sends the *demand persistence* field (number of frames for which request is valid) as a null value.

When it receives the request message, the grant node is responsible for analyzing the bandwidth spectrum to search the maximum amount of earliest contiguous slots for reserving to this transmission. When the granter reaches the amount of requested slots it sends back information elements of reserved slots. If the number of granted/reserved slots extends over several (contiguous) frames, the granter uses the *persistence* field (in Grant_IE) to send in grant message a single Grant_IE entry instead of multiples Grant_IEs with the same *Minislot Start* and *Minislot Range* values.

When the requester node receives the response (grant message) and sends the confirmation message it probably has more packets to transmit than those required initially (if data flow is not reduced or stopped). This surplus of data packets only can be requested on next transmission opportunity which can take the time of several frames, depending of network density.

We observed in the described process that the transference of data is performed in burst periods. The requester/transmitter makes the multiples requests according to the amount of data in the output buffer, but between requests (transmission opportunities) there is a hold off transmission period, as shown in Fig. 16.

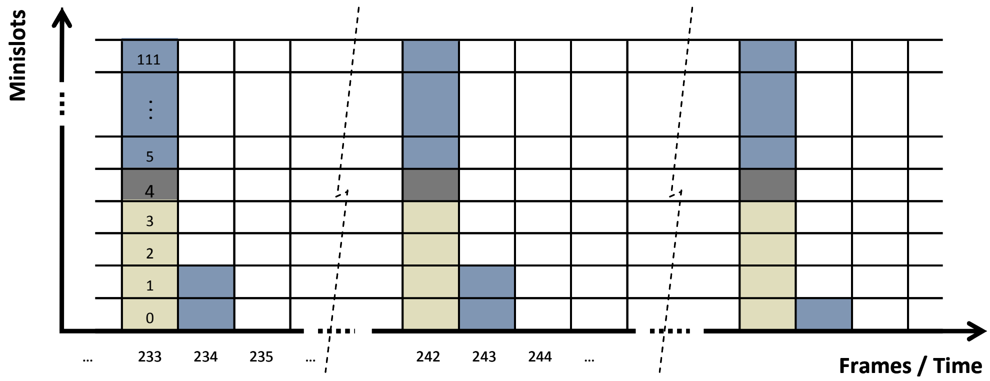


Fig. 16 – Map of granted slots in the ns2mesh80216

Thus we conclude that the bandwidth reservation module used by nsmes80216 simulator ensures the transmission of data according to the messages proposed in standard for the Mesh mode, but does not have the ability to specify QoS parameters such as jitter and delay for exchanged packets in the network.

5.3 Main changes to ns2mesh80216 project to support QoS architecture

In this section we will present the main changes implemented on the ns2mesh80216 model to support the QoS architecture described in chapter 4.

5.3.1 Incoming Traffic Forecast

As shown in our architecture diagram (Fig. 14) we implemented a function in the *Data Management Module* (see *recomputecbr()* header function in APPENDIX VI) to determine the amount of data that is coming to output buffer for each service class.

This value is computed in time intervals corresponding to 32 frames and then communicated to *Bandwidth Management Module*. If the calculated incoming traffic rate changes more than 20% or reduces to zero, the *Bandwidth Management Module* will be informed to proceed with the appropriate bandwidth request change or cancellation.

This forecast function is more suitable for type of services for which constant flow rates are expected, like UGS or nrtPS, but we used that for all services traffic forecast. So, the *Bandwidth Management Module* will know exactly the transmission needs for actives service flows in *Data Management Module* and will reflect them in more accurate bandwidth requests.

5.3.2 Changes to MSH-DSCH Messages

In order to internally schedule the requests, grants and confirmations for different types of service, in an independent way, we added a *Service Class* field (2 bits) to all Request_IE, Availability_IE, Grant_IE entries in each MSH-DSCH message. So, the scheduling control messages (MSH-DSCH) may contain multiple IEs entries referring to the different service flows, with the exception for the MSH-DSCH uncoordinated message that reports only rtPS service information. This message is sent in the data subframe, and must be as smaller as possible to not reduce the bandwidth of transmitting data channel.

The information elements (IEs) of each MSH-DSCH message are internally processed according to the corresponding service class priorities (first UGS and last BE), i.e., the best reservations opportunities are given to priority service classes.

Figures Fig. 17 and Fig. 18 show examples of two control messages containing multiples IEs assigned to different services obtained in our simulation module.

Fig. 17 represents three requests from node 0 to nodes 3 and 4 using the service UGS (demand level 26), nrtPS (demand level 15) and BE (demand level 10).

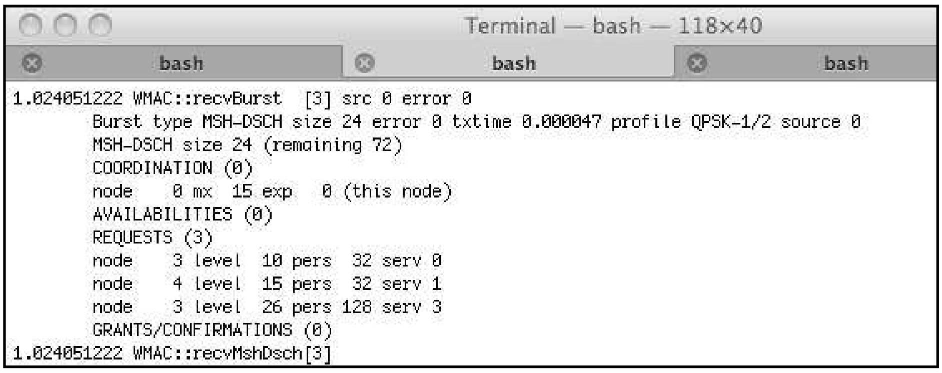


Fig. 17 – MSH-DSCH (request) message with multiple IE services

Fig. 18 (below) shows the response message sent from node 3 back to node 0.

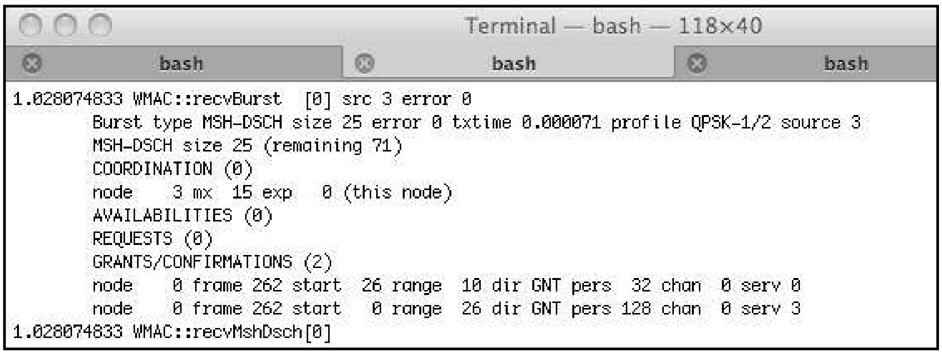


Fig. 18 – MSH-DSCH (grant) message with multiple IE services

5.3.3 Unsolicited Grant Service Grants

According to the specifications defined in 802.16 standard for UGS class in the PMP mode, the packets are of equal size and are sent at a constant bit rate (CBR).

In ns2mesh80216 simulator there is no distinction of service classes. The transmitter node sends a request every time it has an access opportunity, totaling the number of packets accumulated in the output buffer. This procedure is repeated throughout the simulation, whenever the requester has an opportunity to access the schedule control subframe.

In our QoS implementation the transmitter node can preview the amount of information to be sent and then reduces the multiples requests into one. It sets the Demand Persistence field with value 7 (good until cancelled) for UGS service or with value 5 (32 frames) for rtPS, nrtPS and BE services; and in the *Demand Level* field, it sets the forecast traffic rate obtained in the *Data Management Module*. See the example in Fig. 19 compared with the one in Fig. 16.

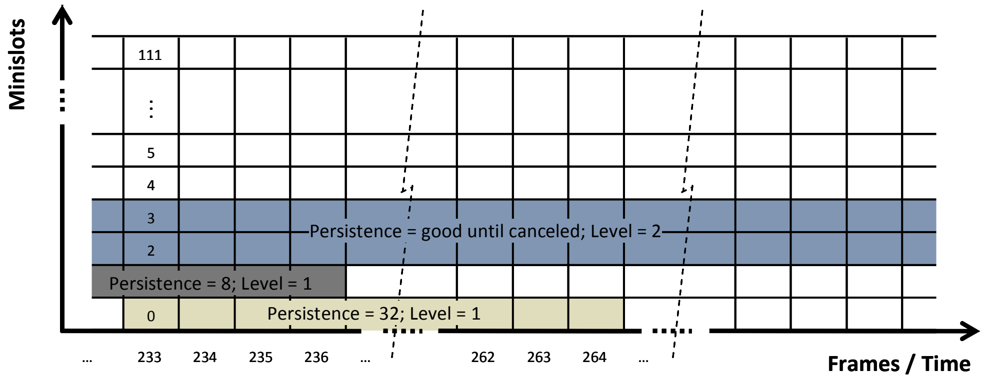


Fig. 19 – Map of granted slots in the QoS implementation

This way it will not be requested the amount of data that is waiting for transmission, but the amount of data to be effectively transmitted in each frame for achieving the desired transfer rate. So, we can reach constant delays (low jitter) and minimize the end to end delays for transmitted packets.

It is also intended that the UGS traffic has priority over nrtPS and BE. For this, the receiver node may have to cancel/discontinued reservations of these services if the bandwidth spectrum is full. If it is the case, the receiver sends in an Availability_IE the range of slots to be cancelled. The *Direction* value is set to zero to inform the transmitter node using these slots that those are now unavailable for transmission.

5.3.4 Real-time Polling Service Grants

To reduce more the end-to-end delay reached for UGS service and thus making possible the transference of voice and image in real-time, we had to reduce the time of tree-way handshake for the bandwidth reservation process.

The option, as referred previously in Chapter 4, was to send the scheduling control messages in the data subframe together with the other data traffic, instead of using the scarce number of control opportunities (control sublayer).

The uncoordinated scheduling was not implemented in the ns2mesh80216, so we assigned the *Bandwidth Management Module* for this process. He has to reserve 3 slots for each link between neighboring nodes and opportunely send the rtPS scheduling messages on these slots.

In this service we also used the bandwidth requests based on information provided by *Data Management Module* forecasts with *Persistence Level* 5 (reservation whit 32 frames of duration).

It can still happen that the receiver has to borrow (transmit in) resources allocated to nrtPS and BE services. In this case the procedure is similar to that outlined for the UGS service class.

5.3.5 Non-real-time Polling Service and Best Effort Grants

In these two services the reservations are made on a 32 frames basis changing the *Demand Level* depending on the traffic necessary.

The differences are on the priorities in scheduling their control messages (requests and grants). The BE available slots for transmission are restricted by the remaining resources after all the other services have made their reservations.

For the nrtPS service, it must be assured a minimum transfer rate (minimum number of slots per frame), defined by network operator. Thus the higher priority classes must take this factor into account when they need to cancel nrtPS reservations. Only the slots that exceed this bandwidth threshold can be canceled.

5.4 Performance Tests and Results

5.4.1 Simulation Environment

For the simulation tests we used the settings described in the IEEE 802.16 standard with *profP3_10* profile [1]. In particular, the bandwidth per channel was 10 MHz, and frames duration equal to 4 ms, including the control subframe and data subframe. The *XmtHoldoffExponent* parameter was set to 0. Further the control frame length was set to 4 slots and the 16-QAM-1/2 MCS profile was used for data modulation, this allowing achieving a transfer rate per channel of 13 Mb/s. Transmission errors in the physical layer were not considered, allowing us to focus essentially on the MAC layer performance.

We used the distributed scheduling for the Mesh network control and we stated the value of 100kB for input buffers of each traffic flow. The simulation tests were undertaken using the independent replication method [14]. For each scenario we run 10 independent simulations, each with 20 s of duration, having a stabilization/initial network configuration period of 2 s.

5.4.2 Performance Evaluation of UGS Service Class

In order to test the performance of UGS class, we used the topology configuration shown in Fig. 20, with 5 SSs all connected with each other, i.e. each SS is in range of the remaining 4, allowing only one-hop communication.

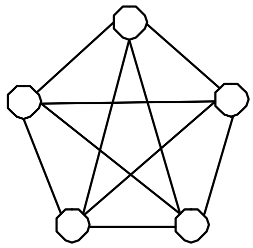


Fig. 20 – 5 nodes clique topology (multiring 5 nodes - 4 branches)

A constant bit rate (CBR) traffic of 1Mb/s was submitted to the network. As all the SSs are on the same conditions it is not relevant defining the transmitter or receiver node, since they are randomly assigned. We used different packet sizes in the various simulations, increasing up to 1700 bytes with 100 bytes intervals.

The average delay per packet obtained in this simulation is represented in Fig. 21.

We therefore noticed that the delay in CBR transmission can be substantially reduced by means of the prediction of the transmission needs and reservation of the same amount of slots in all frames (amount of data calculated for transmissions of 4 ms intervals). In the original model much time is spent with the request for reservation and following renegotiations.

It was also found that in general the delay increases with the packet size used, mainly in the QoS model.

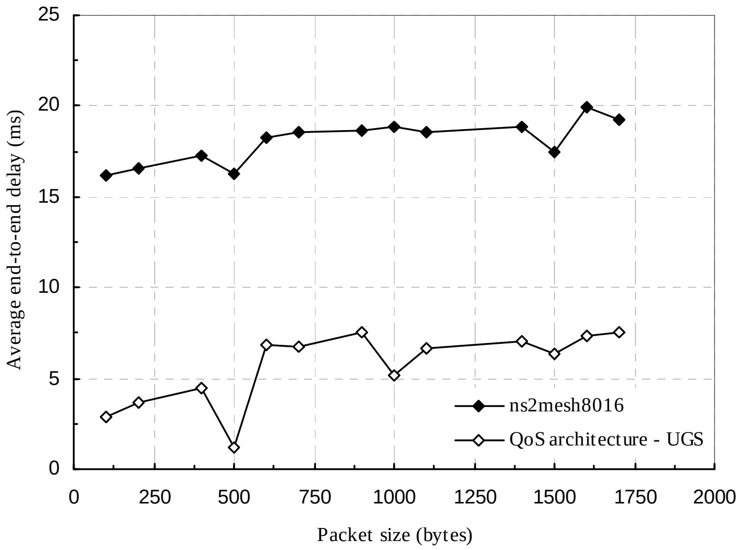


Fig. 21 – Average delay for the UGS service packets

In the above example, for flows of 1Mb/s, it is necessary to send 500 bytes (4000 bits) per frame (4 ms) to avoid throughput loss. The 500 bytes per frame are equivalent to 21 slots[1] per frame.

In the QoS model a reservation for 21 slots was made for an indefinite period.

If the packets are larger than 500 bytes, they have to be fragmented in several frames, which increase the end-to-end delay per packet. Thus, there is a direct relationship between the number of requested slots and the packet size used, as showed in Fig. 21 and further in Fig. 22. In this last example we used packets of 500 Bytes and changed the traffic flow rate. We observed that the minimum delay for packets exchanged was obtained for 1Mb/s flow rate as similarly as in Fig. 21.

[1] The conversion of bytes for slots is given by expression (bytes/α) + 1, with α = NR_BITS PER_SYM × NR_SYM_PER_SLOT / 8 (in bytes). In this case NR_BITS_PER_SYM = 200; NR_SYM_PER_SLOT = 1; α =25.

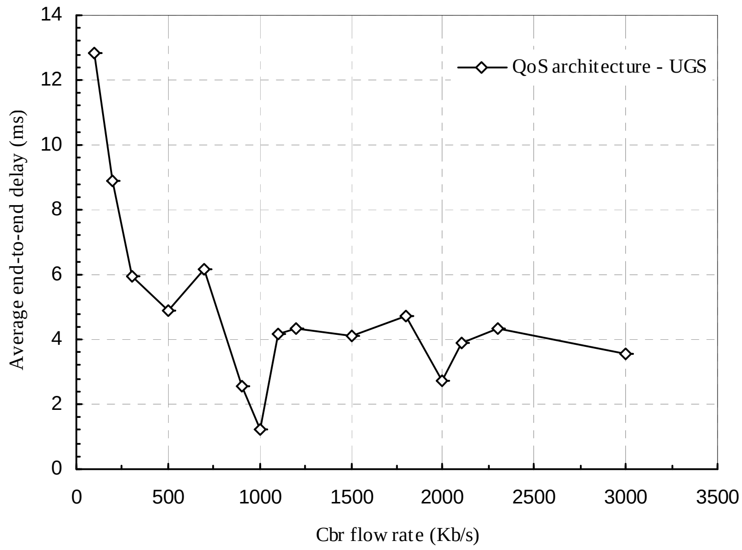


Fig. 22 – Delay for implemented UGS service class

This relationship can be optimized in order to minimize the total delay of traffic flows.

5.4.3 Performance Evaluation of rtPS Service Class

- **Simulation 1**

To test the performance of rtPS class, it was considered again the comparison with the original ns2mesh8016 model. In a first approach, it was used the chain topology represented in Fig. 23, in order to assess the performance evaluation according to the traffic flow length (number of hops in the network until reach the receiver).

We used *telnet*[2] traffic and changed the receiver node along the chain, ensuring that the transmitter node was the first node in the chain.

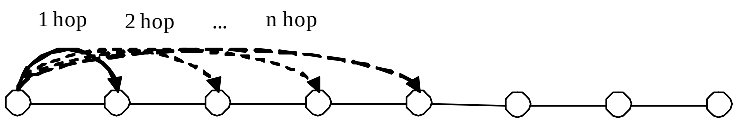


Fig. 23 – Chain simulation topology

[2] Telnet traffic is generated via the Telnet ns2 application and uses the NewReno flavor of TCP, with the default ns2 configuration parameters.

Fig. 24 shows that, even for the rtPS case, the delay per packet decreases when compared with the original model. The difference between both is relatively stable along with the number of jumps variation of the traffic flow. With the rtPS implemented model we achieved an average reduction of 93 ms.

Also note that rtPS flows that extend for more than two-hops reach values of delays not acceptable for real-time voice and images transmission (more than 100ms).

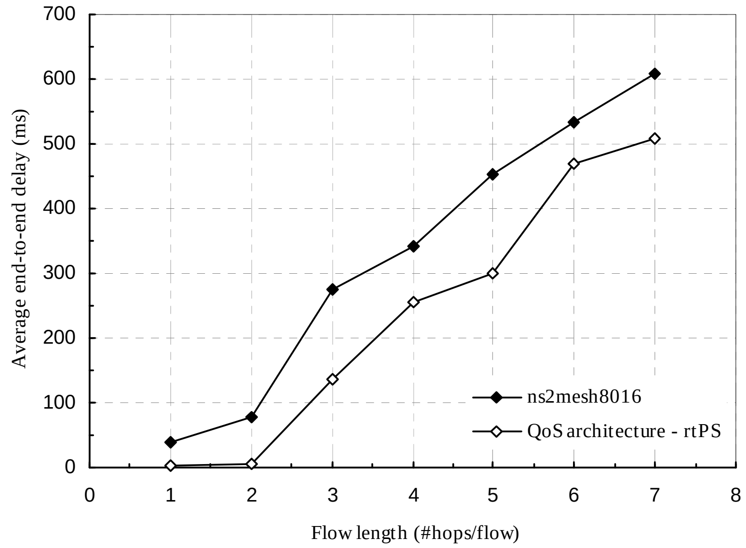


Fig. 24 – Delay for the rtPS service with increasing number of hops/traffic flow

- **Simulation 2**

In this simulation we tested the behavior of the rtPS service while increasing the mesh network density, i.e. increasing the number of neighbors per node. To do this we followed the multi-ring topology. We used this topology in order to successively increase the number of neighboring nodes and allowing flows of two-hop connections, as shown in

Fig. 25.

We used again *telnet* flows.

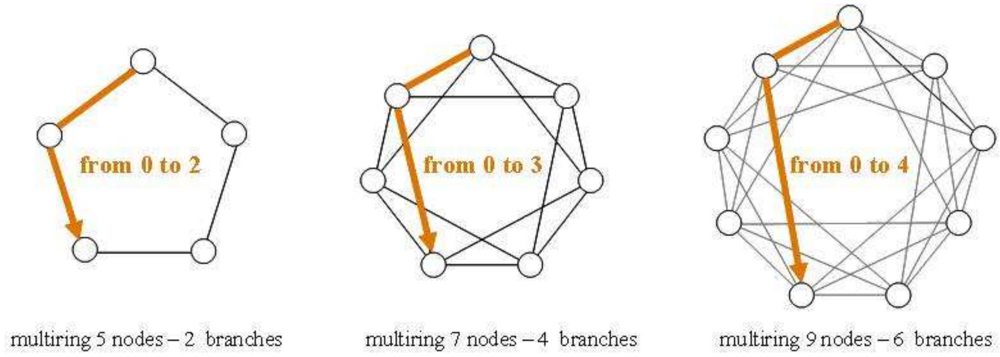


Fig. 25 – Two-hop flows within multi-ring topologies

In Fig. 26 we verify the strong dependence of packets delay with the required time for the three-way-handshake agreements. This is being progressively higher for networks with higher density, as seen in the line with dark rhombs for n2mesh80216 simulations. However, we notice that this does not happen in our implemented rtPS class (line with light rhombs), which uses the data subframe to send control messages (distributed uncoordinated scheduling), avoiding the competition with its neighbors to access control opportunities (coordinated distributed scheduling). In this way, we obtained an rtPS service flows constant delay of 6ms for all network densities tested in this simulation.

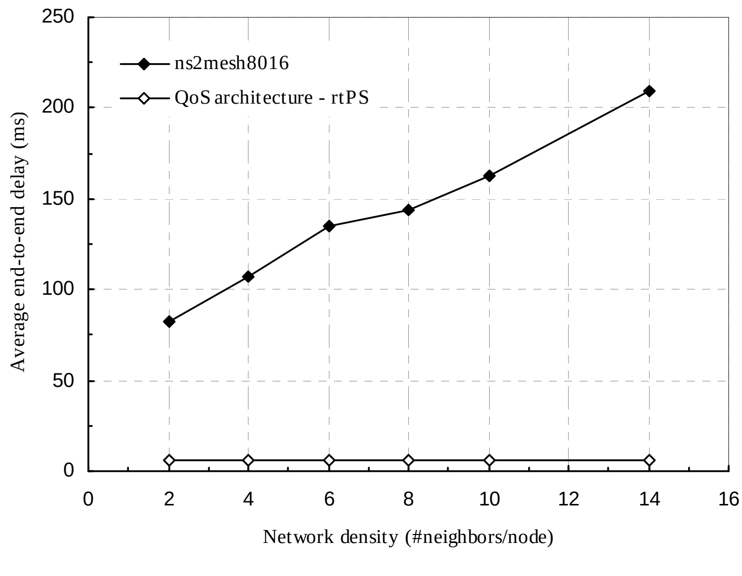


Fig. 26 – Delay for the rtPS service with increasing network density

5.4.4 Mixed Service Class Simulations

- **Simulation 1**

In the following simulations we can observe the behavior of the three service (UGS, nrtPS, BE) classes designed to support higher data transfer rates. Let us see how they react while increasing the transmission rates to the exhaustion of the bandwidth data spectrum.

We used the multiring 5 nodes topology, with all nodes connected and CBR traffic flows for all service classes.

We also set a minimum threshold rate of 1MB/s for nrtPS service.

We verified that for the nrtPS service is guaranteed a transmission rate larger than or equal to 1 Mb/s.

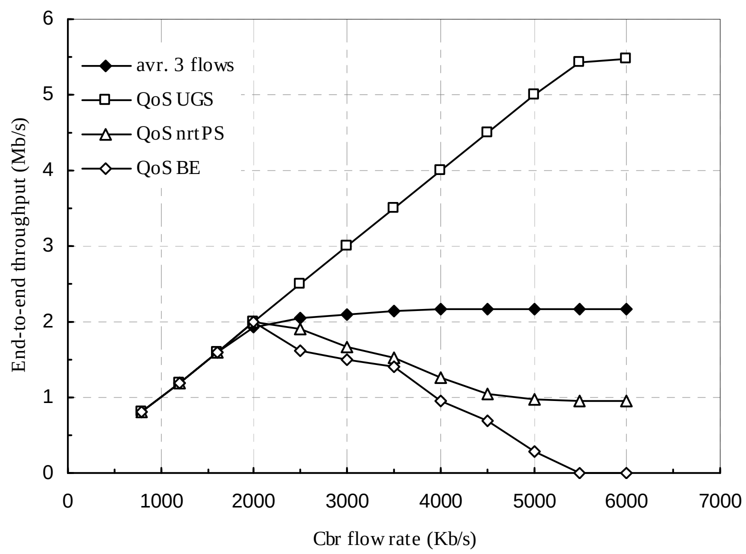


Fig. 27 – Throughput of UGS, nrtPS, BE and original ns2mes80216 flows

We noticed that above 5.5 Mb/s it is no longer possible to transmit BE traffic and the UGS flow stabilizes to allow the co-existence with a nrtPS traffic, achieving approximately 11Mb/s for the total data channel throughput.

In Fig. 28 we analyze the delay of each service flow.

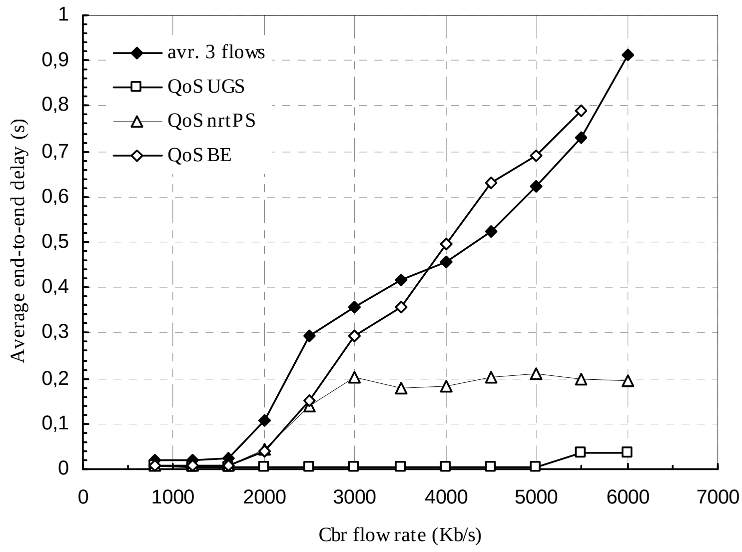


Fig. 28 – Delay of UGS, nrtPS and BE services and original ns2mes80216 flows

The UGS traffic packets have a minimum delay (of 4ms) similar to that achieved in Section 5.4.2 simulations. This value is practically constant, but reaches the 30 ms for above 5.5 Mb/s rates, when the flow begins to be limited by restrictions of the transmission channel and of the nrtPS traffic. In Fig. 29 we also observe that the UGS packets loss is zero until this point and from here starts increasing with 19 packets lost per second for 5 Mb/s and 105 packets lost for 6 MB/s.

The nrtPS flow has always a delay lower than the average obtained for the three simulated flows in the original model. But regarding packets loss, this service reaches higher values than the average obtained in ns2mesh80216. This fact is related to the queuing size used in the original model and in the QoS model. In the original model the buffer is unique and 100000 bytes sized, while in the QoS implementation we used four different buffers (one for each class) with 25000 bytes each.

The obtained delay in BE service is quite unstable, however it accompanies the trend obtained in the ns2mesh80216 simple model and the packet loss is slightly higher than the obtained for nrtPS service.

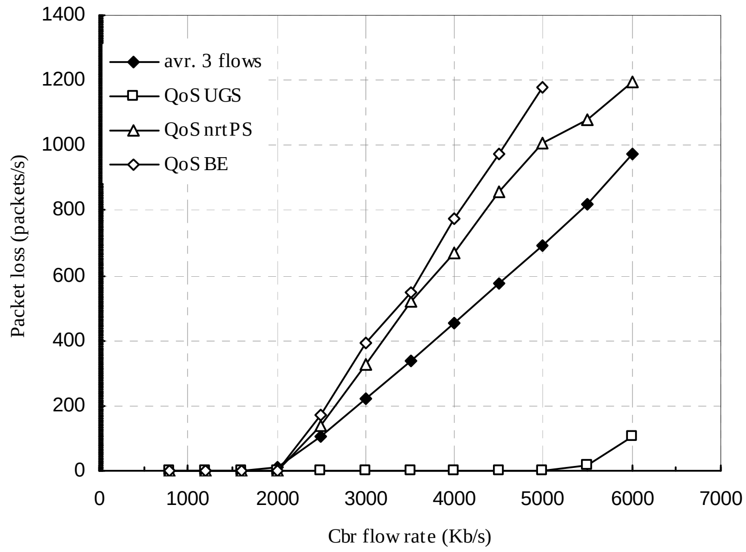


Fig. 29 – Packet loss of UGS, nrtPS, BE services and original ns2mes80216 flows

- **Simulation 2**

In the last test case we simulated a network usage scenario that is very close to the current needs of a telecommunications service provider. We used four traffic types each one with different characteristics: CBR traffic in UGS class, *telnet* calls in rtPS class, video streaming in rtPS class and internet traffic with BE service.

The CBR traffic was configured with a transfer rate of 1Mb/s and packets of 500 bytes. Internet traffic was generated by flows based on a super-imposition of four Interrupted Poisson Processes (IPPs), with packets of 192 bytes. This simulation was supported by the 5 five nodes clique topology.

Fig. 30 shows the throughput achieved by each service class while increasing the number of flows. The UGS and rtPS services obtained the required transferring bandwidth before the transmission channel exhaustion is reached, which was observed with 4 traffic flows per service, approximately 20Mb/s of total bandwidth. It is still observed that for more than two flows per service, the nrtPS and BE services have the bandwidth limited by the increasing use of the two higher priority classes (mainly UGS service). The BE service is totally blocked above the four flows.

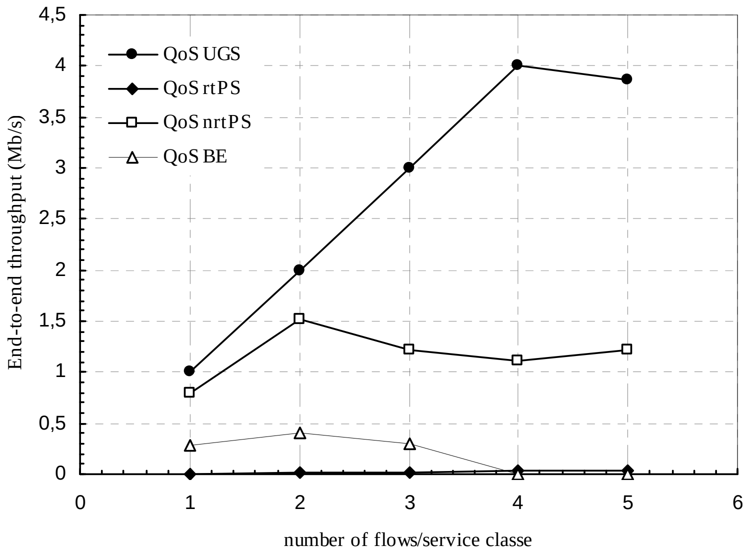


Fig. 30 – End-to-end throughput for UGS, rtPS, nrtPS and BE services

Fig. 31 shows the similar analysis by traffic flow. The throughput reached by nrtPS traffic is not annulled but it gradually decreases with the increase of its flows. The bandwidth range reserved for nrtPS service (corresponding to 1Mb/s) is shared by the several active nrtPS flows. The same applies to the UGS service for more than 4 flows.

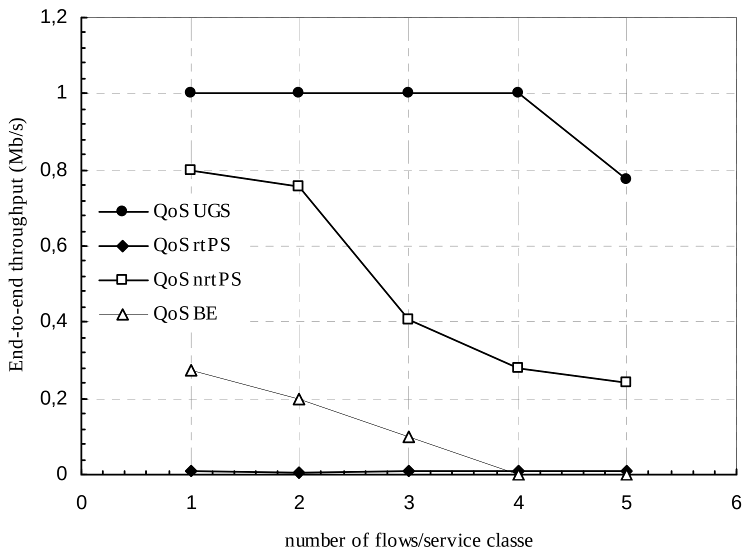


Fig. 31 – End-to-end throughput per traffic flow of each traffic class

In what relates to the average delays observed (Fig. 32), we found that the UGS and rtPS classes obtain delays of the same order of magnitude, 0.3 ms and 0.5 ms respectively.

For the nrtPS service we got a delay of 165ms, for 5 active flows, and with BE service the packets delay reaches 350ms.

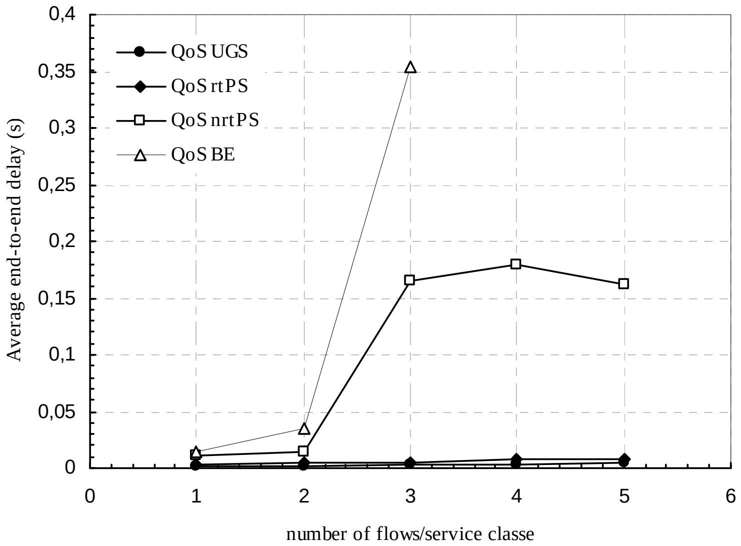


Fig. 32 – Average end-to-end delay for UGS, rtPS, nrtPS and BE services

5.5 Summary

In this chapter we described a set of simulations carried out along this thesis work for testing the implemented module and presented a discussion concerning the performance. We first simulated UGS and rtPS services as the most important services that required more implementation details to agree with similar classes in the PMP mode. Finally we simulated all services in the same scenario to test the response of sharing the bandwidth limited resources. All the simulated scenarios were also tested in the original version presented in ns2mesh8026 and the results were compared with those obtained in our implemented model. We showed the improved results in our QoS model, which gives us the strong conviction that the specified algorithms are in the good way for contributing to solve the open issue of QoS in 802.16 Mesh networks.

Chapter 6

6 Conclusions

In present thesis we implemented a 80216 Mesh QoS simulation module that matches the bandwidth needs of each service class, similarly to the standard Point-to-Multipoint mode. In addition of internal mechanism of the traffic scheduler (internal QoS, as is expected in the standard for the Mesh mode), the presented model allows that the bandwidth is shaped to the performance requirements of each service class.

The bandwidth usage is optimized while the QoS requirements (throughput, delay, jitter, etc) are met for the different services. This QoS model supports the same QoS classes as described for the PMP mode, i.e., UGS, rtPS, nrtPS and BE, but now applied to IEEE 802.16 Mesh topologies.

From the obtained results we identify some advantages resulting from the implemented QoS architecture. For example, the UGS traffic flows (with a constant transfer rate), reserved with *persistence* of 7 (good until canceled) are well suitable for CBR traffic types that maintain a constant transmission rate over a period of time. The reservations with *persistence* of 7 avoid the use of periodic bandwidth requests (as well their respective offers and confirmations) for the same amount of data, thus freeing space in the scheduling control subframe which can be used in another way. As being reserved the bandwidth amount required by the UGS flow (with *persistence* of 7), the QoS architecture permits to achieve the delay and jitter that are characteristic of the UGS class (in PMP mode) for each flow jump along the Mesh network.

For the traffic belonging to the rtPS and nrtPS classes, the QoS architecture does not make reservations for long periods, since there are expected variations in data transfer rate of these flows. In our study we used request reservations with a duration of 32 frames (*persistence* = 5). In this way, the probability of bandwidth wasting or lacking is reduced, which would be the case if it were used unique reservations for undetermined periods of time. Additionally, essential QoS requirement for the rtPS - delay end-to-end - has been optimized, thanks to the abolition of the coordinated three-way handshake delay. These agreements, over the various rtPS flow jumps, are further aggravated with the increase of the number of neighboring nodes (higher density of the network). In this service type the scheduling messages are transmitted in the data subframe along with the rest of traffic data. This can bring some possibility of collisions between the scheduling messages and other traffic, but with our procedure we reduce collisions to a number almost insignificant.

For nrtPS traffic the throughput is more important, so the QoS architecture uses the normal scheduling (coordinated distributed scheduling) in the control subframe to complete the three-way-handshakes.

For BE traffic the QoS architecture attempts to allocate the remaining resources of previous classes reservations. We noticed, however, that bottleneck can occur in this traffic class when the network is overloaded with higher priority streams.

6.1 Further Work

The current work will serve as a sustained basis for continuing studies of QoS in IEEE 802.16 Mesh Networks. Nevertheless there are some points that can be improved to achieve a more stable and robust model. The first point, and perhaps the most critical in this simulation solution of IEEE 802.16 mesh networks, is the development of the position identification model of network components, as it exists in other modules included in NS-2, which enables to specify the cartesian coordinates (X, Y, Z) of each network component, so that the physical layer can be in accordance with links loss and IEEE 802.16 adaptive modulation feature.

This development would also enable the integration with the NAM graphical interface, often used in simulations with NS-2. This graphical interface allows the user to view the representation of built network topology and its operation during the simulation.

Established connections, information exchanged, configuration messages, packet loss, are some of the indicators that may be visible in this GUI model.

References

[1] IEEE Computer Society and IEEE Microwave Theory and Techniques Society, 802.16 IEEE Standard for Local and metropolitan area networks, Part 16: Air Interface for Fixed Broadband Wireless Access Systems, IEEE Std. 802.16-2004, October 2004.

[2] The IEEE 802.16 Working Group on Broadband Wireless Access Standards – URL: http://wirelessman.org/ (or http://ieee802.org/16).

[3] Study Group for High-Speed Unlicensed Metropolitan Area Network – URL: http://ieee802.org/16/human.

[4] WiMAX Forum – URL: http://www.wimaxforum.org/about/.

[5] L. Nuaymi, WiMAX: Technology for Broadband Wireless Access John Wiley & Sons, Ltd. ISBN: 0-470-02808-4 (2007).

[6] P. S. Mogre and M. Hollick. The IEEE 802.16-2004 MESH Mode Explained. Technical Report KOM-TR-2006-08, Multimedia Communications Lab, Department of Electrical Engineering and Information Technology, Technische Universit¨at Darmstadt, Germany, (2006).

[7] P. S. Mogre, M. Hollick, C. Schwingenschloegl, and R. Steinmetz, QoS Architecture for Efficient Bandwidth Management in the IEEE 802.16 MESH Mode, in *WiMax/MobileFi: Advanced Research and Technology*. Auerbach Publications, CRC Press, Dec. 2007.

[8] C. Cicconetti, Ian F. Akyildiz, and L, Lenzini, Bandwitth Balancing in Multi-Channel IEEE 802.16 Wireless Mesh networks, IEEE Network. (2007).

[9] C. Cicconetti, A. Erta, L, Lenzini, and E. Mingozzi, Performance Evaluation of the Mesh Election Procedure of IEEE 802.16/WiMAX, IEEE Network. (2007).

[10] D Sweeney, WiMax Operator’s Manual: Building 802.16 Wireless Networks (Second Edition) (2006).

[11] S.Y. Wang, C.L. Chou, C.H. Huang, C.C. Hwang, Z.M. Yang, C.C. Chiou, and C.C. Lin, The Design and Implementation of the NCTUns 1.0 Network Simulator, Computer Networks, Vol. 42, June 2003, pp.175-197.

[12] Network Simulator (- ns) project - URL: http://nsnam.isi.edu/nsnam/index.php/Main_Page.

[13] L. Lenzini, G. Anastasi, E. Mingozzi, G. Stea, and others ns2mesh80216: an extension to ns2 to simulate IEEE 802.16 Wireless Mesh Networks, Computer Networking Group, Department of Information Engineering, University of Pisa, Italy – URL: http://cng1.iet.unipi.it/wiki/index.php/Ns2mesh80216

[14] A. M. Law and W. D. Kelton, Simulation Modeling and Analysis, McGraw-Hill, (2000)

[15] C. R. Baugh, J. Huang, R. Schwartz, and D. Trinkwon, Traffic model for 802.16 TG3 MAC/PHY simulations, IEEE 802.16 Broadband Wireless Access Working Group, Tech. Rep., Mar. 2001.

Appendices

APPENDIX I

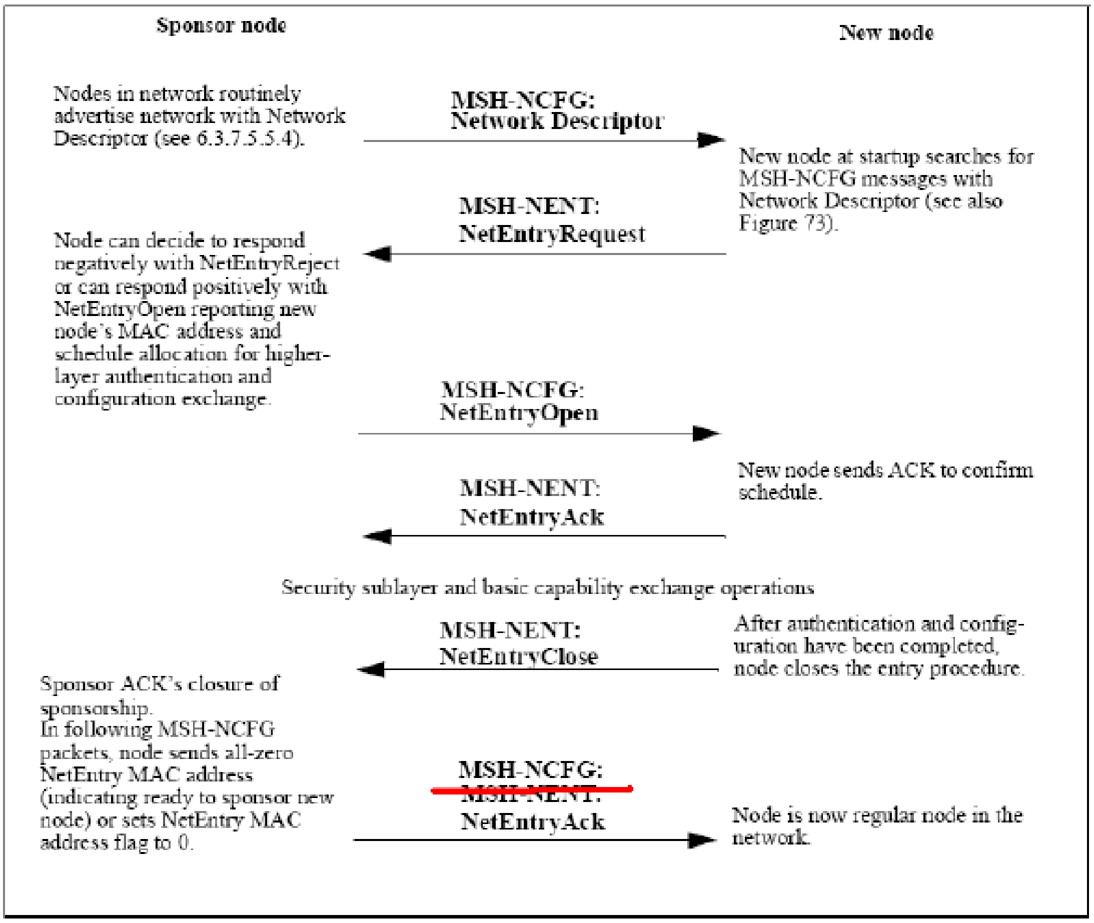
PSEUDOCODE FOR ACCESSING NETENTRY SLOTS [1]

```
/* Variable Definitions */
//MSH-NENT Message queue
Pkt *MSH-NENT_MsgQ =NULL;
//SponsorsState and OthersState record the NetEntry
uint SponsorsState =UNAVAILABLE, OthersState =BUSY;
// Address in the MSH-NCFG packet form the sponsor or other nodes,
// which can be used to determine the availability of the next
// NetEntry transmission opportunity
//SponsorsState can be UNAVAILABLE,AVAILABLE and POLLING.
//OthersState can be AVAILABLE and BUSY.
uint OthersMaxMacAdr =0xFFFFFFFF, OthersMinMacAdr =0x00000000;

void RecvOutgoingMSH-NENT_Msg (Pkt *MSH-NENT_Msg) {
  MSH-NENT_MsgQ->enqueue (MSH-NENT_Msg);
}
void RecvIncomingMSH-NCFG_Msg (Pkt *MSH-NCFG_Msg) {
   if (MSH-NCFG_Msg->sourceMacAdr ==sponsorsMacAdr) {
     switch (MSH-NCFG_Msg->NetEntryAddress)
     {
        case 0x000000000000:SponsorsState =AVAILABLE; break;
        case myMacAdr: SponsorsState =POLLING; break;
        default: break;
     }
   } else {
        switch (MSH-NCFG_Msg->NetEntryAddress)
        {
          case 0x000000000000:break;
          default:OthersState =BUSY;
                    if (OthersMaxMacAdr <MSH-NCFG_Msg->NetEntryAddress)
                         OtherMaxMacAdr=MSH-NCFG_Msg->NetEntryAddress;
                    if (OthersMinMacAdr >MSH-NCFG_Msg->NetEntryAddress)
                         OtherMinMacAdr =MSH-NCFG_Msg->NetEntryAddress;
        }
   }
}

void NetworkControlSubframeStart () {
     boolean xmt =FALSE;
     if (MSH-NENT_MsgQ->qLength()) {
          if (SponsorsState ==AVAILABLE){
```

```
                if (OthersState !=BUSY) xmt =TRUE;
            }
            else if (SponsorsState ==POLLING) {
                      if (OthersState !=BUSY) {
                          xmt =TRUE;
                      }
                      else
                      {
                        if (((mayMacAdr >OthersMaxMacAdr)&&(even supperframe))||
                            ((mayMacAdr <OthersMinMacAdr)&&(odd supperframe))) {
                                xmt =TRUE;
                        }
                      }
            }
        }
        if (xmt) {
            Pkt*MSH-NENT_Msg =MSH-NENT_MsgQ->getHead();
            MSH-NENT_MsgQ->dequeue(MSH-NENT_Msg);
            SendOutPkt (MSH-NENT_Msg, nextNetEntryslot);
        }
        SponsorsState =UNAVAILABLE;
        OthersState =AVAILABLE;
        OthersMaxMacAdr =0x000000000000;
        OthersMinMacAdr =0xFFFFFFFFFFFF;
    }
```

APPENDIX II

MESH NETWORK ENTRY MESSAGE SEQUENCE [1]



APPENDIX III

MSH-CSCF MESSAGE FORMAT[1]

Syntax	Size	Nodes
MSH-CSCF_Message_Format() {		
Management Message Type = 43	8 bits	
Configuration sequence number	4 bits	
NumberOfChannels	4 bits	
for (i=0; i< NumberOfChannels; ++i) {		
Channel index	4 bits	
}		
Padding Nibble	0 or 4 bits	Pad till byte boundary.
NumberOfNodes	8 bits	
for (i=0; i< NumberOfNodes; ++i) {		
NodeID	16 bits	Node index for this node is *i*.
NumOfChildren	8 bits	
for (j=0; j< NumberOfChildren; ++j) {		
Child Index	8 bits	Index of j^{th} child node.
Uplink Burst Profile	4 bits	Burst profile from j^{th} child node.
Downlink Burst Profile	4 bits	Burst profile to j^{th} child node.
}		
}		
}		

APPENDIX IV

Main functions of the MAC Coordinator Module

```
//! Manage MSH-DSCH message from the MAC.
void recvMshDsch (WimshMshDsch* dsch, double txtime = 0);
//! Manage MSH-NCFG message from the MAC.
void recvMshNcfg (WimshMshNcfg* ncfg, double txtime = 0);

//! Election procedure for Dsch called by handle().
void electionDsch ();
//! Election procedure called by handle().
void electionNcfg ();
//! Election procedure called by handle().
void electionNent ();

//! Competition procedure
/*!
 Run the standard mesh election procedure as specified in
 IEEE 802.16-2004 standard, Section 6.3.7.5.5.6 pp. 159-160
 nextXmtTime_ is filled with the slot number relative to
 the node's next Xmt Time
 */
void competition (std::list<NeighInfo>& nghList,MyInfo& my,
                          wimax::BurstType type);

//! Find the competing nodes given a certain XmtTime.
/*!
 Each neighbor that is considered to be a competitor has the competing_
 flag set to one into the nghList vector.
 Return the number of competitors.
 */
unsigned int competingNodes (unsigned int TempXmtTime,
                                          std::list<NeighInfo>&
nghList);

//! Execute the mesh election procedure
/*!
 This function is identical to that in the standard p. 160
 */
bool meshElection (unsigned int TempXmtTime,
                           short unsigned int nodeID,
                           std::list<NeighInfo>& nghList,
                           wimax::BurstType type);

//! Return the holdoff time.
unsigned int computeHoldoffTime(unsigned holdOffExp);

//! Return the XmtTimeMx.
/*!
 Find x in the following formula given NextXmtTime and holdOffExp:
 2^holdoffExp*x < NextXmtTime <= 2^holdoffExp*(x+1)
 */
unsigned int computeXmtTimeMx(unsigned holdOffExp, unsigned NextXmtTime);

//! Return true if a node is eligible.
```

```
bool eligible (unsigned xmtmx, unsigned TempXmtTime, unsigned holdexp);
//! Return the hash for nodeID; see IEEE 802.16 std pp. 160.
unsigned int inline_smear(unsigned short int val);
//! Compute the control slot from the dawn of time.
unsigned int currentCtrlSlot(double txtime = 0);
//! Compute the MSH-DSCH slot from the dawn of time.
unsigned int currentCtrlSlotDsch(double txtime = 0);
//! Compute the MSH-NCFG slot from the dawn of time.
unsigned int currentCtrlSlotNcfg(double txtime = 0);
```

APPENDIX V

Main functions of the Bandwidth Management Module

```
//! Decode grants/confirmations from an incoming MSH-DSCH message.
/*!
 There the following cases:

 - for each grant addressed to this node, a confirmation is added
 to the pending list of confirmations (managed by confirm()),
 the granted minislots are marked as unconfirmed unavailable
 (in the unconfirmedSlots_ bitmap) and the amount of bandwidth
 granted by a neighbor is updated

 - for each grant not addressed to this node, an unavailability is
 added to the pending list of availabilites (managed by availabilities())

 - for each confirmation addressed to this node, update the cnf_in_
 data structure

 - for each confirmation addressed to a node which is not in this node's
 first-hop neighborhood, mark the confirmed minislots as unavailable
 for reception from this node
 */
void rcvGrants (WimshMshDsch* dsch);
//! Decode availabilities from an incoming MSH-DSCH message.
/*!
 We update the status of neigh_tx_unavail_ based on the received
 availabilities.
 */
void rcvAvailabilities (WimshMshDsch* dsch);
//! Decode requests from an incoming MSH-DSCH message.
/*!
 For each request addressed to this node, add the number of minislots
 requested to the req_in_ data structure.
 */
void rcvRequests (WimshMshDsch* dsch);

//! Confirm pending grants.
/*!
 For each confirmation in the unconfirmed list, we:

 - try to send as many confirmation as possible, provided that
 the slots that have been granted are still available for
 transmission by this node (via the self_tx_unaval_ bitmap)
 - update the status of the cnf_out_ data structure
 - set the minislots reserved for transmission at this node, which
 will be used by the handle() function to trigger the packet
 scheduler at the MAC layer. Both self_tx_unavl_ and self_tx_unavl_
 are updated

 Note that the cnf_out_ data structure is updated with the number
 of minislots actually confirmed which will be used for transmission.
 */
void confirm (WimshMshDsch* dsch, unsigned int n, unsigned int
serv_class);
```

```
//! Advertise pending availabilities.
void availabilities (WimshMshDsch* dsch, unsigned int s);

//! Request/grant bandwidth.
/*!
 :TODO: more documentation (come on, this is a critical function!)

 Let H be the average number of frames between two consecutive
 transmission opportunities of this node, and H' the same measure
 for the node to which we are currently granting bandwidth.
 The time window over which we grant bandwidth is NOW + [H',H + 2H'].

 The granted minislots are marked as unavailable for reception.
 The amount of granted minislots are udpated.
 */
void requestGrant (WimshMshDsch* dsch,
                           unsigned int ndx, unsigned int s);


//! Get the interval between two consecutive control opportunities in
frames.
unsigned int handshake (WimaxNodeId x) {
      return (unsigned int) ceil (
                                                (fabs ( mac_->h (x)  -
mac_->phyMib()->controlDuration() ))
                                                / mac_->phyMib()-
>frameDuration()); }

//! Return the quantum value of a given input/output link, in bytes.
unsigned int quantum (unsigned int ndx, wimax::LinkDirection dir) {
return (unsigned int) (ceil(wm_.weight (ndx, dir) * roundDuration_)); }


//! Search the transmit slots reserved for sent uncoordinated messages.
void search_tx_slot (unsigned int ndx, unsigned int reqState);

protected:
//! Invalidate the data structures' entries for the current frame.
void invalidate (unsigned int F);

//! Cancell UGS reservation for requester's neighbours and itself
void cancell_Requester (unsigned int ndx, unsigned char s, WimshMshDsch*
dsch);

//! Cancell UGS reservation for granter's neighbours and itself
void cancell_Granter (unsigned int ndx, unsigned char s);
```

APPENDIX VI

Main functions of the Data Management Module

```
//! Add a MAC PDU to this object.
void addPdu (WimaxPdu* pdu);

//! Schedule a new data burst to a neighbor.
void schedule (WimshFragmentationBuffer& frag, WimaxNodeId dst, unsigned
int service);

//! Drop a PDU (by deallocating PDU/SDU/IP).
void drop (WimaxPdu* pdu);

//! Serve a flow until its deficit or backlog are exhausted.
bool serve (WimshFragmentationBuffer& frag,
                  unsigned int ndx, unsigned int serv, bool unfinished);

//! Recompute the quanta values of a given list of flow descriptors.
void recompute (CircularList<FlowDesc>& rr);

//! Traffic forcast based on data fuffers variation.
void recomputecbr (unsigned int ndx, unsigned char s, unsigned int
bytes);
```