for Giuliana,
"...Oh, kiss me beneath the milky twilight,
Lift your open hand,
Strike up the band and make the fireflies dance,
Silver moon's sparkling.
So kiss me..."

for my
Parents.

Acknowledgements

In this moment, my first thought is for Giuliana, during these three years, she has supported and endured me, she has constantly encouraged me and given the desire and the strength to go forward. Even during the most difficult moments, one of her sweet smile is always enough to strengthen me and start over. Years ago, I still did not met Giuliana and in a thesis I read this inscription: "Behind every great man there is always a great woman", then I did not understand the meaning of this phrase, but now I know what it means to have a special woman at my side. Thank you Giuliana.

Another very special thank I want to say to my parents for their constant presence and for their sacrifices to help and support me.

I have to thank my prof. Marano and Floriano who allowed me to make this PhD experience.

I want also to thank Mauro, Peppino, Fiore, Luana and Franco for each moment spent together, in a particular way a thank to my friend of "office room" Fiore.

Another thank is for Attilio, I spent, working with him, a little period of doctorate. I find in Attilio a "great worker" and a real person. I want say thanks also to Giuseppe, he and Attilio involved me in the "work", I felt me as a part of the "group".

Last but not least, a thought is open to all students and the people which I met during these three years of doctorate, each of them gave me something and I want to thank them.

Andrea

Summary

The wireless networks, characterized by a great potential, represent an increasing attractive world. Via wireless connection and exploiting the wireless advantages, it is possible to eliminate the infrastructure and cabling issues related to wired counterparts. An interesting aspect is the capability to install, relatively quickly, a network which is able to offer services to a wide range of users. With a wireless network, it is also possible to enrich an existing wired network or to create from scratch in a permanent or in a temporary way a new wireless architecture; related to the last advantage we can consider the useful possibility to extol the wireless capability in area affected by natural disasters. In all these cases and many others, that we do not mention, a wireless network is certainly preferable for the ease and speed installation, the cost and the ability to easily extend the number of users.

Within the spectrum of wireless technologies, currently existing, the WiMAX technology and the IEEE 802.16 standard which defines its characteristics, occupies a special place. The IEEE 802.16 is a wireless technology for metropolitan area networks, created to allow access to wireless broadband. In the various versions of the protocol and subsequent corrections published, were introduced several important features, such as the possibility of using mesh network mode, which allows the creation of direct links between users. In the most recent version, IEEE 802.16e, has added the user mobility capability.

This thesis summarizes the issues considered in the PhD period and related to both the physical (PHY) and medium access control (MAC) layers as defined by the IEEE 802.16 standard. The study and analysis of the two protocol layers constitutes the first step to achieve our goal: we want to contribute to the development of WiMAX technology in order to contribute to the creation of an 802.16 network architecture which is able to provide a broad variety of services to users, where each service is characterized by well-defined quality levels. The contribution of this thesis may therefore be expressed in terms of developing of channel error models (related to the physical layer) and algorithms (related to both levels of protocol) that can be a support for the

provision of quality of service. In particular, in this thesis has been examined a set of interesting challenges in WiMAX mesh scenarios as call admission control and metrics to support the route selection. Finally the various solutions tested and developed have been integrated into a single framework that can act as a support for the quality of service.

Contents

List of Figures

List of Tables

Acronyms

AMC Adaptive Modulation and Coding
AODV Ad-hoc On demand Distance Vector
BE Best Effort
BER Bit Error Rate
BPSK Binary Phase Shift Keying
BS Base Station
BWA Broadband Wireless Access
CAC Call Admission Control
CDF Cumulative Distribution Function
CID Connection IDentifier
CP Cyclic Prefix
CPS Common Part Sublayer
CRC Cyclic Redundancy Check
CS Convergence Sublayer
DCD Downlink Descriptor Channel
DIEM Delivering time Interference and ETT based Metric
DIM Delivering time based Interference Metric
DMT Discrete Multi Tone
DSDV Destination-Sequenced Distance-Vector
DSR Dynamic Source Routing
DTMC Discrete Time Markov Chain
EGPRS Enhanced General Packet Radio Service
ETT Expected Transmission Time
ETX Expected Transmission Count
FDD Frequncy Division Duplexing
FFT Fast Fourier Transform
FTP File Transfer Protocol
GCAD Greedy Choice with bandwidth Availability aware Defragmentation
GPC Grant Per Connection
GPSS Grant Per Subscriber Station
GSM Global System for Mobile communications

IBI Inter Block Interference
ICI Inter Carrier Interference
IFFT Inverse Fast Fourier Transform
IP Internet Protocol
ISI Inter Symbol Interference
ISP Internet Service Provider
IWPM Instant Weighed Probability Model
IWPM-2V Instant Weighed Probability Model - 2 Variables
IWPM-3V Instant Weighed Probability Model - 3 Variables
LAN Local Area Network
MAC Medium Access Control
MCM Multi Carrier Modulation
MIB management information base
MPEG Moving Picture Experts Group
MRR Minimum Reserved Rate
MSNEA Mini Slots Number Estimation Algorithm
MSR Maximum Sustained Rate
MSS Mobile Subscriber Station
NLOS Non Line Of Sight
nrtPS not real time Polling Service
OFDM Orthogonal Frequency Division Multiplexing
OFDMA Orthogonal Frequency Division Multiple Access
OSS Operation Support System
PADIEM Priority Aware Delivering time based Interference Metric
PDF Probability Density Function
PDU Protocol Data Unit
PECF Packet Error Correlation Function
PER Packet Error Rate
PHY Physical layer
PMP Point to Multipoint
PKM Privacy Key Management
PSD Power Spectral Density
PSEA Packet Size Estimation Algorithm
QAM Quadrature Amplitude Modulation
QOF QoS Oriented Framework
QoS Quality of Service
QPSK Quadrature Phase Shift Keying
RCT Radio Conformance Test
rtPS real time Polling Service
RX Receiver
SAP Service Access Point
SCN Service Class Name
SDU Service Data Unit
SFID Service Flow IDentifier
SFM Simplified Fritchman Model

SNR Signal to Noise Ratio
SOHO Small Office Home Office
SOS Set of Old Solutions
SS Subscriber Station
SU Subscriber Unit
TDD Time Division Duplexing
TP Test Purpose
TSS Test Set Structure
TT Traffic Table
TX Transmitter
UCD Uplink Channel Descriptor
UGS Unsolicited Grant service
UWB Ultra Wide Band
VoIP Voice over Internet Protocol
WiFi Wireless Fidelity
WLAN Wireless Local Area Network
WMAN Wireless Metropolitan Area Network
WiMAX Worldwide Interoperability for Microwave Access
XDSL X Digital Subscriber Line

1

The advent of a new technology: WiMAX

1.1 Introduction

The expected convergence of fixed and mobile internet services, the emergence of new applications and the growth of wireless subscribers will lead to an ever increasing demand for bandwidth in wireless access. The dream of 3G wireless systems is to provide high-speed multimedia services though mobile cellular technology, enabling subscribers to access the Internet and enjoy videophone, video on demand, games and multimedia chatting. However, the economic efficiency and data performance of 3G wireless systems have not been satisfactory mainly because it was not originally designed for data communications. Thus, burdened by license fees and deployment costs coupled with unsatisfactory performance, many 3G operators suffer from poor profitability. On the other hand, as the market for broadband and mobile communication services attains maturity in some countries, the communications industry has shown a limit in growth based on quantitative expansion. Meanwhile, wireless Internet access service is expected to be the new motivation for overcoming these limitations and increasing revenue. To make this service commercially successful, operators and Internet Service Provider (ISP)s have looked for new solutions for carrying Internet Protocol (IP) packets over the air more efficiently and economically. Nowadays, Wireless Local Area Network (WLAN) and Wireless Metropolitan Area Network (WMAN), which conform to the IEEE802.11 and IEEE802.16 families, respectively, are attracting interest as solutions for wireless Internet access.

WLAN is a high-bandwidth, short-range, two-ways data communications system that uses radio waves rather than fiber or copper cable as its transmission medium. WLAN is a flexible data communications system implemented as an extension to a wired network or as an alternative to a wired Local Area Network (LAN). Thus, wireless LAN combine data connectivity with user mobility. Today, most WLAN use the 2.4GHz frequency band, but the 5GHz band is rapidly emerging. WLAN may be installed to extend or replace a wired LAN in a corporate enterprise, a small or medium sized enterprise,

or a Small Office Home Office (SOHO) environment. A recent application of WLAN technology has been to offer public access to Internet-based services in small public deployment frequently referred to as hotspots.

Currently, the remarkable upsurge in demand for supporting both high-speed and high-quality applications in Broadband Wireless Access (BWA) networks has attracted the attention by both industry and academia. Among a variety of BWA technologies, IEEE 802.16 is a promising one to enable various services to solve the problem of providing enhanced services over the last mile. The IEEE 802.16 protocol, for wireless metropolitan area networks has been recently standardized to meet the needs of wireless broadband access. The 802.16 is also known as Worldwide Interoperability for Microwave Access (WiMAX), which is a no profit association with the scope to accelerate the WiMAX devices diffusion. Behind WiMAX logo there are important companies, for example: Airspan, Alvarion Aperto Networks, Ensemble Communications, Fujitsu, Intel, Nokia, OFDM Forum and Proxim Corporation are a set of these companies. WiMAX can serves Wireless Fidelity (WiFi) hotspots and can provide services to a wide coverage area with a radius of 50 kilometers. It is possible considering also Non Line Of Sight (NLOS) scenarios, in which a rate of 134 Mbps is reachable, this fact implies the possibility to provide broadband services at hundreds of users, using a single sector of a base station. Wireless technologies are becoming significantly: the IEEE 802.16 can increase system performance and decrease the cost of equipments. This technology can also provide the broadband connections 'on demand' to all those places which need temporary connection (conferences, exhibitions, particular events and more).

WiMAX is faster than WiFi and the first technology is also characterized by a widest coverage area: in fact the WiFi coverage area can be measured in the order of square meters, instead for WiMAX we can tell about square kilometers.

A typical scenario obtainable by WiMAX technology is depicted in figure 1.1:

- a Base Station (BS) is connected with a set of Subscriber Station (SS)s;
- each SS can connect to internet a little group of buildings.

The IEEE 802.16 standard defines two protocol layers: Medium Access Control (MAC) layer and Physical layer (PHY); the MAC layer can support two different topologies:

- Point to Multipoint (PMP);
- Mesh.

The first one allow to establish only links between BS and SSs (see figure 1.1). The BS in this way is the central point of the network. Instead in the second topology mode also direct links between SSs are allowed (see figure 1.2). WiMAX technology can be used for creating wide-area wireless backhaul network. When a backhaul-based WiMAX is deployed in mesh mode, it not

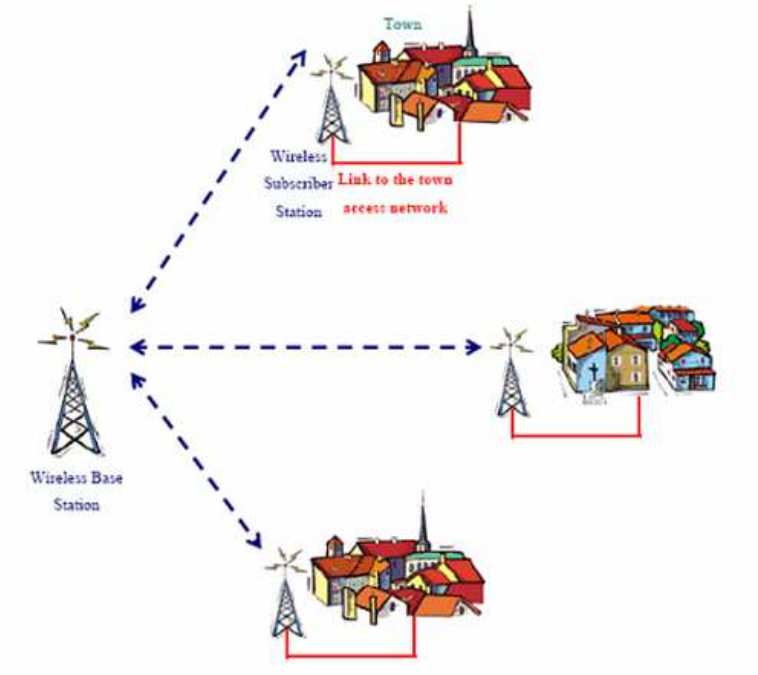

Fig. 1.1. A typical WiMAX scenario in PMP mode

only increases the wireless coverage but also provides features such as lower backhaul deployment cost, rapid building, easy deployment, robustness and re-configurability. This will make it one of the indispensable technology in next generation networks.

The IEEE 802.16 protocol defines operations in both licensed and license-exempt bands. The licensed band deployment is useful for dense and competitive coverage areas, in this case in fact, the interference is the major challenge. The deployment in license-exempt bands is used to cover restricted area and also to limit the initial investments. A significant advantage of WiMAX technology is the great flexibility in the network infrastructure deployment, this is due to the ability to define the width of the channel, the type of duplexing and the transmission techniques. This new technologies are a viable alternative to traditional broadband technologies such as X Digital Subscriber Line (XDSL), cable modems and fiber optics, as they allow to an ISP to create its own network infrastructure with a high scalability in terms of investment and services capacity. Summarizing, the WiMAX technology allow to ensure optimal performance:

- in terms of capacity, even with cell with very high load;
- in terms of coverage, although the presence of indoor Subscriber Unit (SU) reduces the performance.

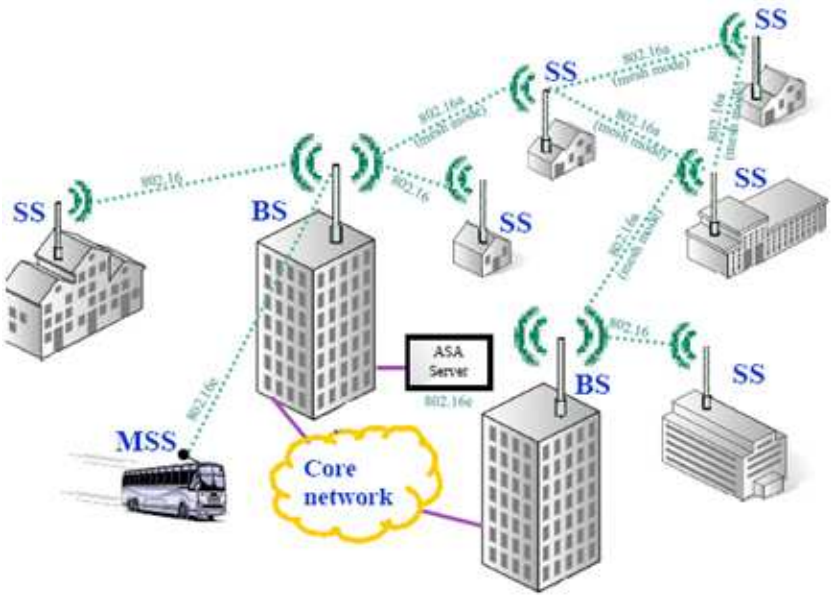

Fig. 1.2. An example of WiMAX scenario operating in mesh mode

WiMAX technology is a very promising technology and it is characterized by a series of advantages. Certainly, it was conceived with the prospect of becoming the technology that could eliminate the digital divide problem, and is proposed, in its own right, to assume a leading role among the existing technologies.

Nevertheless, the solutions that are attracting increasing interest, are also the integrated architecture, in which two or more technologies can be integrated and can cooperate in order to guarantee high quality of services over large areas and to a large number of users. The chances to create integrated architectures are different; cooperation such as WiMAX - WiFi, WiMAX - Ultra Wide Band (UWB), or WiMAX - 3G or other kind of cooperation can be considered. Surely a basic problem, which is common to all the integrated architecture, is to ensure quality of service to users. Each architecture, each protocol is characterized by its own mechanisms to ensure Quality of Service (QoS) in a network segment. But what happens when a data stream of a user must go through more than one segment of the integrated network? Once a protocol of a specific segment, admits a new call, and once the call has been moved to another segment, how can the QoS levels guaranteed at the instant of the admission call be maintained? The problem is to guarantee an end-to-end QoS.

The introduction of the promising technology IEEE 802.16 is related to the focus of this thesis. Our intent is to study the IEEE 802.16 protocol, considering both PHY and MAC layers, in order to elaborate a set of solutions

useful to improve and enrich this protocol. The following of this chapter is conceived as an introduction to this technology and also as an introduction to the issues which have represented the our research challenge.

1.2 IEEE 802.16 standard evolution and related documents

The IEEE 802.16 Working Group has defined the standard protocol which is behind the commercial name WiMAX. In particular the advent of the actual state of protocol was developed by publication of a series of subsequent amendments. This process, at the actual state of the art, has produced four different network architectures as specified by IEEE 802.16 protocol, and other new kind of architectures are under study. In the following, in order to make the reader able to distinguish the various amendments and protocol versions, we introduce a brief description for each document related to IEEE 802.16 protocol:

- *IEEE 802.16-2001, Air Interface for Fixed Broadband Wireless Access Systems.*
  This is the first standard proposed by 802.16 task group and it is approved on 6 December 2001 [1]. This standard specifies MAC and PHY features for a point-to-multipoint broadband wireless access systems providing multiple services. It is designed to support small office/home office applications and it is capable to guarantee a data rates of 134 Mbit/s. The protocol features are described by a layered structure organized in layers and sublayers. The PHY is characterized by a set of air interfaces operating in frequencies range from 10 to 60 GHz, which is able to support data transmission in line-of-sight scenarios. MAC layer is instead organized by a set of sublayes in which very interesting is the presence of Privacy sublayer, it has the task to provides secure service supported by data encryption and privacy keys management.
- *IEEE 802.16c-2002, Air Interface for Fixed Broadband Wireless Access Systems - Amendment 1: detailed system profiles for 10-66 GHz.*
  The version *c* of IEEE 802.16 protocol [2] is approved on 11 December 2002 and it is an amendment updates and expands IEEE 802.16-2001 protocol. It presents sets of features and functions to be used in typical implementation cases; also it represents an improvement to eliminate errors and inconsistencies. Obviously it is referred to 10-66 GHz licensed band.
- *IEEE 802.16a-2003, Air Interface for Fixed Broadband Wireless Access Systems - amendment 2: medium access control modification and additional physical layer specifications for 2-11 GHz.*
  The work group *a* of IEEE 802.16 protocol [3] started its work before group *c* but its results were approved only on 29 January 2003. This version is an amendment to 802.16-2001 protocol and adds to it a series of

important features. The first one is the mesh concept; in this way the capability to consider a different topology is introduced. The addition of this characteristic causes many changes in the MAC and PHY functionality, in fact with the introduction of mesh concept was consequently introduced a complication in bandwidth and QoS management mechanisms. The other important improvement introduced by this amendment is related to PHY layer, in fact, the physical layer specification to operate in 2 to 11 GHz band, also in license-exempt bands, is specified. To operate in this band, new channel impairment phenomena, as multipath, has to be considered and to contrast it new air interfaces using Orthogonal Frequency Division Multiplexing (OFDM) and Orthogonal Frequency Division Multiple Access (OFDMA) technique are introduced. With this two novelties the protocol is projected toward different scenarios.

- *IEEE 802.16.2-2004, Coexistence of fixed broadband wireless access systems.*
  This document [4] is a recommended practice, approved on 9 February 2004 and defines recommendations for the design and coordinated deployment of fixed broadband wireless access systems, with the focus to verify and control interference. In practice, the task of this work group is to define a document to promote coexistence for fixed broadband wireless systems and to specify how to manage coexistence in a shared environment with acceptable mutual interference. This document in particular address spectrum from 2 to 60 GHz.
- *IEEE 802.16-2004, Air Interface for Fixed Broadband Wireless Access Systems.*
  This protocol citeref.5 is a revision of standard IEEE 802.16-2001 and it is approved on 24 June 2004. This standard can be considered as the final version for PMP and mesh network architectures. This documents is an improved version of protocol IEEE 802.16-2001 and contain also the revision and corrigenda introduced in the subsequent IEEE 802.16a-2003 and IEEE 802.16c-2002 versions. It summarizes all the MAC and PHY mechanisms in both PMP and mesh mode and also each air interface developed for line-of-sight and non-line-of-sight scenarios, considering also licensed and license-exempt bands.
- *IEEE 802.16f-2005, Part 16: Air Interface for Fixed Broadband Wireless Access Systems - Amendment 1: Management Information Base.*
  This standard [6] is approved on 22 September 2005 and amends IEEE 802.16-2004 standard. It specifies a management information base (MIB) for the MAC and PHY and associated management procedures. This document is produced taking into account the focus of defining the management object and the topics related to managed devices.
- *IEEE 802.16e-2005, Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems - Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands and Corrigendum 1.*

This document [7], approved on 7 December 2005, updates the IEEE 802.16-2004 standard. The main feature which is introduced in this document is the user mobility. The task group *e* specifies a system for combined fixed and mobile BWA supporting subscriber stations moving at vehicular speeds in licensed bands under 6 GHz. It is based on OFDM transmission method with 256 Fast Fourier Transform (FFT) points, i.e. 256 subcarriers. It should operate in this bands supporting bit rates up to 15 Mbit/s to mobile SS and also higher layer handover between base stations or sectors are specified. This standard specifies also corrections to IEEE 802.16-2004.

- *IEEE 802.16k-2007, Media Access Control (MAC) Bridges, Amendment 2: Bridging of IEEE 802.16.*
  This version [8] is the shortest standardized document related to IEEE 802.16 protocol. It amends 802.1D protocol to support the bridging of the IEEE 802.16 medium access control. It is approved on 22 March 2007 and mainly specify a little set of additions and improvements for 802.1D. The IEEE 802.1D is the IEEE MAC bridge standard and allows communications between two end stations belonging to separate LAN.
- *IEEE 802.16g-2007, Part 16: Air Interface for Fixed Broadband Wireless Access Systems - Amendment 3:Management Plane Procedures and Services.*
  This standard [9] amended the IEEE 802.16-2004 standard and it is elaborated to specify each management aspect related to fixed and mobile broadband wireless systems. It specifies the management functions, interfaces and protocol procedures. The main features are related to the enhancements of the radio interface MAC Management messages, enhancements of the radio interface data plane capabilities and introduction of a set of primitives for the entities described in IEEE 802.16 protocol. It is approved on 27 September 2007.
- *IEEE 802.16 Conformance protocols.*
  All the application cases, the base stations and the subscribers stations implementations, have to compliance with the protocol constraints and guidelines. The focus is to guarantee the interoperability between different system implementations. To verify the effective interoperability there is the need of a well defined Test Set Structure (TSS), Test Purpose (TP) and Radio Conformance Test (RCT). These tests are specified in a set of documents published in different times along the whole protocol process development. In the following the actual set of conformance protocols are listed:
  - *IEEE 802.16 Concormance01-2003, Part 1: Protocol Implementation Conformance Statement (PICS) proforma for 10-66 GHz WirelessMan-SC air interface* [10], it is approved on 12 June 2003, it is a conformance to IEEE 802.16-2001.
  - *IEEE 802.16 Conformance02-2003, Part 2: Test Suite Structure and Test Purpose for 10-66 GHz wirelessMan-SC air interface* [11], it is

approved on 11 December 2003, it is a conformance to IEEE 802.16-2001 as amended by IEEE 802.16a-2003 and IEEE 802.16c-2002.
- *IEEE 802.16 Conformance03-2004, Part 3: Radio Conformance Tests (RCT) for 10-66 GHz WirelessMAN-SC Air interface* [12], it is approved on 12 May 2004, it is a conformance to IEEE 802.16-2001 as amended by IEEE 802.16a-2003 and IEEE 802.16c-2002.
- *IEEE 802.16 Conformance04-2006, Part 4: Protocol Implementation Conformance Statement(PICS) proforma for frequencies below 11 GHz* [13], it is approved on 15 September 2006, it is a conformance to IEEE 802.16-2004.

For 802.16 protocol other work groups, not cited previously, have to be considered; these groups do not have yet terminated the standardization process and thus no standardized documents are produced. The future versions of IEEE 802.16 standard are the following:

- *IEEE 802.16j, Amendment to IEEE 802.16e-2005 on Mobile Multihop Relay.*
  Multihop relaying for coverage extension in wireless networks is an old concept, in a relay networks, several relay stations between transmitter and receiver work together to forward the signal transmitted from transmitter to receiver. The IEEE 802.16 working group has devoted a task group to incorporating relay capabilities in the foundation of mobile IEEE 802.16e-2005. Currently, this task group is in the process of finishing IEEE 802.16j, the Multihop Relay Specification for 802.16. This amendment will be fully compatible with 802.16e-2005 mobile and subscriber stations, but a BS specific to 802.16j will be required to operate for relays.
- *IEEE 802.16h, Improved Coexistence Mechanisms for License-Exempt Operation.*
  This task group is still far to realize the final document in which the focus is to develop the coexistence mechanisms in license-exempt bands. The coexistence word is related to the environment sharing between entities providing wireless broadband service using the same frequency spectrum.
- *IEEE 802.16m, Air Interface for Fixed Broadband Wireless Access Systems - Advanced Air Interface.*
  The task group *m* is studying the development of an advanced air interface. This document will introduce a layered cell structure and also improvement in data rates achieving 100 Mbit/s for mobile users and 1 Gbit/s for fixed users.

1.3 An overview of MAC and PHY protocol layers

In the following subsections, we briefly describe the protocol stack as delineated by the IEEE 802.16 standard protocol. In the first subsection the salient

points of MAC layer will be introduced and in the second subsection the PHY layer, with its five air interfaces, will be summarized.

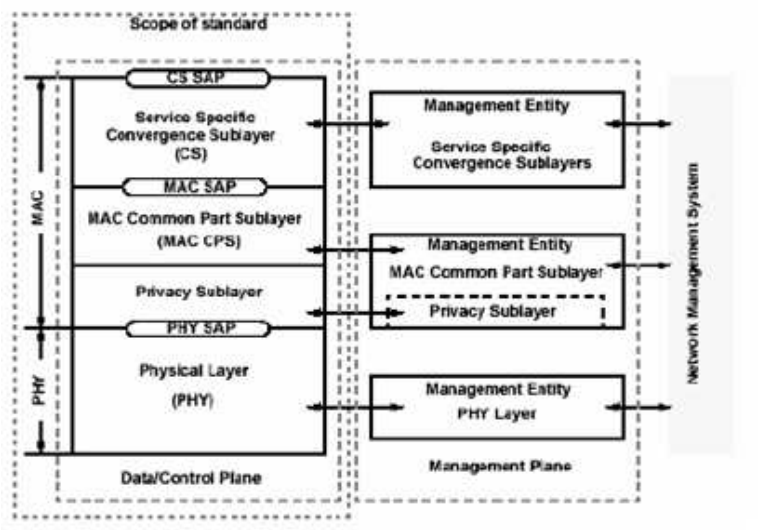

Fig. 1.3. IEEE 802.16 protocol stack

1.3.1 MAC layer

IEEE 802.16 defines a single-level MAC with various modifications and improvements published in various steps, which adds various physical layer specifics, covering both licensed and license-exempt bands. The IEEE 802.16 protocol was specified through a stack architecture, visible in figure 1.3.

The various sublayer can interact with each other and access to the services of the lower layers through the Service Access Point (SAP), so for example the Convergence Sublayer (CS) provides a set of services to higher layers through the CS-SAP, and in turn it enjoy the services of the Common Part Sublayer (CPS) through the services access point called MAC-SAP. The whole course in order to allow communications between equal entities, typical of a protocol defined by a stack architecture.

The protocol offers QoS mechanisms at both MAC and PHY protocol layer. In this section, we will see what is the MAC point of view and we are going to introduce what are the mechanisms offered by this layer in order to guarantee well defined levels of quality of service. In the first step we must distinguish between the different topologies supported by the IEEE 802.16 protocol. The protocol supports two different modes, the Point-to-Multipoint mode and the optional mesh mode. To correctly distinguish the two modes

we define what are the different entities that come into play in a WiMAX network:

- BS: Base Station;
- SS: Subscriber station;
- MSS: Mobile Subscriber Station.

The SSs and the Mobile Subscriber Station (MSS) are the users stations, the last stations are equipped with mobility capabilities, while the BS is the base station and has a central role for different reasons in both operational modes.

In PMP mode case, the only connecting links existing between the various entities, are the links of BS with the various user stations, fixed or mobile. It is not possible any direct link between the various user stations, consequently all stations must be submitted by BS which act as a central entity for the bandwidth allocation and to registry the user stations. In the mesh mode case there are also the possibility to create links between the various SSs. In practice, a user station (SS), which does not fall within the range of a BS, can reach it by exploiting the presence of any link with the near user stations. In this way, the SS in order to communicate with the BS, can exploit a multi hop path built on a set of mesh links. A mesh link is a connection between two SSs. This is not applicable to MSSs stations that continue to be binding to the BS, although in a later versions of the protocol in mesh mode, under designing, the protocol designer want to introduce the possibility to use the mesh mode also with MSSs.

In mesh mode the BS loses the central role of the only entity capable of managing the bandwidth allocation but retains a certain importance because it is the only station to have access to the "rest of the world", making the role of gateway to Internet. The distinction between the two operational modes is necessary because in both cases the quality of service, is managed and assured in a different way and using different MAC mechanisms.

The MAC layer, as visible in figure 1.3, is divided into 3 different sublayers, the first of these, or the upper layer, is the Convergence Sublayer. The main task of the Convergence Sublayer is to ensure to different types of higher protocol layers, for example:

- packet protocols such as IP protocol;
- ATM protocol;

the ability to communicate with the lower stack layers. The IEEE 802.16 protocol is a connection-oriented protocol, and between BS and SS can be created more than one connection (PMP mode). In this context, the convergence sublayer performs the delicate task of classification of SDUs, mapping the various SDUs from higher layers on the proper connection. In order to make effective this mapping, a set of classifiers is defined, and each SDU must be submitted to it, before being assigned to a connection. Among the various classifiers must be defined an application order, and if an SDU cannot

be mapped on any connection, it will be discarded. Another special feature done by CS sublayer is the deletion of parts of the Protocol Data Unit (PDU) header that are repeated packet by packet, which can be rebuilt once reached the destination. Please note that this is possible because the MAC is linked to the connections, hence the packets sent over a connection have some repetitive fields, and this because they belong to the same data flow. The purpose of this, is to optimize the data transmission saving bandwidth.

The central sublevel is the Common Part Sublayer (CPS). It performs typical tasks of the medium access control layer, thus providing algorithms to ensure efficient coordination between the various entities that require bandwidth allocation.

The last sublevel that includes the MAC is the Privacy sublayer that give to service providers a strong protection from theft of service. Moreover, it protects the data flow from unauthorized access by strengthening the encryption of the flows passing through the network. The Privacy sublayer provides a client / server management protocol authentication key where the BS (server), monitors the keys distribution to the clients. This sublayer is characterized by two main components:

- an encapsulation protocol for the encryption of data packets that are sent over the network: this protocol defines a set of encryption suites;
- a key management protocol: Privacy Key Management (PKM).

The MAC PDU is shown in figure 1.4, and consists of a fixed length header equal to 6 bytes, a payload that can contain one or more SDUs or SDU fragments or even can be absent, and finally optionally can appear the CRC field (Cyclic Redundancy Check). Please note that the Service Data Unit is the information coming from higher-layer protocol. In figure 1.4 is also visible the generic PDU header, which is different from header used to request bandwidth in PMP mode. The represented header is characterized by fixed length and contains several fields:

- HT: header type, which is used to distinguish between a generic header and bandwidth request header used in PMP mode;
- EC: Encryption Control, which is used to indicate if the payload is encrypted;
- Type: it is used to indicate if the payload contains one or more subheaders;
- Rsv: it is a reserved field, not used;
- CI: it indicates if the payload end with a CRC portion;
- EKS: it is used to indicate the payload encryption key;
- LEN: the length of the PDU, including header and CRC;
- CID: it is the connection identifier, it in mesh mode contains link and network identifier;
- HCS: header check sequence, it is used to detect header errors.

Inside the payload of a MAC PDU, can be carried both data and management messages. The format of the management message is represented in

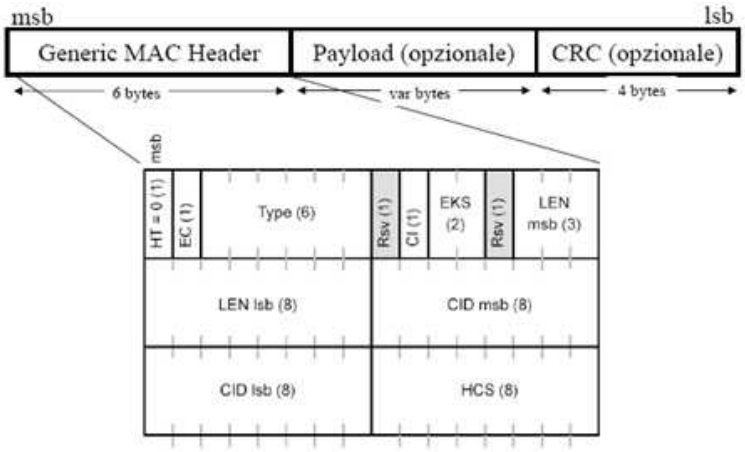

Fig. 1.4. MAC PDU

figure 1.5. The management message type contains the type of message conveyed, and the "management message payload" carries the actual message.

Fig. 1.5. Management message

In PMP mode each communication must be associated to a single service flow. The service flows are created after the SS has completed the registration protocol with the BS. The registration is completed if the SS is able to maintain synchronization with the uplink and downlink channel, while receiving DL-MAP, UL-MAP, Uplink Channel Descriptor (UCD) and Downlink Descriptor Channel (DCD) messages in regular way. The first two messages describe the allocations of bandwidth in downlink and uplink respectively, while the latter two messages describe the channel characteristics. The concepts of service flow and connection will be treated later in the subsequent section when the QoS mechanisms will be illustrated.

The PMP mode of 802.16 protocol is strongly oriented to the connection, each station is characterized by a 48 bits MAC address, and each connection is identified by a 16-bit Connection IDentifier (CID). Unlike the mesh mode, PMP mode has well distinguishable uplink and downlink. In downlink, the BS is the only station that is able to transmit, in broadcast way, without coordination with the other stations, and each user station, SS or MSS, retaining only what is directly to itself. The various SSs stations should instead share the uplink channel. The bandwidth request, by a general SS, may occur in several ways:

- Bandwidth request header;
- Piggyback request (using the grant management subheader).

The bandwidth requests can be sent during the following transmission opportunities:

- Request IE;
- Any data grant burst type IE.

The BS can allocate bandwidth to SSs, periodically, in order to allow to the SSs the bandwidth requests sending. This mechanism is called polling and it can be of two types:

- broadcast polling;
- unicast polling (including the Poll Me bit: PM).

Obviously, in the case of broadcast polling, in the same transmission slot may be a contention; in which case the contention resolution method is the use of the exponential backoff. Once the various stations are sent the bandwidth requests to the BS, it can allocate the bandwidth in two ways:

- Grant Per-Connection (GPC), in which case the BS allocates bandwidth to the single connection;
- Grant Per-SS (GPSS): in this mode of bandwidth allocation, the BS includes all the bandwidth requests, made by the same SS for all its connections, and gives to the SS a single aggregate grant, thus the user station can divide to the various connections, the granted bandwidth.

In mesh mode, as previously anticipated, there is the ability to create and manage direct links between the SSs stations. In particular, in this mode, each entity is generically named "node" and new concepts and terms absent in PMP mode are introduced. These new introduced terms are the following: neighbor, neighborhood and extended neighborhood of a node. A node is said to be neighbor of another node if there is a direct link between the two nodes, the neighborhood of a node is the set of all neighbor, or in another way is the set of nodes that are one hop away from the node, and the extended neighborhood, in addition to neighboring nodes, contains additionally all the neighbors of the neighborhood, or in other way, we can say that the extended neighborhood contains all the nodes that are two hop away from the node itself. As earlier mentioned, the BS loses the central role that characterizes the PMP mode, and in fact the basic principle that governs the mesh is the follows:

no one node can transmit on its own initiative, including the BS node, without coordinating its transmission within its extended neighborhood.

The BS does not have the central role of the only manager of bandwidth allocation, so all the network nodes have equal importance. In a network that operates in mesh mode, there are two different ways of allocating bandwidth

according to a kind of distributed or centralized scheduling. In the distributed scheduling, which in turn can be either coordinated or uncoordinated, all stations must coordinate their transmissions in their extended neighborhood. This type of scheduling uses all or a portion of the scheduling control subframe, to send its regular schedule and to propose changes of the same in a PMP mode, i.e. the messages used in this phase are sent in broadcast way.

Within a channel, all neighbors receive the same transmission schedule. All stations in a network, use the same channel to transmit the schedule information. This information will be issued in the format requests - grants. The distributed coordinated scheduling ensures that all the transmissions will take place without having to rely on the base station. The uncoordinated scheduling, respecting the constraints of coordinated distributed scheduling, can ensure communications with fast setup on the basis of individual links. The uncoordinated scheduling is determined by requests and grants between two nodes, it must also take place in a manner that does not cause collisions with messages of coordinated scheduling and its traffic. Both modes of distributed scheduling, use a *three - way - handshake* protocol.

In summary, the differences between distributed coordinated and not coordinated scheduling, are the following:

- in coordinated scheduling the control messages are scheduled in scheduling control subframe in collisions free manner;
- in not coordinated scheduling, the messages must be sent in the data traffic portion frame, and may collide with each other message.

The second mode of bandwidth allocation is based on centralized scheduling. In this case, the BS determines the flow assignments on the basis of requests received by SSs. The BS works so as in PMP mode, the only difference is that in this case not all the SSs can rely on a direct connection with the BS, hence the requests - grants message must be issued within the system in broadcast mode. The grants and the requests messages, in accordance with centralized scheduling, are broadcast only within their assigned transmission opportunities, in the scheduling control subframe. Obviously, the bandwidth allocation rules, described above, can be combined to achieve the goal of optimizing the best allocation of bandwidth resources.

The scheduling mechanisms described above, use a series of messages that are exchanged within the node extended neighborhood. Here we see these messages in order to create a complete picture of the transmission mechanism described by the protocol. The messages that we look in detail are the messages:

- MSH-NCFG;
- MSH-NENT;
- MSH-DSCH;
- MSH-CSCH;
- MSH-CSCF.

The MSH-NCFG messages has a particular importance, because they have the task to carry the configuration information and the setting parameters of the network. This type of messages can be sent in control network subframe and therefore cannot be present in every frame, because this network alternate frames containing network control subframe and scheduling control subframe. When a new node is active and it want to start registration phase, it should listen to receive MSH-NCFG messages from neighbors. The receipt of this message is essential for a new node or a node that needs to repeat the synchronization phase. In fact in this message there are the descriptions of all the network parameters: the frame slots description, the neighbor nodes, etc.

The dispatch of such messages is made in a collision free mode and this is granted by the presence of two fields in MSH-NCFG message, these fields allow the calculation of the next transmission time of each neighboring node:

- *xmt holdoff exponent*;
- *next xmt mx*.

Each node, at the instant which sends an MSH-NCFG message, will calculate its next transmission instant and expresses it in a range form by the two previous mentioned terms. In practice, the node does not say to the neighbors the next transmission instant, but sends an interval time in which the next transmission take place, this interval is defined by the following constraints:

$$nextxmttime > 2^{xmtholdoffexponent} * nextxmtmx \tag{1.1}$$

$$nextxmttime \leq 2^{xmtholdoffexponent} * (nextxmtmx + 1) \tag{1.2}$$

Between a transmission and the next one, a node must waiting for in silence for an interval time equal to:

$$xmtholdofftime = 2^{xmtholdoffexponent+4} \tag{1.3}$$

When a node sends an MSH-NCFG message, in addition to sending information about himself, it will sends also information about its neighborhood, so each node, collecting the information received from all the neighboring nodes will be able to reconstruct information about the 2 hop neighborhood (called also extended neighborhood). Within the extended neighborhood and in a certain slot, only one node can transmit.

The MSH-NENT message is used by a new node in order to carry out the requests for admission and registration in the network. When a new node is active and it wants to register itself at network, is to listen to MSH-NCFG messages, and after receiving two messages from a single source node, it can select a node to make the request for entry into the network. The selected node, to which the node make the request, is defined *sponsor node* and the new node is defined *candidate node*. The *candidate node* can use MSH-NENT messages to request the opening and subsequent closure of a channel through

which candidate acquire what is necessary to its configuration. Consequently, the MSH-NENT message may contain:

- net entry request;
- ack net entry;
- net entry close.

The MSH-NENT message as the previous MSH-NCFG, cannot be sent in any transmission opportunities, but only in a specific opportunity within a frame. This is the first opportunity that is present in the frame containing control network subframe, this opportunity is not present in every frame but appears regularly, with the same frequency, as the opportunities of the MSH-NCFG messages. The MSH-NENT message is subject to an algorithm, which decides whether a node can transmit or not, but unlike the previous message, will not necessarily happen in a collision-free manner, and if this happen, the node has to use an exponential backoff algorithm.

Other types of messages that hold great importance in the mesh mode are MSH-DSCH and MSH-CSCH messages. The MSH-DSCH messages are used in association with use of distributed scheduling, they can be sent at regular intervals to inform the neighboring nodes about the scheduling of the transmitting station. The transmission mode of the message is the same of MSH-NCFG message. In fact both types of message should be subject to the same algorithm that determines the instant of transmission of the various nodes. Also this message contains the same values, described for MSH-NCFG message, and used to determine the next transmission instant; a node must also coordinate the transmission in the extended neighborhood. Unlike the MSH-NCFG message, the MSH-DSCH message can be sent in one of the transmission opportunity present inside the scheduling control subframe. Such messages can be used for both the coordinated distributed and uncoordinated distributed scheduling, and that is for requests that are negotiated directly between two nodes. Now we describe the MSH-CSCH message. This type of message is used in case of centralized scheduling, it is sent in broadcast way from the BS to its neighbors, and the neighboring nodes will continue the processing to transmit bandwidth grants made by the BS.

In addition to transport bandwidth grants, these messages are used to transport the requests. In fact, each node can send an MSH-CSCH message containing its bandwidth request and the request of all the children nodes in its reachability subtree. Even MSH-CSCH, as well as the MSH-DSCH messages are sent in scheduling control subframe. In support of centralized scheduling mode, there is also the MSH-CSCF message, which allows the transmission of configuration messages for centralized scheduling mode. Indeed, this message contains information about the child nodes of the sender node, depending on the particular considered reachability tree. That message must be sent in the opportunities, dedicated to it, into the scheduling control subframe.

1.3.2 PHY layer

The physical layer is the lowest layer found in the protocol stack. In particular, the protocol defines a single IEEE 802.16 MAC layer but different air interfaces. Different air interfaces are defined to support the MAC level which take into account different characteristics because of the various frequency bands ranging they consider. Any system, that implements this layer, must respect the constraints set in terms of transmission techniques, supported modulation and many other specific characteristics.

The protocol provides for the possibility of using both single carrier modulation techniques and multi-carrier modulation techniques such as OFDM technique (Orthogonal Frequency Division Multiplexing). The presence of so different air interfaces is due to the will of the protocol designers, in fact in this way they want to make the transmission robust and able to adapt to the type of scenario in which network devices are operating. Considering the single carrier modulation, it is perfect for an environment where there is not a high impact of multipath fading, and therefore we can consider an environment characterized by a not frequency selective transmission channel, while the OFDM modulation, which is a very efficient multicarrier modulation, is right to the most difficult and frequency selective transmission channel. In the table 1.1 you can see the interfaces provided by protocol.

Table 1.1. Air interface nomenclature

Designation	**Applicability**	**Duplexing alternative**
WirelessMAN-SC	10-66GHz	TDD FDD
WirelessMAN-SCa	below 11GHz licensed bands	TDD FDD
WirelessMAN-OFDM	below 11GHz licensed bands	TDD FDD
WirelessMAN-OFDMA	below 11GHz licensed bands	TDD FDD
WirelessHUMAN	below 11GHz license exempt bands	TDD

The supported modulations are Binary Phase Shift Keying (BPSK), Quadrature Phase Shift Keying (QPSK) and from 16 to 256 Quadrature Amplitude Modulation (QAM) with the possibility to obtain different data rates varying the encryption type. The 802.16 technologies support both Time Division Duplexing (TDD) and Frequncy Division Duplexing (FDD) mode, allowing greater flexibility in deploying the network. In the TDD mode, downlink (related to communication from BS to SS) and uplink (related to communication from the SS to the BS) operating in the same frequency band at different

times, alternate transmission of downlink and uplink frames. Since this alternation is very quick, you have the perception that the channel is active both in uplink and in downlink in the same instant. As stated above, the TDD is used for services that have an asymmetric traffic into the two different link, such as access to the Internet. In FDD mode downlink and uplinks signals are transmitted simultaneously on two different frequency channels, and this results in an inefficient usage of resources, where the traffic is asymmetric, because the downlink and uplink spectra are unused for a long time. Therefore, summing up, while the TDD is very helpful in the case of asymmetric traffic (i.e. Internet access), or in scenarios where there is not the pair of channels; the FDD, on the other hand, is more appropriate in the case of symmetric traffic (such as Voice over Internet Protocol (VoIP)). These suggest consideration as well as the physical layer is designed with attention to the quality of transmission, the wide range of choices, allows to developers the chance to play with all possible configurations in order to achieve high quality standards.

All the 802.16 technologies use AMC (Adaptive Modulation and Coding). This feature allows you to improve performance, and optimize the throughput and the range of coverage. The AMC, in fact, provides a dynamic range of modulation and code rate for each user, depending on the condition of the radio link. When the received signal is low, as in the case of terminal far from the BS, the system automatically selects a modulation more robust but less efficient in terms of capacity (such as QPSK), in order to keep the probability of error equal to the target level. When the signal level received is high, then high modulation (such as 64 QAM) are chosen without increasing the probability of error. The capacity of WiMAX networks to use a robust scheme of adaptive modulation type, ensures broad benefits to large distances, with a high level of spectral efficiency and tolerance to the reflections of the signal. For example, if the base station is unable to establish a stable connection to a remote user using the modulation scheme of the highest level, 256 QAM, the modulation level is reduced to 16 QAM or QPSK with reduction of supply of throughput, but with increased efficiency on the distance. The so-called Adaptive Modulation and Coding (AMC) technique, have been proposed in order to be chosen the most effective scheme based on the state of the channel. The choice of levels of modulation encoding optimizes the required service. The 802.16 standard can achieves its high data rate and efficiency by using multiple orthogonal (overlapping) carrier signals instead of a single carrier approach. This parallel carrier ability is called multi-carrier modulation (MCM) or discrete multi-tone (DMT), and is ideal for addressing errors that may arise in indoor and outdoor wireless environments.

1.3.3 QoS mechanisms

As we have previously introduced, the focus of this thesis, which will discussed again in detailed way in the following of this chapter, is to extol the protocol mechanisms to create a cross layer framework to provide services with well

defined QoS levels. At this purpose, up to this point, we have introduced the basic concepts of the two layers and now we will illustrate the basic mechanisms supported by protocol to provide QoS.

The presence of these mechanisms gives the opportunity to provide services with high levels of quality, which is not achievable with other wireless standards. What makes IEEE 802.16 a strong protocol, in PMP operative mode, are few and well-defined concepts, such as the connection, the scheduling data service and the service flow. As mentioned above, the 802.16 protocol in PMP mode is strongly connection oriented, everything happens within it, and which is associated with the concept of service flow. Between BS and SS, everything happens within the connection. Let's look in the following the various concepts just introduced.

The QoS parameters are linked to the service flow, but a service flow cannot exist if not associated with a connection. And the scheduling data service, which in a certain way classify the connections, are the completion of this complex structure. A single SS may provide services to an entire building, as a result, each SS can embrace within a single connection, all types of traffic of different users, with the same characteristics. So everything revolves on the concept of connection and service flow. The connections, identified by a CID, occur between CS levels, and it create a communication channel between convergence sublayer entities. The connections can operate in a dynamic way, they can be created, their parameters can be changed and finally, a connection can be deleted. In figure 1.6 the mechanism implemented at CS is shown, i.e. the mapping of the SDUs over the corresponding connections. Obviously, this game also contributes to the QoS classification , because in such a way, the SDU not delay tolerant will never be mapped on a connection that carries best effort traffic, and consequently, an SDU of delay tolerant application will not be mapped on a connection that can handle traffic with stringent delay constraints. All this in order to optimize the quality of services.

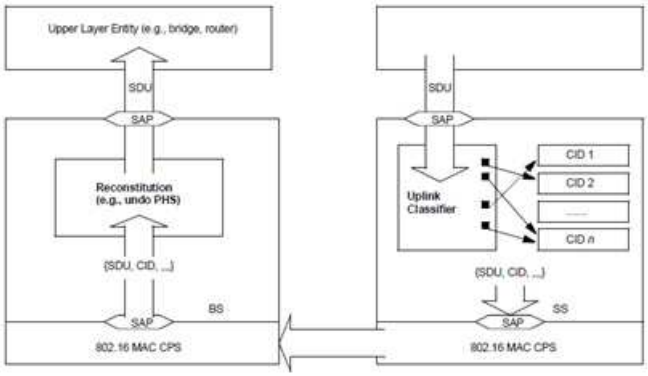

Fig. 1.6. PDU classification

You may notice that the protocol 802.16 was born with QoS in the soul also in the quality of service offered to management message traffic. In fact, after that an a SS registers itself at the BS, between them will be instantiated three different management connections with different QoS levels:

- Basic management connection: used to exchange short urgent messages;
- Primary management connection: carrying longer messages and delay tolerant;
- Secondary management connection: used to carry delay tolerant messages standards based.

Each connection is associated with a single scheduling data service and each data service is associated with a set of QoS parameters that quantify aspects of its behavior. Moreover, each scheduling data service is associated with specific bandwidth request mechanisms that allows it the respect of qualitative constraints imposed by the particular application. The following are the four types of scheduling data service supported by 802.16:

- Unsolicited Grant Service (UGS): used with real time traffic that generates fixed-size packets on a periodic basis. Example of this kind of traffic is voice over IP. A connection, mapped on this type of scheduling, has an amount of bandwidth allocated by the BS, constant over time. An SS may use this connection to request bandwidth for other connections by setting Poll Me bit in one of the MAC PDU subheader. By setting this bit, SS require a polling by BS for connection of the same SS. In this way is obtained an optimization of bandwidth and this because there is not a bandwidth waste to send bandwidth request messages;
- Real-time Polling Service (rtPS): used for real time traffic that generates packets of variable size on a regular basis, an example is an Moving Picture Experts Group (MPEG) video. The mechanisms of bandwidth request associated to this scheduling data type are piggyback request and the unicast polling. The first of these mechanisms can include a bandwidth request for the connection within a PDU that carries data. This is also a mechanism that allows bandwidth saving. The unicast polling, instead, is realized by allocating, by the BS, a transmission opportunity to the SS. In this opportunity the SS can send the amount of bandwidth request that it need;
- Non-real-time Polling Service (nrtPS): suitable for not real time traffic with packets of varying size sent on a regular basis. An example could be the FTP traffic. The mechanisms of bandwidth request allowed in this case are the piggyback and unicast and broadcast polling.
- Best Effort (BE): The Best Effort scheduling data service is used for types of traffic that have no one stringent qualitative constraints of any kind. For example we can consider data traffic generated during an Internet session. In this case, all the bandwidth request mechanisms available by the Protocol, are allowed. Generally, a base station, once accommodate

all the above types of traffic, assignes to best effort services the remaining bandwidth.

Analyzing the previous concepts, it seems clear the structure built by the protocol for the provision of quality of services. The priority offered, for example, to Voice over Internet Protocol (VoIP) or real time traffic, highlights a quality of service inherent in the nature of classes structure.

The framework will be completed and will appear in all its beauty when we go to describe the service flow concept. It represents the points of contact with the structure of the real and practical applications constraints. In fact, with the scheduling data service, we have not done anything other than a qualitative classification of traffic classes. Instead, the service flow, will make dirty its hands with the real constraints of user applications.

The QoS in IEEE 802.16 protocol is closely linked to the service flow concept: a service flow is a bi-directional flow of packets that provides a particular quality of service. Each service flow is characterized by specific qualitative constraints (time, bandwidth, etc.). A service flow is enabled between an SS and a BS and to it are assigned the necessary characteristics for the particular type of transmission required by the SS; once activated one and only one connection will be associated with it. In this way, all communications will take place between SS and BS, with certain restrictions, can be sent in a single connection within a single service flow. They are created after the SS has completed the registration protocol with the BS. Service flow of various kinds can be created:

- provisioned, is the provided service flow that has not bandwidth reserved to it. These service flow is activated in deferred way;
- admitted: is a service flow that is not activated, but with reserved bandwidth;
- activated: is an active service flow.

When a service flow is admitted it is characterized by a given CID. Each service flow is mapped onto a connection, and each connection will belong to one of scheduling data service offered by the protocol in basis of required QoS. Only an activated service flow may forward packets. For each service flow, and thus connection CID, will be associated a set of parameters, the main parameters defining the qos for the particular services are:

- MSR: Maximum Sustained Rate;
- MRR: Minimum Reserved Rate;
- maximum-latency;
- maximum jitter;
- priority.

In downlink, once the MSR parameter is defined, the BS does not need any more. For a given connection, there is a mapping with an active service flow and the minimum data rate is guaranteed by MRR parameter. Each

connection can try to transmit with a higher data rate, of course, by making a request in accordance with the scheduling data service. The BS will enable this increase until the value expressed by MSR parameter is achieved. Therefore we can say that the MRR associated with the different services, act as the "guarantee", while the MSR serves to limit a connection. In 802.16, all the service flows have a 32-bit service flow identifier (SFID). Active service flows also have a 16-bit connection identifier (CID) which is in turn associated with a connection.

A service flow is characterized by the following attributes:

- Service Flow ID: As mentioned above, a SFID is assigned to all existing service flows. The SFID serves as the principal identifier of a service flow for the subscriber station and the base station. A service flow has at least an SFID and an associated direction;
- Connection ID: Mapping to an SFID exists only when the connection has an admitted service flow;
- a QoS parameter set provisioned via the network management system;
- a set of QoS parameters for which the base station (and possibly the subscriber station) is reserving resources. The principal resource that is reserved is bandwidth, but may also offer required guarantees on latency.

Service flows can be either dynamic (created using native 802.16 control messages) or static (provisioned through the network management system). A dynamic service can be created, modified or deleted using a set of management primitive function provided by protocol.

Since multiple service flows may need to share a common set of QoS parameters, the protocol developers have introduced the concept of service classes or service class names. A service class is an optional object that may be implemented at the BS. A service class is defined in the base station and has a particular QoS parameters set. The QoS parameters set of a service flow may contain a reference to the service class name (SCN) that selects all of the QoS parameters of the service class. The service classes also identify the service characteristics of a service flow to external systems such as a billing system or customer service system. For consistency in billing, operators should ensure that SCNs are unique within an area serviced by the same operation support system (OSS) that utilizes this interface.

Unlike the PMP mode, in the mesh mode case, the service flow concepts related to connection and the resulting service flow characteristics, do not exist anymore. The only claims made by the protocol for QoS issues, say that the quality of service must be guaranteed, in the link context, packet by packet. It must be the transmitting node, within the constraints of the distributed bandwidth allocation algorithm, to ensure compliance with the constraints of the individual application quality.

Thus, to realize end satisfy the qos constraints, the protocol defines particular fields within PDU header. As we have seen, the header of a generic MAC PDU, contains a 16 bits CID field. In PMP mode, this field contains

the identifier of the BS - SS connection, rather in the mesh mode, CID field is split into two parts, the first portion of 8 bits is the logical network identifier, the second portion of the same size contains the link identifier. This is true in the case of MAC management broadcast message. If the MAC PDU contains a data payload, the first 8 bit portion of the CID is redistributed on a set of four fields used to implement QoS policies. The four fields are:

- Type: indicates whether the PDU is a management message or an IP datagram; it is 2 bits long. The other two configurations of the field are reserved for future developments;
- Reliability: indicates the number of admitted retransmissions for the MAC PDU in question, there are two different configurations, the first has no chance of retransmission and the second one allows a maximum number of retransmissions equal to 4;
- Priority / Class: it indicates the priorities associated with the membership class of the message;
- Drop Precedence: a message with a high drop precedence value has a high probability of being eliminated in case of network congestion.

The presence of these four fields, and especially the last two, provides to the protocol the capabilities to create services classes in which to map the various user applications, defining a priority and providing to the nodes the capability to drop a packet belonging to a particular class, according to its weight. The MAC layer of IEEE 802.16 protocol does not define any explicit instrument for the management and for the assurance of the guaranteed quality of services. The protocol voluntarily leaves gaps that implementers will go to fill, creating algorithms for bandwidth allocation and qos management, exploiting the basic mechanisms offered by protocol. The implementer has other mechanisms available under the protocol; these mechanisms can be used to improve the provided quality of service. These factors are closely linked to nature and structure of the frame designed in the protocol.

As just said shows that the nature of cross-layer architecture, designed to ensure the QoS, is inherent to the protocol itself and it is not an abstract idea away from it. In order to understand these new mechanisms, is required to define the structure of the frame used on the physical layer. A frame in mesh mode is divided into two parts:

- control subframe
- data subframe.

In turn, we can individuate two types of control subframe: the first is used to create and maintain the cohesion of the structure, and it is called, as shown in figure 1.7, Network Control subframe. The second type is used to coordinate the centralized and/or distributed scheduling within the network and is defined Schedule Control subframe. Not all the frames are therefore used to hold any kind of message. A frame can contain, exclusively, a network control

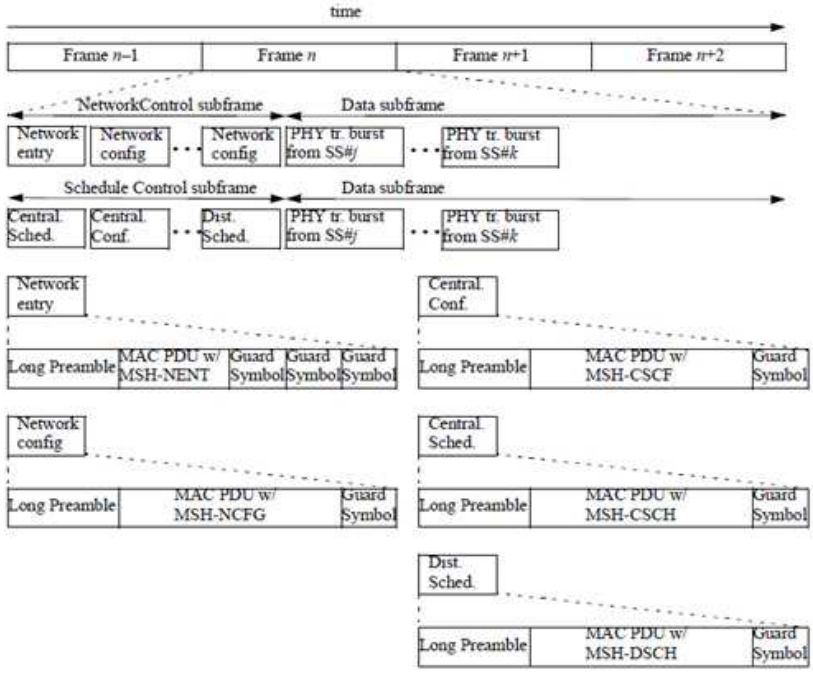

Fig. 1.7. Mesh frame

subframe or a scheduling control subframe, in alternate way. The second type of subframe is more frequent than the first one. The periodicity of the subframe type, the number of transmission opportunities to put a certain type of messages and other network parameters are dynamically settable and are broadcast in the Network Descriptor contained in the MSH-NCFG messages; it can be transmitted within the transmission opportunities of the Network Control subframe, except for the first opportunity, which can be used to send only MSH-NENT messages. The frames that contain the Schedule Control subframe are authorized to carry MSH-DSCH, MSH-CSCH and MSH-CSCF messages.

The protocol is able to define how many frames containing scheduling control subframe occur between two frames containing network control subframe. Of course the idea is to find a value that represents a good compromise of alternation between the two types of subframe. This is because a little number of scheduling control subframe make slow the bandwidth allocation process: there would be a little number of transmission opportunities for MSH-DSCH, MSH-CSCH and MSH-CSCF messages and therefore should be to collide with the efforts to obtain certain levels of quality of service. On the other hand, however, a great number of scheduling control subframes, could make excessively slow the responses to requests for network reconfiguration, because this

would decrease the transmission opportunities for MSH-NENT and MSH-NCFG message.

Another factor which affect the quality of any algorithm of bandwidth allocation and therefore any mechanism used to meet certain QoS constraints, is to establish the behavior of the *"xmt holdoff exponent"* parameter. This parameter, as previously introduced, determines the ineligibility time of a node, that is, determines the time of silence between an MSH-DSCH message transmission and the next. Consequently, high values of this parameter makes a node too slow to send bandwidth requests, consequently, also the granter node is slow in responding. Optimization is therefore very important to calculate the range of *xmt holdoff time* value. Looking at the equation proposed in the protocol that allows the calculation of that interval (1.3), we can note the presence of the *"4"* as a fixed part of exponent. This fixed part can lead to continued growth of silence, which is the time interval between two successive transmissions. Consequently, an ad-hoc algorithm that can calculate the exponent value in a dynamic and adaptive way, based on traffic and depending on the network nodes density, it would be an interesting solution to obtain an optimized MAC layer for QoS provided to users.

1.4 A brief introduction to the addressed issues

Previously we have introduced the basic concepts related to PHY transmission techniques and to QoS mechanisms of the IEEE 802.16 protocol. The analysis of both PHY and MAC layer, represents the first step to reach our focus: to enrich the 802.16 technology in order to contribute at the creation of an 802.16 network architecture capable to provide a wide variety of services characterized by well defined QoS levels; our contribute will be expressed by designing and developing models (related to PHY layer) and algorithms (related to both PHY and MAC layers) to support QoS improvement.

1.4.1 Channel error models: to improve the QoS and to make easy the systems simulations

Related to PHY layer, our intent is to design channel error model useful to improve QoS of overall system and to facilitate the software simulation which want to take into account also environmental impairment effects.

Wireless propagation channels can be classified in two major categories. The first category is analog or physical channels, where the parameters of interest are the received signal strength, the noise and/or interference power, the mobile speed, etc. Channel models for physical channels place emphasis on describing the fading characteristics of the received signal. The second category is digital channels, where we are interested in the number and distribution of error events in a sequence of packets. A digital errors encountered in digital wireless channels are not independent but occur in bursts or clusters.

Channel models for digital channels are called error models, which aim at describing the statistical properties of the underlying bursty error sequences. Error models are either descriptive or generative. A descriptive model analyzes the statistical behavior of target error sequences obtained directly from a real digital channel or by a computer simulation of the overall communication link. A generative model specifies a mechanism that generates error sequences statistically similar to the target error sequences. Compared with a descriptive model, the main advantage of a generative model is that it can greatly reduce the computational effort for generating long error sequences, and therefore speed up simulations.

The creation of a generative model for a given channel is interesting for different point of views:

- When compared with standard approaches simulating the overall communications link, generative models provide a method to reduce the computational load of generating long error sequences.
- Many relevant channel statistics that are utilized to evaluate communications systems can be analytically derived from the generative model. Therefore, as long as the generative model is not too complex, this procedure will be more efficient and accurate than the bit-by-bit processing of the original error sequences.
- The generative model can be used to predict channel behavior. The prediction of a received packet state can be intended as the probability related to the received packet state and thus consequently expressed as the probability that the packet can be lost or in a complementary way that the packet can be received as error free. The need to have this kind of model is related to the ability to predict channel behaviour in certain situations in order to optimize data transmission instant by instant; in fact, the presence of a model that is able to predict whether the next transmitted packet arrives to the receiver side as wrong or as error free allows modification of such parameters, as the packet dimension, QoS constraints and others, to optimize network throughput. Also route selection metric can benefit by this model.

1.4.2 Call admission control: to improve bandwidth management

In the mesh mode supported by IEEE 802.16 standard protocol, the network can be constituted by a set of mesh nodes which can communicate among them respecting the constraints of a centralized or distributed coordination. If each node has a large amount of data to be disposed, every node will make constant demands for bandwidth. If these requests are not properly managed and regulated, in centralized or distributed way, the network can fall in a congested state; this obviously, causes performances and QoS degradations. In order to avoid congested state it is possible to act in two ways:

- in proactive way, trying to avoid the occurrence of congested state;

- in reactive way, reacting when the dangerous situation occurs.

In proactive way it is possible to prevent the onset of the problem in these ways:

- decreasing the bursty property of traffics, i.e. the traffics must be made more regular;
- planning which data packets are to be discarded;
- planning the admission decision for the transmission requests;
- planning the scheduling for the admitted data flows.

As said so far can be summarized in the implementation of efficient call admission control and scheduling algorithms, which are able to optimize the transmission in the network and to grant the compliance of each application with its QoS constraints. Our contribution, in this case is represented by the design of a particular call admission control (CAC) algorithm. This algorithm introduces a new concept: the bandwidth defragmentation, which at the best of our knowledge, not yet been applied to IEEE 802.16 protocol.

1.4.3 Are there multiple routes?

In an 802.16 mesh network, as in other kind of wireless network, when a source node must transmit an amount of data, it invokes a routing algorithm to individuate a route or a set of routes to reach the destination node. When the routing algorithm returns a set of routes, the source node must select one of the routes to submit data to the destination.

In order to successfully and efficiently support data transfers between source and destination nodes, a route metric plays a significant role in selecting a route with high throughput or route which guarantees the compliance with QoS constraints. However, the design of such a routing metric is difficult compared with metrics on wired networks because of unreliable links and the shared nature of wireless links. In a superficial point of view, the minimum hop count metric can appear the most suitable choice, but making a depth analysis, we can say that the minimum hop count metric tends to choose routes with longer distant links that generally have higher loss rates, which impairs the overall path performance. Therefore, it is crucial to reconsider the quality of each wireless link when designing a routing metric. The most popular routing metric in multi-hop wireless networks is the minimum hop count metric, which is used by many existing ad-hoc routing protocols such as Dynamic Source Routing (DSR) [14], Ad-hoc On demand Distance Vector (AODV) [15], Destination-Sequenced Distance-Vector (DSDV) [16]. The primary advantage for the minimum hop count metric is its simplicity and no extra measurements or overhead are needed for selecting the appropriate route. However, it has been shown that a route with minimized hop count does not necessarily guarantee the high throughput for that path also it is not useful for application with qualitative constraints. This is due to the fact that

the minimum hop count metric does not consider the different frame loss rates along the wireless links. It tends to select a path with minimal hop count but normally suffers from high loss rate link. It is well known that the wireless links with high loss rate usually require more retransmissions to successfully deliver a data packet, thus consume more medium resources.

Our contribution in this case is a study about a set of metrics applied to a wimax scenario and a proposal of new metrics to select the best route taking into account the packets loss due to the transmission channel.

1.4.4 A team effort to achieve a common goal

All the previous introduced elements as the call admission control algorithm, the channel error model and the route selection metric can work together to reach the our common goal. In fact in the last chapter of our thesis, we design and present a framework which can play the role of supporter to reach well defined level of quality of service. We utilize the various elements as a set of bricks to built a framework supported also by other two interesting bricks:

- when a node has to trasmit a set of data units to another neighboring node, it must consider the presence of a real channel between the nodes; the channel can corrupt the transmitted packets and obviously, the effect of the channel it is even more evident for packets with great size value and it could ideally be negligible for packets with very small size value; consequently, in order to guarantee the compliance with the QoS constraints, it is necessary the presence of an algorithm which is able to decide the best packet size value;
- when a node has a set of queued packets that must be sent to a destination node and the node must guarantee the compliance with delay constraints, it is necessary the presence of an algorithm which is able to estimate, as efficiently as possible, the right value of mini slot number that will be the bandwidth requirement.

These elements bring togheter the two layers of protocol stack defined by IEEE 802.16 standard: PHY and MAC; the resulting instrument can be defined as a QoS oriented cross layer framework.

2

Channel error models for WiMAX scenarios

2.1 Introduction

The need to have a channel error model is related to the ability to predict channel error behavior in certain situations in order to optimize data transmission instant by instant; in fact, the presence of a model that is able to predict whether the next transmitted packet arrives at the receiver side as wrong or as error free allows modification of parameters such as the packet size, QoS constraints and others, to optimize network throughput. Using a generative model we could predict the probability that a packet is lost (defined as a bad packet), or that the packet is received without errors (defined as a good packet). Following these concepts, the model type choice appears clear and well defined: the promising models from this point of view are the Markov chain based models. In a channel Markov chain based model, each state represents a certain channel condition with a corresponding probability of good or bad packet reception. State transition probabilities represent transitions between channel conditions. The state transition at each time instant is even more evident in a time variant channel characterized by impairment phenomena, such as the Doppler effect and multipath fading, where the latter makes the channel frequency selective.

In the literature there are various Markov chain based models applied to different scenarios, such as the WiFi or GSM (Global System for Mobile communication) scenario, but our contribution to generative models study is not only the application of a set of Markov chain based models to a WiMAX scenario, which to the best of our knowledge, is not present in the literature, but also the presentation of new models designed to achieve good performances in artificial trace generation. The selected Markov chain based models are the following: Full State Markov (FSM), Gilbert-Elliot, Hidden Markov Model (HMM) and MTA (Markov-based Trace Analysis) model. These models are used to model the channel under different configurations and the model performances were evaluated. The evaluation of Markov chain based models was carried out through statistical parameters. Each model, once defined, reflects

only one specific channel configuration. As a consequence two new models are proposed in this chapter: the first one is able to obtain good performances in a single channel configuration, the second aims to extend the model to a more general case.

2.2 Channel model: state of the art

Various models to describe channel error behavior are present in the literature. In [17] an introduction to channel modelling with Markov chain based model is described.

Among the various models, probably the most simple, is the Gilbert - Elliot model [18]. It is characterized by a clear simplicity, in fact, it is realized with a Markov chain with only two states: the good and the bad state, and by the elements of transition probabilities matrix. Through this model a generic packet can be introduced at the receiver as wrong or as error free. The results obtainable with the Gilbert - Elliot model are satisfactory, also taking into account the low complexity of the model.

Another particular model known in the literature is explained by the MTA algorithm. The MTA algorithm was proposed and analyzed in [19], and it was proposed for designing a channel error model and then was applied to the GSM system. The MTA algorithm allows a more accurate analysis of network trace. This algorithm gives the possibility of creating a channel error model that generates an artificial trace which follows the statistical property of simulation trace more faithfully than other classical Markov chain models, such as the Gilbert-Elliot. Obviously the negative side is the greater complexity of the model. In fact, through the MTA, the complete model is constituted of two states: "lossy state" and "error free state". The error free state is characterized by a deterministic process, because it represents burst of only error free packets, instead the lossy state is then expanded in another DTMC (Discrete Time Markov Chain) that generates a burst of wrong and error free packets. This model is more complex than the Gilbert - Elliot model, but it incorporates also more concepts inherent to the stationarity of the trace.

More complex Markov chain models are presented in [20] and in [21]. In the first paper an improved Markov chain model is described, in which a fading process is modelled with a finite - state Markov chain with two dimensional states given by the quantized value of the amplitude and its speed of variation. Also the model presented in [21] is more complex than the previously described models. In this paper, an improvement of the MTA algorithm is presented, i.e. the channel modelling is based on Multiple state MTA (MMTA). Also this model is characterized by high complexity. In addition, in [21] a comparison of a set of models applied to the GSM scenario is presented and the authors conclude their work electing the best performing model for the considered scenario. Work [22] follows the paper [21] guidelines, thus the authors apply

a set of other Markov chain based model to the WiFi scenario, proving that the proposal has the best behaviour.

In [23] the authors present a finite state Markov model for Rayleigh fading channels. The model is constituted of a generic number of states equal to K, and where each state is characterized by a different channel condition expressed as a function of SNR (Signal-to-Noise Ratio). The channel remains in the current state so long as the present value of SNR remains within the SNR range assigned to that particular state. A specific state, in this way, has a well defined BER (Bit Error Rate) value associated with it. The transition in this model, is only possible between two adjacent states. The validation of this model is obtained by the authors, simulating a Rayleigh fading channel with mobility user with the only speed constant value of 5 km/h, and thus no single proof of a realistic and dynamic scenario with a rapid change of user speed value is provided. The authors propose a good mathematical elaboration, but the model appears very complex in the number of states: a state for each channel condition (in BER or SNR terms). In the case of a large number of states, even the incorporation of states with similar BER values leads to effective reduction of the number of states.

In [24] first order Markov models for received signals amplitude for flat fading channel are examined; the authors present the models and conclude that these models can be used only in very slow fading channel analysis related to a short time interval. To prove this, they use autocorrelation functions. The author of [25] introduce new concepts (respect to [24]) to model, in more accurate way, the fading channel correlation; and in [26] this study is extended telling about new Markov chain based model. In [26] the authors describe a quadrature Markov chain model where the channel fading is described by its amplitude and rate of change.

In [27] the authors propose an ARMA model to realize a digital filter to simulate a multipath fading scenario, and in [28] Markov chain models and statistical analysis are applied also in free space optical link to describe channel fading.

The authors of [29] propose a channel model based on Markov process, but this model is created for indoor scenario. The model is designed using measurement data. Finally the proposed model is validated by a comparison between simulation and measurement results.

In paper [30] the authors presents an HMM technique to simulate the bit errors in wireless scenario (in particular three different cases were considered: AWGN channel, flat fading and vehicular channel); they focus attention on BPSK modulation and CDMA because these are the guidelines for the 3G. The authors demonstrate the relation between HMM order and Eb/N0 channel condition.

In work [30] the authors propose a Markov based generative model characterized by two processes: the first one is dedicated to combine error bursts with error free bursts while the second is dedicated to create individual error bursts. This model is validated considering Enhanced General Packet Radio

Service (EGPRS) transmission system. Also in paper [31] rural and urban scenarios with EGPRS transmission system are considered and the authors propose and analyze a new class of generative models to describe channel error behavior. The proposed model performances are compared with SFM (Simplified Fritchman Models) models. Other interesting works related to channel error behavior are [32] - [36].

2.3 What we propose in this regard?

In the literature there is a great number of channel models, both Markov chain based models and other kind of channel models designed as a filter. Considering all the previous works, collected by literature, no one of these apply a channel model to a WiMAX scenario, in this way there are not works about performances analysis of Markov chain based model in a IEEE 802.16 scenario.

We select a set of models that are characterized by not great complexity and which present good performances in scenario in that they have already been applied; we analyze their performance in a WiMAX scenario. It is true, the performance analysis of Markov models has been studied in other scenarios different from the WiMAX scenario, then, in a first consideration, the same analysis in a WiMAX scenario may seem repetitive and unnecessary. In reality this is not true, because there is no clear solution about the choice of the Markov model and its states number. This choice may depend by many factors, such as the particular application, the complexity of the model, the accuracy, the modulation/demodulation and the encoding used.

To analyze the models performance, the scenario is set with particular conditions of user mobility, packet dimension, multipath effect etc; by simulation thus the packet state trace can be obtained, i.e. a sequence of *"1"* and *"0"* flags that indicates whether the packet is received as wrong or as error free respectively, by this trace it is possible to carry out the parameter values that describe the selected model. After this first step it is possible to make a performance comparison. But, as we have already anticipated, each model, once defined, reflects only one specific channel configuration, in particular the Gilbert-Elliot model is characterized by only the transition probabilities matrix M. The M matrix is the "parameter" that "configures" the model and it is strictly related to the configuration scenario, thus if a change to scenario configurations is required, even for only one of its parameters, for example the user mobility speed value, consequently the model and thus the M matrix must be recalculated. In conclusion if the probability value is required in order to obtain a wrong packet in a specific time instant, under a specific value of user mobility speed, a model is needed for every speed value.

Obviously this situation is not practical in a realistic application. The novelty of our work is not only the application of a set of Markov chain based models to a WiMAX scenario but also the presentation of a new model that

overcomes the previous explained problem. At each time instant the IWPM (Instant Weighed Probability Model) is able to calculate the probability value to obtain a wrong or an error free packet at the receiver side, and obviously under any variables configuration that describes the scenario.

Another novelty presented in the chapter is the design of a Hybrid Markov chain based model obtained by the merging of the MTA and the Gilbert-Elliot model. The idea of Hybrid is obtained by observing performances of evaluated models. Summarizing, our proposals are listed below:

- A comparison among the most know channel Markov modeling techniques to evaluate the link error model of a wireless channel under IEEE 802.16 PHY layer.
- A definition and evaluation of an hybrid model that tries to combine the benefits of the best Markov models considered in the performance evaluation in order to improve the model approximation.
- A proposal of a novel channel model able to consider the node mobility and non stationary conditions of the trace files generated by the channel model implemented in Matlab tool.

2.4 Markov chain based models

In this subsection we, first introduce briefly the markov chain basic concepts and then we illustrate the selected models used to make the performance comparison in the WiMAX scenario.

A Markov chain is a stochastic process, where if *"t"* is the observation instant, the process evolution from instant *"t"* depends only on this instant and not on previous temporal instants. To define precisely a Markov chain, or process, it is necessary to define the following value:

$$P(X_N = x_N \bigcap X_{N-1} = x_{N-1} \bigcap ... \bigcap X_1 = x_1 \bigcap X_0 = x_0) \tag{2.1}$$

the previous value is the probability that the system is in state x_0 at the time t_0, in state x_1 at the time t_1 and so on up to the value x_N at the time t_N. Using conditioned probability relation and indicating this value with P_c:

$$\begin{aligned} P_c = P(X_N = x_N | X_{N-1} = x_{N-1} \bigcap ... X_1 = x_1 \bigcap X_0 = x_0) * \\ * P(X_{N-1} = x_{N-1} \bigcap ... \bigcap X_1 = x_1 \bigcap X_0 = x_0) \end{aligned} \tag{2.2}$$

Because we hypothesize that the considered process is a Markov process, and considering the conditioned probability relation, it is possible to obtain the final relation as:

$$P_c = P(X_N = x_N \bigcap X_{N-1} = x_{N-1} \bigcap ... \bigcap X_1 = x_1 \bigcap X_0 = x_0) =$$
$$= P(X_0 = x_0) \prod_{K=1}^{N} P(X_K = x_K | X_{K-1} = x_{K-1}) \tag{2.3}$$

This value is a generic element of matrix M that describes the Markov chain. This matrix is the transition probability matrix and it is defined as the stochastic matrix, because it respects the condition that the sum of the elements of each row must be equal to one. This is expressed by the following equation:

$$\sum_{j=1}^{N} p_{i,j} = 1 \tag{2.4}$$

This chapter is not intended to be a tutorial on different treated models, the various selected models are only briefly introduced and the attention is focused on the evaluation of their performances.

For a tutorial about theoretical basis of Markov modeling of fading channel you can see [37]. Here we introduce the principles of channel modeling and treat some application examples.

2.4.1 Gilbert - Elliot model

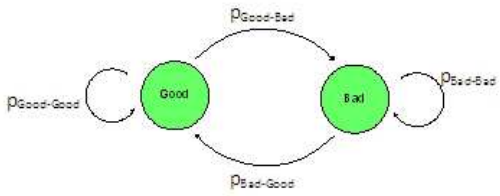

Fig. 2.1. Gilbert - Elliot model

The Gilbert - Elliot model is a simple two-state model, see [18]. The two states are "Bad" state and "Good" state. Thus, the chain can be described with a matrix of $2 * 2$ elements (see equation (2.5) and figure 2.1). The Bad state corresponds to a packet received incorrectly, whereas the Good state corresponds to a packet received in error free manner; thus, Good and Bad are associated with "0" and "1" trace element respectively. Each element of M represents a probability. For example, in equation (2.5) $p_{Good-Bad}$ represents the probability of transiting from Good state of the chain to Bad state.

$$M = \begin{bmatrix} p_{Good-Good} & p_{Good-Bad} \\ p_{Bad-Good} & p_{Bad-Bad} \end{bmatrix} \tag{2.5}$$

The elements of the matrix, can be calculated counting the occurrences of states in the simulation trace, thus they can be calculated by:

$$p_{Good-Bad} = \frac{O_{Good-Bad}}{O_{Good}} \tag{2.6}$$

$$p_{Bad-Good} = \frac{O_{Bad-Good}}{O_{Bad}} \tag{2.7}$$

$$p_{Good-Good} = 1 - P_{Good-Bad} \tag{2.8}$$

$$p_{Bad-Bad} = 1 - P_{Bad-Good} \tag{2.9}$$

In equation (2.6) $O_{Good-Bad}$ is the number of occurrences of good packets followed by bad packets; $O_{Bad-Good}$ in (2.7) is the number of occurrences of bad packets followed by good packets; instead the O_{Bad} and O_{Good} are the total occurrences of bad packets and good packets respectively.

The equations (2.8) and (2.9) allow calculation of the $p_{Good-Good}$ and $p_{Bad-Bad}$ values using the stochastic condition expressed by (2.4).

2.4.2 FSM (Full State Markov)

FSM is a k^{th} order chain model, see [38]; its order is related to the memory property of the trace generator process. A k^{th} order model considers a memory packets number equal to k. To individuate the FSM states from trace, a k size sliding window has to be considered, thus at each step, scanning the trace, a packet flag (on the right) enters the window and a packet flag (on the left, i.e. the most significant in the window) leaves the window. In figure 2.2 an example of state sequence individuation is depicted.

The red square represents the sliding window, and at each step it scans the trace and individuates a new state. In this way, scanning the trace, it is possible to calculate states occurrences, and then in the same way, as Gilbert - Elliot, it is possible to compute a transition probabilities matrix (see equations (2.6) - (2.9)).

2.4.3 HMM (Hidden Markov Model)

The Hidden Markov model [39], [40] in the literature is related to various theoretical problem solutions, but in our application, using HMM to model a link error behavior it is necessary to estimate three parameters. The first one estimation is related to the number of states constituting the Markov chain;

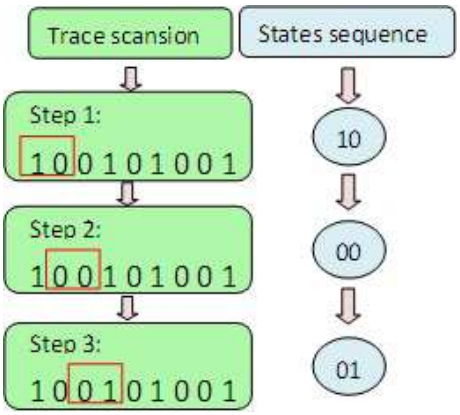

Fig. 2.2. Simulation trace scansion and states sequences individuation

in fact, during its evolution the channel switches between a set of k states following a Markov chain, thus it is need to define the space state $S = 1, 2, ...k$. In our analysis we estimate the S cardinality in a practical way: we individuate three channel states because we associate to the channel three different degradation levels useful to evaluate QoS for user applications. We select two bounds which are related to Packet Error Rate (PER) bounds for multimedia ($PER < 1\%$) and for voice applications ($PER < 4\%$); consequently three different channel degradation levels and obviously three states for the Markov chain can be individuated as *"low"*, *"medium"* and *"high"* degradation. Thus scanning the trace with a particular sliding window size, to each window, it is possible to relate a specific degradation condition. The window size is determined by:

$$U = 2 * mGL \tag{2.10}$$

where mGl is the maximum value of error free burst length, i.e. this value is the length of the longest sequences of *"0"* in the packet error trace. The choice of this value is related to the idea, expressed in [19], that a sub-trace with this size, maintains the same statistical property as the only original trace. The transition of channel, from one state to another, is not visible to observer and is due to a presence of transition probabilities matrix M. The second estimation is thus the estimation of the matrix M. This matrix can be evaluated in experimental way considering the simulation trace. The first step is to scan the trace to individuate the degradation states previously described and with a second step it is possible to build a new trace in which we have a sequence of states: *"high"*, *"medium"* and *"low"*. From the new trace, counting the state occurrences and considering equations similar to (2.6), (2.7), (2.8) and (2.9), it is possible to elaborate the transition probabilities matrix. The observer does not know the actual state of the channel, governed by transition probabilities matrix, but it can see, instant by instant, a symbol showed by

model. Each state of chain is related to a set of symbols defined symbols alphabet: *Z*. we are using HMM to model a wireless communication channel, thus, we identify the symbols alphabet with the set of admissible states for the transmitted packets. In this way we identify Z with the set: $\{0, 1\}$. The last estimation phase necessary to complete the model is to define the observation matrixes which define the symbol extracted by the model when the channel is in a particular state. This estimation can be made experimentally, as we made for the evaluation of M matrix, counting the occurrences of bad or good packet in each channel state individuated in the simulation trace.

For a complete description of HMM see [39].

2.4.4 MTA (Markov-based Trace Analysis)

MTA is a more complex Markov model [19], as shown in figure 2.3. This model is composed of two states: free error state and lossy state. These two states can be obtained by packet error trace in this way: the original trace must be partitioned into sub-states (or sub-sequences) with the help of a constant state change *"C"* defined as:

$$C = M_x + D_x \tag{2.11}$$

where M_x is the mean value and D_x is the standard deviation of burst error lengths in packet trace. For greater clarity a burst error is a sequence of *"1"* in the error trace. Once this change of state constant is obtained, the original trace partition can be made in this way: two states can be individuated in the original trace: error free state and lossy state. A lossy state is a burst of *"1"* and *"0"* that starts with *"1"* and can contain *"1"* and also a sequence of *"0"* with length less or equal than constant state change C. An error free state instead is a burst of *"0"* with length greater than *C*. Once the trace is partitioned, then it is possible to obtain two different traces or processes: the first by concatenating the lossy states and another by concatenating the error free states. These two new processes have the stationarity property for construction. Specifically, the second one is a deterministic process.

Thus two states have been individuated: the first describes the lossy trace and the second describes the error free trace (see figure 2.3). The lossy process is not deterministic as the error free process, thus, it must be modeled by another DTMC (Discrete Time Markov chain) "inserted" in the lossy state. The criterion to calculate the DTMC order for lossy trace model is the conditional entropy (such as considered later). The conditional entropy is an indication of the randomness of the next element in the generated trace. Thus considering a very low value of conditional entropy, or "randomness", the next generated value stretches to become deterministic. However, it must be kept in mind that the conditional entropy value is related to the DTMC order. In this way, decreasing randomness causes an increasing order of DTMC. At this point, determining a compromise between the two trends becomes necessary. The equation to calculate the conditional entropy is taken by [19]:

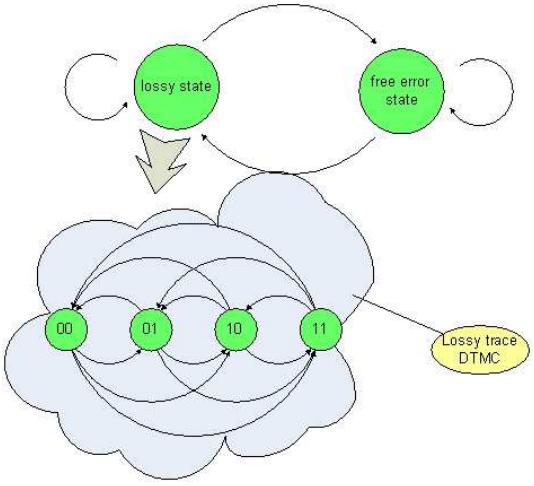

Fig. 2.3. Markov chain model obtained by MTA algorithm

$$H(i) = -\sum_{\mathbf{x}} \frac{\epsilon(\mathbf{x})}{T_{samples}} * \sum_{y\in\{0,1\}} \frac{\epsilon(y,\mathbf{x})}{\epsilon(\mathbf{x})} * \log_2 \frac{\epsilon(y,\mathbf{x})}{\epsilon(\mathbf{x})} \tag{2.12}$$

where i is the DTMC order; $\mathbf{x}$ is a vector of elements and represents one of the 2^i different sequences of i consecutive elements in the trace; $T_{samples}$ represents the number of sequences of length i in the trace; $\epsilon(\mathbf{x})$ represents the number of occurrences of $\mathbf{x}$ in the trace; $\epsilon(y,\mathbf{x})$ represents the number of occurrences of $\mathbf{x}$ in the trace, followed by y, with $y \in \{0,1\}$. For convention, during calculation of $(arg)log_2(arg)$ term, $arg = 0$, $0 * log_2(0)$ must be considered equal to *0* (see [19]). For greater clarity a DTMC of order i corresponds to a DTMC with 2^i number of state. Thus the final model is that seen in figure 2.3.

2.5 Markov chain based model performance evaluations

In this section, as first step, we will present the WiMAX scenario, the channel transmission model and the simulation settings implemented in the Matlab simulator, subsequently we will introduce the statistical parameters used for the performance evaluation and finally a performance comparison for the models introduced in section 2.4 will be commented.

2.5.1 WiMAX scenario and transmission channel implementation

The considered scenario is a low population density scenario, such as a small town or a rural environment, in which a Base Station provides wireless broadband services to mobile users. The user mobility is considered in a speed range

of 0 - 120 km/h, being typical vehicular mobility. The scenario is characterized by wireless transmission that is affected by phenomena that are different from the wired counterpart. Thus a set of impairment effects contribute to deteriorate the signals. The effects that can be individuated in our scenario are multipath effect, Doppler effect and path loss. These effects are not negligible in a realistic simulation.

The first of these is due to the objects around the environment where wireless communication takes place. It is the result of the reflections and refractions of the waves against the obstacles between a transmitter and a receiver. A classical approach to represent a multipath channel is the channel impulse response characterization as a pulses train at different amplitudes. By these pulses it is possible to obtain the delay spread value that represents the spreading delay between the arrival of the first signal and the last. A mathematical model of the multipath effect is presented in the following expression:

$$g(t,\tau) = \sum_{i=1}^{N} g_i(t)\delta(t-\tau_i) \tag{2.13}$$

In the previous equation g_i is the impulse response of only one path, and with δ function the delay spread is represented. Altogether, N different paths can be considered. Since the receiver and some of the objects that reflect the signal in the environment can move, channel impulse response is a function of time t and delay τ_i.

The path loss is the signal attenuation due to the transmitter (Tx) - receiver (Rx) distance and to the type of transmission scenario. In this case the Walfish-Ikegami model has been applied [41]. This model allows the path loss calculation for a distance between Tx and a Rx that falls in the range [0.02-5Km]. The applied equation is presented in the following:

$$L_{[dB]} = 42.6 + 26\log(d_{[km]}) + 20\log(f_{[MHz]}) \tag{2.14}$$

Due to mobile receiver motion as well as the nature of the path, the transmitted frequencies undergo Doppler frequency shifts [42], [43]. In the case of user mobility, to take this effect into account, the Clark spectrum calculation is adopted. In order to obtain the Clark's Power Spectral Density (PSD) the equation (2.15) is applied:

$$P(f) = \frac{1}{\pi}\frac{1}{\sqrt{f_d^2 - f^2}} \tag{2.15}$$

under the following condition:

$$|f| < f_d \tag{2.16}$$

where f_d (frequency shift) can be calculated using the following expression:

$$f_d = \frac{v}{c} f_c \tag{2.17}$$

where v is the relative speed between the transmitter and receiver, c is the speed of light and f_c is the carrier frequency. This spectrum with the path loss calculation will provide the impulse response of the channel where the time-varying nature is associated with the coherence time. The channel transfer function changes only slightly during coherence time. The commonly used approximation is:

$$T_c = \frac{1}{f_{Dmax}} \tag{2.18}$$

where f_{Dmax} is the maximum Doppler frequency. Another way to take into account this effect is to consider an over-modulation of carrier frequency, and the entity of this modulation is the f_d value.

2.5.2 Simulation settings

To evaluate the selected Markov chain based models, applied to channel behavior, a simulator that takes into account each previously described scenario aspect and also IEEE 802.16 PHY indications has to be designed.

The transmission chain as indicated in IEEE 802.16 protocol is modeled in Matlab tool [44]. In particular, WirelessMAN-OFDM air interface (without MIMO) is chosen, and this is because we must try to make the system robust against the mitigation effects. In fact OFDM is developed to be used in time-variant multipath scenario. As indicated by protocol, the input data sequence is baseband modulated, using a digital modulation scheme, then the data symbols are parallelized in n different substreams. Each substream will modulate a separate carrier through the IFFT (Inverse Fast Fourier Transform) modulation block, which is in fact the key element of an OFDM scheme. A cyclic prefix (CP) is inserted in order to eliminate the inter-symbol (ISI) and inter-block interference (IBI) (See figure 2.4). Data are back-serial converted, forming an OFDM symbol that will modulate a high-frequency carrier before its transmission through the channel. To the receiver, the inverse operations are performed. On the receiver side, the CP is removed before any signal processing starts. If the length of the CP interval is larger than maximum expected delay spread, all reflections of previous symbols are removed and orthogonality is restored.

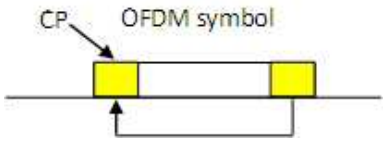

Fig. 2.4. OFDM symbol

The orthogonality is lost when the delay spread is larger than length of CP interval. Inserting CP has its own cost, we lose a part of signal energy since it carries no information. The loss according with [45] is measured as follows:

$$SNR_{lossCP} = -10\log_{10}\left(1 - \frac{T_{CP}}{T_{sym}}\right) \tag{2.19}$$

where T_{CP} is CP interval length and T_{sym} is the OFDM symbol duration. Although we lose part of signal energy, the fact is that zero Inter Carrier Interference (ICI) and ISI situation pay off the loss.

The simulator was realized through the Matlab tool and considers the multipath scenario under the WirelessMAN-OFDM PHY air interface. The simulator considers a transmitter (BS) and a receiver (SS) in a low densely scenario, in which relative speed and distance are varied in order to obtain simulation results for different scenario configurations. The simulator is set as shown in table 2.1. The frequency carrier f_c and the modulation used in each scenario are 2GHz and QPSK respectively; the paths number is three, the first path is the Line Of Sight (LOS), the second path has $0.1\mu s$ delay and the third has a delay of $1.4\mu s$. With the introduction of multipath fading, the channel is frequency selective, and OFDM modulation, as explained in the IEEE 802.16e protocol, was implemented in the simulator. Thus, the other parameters listed in table 2.1 are related to OFDM: F_s is the sampling frequency; T_s is the OFDM symbol duration; T_g is the cyclic prefix (CP) duration; and the resulting subcarrier spacing is obtained by:

$$\Delta f = \frac{F_s}{NFFT} = 11.4KHz \tag{2.20}$$

with $NFFT = 256$. The simulator is able to carry out a set of packet error traces. A packet error trace is a sequence of flags *"1"* or *"0"*, where flag *"1"* indicates that the packet is received as wrong, whereas flag *"0"* indicates that a packet is received as error free. Thus, a trace describes the channel error behavior, or so we can say that it depicts the MAC - to - MAC link; in fact in the considered simulator transmission chain, also physical layer error detection and correction instruments are involved, thus a packet is considered as error free, if the physical error correction techniques (as Reed-Solomon code) fail the data unit reconstruction and consequently it is not able to deliver an error free data unit to MAC upper layer. We indicate generally this unit as "packet". Obviously no one ARQ (Automatic Repeat Request) mechanism is considered. These traces are defined as "simulation traces", and by these traces, the parameters that define each Markov model are calculated. Once the models are set (by transition probabilities matrix computation), they are subsequently tested, and to do this a set of packet error traces are generated by models. These traces are defined as "artificial trace", and then the performances model evaluation is effected by comparing simulation traces

with artificial traces. To make this comparison a set of statistical parameters is used.

Table 2.1. Simulation settings

PARAMETER	VALUE
Modulation	QPSK
$BW(MHz)$	2.5
$Fs(MHz)$	2.92
$Ts(\mu s)$	109.59
$Tg(\mu s)$	21.92
$Bitrate(Mbps)$	1.8
$No.of path$	3
$Eb/N0(dB)$	22
Delay spread(μs)	$0 - 0.1 - 1.4$

2.5.3 Performances Parameters

In this section the performance parameters used to make performance evaluation are discussed. The parameters of each model are calculated through packet error traces obtained by simulations. Thus, each model describes channel error behavior and has the capability to generate an artificial packet error trace. For each model a number of artificial traces are obtained and to compare models performance, the artificial traces are statistically analyzed and compared with the simulation traces. To evaluate performances a set of statistical properties is considered and applied to two different random variables elaborated by traces. The variables are B and G; the first indicates the error burst length and the second indicates the error free burst length. The statistical property considered to evaluate the model is the following:

- Entropy Normalized Kullback-Leibler distance: this value, indicated below as the *ENK* value, is a statistical divergence measure between two probability distributions. The *ENK* value is a metric derived from the Kullback-Leibler distance and it is presented in [27]. The relation (2.21) allows calculation of the *ENK* value:

$$ENK(p(x)||q(x)) = \frac{D(p(x)||q(x))}{H(p(x))} \tag{2.21}$$

  here $H(p(x))$ is the entropy value that normalizes the Kullback-Leibler distance $D(p(x)||q(x))$. The former is defined by:

$$H(p(x)) = -\sum_{x \in S} p(x) \log(p(x)) \tag{2.22}$$

  and instead, the second is:

$$D(p(x)||q(x)) = \sum_{x \in S} p(x) \log \left(\frac{p(x)}{q(x)} \right) \tag{2.23}$$

In the relation (2.21) x is a random variable defined over a letter set S. Instead $p(x)$ and $q(x)$ are two probability distributions defined for the random variable x. The *ENK* value, as defined by equation (2.21), can be computed between two distributions. Initially three packet error traces obtained by simulations are considered; these traces are denominated s_1, s_2 and s_3, and then *ENK* values are computed on these traces in this way:

- $ENK(S_1||S_3)$: S_1 is the probability distribution of a random variable, elaborated by trace s_1. S_3 instead is the probability distribution elaborated by trace s_3.
- $ENK(S_2||S_3)$: similarly S_2 and S_3 are the probability distributions evaluated on random variable elaborated by trace s_2 and s_3 respectively.

These two values are considered as reference values for *ENK* values computed over distributions extracted by artificial traces. Thus, for each model an artificial trace is generated and $ENK(S_1||X_m)$ and $ENK(S_2||X_m)$ are computed, where X_m is the probability distribution derived from the artificial trace. This procedure is repeated for each model and, then, the *ENK* values obtained from each model are compared with the pair of values initially computed (reference values). If the $ENK(S_1||X_m)$ and $ENK(S_2||X_m)$ are smaller than the reference values then the considered Markov chain based model is a good model for the channel, i.e. it models channel error behavior with good approximation. Obviously the *ENK* values are related to a specific random variable and also the goodness of the model is related to a variable choice. Thus, two random variables are considered, and the procedure is repeated for both B and G. In order to avoid to relate the computed value to a particular trace, the $ENK(S_1||S_3)$ and $ENK(S_2||S_3)$ values are computed as mean value by a set of simulation traces, instead the $ENK(S_1||X_m)$ and $ENK(S_2||X_m)$ are computed as mean value by a set of artificial traces.

- Standard error: it is an error measure that can be computed between two random variable distributions. As previously explained, B and G random variables are considered, and standard error is used to calculate the "distance" between artificial trace burst lengths distribution and simulation trace burst length distribution related to both B and G variables. The relation (2.24) allows to calculate this error.

$$Er = \sqrt{\frac{(n_1 + n_2)\left[\left(\sum_{x \in S} x^2 - \frac{\left(\sum_{x \in S} x\right)^2}{n_1}\right) + \left(\sum_{y \in S} y^2 - \frac{\left(\sum_{y \in S} y\right)^2}{n_2}\right)\right]}{(n_1 * n_2)(n_1 + n_2 - 2)}} \tag{2.24}$$

In equation (2.24) x and y are random variables defined over an letter set S, and n_1 and n_2 are the number of values that x and y random variables assume respectively.

- Mean and standard deviation: these statistical values are calculated, as before, both on simulation traces random variable distributions and both on random variable distributions related to artificial traces generated through Markov chain based models. The comparison of these values gives an idea about approximation quality between artificial traces and simulation traces, consequently about models and simulation results.

2.5.4 Performance evaluations

In table 2.2 performance evaluation results are summarized. In this table the results are presented related to a particular scenario with a fixed user speed value equal to 20 km/h and a transmitter receiver distance equal to 1 km. The performances results, related to other scenarios, are not presented here because they do not enrich the work with qualitative results different from those that can be deduced from table 2.2. Table 2.2 contains the values of statistical parameters previously described. In the first column the models are indicated and the second one contains the evaluated random variables. As first step the *ENK* calculation is considered, in the rows labeled as simulation trace. The reference values are expressed, hence if a model has *ENK* values smaller than reference values, it is possible to summarize that the model represents a good channel behavior approximation.

Table 2.2. Markov chain based models performances results

Model	Random variable	ENK $(s_1 \| \| x_m)$	ENK $(s_2 \| \| x_m)$	Standard error	Mean	Standard deviation
MTA	G	126.3097	129.7719	19.8786	113.8241	145.7262
	B	0.0213	0.0112	0.0081	1.0257	0.1314
Gilbert	G	70.5306	64.7772	7.4150	67.2401	67.7231
Elliot	B	0.0341	0.0212	0.0162	1.0223	0.1457
FSM	G	71.7292	65.1174	6.8663	64.4981	60.2305
	B	0.0445	0.0350	0.0258	1.0209	0.1526
HMM	G	74.8204	66.9982	8.4939	66.9143	66.9844
	B	0.1168	0.0930	0.0328	1.0136	0.1052
Reference values						
Simulation	G	81.1095	67.0041	/	64.9591	67.0042
trace	B	0.0213	0.0204	/	1.0269	0.1599

Considering the MTA model and B random variable, it can be said that the MTA is a good model, because MTA *ENK* values are smaller than reference values and observing the *ENK* columns no one model has the same good results for this statistical parameter. The Gilbert - Elliot model instead

presents *ENK* values that are not smaller than reference ones, they are small but not enough; also FSM values are greater than reference values, so FSM is not a good model for the B random variable. Also the HMM, considering the B random variable, is not a good model.

Observing the G random variable the previous considerations on the MTA are not valid. In fact, *ENK* values demonstrate that the MTA is not a good model inherent to the G random variable because the *ENK* values are excessively greater then reference ones. All the other models have good *ENK* values but no model has a good $ENK(S_2||X_m)$ value and, in particular, the Gilbert-Elliot model has the best *ENK* values.

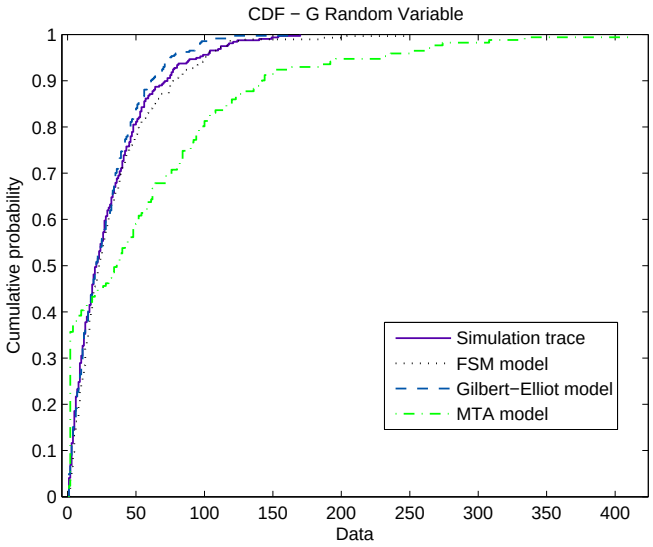

Fig. 2.5. CDFs of the G random variable related to the simulation trace, FSM, the Gilbert-Elliot and MTA model

The standard error column confirms the best results of the MTA for the B random variable, whereas the other models present small errors, but greater than the MTA. Also for the G random variable, the standard error confirms that the MTA is not a good channel behavior approximation. The other models, in this case, are approximately on the same performance level.

Mean and standard error in the sixth and last columns respectively, confirm the previous consideration. It is interesting to note the good Gilbert-Elliot results because this model presents values that are close to reference ones.

In figure 2.5 CDFs [46] functions (Cumulative Density Function) obtained by the simulation traces, FSM, the Gilbert-Elliot and MTA models are depicted. In this figure it is possible to confirm the best FSM and Gilbert-Elliot results and the worst MTA results inherent to the G random variable.

2.6 Hybrid and IWPM models: the our idea to design new generative models

The focus of this section is to illustrate two new channel error models. The first one model, i.e. the Hybrid model, is a Markov chain based model and it is designed observing the performance analysis proposed in the previous section. The target of this model is to generate artificial traces which may present good performances in both random variables. The second model is thought in order to eliminate the dependence of markov chain based models parameters from the scenario configuration.

2.6.1 The Hybrid Model

The basic idea of hybrid trace generation is the following: the MTA obtains good results in the B case, instead the Gilbert - Elliot and HMM in the G case; consequently, if a hybrid model, that follows the MTA to generate corrupted packet bursts and the Gilbert - Elliot or HMM to generate error free packet bursts, is considered, excellent results in all conditions could be achieved. The HMM is more complex than the Gilbert - Elliot, thus the hybrid generation is made by hybrid "MTA - Gilbert-Elliot" generation. In figure 2.6, the flow chart describes the generation process.

The final generated trace is indicated as R-Trace. The first step is the generation of a state ($S_{MTA(i)}$) by MTA model. After this, if the generated state is a "lossy state" the corresponding sub-trace must be generated (using MTA), related to "lossy state", and this must be appended to the R-Trace. In this case generating a "lossy state" by MTA, creates the need to generate a burst of corrupted data packets, and in this case MTA presents a good behavior. Alternatively, if the generated $S_{MTA(i)}$ is an "error free state", it is necessary to generate a burst of error free data packets and thus a sequence of "good states" created by the Gilbert-Elliot model must be appended to the R-Trace. The sequence length is established by the occurrence of a "bad state", i.e. the sequence generation of "good states" continues until there is a "bad state". The single generated Gilbert-Elliot state is indicated in flow chart as $S_{G-E(i)}$; obviously the $S_{G-E(0)}$ is equal to *"0"* because a "good states" sequence must be generated. The flow chart explains clearly the process generation with the various conditions.This hybrid model is used to generate a set of artificial traces that are useful to evaluate the model performances.

Hybrid model performances

In order to test the hybrid model behavior, it was evaluated in a wide series of scenarios; the table 2.3 presents the evaluation results for scenarios set with a packet size value belonging to range 6 - 120 byte and also for a user speed value belonging to the following ranges labeled as:

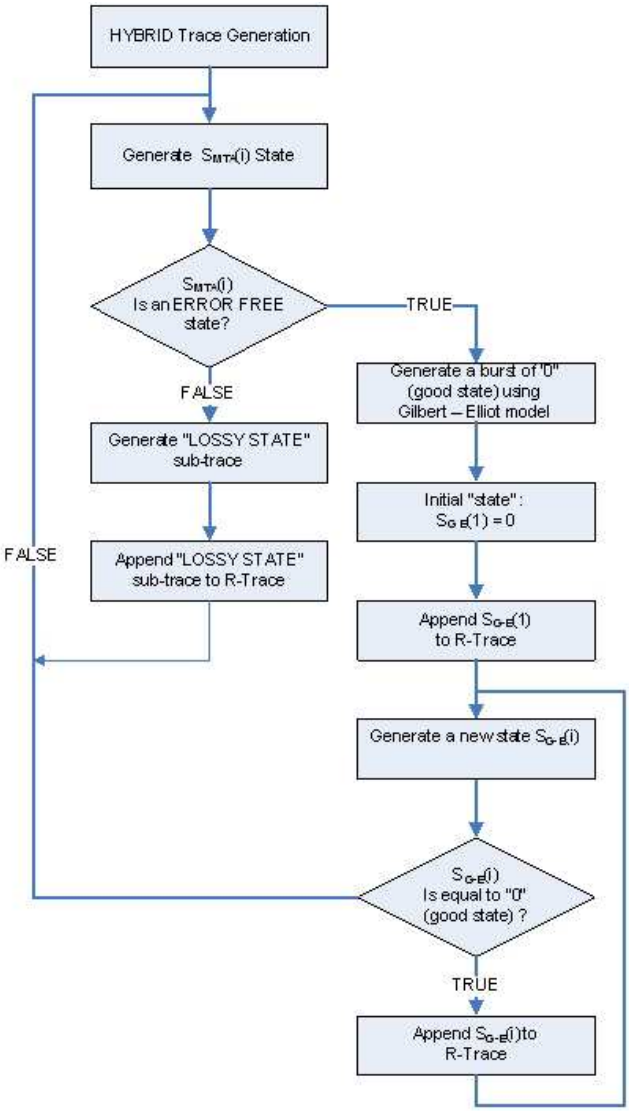

Fig. 2.6. Hybrid model flow chart

Table 2.3. Performances results for packet size values belonging to 6-120 byte range

Packet size interval: 6 - 120 byte						
Model	Random variable	ENK $(s_1 \| \| x_m)$	ENK $(s_2 \| \| x_m)$	Standard error	Mean	Standard deviation
Low speed scenario: 0 - 40 km/h						
MTA	G	79.6346	83.2448	26.7207	366.887	451.9657
	B	0.05075	0.04175	0.00115	1.0311	0.1592
Gilbert	G	66.3623	62.1193	16.7438	199.9496	179.1989
Elliot	B	0.07622	0.07713	0.0062	1.029	0.1444
Hybrid	G	61.4833	55.5892	17.7655	201.1941	193.4311
	B	0.06347	0.04845	0.00121	1.0303	0.1578
Reference values						
Simulation	G	74.7565	56.3255	/	200.4736	201.7237
trace	B	0.079	0.049	/	1.0395	0.1942
Average speed scenario: 40 - 80 km/h						
MTA	G	69.5135	67.7729	18.7035	272.7918	370.5034
	B	0.02235	0.02616	0.00117	1.0484	0.1883
Gilbert	G	48.8973	56.996	16.4158	190.9236	191.0328
Elliot	B	0.03383	0.03684	0.0605	1.0475	0.2254
Hybrid	G	49.1088	53.8098	15.9109	181.8953	176.3525
	B	0.03166	0.03354	0.0292	1.0494	0.1896
Reference values						
Simulation	G	57.4372	57.978	/	181.9615	181.2811
trace	B	0.03225	0.0349	/	1.0511	0.2089
High speed scenario: 80 - 120 km/h						
MTA	G	68.9385	71.1747	28.5214	244.759	344.3488
	B	0.03201	0.02211	0.009	1.0867	0.287
Gilbert	G	46.1792	40.5316	18.8474	150.0103	146.2052
Elliot	B	0.03982	0.03394	0.0121	1.0544	0.2131
Hybrid	G	46.1782	40.2286	17.2287	156.9054	149.8467
	B	0.03531	0.03133	0.01113	1.0782	0.2811
Reference values						
Simulation	G	52.5987	41.023	/	156.3124	148.5627
trace	B	0.0419	0.03619	/	1.0883	0.3328

- Low speed: 0 - 40 km/h;
- Average speed: 40 - 80 km/h;
- High speed: 80 - 120 km/h.

While the table 2.4 presents the evaluation results for the model applied in scenarios with packet size values in range 120 - 216 byte and with the same previously introduced speed ranges. The Hybrid model results are compared with the results of the most promising models previously individuated. In the first and second row of both tables the MTA performances are summarized; in the third and fourth raw instead the Gilbert - Elliot model performances

Table 2.4. Performances results for packet size values belonging to 120-216 byte range

Packet size interval: 120 - 216 byte						
Model	Random variable	ENK $(s_1\|\|x_m)$	ENK $(s_2\|\|x_m)$	Standard error	Mean	Standard deviation
Low speed scenario: 0 - 40 km/h						
MTA	G	66.9343	68.2632	23.8476	281.3978	379.6075
	B	0.0485	0.02743	0.0078	1.1044	0.3496
Gilbert	G	50.8044	42.4422	5.5225	167.7126	164.1605
Elliot	B	0.06241	0.04364	0.0886	1.0893	0.2868
Hybrid	G	54.9501	43.0509	5.3081	172.414	169.8216
	B	0.05733	0.03747	0.0067	1.0967	0.2862
Reference values						
Simulation	G	57.1266	43.2536	/	169.0134	165.9264
trace	B	0.064	0.04279	/	1.122	0.3329
Average speed scenario: 40 - 80 km/h						
MTA	G	64.9954	62.0321	20.1434	289.2966	378.7408
	B	0.0384	0.04132	0.01341	1.0881	0.2753
Gilbert	G	35.942	41.9103	9.0311	161.2668	165.2284
Elliot	B	0.04928	0.04831	0.0588	1.074	0.2602
Hybrid	G	34.976	43.7392	7.976	159.493	161.1681
	B	0.03687	0.03706	0.0469	1.1234	0.3607
Reference values						
Simulation	G	40.9777	49.0768	/	159.7492	164.7831
trace	B	0.0429	0.04391	/	1.1193	0.3688
High speed scenario: 80 - 120 km/h						
MTA	G	62.1538	62.875	13.5368	99.7276	141.5732
	B	0.0309	0.026	0.0157	1.2963	0.5731
Gilbert	G	47.8551	50.4594	4.0582	61.5622	62.2825
Elliot	B	0.0431	0.0428	0.07301	1.2357	0.6304
Hybrid	G	47.2367	49.1528	4.0509	61.8979	61.8372
	B	0.0299	0.02938	0.0276	1.284	0.5628
Reference values						
Simulation	G	50.4082	55.2345	/	62.4111	61.2921
trace	B	0.0317	0.0314	/	1.3024	0.5888

are described and compared with the values collected in the fifth and sixth raw related to the Hybrid model.

The G and the B random variables results are summarized in the two tables which contain also the reference values for each scenario. If we consider the two variables in a separate way, in B case the best performances are reached with the MTA model, instead in the G case it is possible to see how the $ENK(S_1||X_m)$ and $ENK(S_2||X_m)$ for Gilbert - Elliot model are both smaller than the reference values; but neither of the two models can be considered a good model for the channel error behavior in both B and G

cases. This conclusion is derived by *ENK* values analysis. The Hybrid model instead is a good model because it in B case reflects the MTA behavior and in G case it presents the Gilbert - Elliot advantages. These consideration can be conducted observing both the tables in which appear in evident way the usefulness of Hybrid model. The considerations derived by *ENK* comparison can be also confirmed by the other statistical parameters: standard error, mean value and standard deviation confirm the best behavior of MTA model in B case and of Gilbert - Elliot model in G case; the best behavior, considering both the cases B and G is showed by Hybrid model.

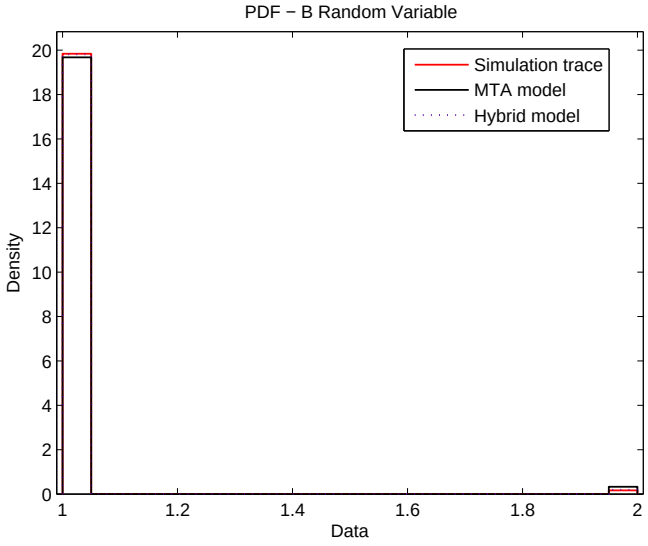

Fig. 2.7. PDFs of the B random variable for the simulation trace, Hybrid and MTA model

In figures 2.7 and 2.8 PDFs (Probability Density Function) [46] of the B random variable and the CDFs of the G random variable, for Hybrid and MTA models and simulation trace are depicted respectively. Both figures proof the best results of the Hybrid model. These figures represent an example of a particular application case.

In figure 2.9 the Packet error correlation functions (PECF) related to simulation trace, Gilbert Elliot, Hybrid and MTA artificial traces are depicted. The PECF is defined as a function *P(K)* and it is the conditional probability that the K_{th} packet following a wrong packet is also in error. The comparison between these functions is another way to proof the best behavior of Hybrid model. In fact the PECF related to Hybrid model match, with PECF of simulation trace, more better than other depicted PECF. These function obviously are calculated taking into account a single well defined system con-

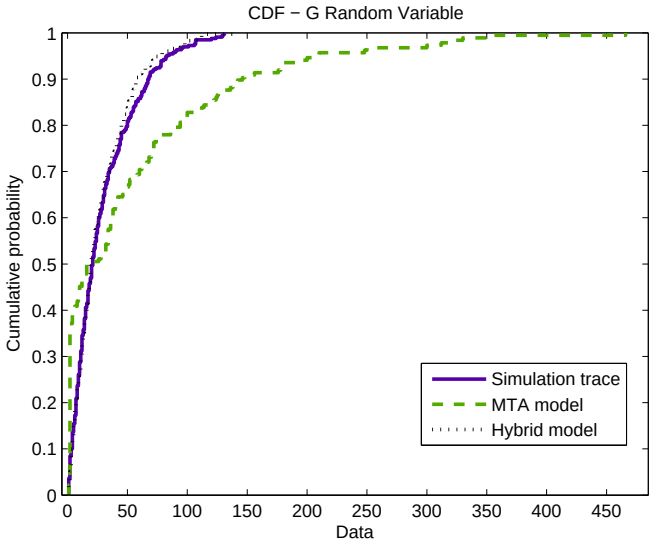

Fig. 2.8. CDFs of the G random variable for the simulation trace, Hybrid and MTA model

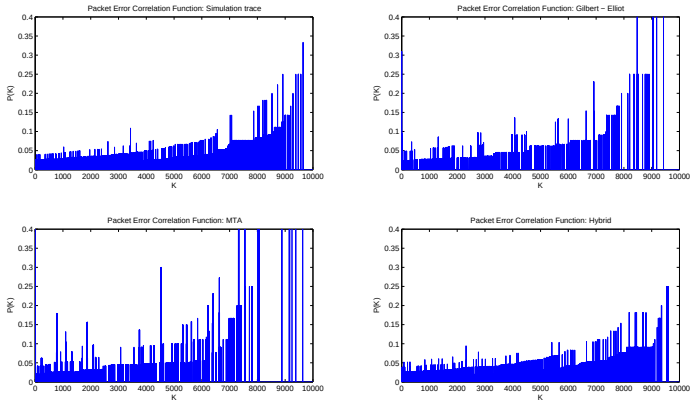

Fig. 2.9. Packet error correlation functions (PECF) for the simulation trace, Gilbert Elliot, Hybrid and MTA artificial traces

figuration (packet size: 120 byte; speed: 20 km/h; distance: 1 km), we could evaluate each configuration, but a single configuration is satisfactory to make our qualitative analysis.

2.6.2 Instant Weighed Probability Model (IWPM)

All the previous models are linked to a particular scenario configuration, i.e the Markov chain based models reflect a well-defined scenario with its particular configuration. When there is a change in the scenario configuration, it is necessary to recalculate the model parameters. Hence there is not a model that is able to say what is, instant by instant, the probability of receiving a wrong or an error free packet, for a given speed value. In this context we are talking about the speed variable but any interesting variable could be considered. IWPM model allows us to disengage from the particular system configuration. In this section the IWPM model is described.

The IEEE 802.16e protocol allows vehicular mobility to be supported, thus a speed value range of 0 - 120 km/h can be considered. This interval can be divided into a series of sub-intervals as shown in figure 2.10. Thus, with a sub-interval size of 20 km/h, 6 equal size sub-intervals are obtained. As clearly shown in figure 2.10, it is possible to assign a specific state of the model to a sub-interval, hence the model is characterized by 6 states. Each state is thus closely related to a specific sub-interval. For example, state *"1"* corresponds to sub-interval 0 - 20 km/h, state *"2"* corresponds to sub-interval 20 - 40 km/h and so on. From simulation traces the probability of generating a bad packet at limit speed values of the various sub-intervals can be evaluated. Each state can be enriched with these two probability values and considering that with $P_B(v_u)$ it is indicated the probability, in a particular instant, to obtain a bad packet with a user speed value equal to "v_u", at state *"2"* the $P_B(20)$ and $P_B(40)$ values must be attributed; instead at state *"3"* the $P_B(40)$ and $P_B(60)$ probability values must be attributed and so on for the other states. These probability values are indicated in a simplified manner as Pi and represents the probability to obtain a bad packet in the scenario configuration: (v_i); in this way P_i correspond to $P_B(v_i)$.

In this way we design a model constituted of 6 states and where each state has a pair of probability values. The transition from one state to another is possible only between two adjacent states; obviously this is realistic because a user speed profile can change only in a continuous way. The transition is controlled by speed value, i.e. a threshold value is associated with each arch of the model, thus if a user is situated in state *"2"* and its speed value grows to 50 km/h, the user transits to state *"3"* at the instant when its speed value exceeds the threshold value associated with the arch (2,3). Instead if its speed value decreases to 10 km/h, the user transits to state *"1"* at the instant when its speed values falls below the threshold value associated with the arch (2,1).

Figure 2.10 can explain the described behavior in a clear way; furthermore in the previous figure the transition thresholds are also depicted. Let us con-

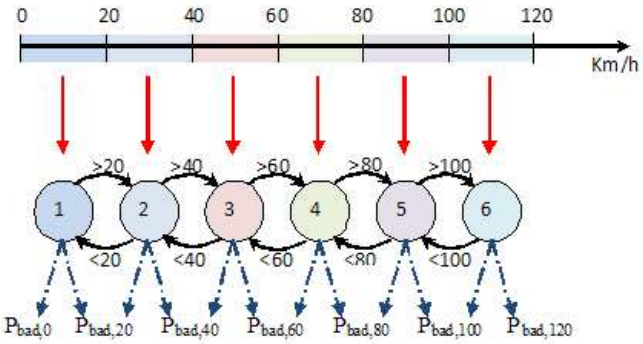

Fig. 2.10. IWPM scheme

sider a user with a speed value, in a specific instant, contained in an interval depicted in figure 2.10. The first step is to individuate the state of the model where the user is situated; the following relation allows the computation of this:

$$i = I(v(t)) = \left\lfloor \frac{v(t)}{20} \right\rfloor + 1 \tag{2.25}$$

The previous relation explains how the state is related to the instant speed value. Once the state is defined, there are two probability values related to it and the probability value of interest must be calculated by these two values. $P_B(t)$ is related to the pair of probability values of state *"i"* expressed by (2.25). The following illustrates the dependences between probability values:

$$P_B(t) = \varphi(P_{i(t)}, P_{i(t)+1}) \tag{2.26}$$

$$P_B(t) = \varphi(P_{i(v(t))}, P_{i(v(t))+1}) \tag{2.27}$$

$P_B(t)$ is related to two time and speed dependent values. Its value obviously belongs to an interval where its bounds are $P_{B,i(t)}$ and $P_{B,i(t)+1}$. If the user speed value is closer to the lower bound of speed sub-interval associated with the state, then $P_B(t)$ is close to $P_{B,i(t)}$; instead, if the symmetric case is verified $P_B(t)$ is close to $P_{B,i(t)+1}$. The idea is to relate $P_B(t)$ to a weighed sum of the other two values, thus this can be expressed as:

$$P_B(t) = w_{v,i} * P_{i(v(t))} + w_{v,i+1} * P_{i(v(t))+1} \tag{2.28}$$

where w_i and w_{i+1} are:

$$\begin{aligned} w_{v,i} &= w_{v,i}(v(t)) = \\ &= f_{v,odd}(v(t)) * mod(i,2) + f_{v,even}(v(t)) * (1 - mod(i,2)) \end{aligned} \tag{2.29}$$

$$w_{v,i+1} = w_{v,i+1}(v(t)) = \\ = f_{v,odd}(v(t)) * mod(i+1,2) + f_{v,even}(v(t)) * (1 - mod(i+1,2)) \tag{2.30}$$

The weights are designed to respect the previously expressed conditions, i.e. for greater clarity the idea that is at the basis of model design will be repeated: *"The $P_B(t)$ belongs to an interval defined by bounds: $P_{i(t)}$ and $P_{i(t)+1}$; this interval is identified by a speed sub-interval to which the user speed belongs. Thus, the probability $P_B(t)$ is close to the $P_{i(t)}$ value if the user speed is close to the lower bound speed value; instead, it is close to the $P_{i(t)+1}$ value in the symmetric case."*

The $f_{v,odd}$ and $f_{v,even}$ are designed to obtain the right weights that allow the previous conditions to be obtained, and are defined as:

$$\begin{aligned} f_{v,odd}(v(t)) &= v_{o0} + v_{o1} * \cos(v(t) * \omega_v) + v_{o2} * \sin(v(t) * \omega_v) \\ v_{o0} &= 0.5 \\ v_{o1} &= 0.5 \\ v_{o2} &= -3.079e - 16 \\ \omega_v &= 0.1571 \end{aligned} \tag{2.31}$$

$$\begin{aligned} f_{v,even}(v(t)) &= v_{e0} + v_{e1} * \cos(v(t) * \omega_v) + v_{e2} * \sin(v(t) * \omega_v) \\ v_{e0} &= 0.5 \\ v_{e1} &= -0.5 \\ v_{e2} &= 2.937e - 16 \\ \omega_v &= 0.1571 \end{aligned} \tag{2.32}$$

the v_{o2} and v_{e2} terms can be neglected. The function $f_{v,odd}$ and $f_{v,even}$ are depicted in figure 2.11.

In this way, for example, considering a user speed value v_0 equal to 38 km/h, *"i"* which proves to be equal to *2* can first be calculated and thus it can be said that the user is in state *2*. Then, after various simplifications the following formula is obtained:

$$P_B(t) = f_{v,even}(v_0) * P_2 + f_{v,odd}(v_0) * P_3 \tag{2.33}$$

As results a probability value is obtained which is closer to the probability value at a speed value of 40 km/h (i.e. the P_3 is heavier than P_2 and this is due to $f_{v,even}$ and $f_{v,odd}$ values).

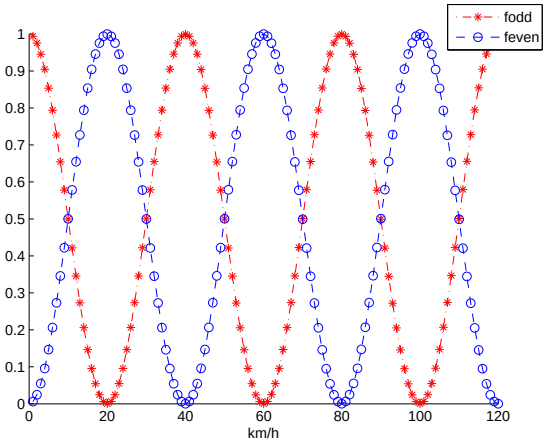

Fig. 2.11. $f_{v,odd}$ and $f_{v,even}$ behavior

IWPM simulation results

The first experiment in which the IWPM model is tested considers the mobile user with a variable speed. The user is initially stopped, and then he increases his speed with a constant acceleration until the user reaches the speed value of 55 km/h; once this value is reached he decreases his speed with a constant deceleration until stopping. The simulation scenario and the PHY settings are described in section 2.5.2 and in table 2.1 respectively.

The speed trend is depicted in figure 2.12 and it is represented by a triangle; the slopes of the speed lines are identical and they are the user acceleration and deceleration respectively.

To simulate this scenario and to apply IWPM to it, the user speed is sampled. The speed sampling is made with a Δt period of 10s; the basic idea is that the scenario is simulated in sampled points in a steady state condition: i.e. the user has an instant speed value equal to *"v"* and he remains at this speed for a time interval that allows him to reach a stable PER value. The choice of sampling period reflects this condition and the value of 10s is calculated as the necessary time to send approximately 10000 packets of 192 bytes at a rate equal to 1.5 Mbps. Thus, the real speed characteristic is depicted in figure 2.13 as a series of steps.

The acceleration and deceleration are chosen to have the rounded speed values: 5, 10, 15, ... 45, 50 and 55 km/h; its absolute value is $1.355 * 10^{-4} km/s^2$.

In figure 2.14 two trends are depicted: the red line represents the probability to obtain a bad packet, on the receiver side, as measured by simulation; instead the blue line is the same parameter as evaluated by the IWPM model. The trends confirm the good results of the proposed model; this fact is also visible in figure 2.15, where the percentage relative error, that is present if

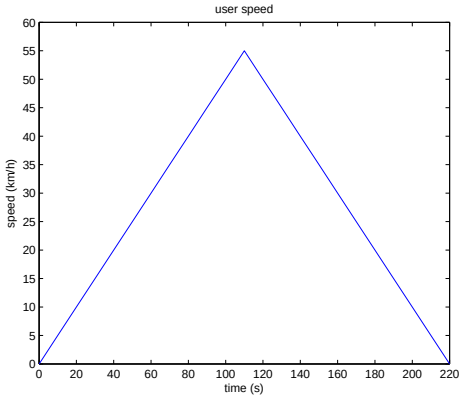

Fig. 2.12. User speed characteristic

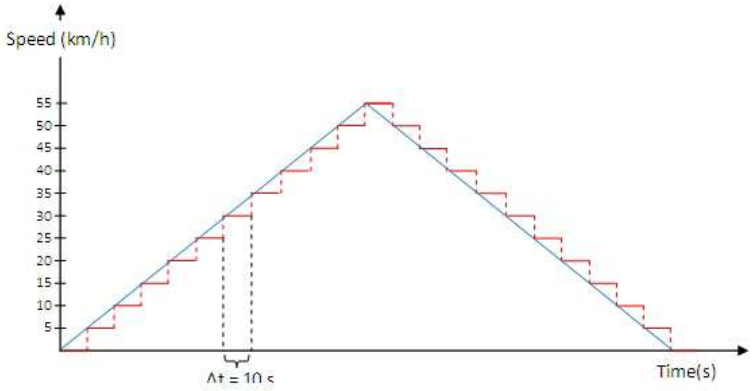

Fig. 2.13. Speed sampling

the IWPM is used to calculate the bad packet probability, is depicted. The maximum error is less than 3%.

The second experiment, used to validate the IWPM, removes the limitations of the first experiment. In this one, the idea of user speed steady state is removed. With the user speed steady state concept, in the first experiment, the user maintains the same speed value until the transmission reaches the steady state; the steady state allows estimation of the simulation bad packet probability value with a mean value to compare with IWPM expected value.

In this case a user is considered that has the initial speed value equal to 5 km/h and he increases the speed with a constant acceleration of $0.0014 km/s^2$; this acceleration is a realistic value and allows to a generic user, starting from boarding, to reach the speed of 100 km/h in 20s. In the experiment the speed interval 5 - 20 km/h is considered and this does not represent a limitation for the experiment. The user speed is visible in figure 2.16. As can be seen

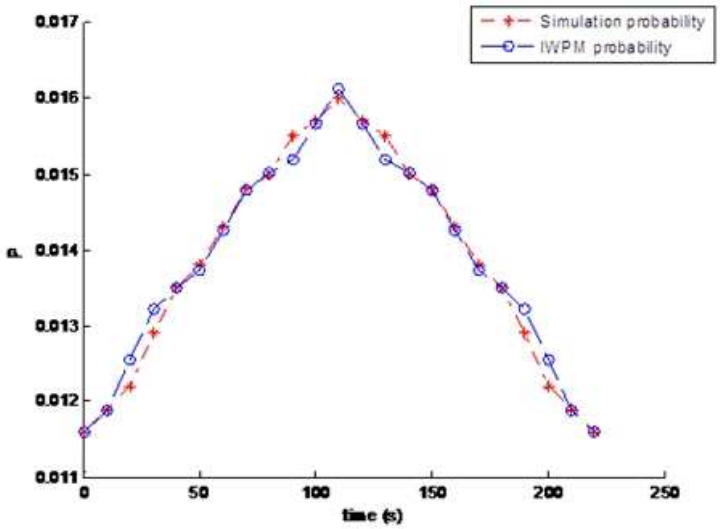

Fig. 2.14. Probability of having a bad packet obtained by simulation and IWPM model

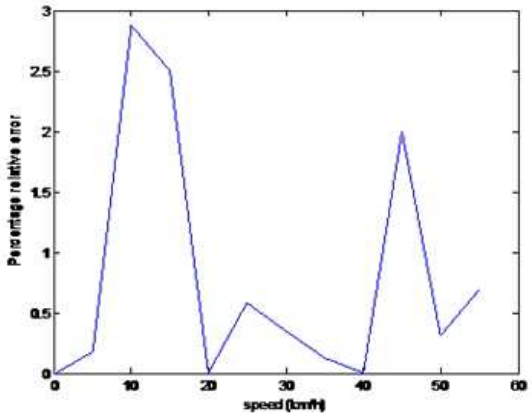

Fig. 2.15. Percentage relative error between simulation values and IWPM values vs user speed

in figure 2.16, the speed is sampled at an interval of 3 km/h and the user is considered as maintaining the sampled speed for a time interval necessary to pass to a subsequent sampled speed. For greater clarity : the user starts at 5 km/h and it is supposed that he maintains this speed value for Δts, after this time his speed is 8 km/h. This Δt equal to 0.5952 s is the necessary time to reach 8 km/h, starting from 5 km/h, with the acceleration value of $0.0014 km/s^2$.

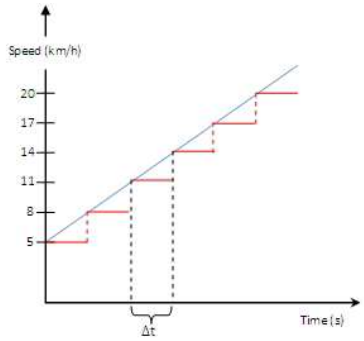

Fig. 2.16. User speed characteristic and speed sampling

This supposition is not a limitation for the experiment, because during this interval the transmission does not reach a steady state and thus the system is dynamic. During Δt interval, various simulation run results, for each speed value, are collected. On the basis of previous suppositions and settings, the following user speeds are simulated: {5, 8, 11, 14, 17}. For each speed value a set of bad packet probability values are obtained. In addition to the values obtained by simulation there are the values predicted by IWPM; this model extracts a bad packet probability for each speed value. The final step of the experiment is to verify that the IWPM predicted values are acceptable. There is this situation: there is a random variable, i.e. the probability of receiving a bad packet, characterized by a normal distribution obtained by simulation, and also there is a predicted value, by IWPM, for the bad packet probability; to verify whether this value is acceptable, a confidence interval can be designed for probability distribution and if the IWPM predicted value falls in the confidence interval then it means that the prediction is good. Obviously this must be repeated for each analyzed speed value.

The horizontal lines depicted in figure 2.17 are the 95% confidence intervals calculated for all the speed values, while the vertical lines represent the predicted IWPM values. On the vertical axis it is possible to note the speed values: 5, 8, 11, 14 and 17 km/h.

For example observing the figure, it can be seen how the predicted value related to 17km/h, the vertical yellow line ending with a circle, intersects the

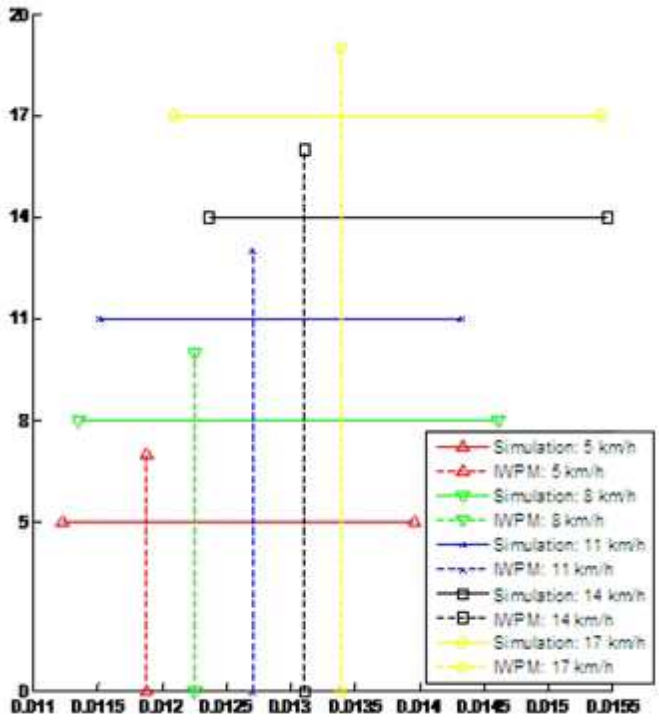

Fig. 2.17. Confidence intervals and IWPM predicted values

confidence interval related to the same speed value, which is depicted with a yellow horizontal line ending with a circle. In general all the predicted values respect the membership condition.

Both the experiments have good results and prove that IWPM is able to model a time variant channel error behavior in a faithful way. To the best of our knowledge, in the literature, there is no similar model either in the scenario with user transmission in steady state or in a realistic dynamic scenario where the channel behavior rapidly changes conditions which have been successfully tested.

Sub-interval Size Evaluation

In the IWPM section, when the model was described, a particular concept was voluntarily neglected. Why did we choose a sub-interval size equal to 20 km/h without justifying this value? This omission was voluntary because in a previous section attention was focused on the model description. Now, to motivate

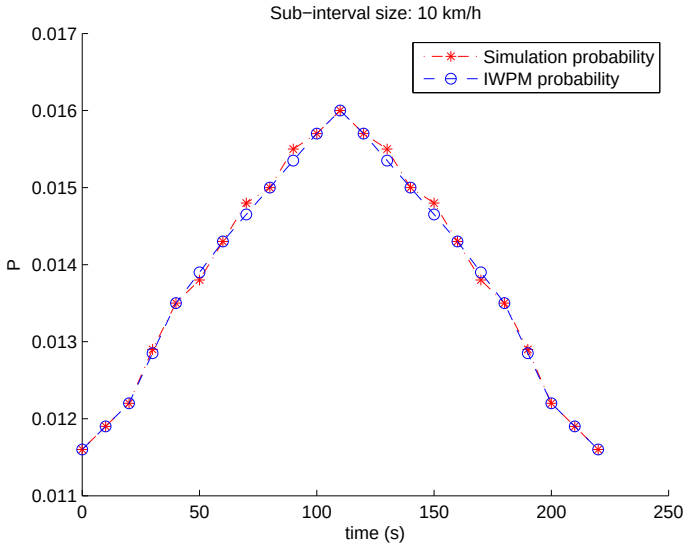

Fig. 2.18. Probability of having a bad packet obtained by simulation and IWPM model with 10 km/h sub-interval size

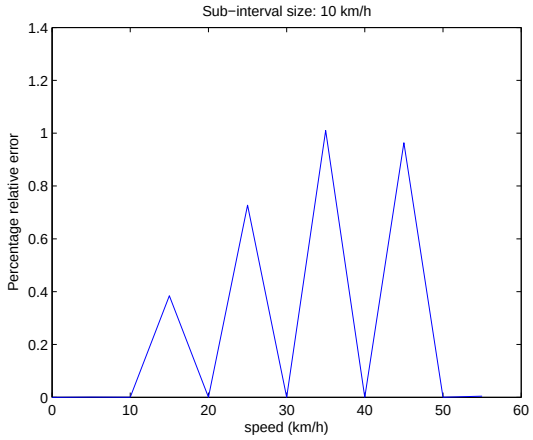

Fig. 2.19. Percentage relative error between simulation values and IWPM values vs user speed with 10 km/h sub-interval size

the size choice, other simulation results are illustrated. The first experiment, with 10 and 40 km/h sub-interval size values, is repeated and figures 2.18 and 2.19 illustrate the simulation results compared with the IWPM results and percentage relative error respectively; these figures are related to the 10 km/h

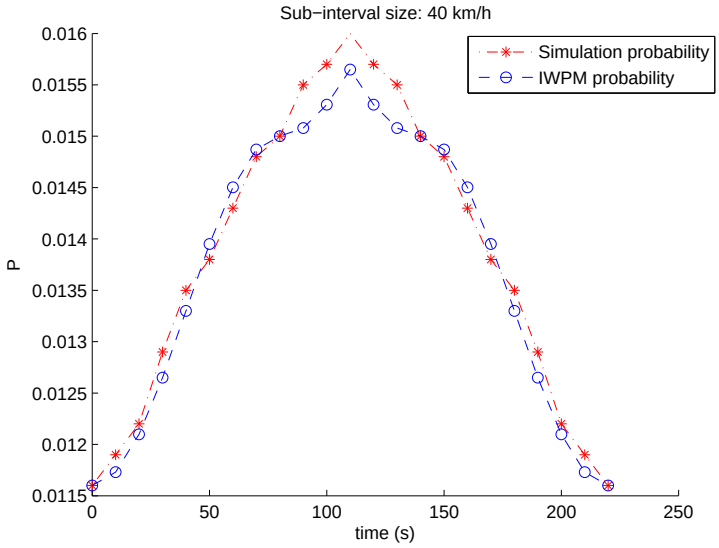

Fig. 2.20. Probability of having a bad packet obtained by simulation and IWPM model with 40 km/h sub-interval size

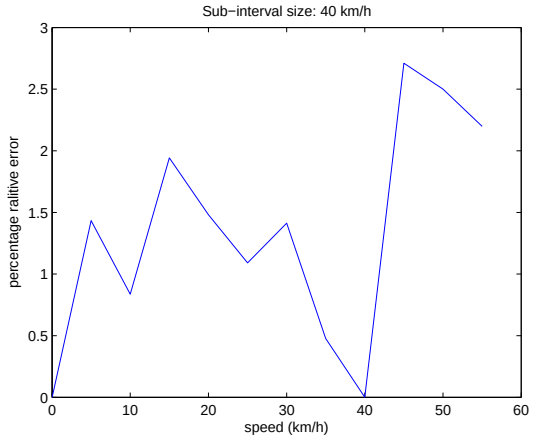

Fig. 2.21. Percentage relative error between simulation values and IWPM values vs user speed with 40 km/h sub-interval size

sub-interval size, instead figures 2.20 and 2.21 depict the same parameters for the 40 km/h case.

In figure 2.18 it is possible to note how the IWPM results strictly match the simulation results. The percentage relative error behavior is also improved (see figure 2.19), hence if the model is set with a 10km/h sub-interval size, with

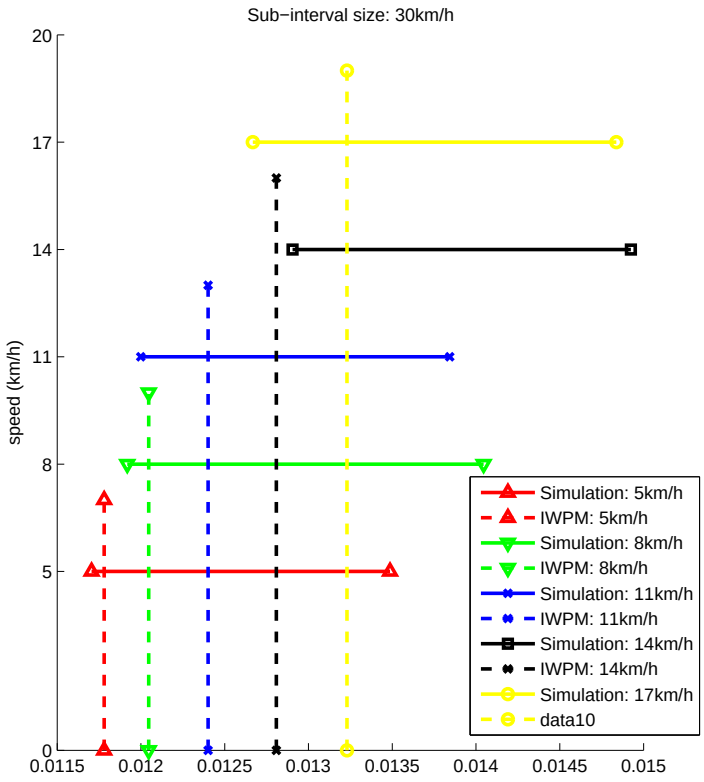

Fig. 2.22. Confidence intervals and IWPM predicted values with 30km/h sub-interval size

decreasing the size, the IPWM behavior is improved. A negative aspect is the increased number of states of the IWPM; in fact, in this case it is characterized by 12 states. In figure 2.20 the depicted IWPM behavior is good but there is a worsening compared to 10 and 20 km/h sub-interval sizes cases. Figure 2.21 also confirms this trend, in fact, the maximum percentage relative error is less than 3% but its general trend is worsened. Repeated experiments illustrate how there is a worsening in the IWPM performance when the sub-interval size is increased; however, the model can still be used.

To verify further the effects of increasing the sub-interval size the second experiment is repeated with 30 and 40 km/h size values. The results of the experiments are illustrated in figures 2.22 and 2.23, where the 95% convergence intervals and the provisioned IWPM values are represented. The results obtained in the 30 km/h case, and illustrated in figure 2.22, are not accept-

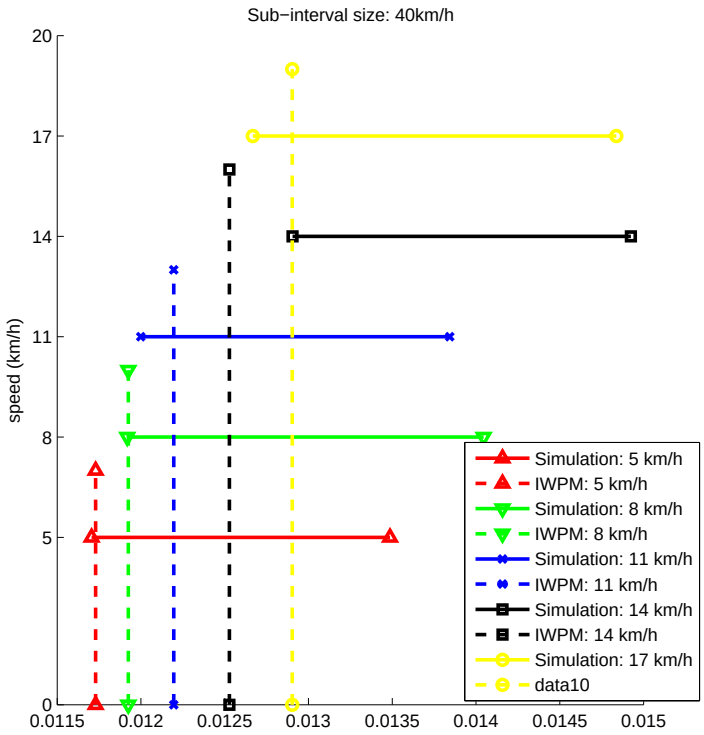

Fig. 2.23. Confidence intervals and IWPM predicted values with 40km/h sub-interval size

able; in fact, the IWPM predicted value, for a user speed equal to 14 km/h, falls outside the confidence interval; also the predicted values related to 5 and 8 km/h user speed are not good because the values are too close to the confidence interval lower boundary. This negative IWPM behavior is even more evident in figure 2.23.

Thus in conclusion IPWM performances are related to the sub-interval size choice. The dynamic scenario without steady state (second experiment) is a realistic test-bed for size choice, and the size selection must be made with a trade-off between the model states number and performance. Not all values can be used to set the sub-interval size and a good choice, as proven by results, is thus the 20 km/h value.

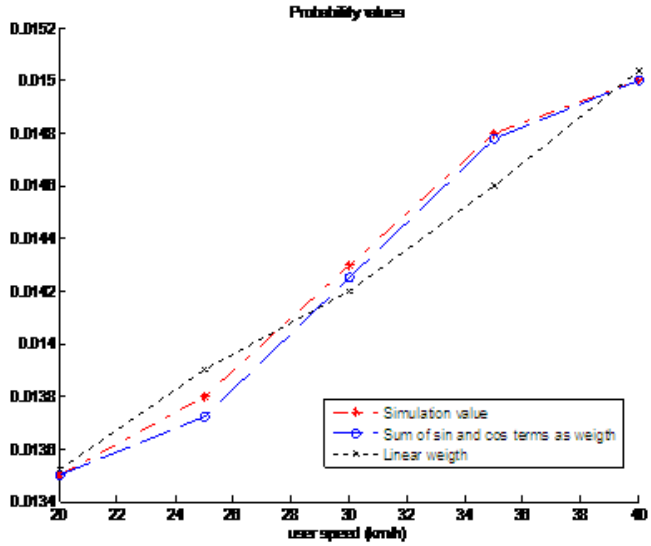

Fig. 2.24. probability values obtained by simulation, by IWPM model with sin() and cos() weight function and by linear function

Weigh function motivation

Another neglected concept is referred to the particular function chosen to built weight functions. We represent this function as sum of *sin()* and *cos()* terms because it allows us to obtain the best model performance. As an example, in figure 2.24 the probability values obtained by simulation (red dashed line), and the probability values obtained from the model IWPM for two different weight functions are represented. One weight function is that obtained with the relations (2.31) and (2.32), and the second is a linear weight function. In the first case the maximum percentage relative error is equal to 2.8%, instead with linear weight function we obtain a value equal to 6.71%. The best behaviour is obtained with (2.31) and (2.32) functions (blue dashed line).

IWPM and Markov chain based model performance comparison

In the figure 2.25 the results comparison between IWPM, Hybrid and Gilbert-Elliot models are illustrated. To make the comparison a set of four speed values are considered: $\{25, 50, 75, 100\}km/h$; as function of these values the probabilities to receive a bad packet are depicted. The best behavior is obtained by Hybrid model, but also IWPM presents a good result. The best behavior obtained with Hybrid model is confirmed by percentage relative errors depicted in figure 2.26. Apparently the presence of IWPM is not useful but we

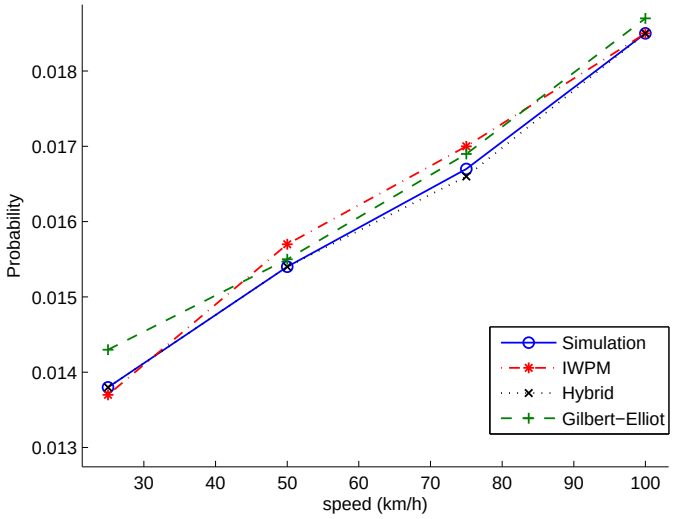

Fig. 2.25. Probability to have a bad packet

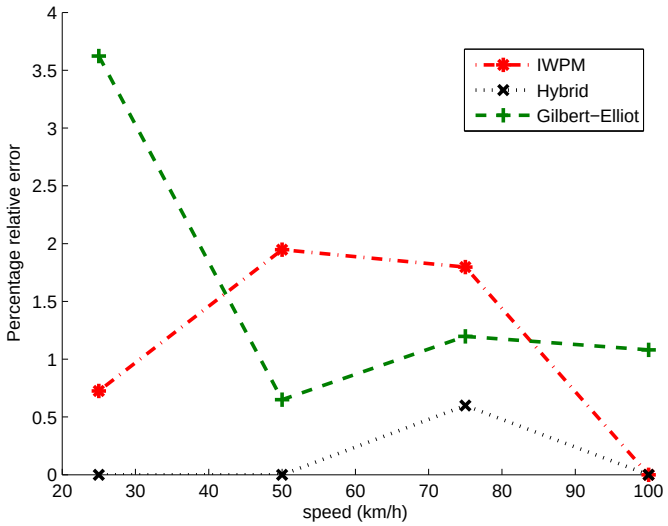

Fig. 2.26. Percentage relative errors

want to focus about the real advantage introduced by IWPM model. IWPM allows to calculate the probability to receive a bad packet applying the equation (2.28); in turn, to use the (2.28) it is need to calculate its coefficients

using as input the simulation traces. Obviously these coefficients have to be evaluated only one time and then we have a function useful for each speed value. To evaluate the probabilities by Gilbert-Elliot model we have to configure the model calculating the transition probabilities matrix, but this become a repetitive process because the elements of transition probabilities matrix must be recalculated every time there is a change in scenario configurations and i.e. we must recalculate the matrix for each speed value in the selected set. In this way, to evaluate the probability for the defined speed set, four different matrixes and consequently, four different Gilbert-Elliot models must be evaluated. The same consideration can be made for Hybrid model. Obviously this situation is not practical. The advantage of IWPM can be expressed in term of computational complexity.

In table 2.5, the computational complexity for each model as function of speed set size are summarized.

Table 2.5. Computational complexity

Model	Complexity
IWPM	$\Theta(a) = \Theta(1)$
Gilbert-Elliot	$\Theta(b * n) = \Theta(n)$
Hybrid	$\Theta((b + c) * n) = \Theta(n)$

The computational complexity of IWPM is not related to the set size *"n"*, there are only a number of *"a"* coefficients which must be computed only one time. With Gilbert-Elliot model there is the need to evaluate *"b"* coefficients for each speed value; instead Hybrid model considers a number of coefficients equal to sum of *"b"* and *"c"*, where *"c"* is the number of MTA coefficients.

Improving the IWPM: IWPM-2V

IWPM model is able to say what is, instant by instant, the probability of generating a wrong packet or not, for a given speed, in this section our intent is to present an improvement of IWPM. IWPM is designed to disengage the channel model from scenario configuration and this becomes increasingly true with the add of another variable. The improved IWPM is defined as IWPM-2V (IWPM- 2 Variables) and in this context we are talking about two variables: speed and packet size, but any interesting variable, as scenario parameter, could be considered.

To design the model, we consider the speed values range of $0-120km/h$ and divide this interval into a series of sub-intervals with a sub-interval size of 20 km/h, 6 equal size sub-intervals are obtained as in IWPM. Instead, considering the packet size range, we choose arbitrarily the range $6 - 216$ byte that is contained in range $6 - 255$ byte indicated in IEEE 802.16 PHY. As in speed value case also in this one it is possible to divide the range into 6 sub-intervals

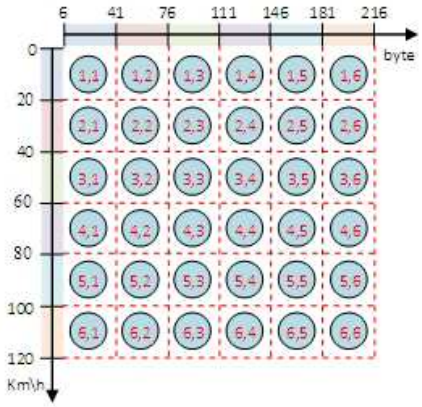

Fig. 2.27. IWPM scheme

using a 35 byte sub-interval size. At each instant, a user "configuration", is characterized by a pair of values: *(speed, packet size)*, thus it can be associated with a pair of sub-intervals: speed and packet size sub-interval (where the two values fall in). At each sub-interval pair can be associated a model state, thus the total model states are all the possible pairs combinations that can be created. In simple words, if we have m sub-intervals related to speed and n sub-intervals related to packet size, the total state number is $m * n$. Each state is thus closely related to a specific user "configuration". For example, state *(1,1)* corresponds to speed sub-interval 0 - 20 km/h and packet size sub-interval 6 - 41 byte and so on. For each state, 4 probability values are associated, and these are the probability to obtain a "bad" packet at the related sub-interval bounds values, i.e. for example, the *(1,5)* state is equipped with the probabilities to obtain a corrupted packet at the following pair *(speed, packet size)* configurations: *(0,146); (0,181); (20,146); (20,181)*. We can indicate these values as *P(v,s)*, and thus the values related to the state *"5"* can be indicated as *P(0,146), P(0,181), P(20,146)* and *P(20,181)*. All the probability values can be evaluated by simulation traces. Considering the dashed matrix, containing the model states, depicted in figure 2.27, the transition from one state to another is possible only following these rules:

- the final state must belong to a raw adjacent to start state raw and this because the user speed profile can vary only in a continuous way;
- the final state may belong to any matrix column and this because the choice of the new packet size value is not related to the previous one used to transmit data;
- the transition is controlled by speed and packet size values.

In this way, if a user is situated in state *(1,2)* and its speed value grows to 30 km/h, the user transits in state *(2,2)* at the instant when its speed value exceeds the upper sub-interval threshold value of 20 km/h. Following the

previous rules, for example, the transitions from state *(1,3)* to state *(2,3)* or to *(2,4)* are possible, but transitions from state *(1,3)* to state *(3,5)* or to *(3,3)* are not possible. In figure 2.27, to avoid to create a confusing figure, the transition arches are not depicted. Now consider the equations related to the model and their use to calculate the mobile user instant probability to lose a packet, this value is indicated with $P_B(t)$. Consider a user that has a speed and a packet size value, in a specific instant (v_{user} and s_{user}), contained in sub-intervals depicted in figure 2.27 and specifically each configuration parameter must belong to a particular sub-interval. The first step to identify the model is to create the matrix *P*. *P* is a matrix of $(n+1)(m+1)$ elements, where $P_{i,j}$ is the generic element, and it represents the probability value to obtain a bad packet with v_i and s_j packet size and speed value respectively ($P_{i,j} = P(v_i, s_j)$), with $v_i \in \{0, 20, 40, 60, 80, 100, 120\}$ and $s_j \in \{6, 41, 76, 111, 146, 181, 216\}$.

In practice the matrix P contains the probability values related to the states; in particular the state *(i,j)* has the following associated values: $\{P_{i,j}$; $P_{i,j+1}$; $P_{i+1,j}$; $P_{i+1,j+1}\}$. Then it is necessary to individuate the state of the model where the user is situated, the following relation with the relation defined in (2.25) allows computation of index i and j:

$$j = J(s(t)) = \left\lfloor \frac{s(t) - 6}{35} \right\rfloor + 1 \tag{2.34}$$

The previous relation explains how the state is related to instant speed and packet size value (*s(t)* indicates the packet size value at time instant t). Once the state is defined, there are 4 probability values related to sub-interval bounds that individuate it, and from these the probability value of interest can be calculated. The selected state is the *(i,j)* state and it can be individuated in dashed matrix of figure 2.27. $P_B(t)$ is related to the four probability values related to the four sub-interval bounds (two bounds for speed value and two for packet size):

$$P_B(t) = \varphi(P_{i,j}, P_{i,j+1}, P_{i+1,j}, P_{i+1,j+1}) \tag{2.35}$$

Its value belongs to an area, depicted in figure 2.28, delimitated by $\{P_{i,j}$; $P_{i,j+1}$; $P_{i+1,j}$; $P_{i+1,j+1}\}$ vertices. The area is delimitated by the four sub-interval bounds, that correspond to *(i,j)* state. The combination of these bounds individuate 4 points and consequently the area that contains the user configuration (s_{user}, v_{user}). In fact, if the user speed value is closer to the lower bound of speed sub-interval (v_i, v_{i+1}) and the packet size is near to lower bound of packet size sub-interval (s_j, s_{j+1}) associated with the *(i,j)* state, then $P_B(t)$ is near to $P_{i,j}$, instead if the speed is near to upper bound of speed sub-interval and the used packet size is near to lower bound of packet size sub-interval associated with the state then $P_B(t)$ is near to $P_{i+1,j}$ value. The idea is to relate $P_B(t)$ to a weighed sum of the other four values. To do this we consider two different steps: in the first one we evaluate two weighed sums to obtain P_C and P_D points depicted in figure 2.28.

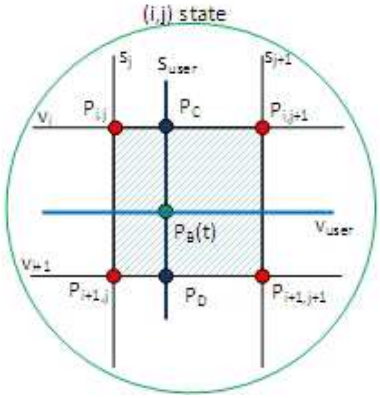

Fig. 2.28. Area individuated by the four sub-interval bounds

P_C represents the weighed sum of the probabilities value related to the upper and lower bounds of sub-interval packet size values and also considering fixed and equal to v_i (speed value lower bound associated to the state *(i,j)*) the value of packet size, to more clearness P_C is the probability to obtain a bad packet with a fixed value of speed equal to vi and with a packet size value equal to s_j; P_C can be computed by:

$$P_C = w_{s,j} * P_{i,j} + w_{s,j+1} * P_{i,j+1} \tag{2.36}$$

in similar way P_D can be obtained considering fixed and equal to v_{i+1} (speed value upper bound associated to the state (i,j)) the value of user speed and it represents the probability to obtain a bad packet with a packet size value equal to s_j and a speed value equal to sub-interval upper bound. P_D can be computed by:

$$P_D = w_{s,j} * P_{i+1,j} + w_{s,j+1} * P_{i+1,j+1} \tag{2.37}$$

The weights are designed to respect the previously expressed conditions:

$$\begin{aligned} w_{s,j} &= w_{s,j}(s(t)) = \\ &= f_{s,odd}(s(t)) * mod(j,2) + f_{s,even}(s(t)) * (1 - mod(j,2)) \end{aligned} \tag{2.38}$$

$$\begin{aligned} w_{s,j+1} &= w_{s,j+1}(s(t)) = \\ &= f_{s,odd}(s(t)) * mod(j+1,2) + f_{s,even}(s(t)) * (1 - mod(j+1,2)) \end{aligned} \tag{2.39}$$

where $f_{s,odd}$ and $f_{s,even}$ and other parameters are defined by the following:

$$\begin{aligned} f_{s,odd}(s(t)) &= s_{o0} + s_{o1} * \cos(s(t) * \omega_s) + s_{o2} * \sin(s(t) * \omega_s) \\ s_{o0} &= 0.5 \\ s_{o1} &= 0.4292 \\ s_{o2} &= 0.2564 \\ \omega_s &= 0.08976 \end{aligned} \tag{2.40}$$

$$\begin{aligned} f_{s,even}(s(t)) &= s_{e0} + s_{e1} * \cos(s(t) * \omega_s) + s_{e2} * \sin(s(t) * \omega_s) \\ s_{e0} &= 0.5 \\ s_{e1} &= -0.4292 \\ s_{e2} &= -0.2564 \\ \omega_s &= 0.08976 \end{aligned} \tag{2.41}$$

With this first step the segment $P_C - P_D$ is obtained, now in the second step $P_B(t)$ can be obtained as a weighed sum of P_C and P_D probability values because now we want the probability value at user speed value that is contained in sub-interval (v_i, v_{i+1}) and thus it is on segment $P_C - P_D$. $P_B(t)$ is thus defined as:

$$P_B(t) = w_{v,i} * P_C + w_{v,i+1} * P_D \tag{2.42}$$

and where $w_{v,i}$ and $w_{v,i+1}$ are defined in IWPM model by equations (2.29) and (2.30). In this way, for example, if the user speed is 55 km/h and the packet size is 192 byte, *"i"* and *"j"* can first be calculated, which result equal to *3* and *6* respectively, and thus it can be said that:

- the user is in state *(3,6)* because the speed bounds are 40 and 60 km/h and the packet size bounds are 181 and 216 byte;
- the 4 probability values associated with the previous bounds are: $P_{3,6}$ is associated with configuration ($v_i = 40$, $s_j = 181$); $P_{4,6}$ is associated with configuration ($v_{i+1} = 60$, $s_i = 181$); $P_{3,7}$ is associated with configuration ($v_i = 40$, $s_{j+1} = 216$) and $P_{4,7}$ is associated with configuration ($v_{i+1} = 60$, $s_{j+1} = 216$); P_C can be calculated as: $P_C = P_{3,6} * f_{s,odd}(192) + P_{4,6} * f_{s,even}(192)$; P_D can be calculated as: $P_D = P_{3,7} * f_{v,odd}(192) + P_{4,7} * f_{v,even}(192)$;
- finally:

$$P_B(t) = w_{v,i}(55) * P_C + w_{v,i+1}(55) * P_D \tag{2.43}$$

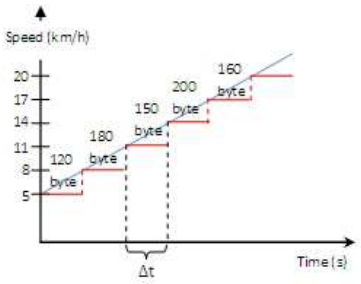

Fig. 2.29. User speed characteristic and speed sampling

To validate the presented model, we present the following experiment in which a user has the initial speed value equal to 5 km/h and he increases the speed with a constant acceleration of $0.0014 km/s^2$.

In the experiment the speed interval $5 - 20 km/h$ is considered and this does not represent a limitation for the experiment. The user speed is visible in figure 2.29. This experiment is the same experiment described in section 2.6.2, in which there is not the possibility to reach a steady state condition. The novelty is: at each speed value, as can see in figure 2.29, the user changes the packet size. During Δt time interval various simulation run results, for each speed value and relative packet size, are collected. On the basis of previous supposition and settings thus the following speed values and packet size user configuration are simulated: {(5km/h, 120byte); (8km/h, 180byte); (11km/h, 150byte); (14km/h , 200byte); (17km/h ,160byte)}. For each user configuration a set of bad packet probability values are obtained and the final experiment step is to verify that the IWPM-2V predicted values are acceptable; there is this situation: there is a random variable bad packet probability, this variable is characterized by a normal distribution obtained by simulations, and also there is a predicted value, by IWPM-2V, for the bad packet probability; to verify whether this value is acceptable a confidence interval can be designed for probability distributions and if the IWPM-2V predicted value falls in the confidence interval then it means that the prediction is good.

This obviously must be repeated for each analyzed configuration. The horizontal lines depicted in figure 2.30 are the 95% confidence intervals calculated for all the user configuration, instead the vertical lines represent the predicted IWPM-2V values. In general all the predicted values respect the membership condition. This experiment presents good results and prove that IWPM-2V is able to model a time variant channel behaviour in a faithful way. As we did in section 2.6.2, also in this case we verify what happens varying the subinterval size for both the variables.

To verify the sub-interval increasing size effects, now the experiment is repeated with 40 km/h speed sub-interval size and 70byte packet sub-interval size. With these two values the model states number became 9. The results of this experiment are illustrated in figures 2.31 where the 95% convergence

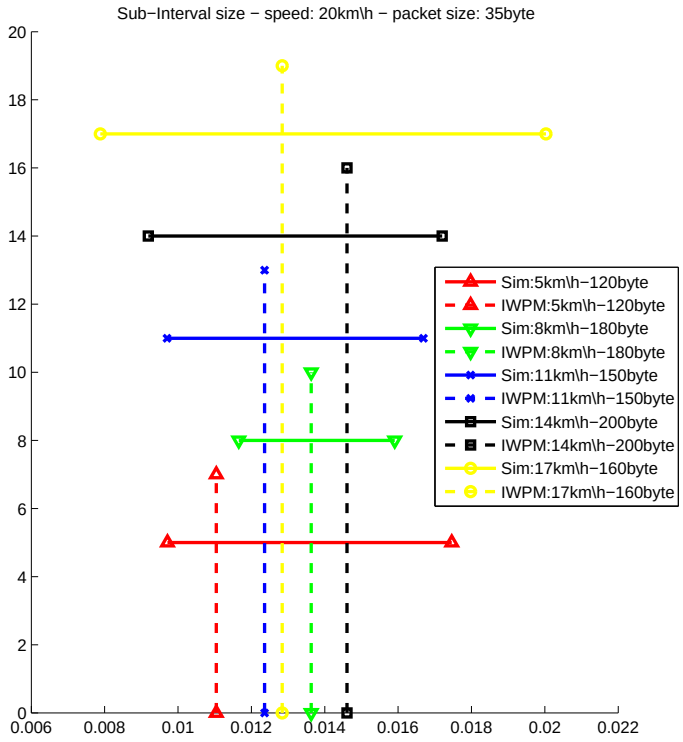

Fig. 2.30. Confidences intervals and IWPM-2V predicted values

intervals and the provisioned IWPM-2V values are represented. The results obtained are not acceptable, in fact, the IWPM-2V predicted value, for a user speed equal to 5 km/h and packet size equal to 120byte, falls outside the confidence interval, also the predicted value related to 8 km/h user speed is not good because the value is too close to the confidence interval lower boundary. Instead, obviously, decreasing sub-interval size the model behavior improves its performances but the model states number tends to grow. Thus in conclusion IWPM-2V performances are related to the sub-interval size choice. The dynamic scenario without steady state, is a realistic test-bed for size choice and also in this case, as in IWPM case, the size selection must be effectuated by a trade-off between the model state number and performance. Not all values can be used to set the sub-interval size. For example, a good choice, as proofed by results, are thus the 20 km/h and 35 byte values.

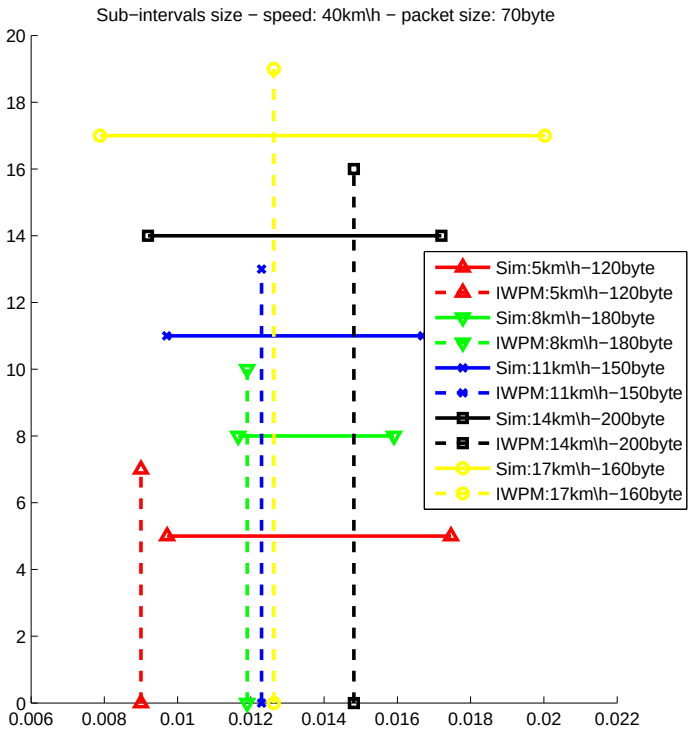

Fig. 2.31. Confidence intervals and IWPM-2V predicted values with 40km/h and 70 byte sub-intervals size

A further improvement: IWPM-3V

The IWPM in the previous section is improved as IWPM-2V with the addition of the second variable, in this way IWPM-2V is able to predict the packet state as function of scenario configuration, where the scenario configuration can be expressed by the value of two variable. This model can be further improved adding also the third variable: the transmitter - receiver distance. In this section we describe the introduction of this new variable, illustrating graphically and analytically how to work with the Instant Weighed Probability Model - 3 Variables (IWPM-3V) model. It is important to note as IWPM-3V became thus an important generative model because it is able to consider a wide variety of scenario configuration. To add the third variable the following interval for the transmitter - receiver distance is considered: $[1000 - 7000]m$ and this interval is divided into 3 subinterval with size equal to 2000 m.

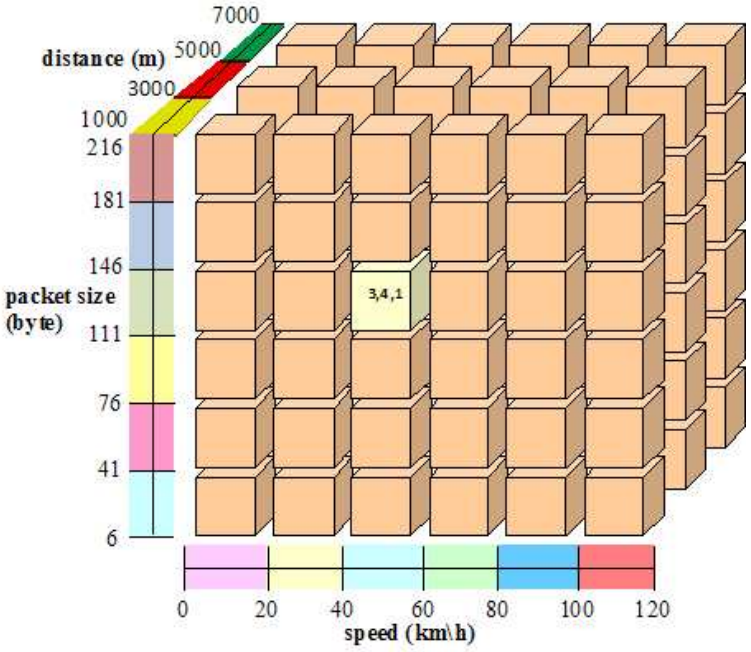

Fig. 2.32. IWPM-3V model

The new IWPM-3V is depicted in figure 2.32. Each "cube", depicted in the figure, represents a state of the model and instant by instant, the user state corresponds to one state of the model. For example, a scenario configuration falls in the "cube" labeled in the figure 2.32 as *(3,4,1)* if the following three conditions are verified:

- the user speed value falls in the interval: $[40, 60]$ km/h;
- the packet size falls in the interval: $[76 - 111]$ byte;
- the transmitter - receiver distance falls in the interval: [1000 - 3000] m.

$P_B(v_i(t0), s_j(t0), d_k(t0))$ indicates the probability to obtain a corrupted packet at t_0 instant in a scenario with a speed equal to v_i, a transmitted packet size equal to s_j and a transmitter - receiver distance equal to d_k. The notation can be simplified considering as obvious the time dependence of the three variable, obtaining thus the notation: $P_B(v_i, s_j, d_k)$. To each state are associated 8 different probability values which correspond to 8 different scenario configurations and to calculate the probability in a particular configuration, it is need to compute a weighed sum of these 8 probability values associated with the user state. One of these 8 probability values is indicated in a simplified manner as $P_{i,j,k}$ and represents the probability to obtain a bad packet in the scenario configuration: (v_i, s_j, d_k); in this way $P_{i,j,k}$ correspond

to $P_B(v_i, s_j, d_k)$. Continuing the previous example to the *"cube" (3,4,1)* there are associated these 8 probability values:

- top face of the *"cube"*: $P_B(40, 111, 1000)$, $P_B(60, 111, 1000)$, $P_B(40, 111, 3000)$, $P_B(60, 111, 3000)$;
- lower face of the *"cube"*: $P_B(40, 146, 1000)$, $P_B(60, 146, 1000)$, $P_B(40, 146, 3000)$, $P_B(60, 146, 3000)$.

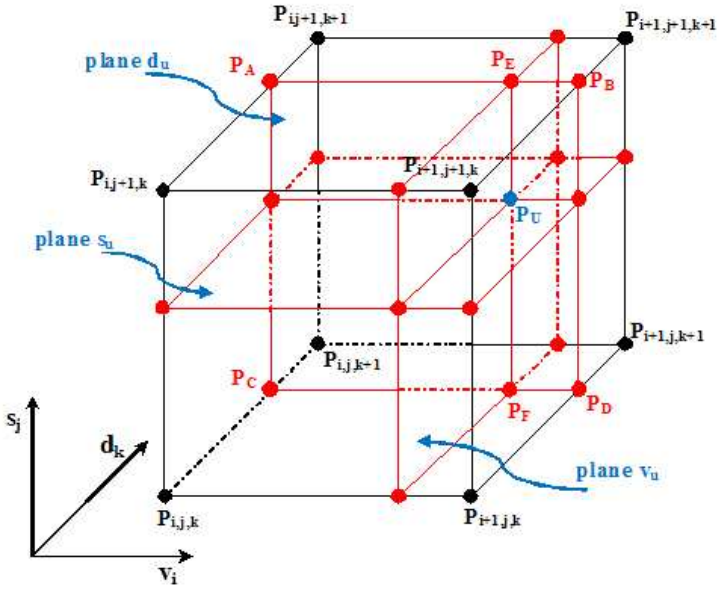

Fig. 2.33. A particular "cube" (state) of the model

Assume that, in a particular instant, the scenario configuration is (v_u, s_u, d_u). To calculate the corresponding probability value, indicated as P_U, it is necessary to individuate the stated in which the user configuration fall in. To individuate the state the following equation with the equations (2.25) and (2.34) must be used:

$$k = K(t) = \left\lfloor \frac{(d(t) - 1000)}{2000} \right\rfloor + 1 \tag{2.44}$$

At this point we know what is the state with the 8 probability values associated to it. It is possible evaluate P_U evaluating graphically the interception point among the three planes related to v_u, s_u and d_u; consequently to evaluate P_U is possible to follow (considering now a 3D case) the same process used in IWPM-2V. The following step is to evaluate the P_A, P_B, P_C and P_D values, in this way the scenario become independent by distance value:

$$P_A = w_{d,k} * P_{i,j+1,k} + w_{d,k+1} * P_{i,j+1,k+1} \tag{2.45}$$

$$P_B = w_{d,k} * P_{i+1,j+1,k} + w_{d,k+1} * P_{i+1,j+1,k+1} \tag{2.46}$$

$$P_C = w_{d,k} * P_{i,j,k} + w_{d,k+1} * P_{i,j,k+1} \tag{2.47}$$

$$P_D = w_{d,k} * P_{i+1,j,k} + w_{d,k+1} * P_{i+1,j,k+1} \tag{2.48}$$

subsequently we can evaluate P_E and P_F (in this way is eliminated the dependence from speed variable) :

$$P_E = w_{v,i} * P_A + w_{v,i+1} * P_B \tag{2.49}$$

$$P_F = w_{v,i} * P_C + w_{v,i+1} * P_D \tag{2.50}$$

finally the desired P_U value is obtained by:

$$P_U = w_{s,j} * P_F + w_{s,j+1} * P_E \tag{2.51}$$

The adopted process can be followed graphically in the figure 2.33.

The weigh functions $w_{v,i}$ and $w_{v,i+1}$ used in the equations (2.49) and (2.50) are defined by equations (2.29) and (2.30); the weigh functions $w_{s,j}$ and $w_{s,j+1}$ used in (2.51) are defined by equations (2.38) and (2.39); instead, the weigh functions used in (2.45 - 2.48) are defined by the following:

$$\begin{aligned} w_{d,k} &= w_{d,k}(d(t)) = \\ &= f_{d,odd}(d(t)) * mod(k,2) + f_{d,even}(d(t)) * (1 - mod(k,2)) \end{aligned} \tag{2.52}$$

$$\begin{aligned} w_{d,k+1} &= w_{d,k+1}(d(t)) = \\ &= f_{d,odd}(d(t)) * mod(k+1,2) + f_{d,even}(d(t)) * (1 - mod(k+1,2)) \end{aligned} \tag{2.53}$$

$$\begin{aligned} f_{d,odd}(d(t)) &= d_{o0} + d_{o1} * \cos(d(t) * \omega_d) + d_{o2} * \sin(d(t) * \omega_d) \\ d_{o0} &= 0.5 \\ d_{o1} &= -3.294e-11 \\ d_{o2} &= 0.5 \\ \omega_d &= 0.001571 \end{aligned} \tag{2.54}$$

$$\begin{aligned} f_{d,even}(d(t)) &= d_{e0} + d_{e1} * \cos(d(t) * \omega_d) + d_{e2} * \sin(d(t) * \omega_d) \\ d_{e0} &= 0.5 \\ d_{e1} &= 3.294e - 11 \\ d_{e2} &= -0.5 \\ \omega_d &= 0.001571 \end{aligned} \tag{2.55}$$

The d_{o1} and d_{e1} terms can be neglected.

An application case for IWPM

The IWPM model can be used to obtain an estimate of probability to loss a packet referred to a particular link between two network nodes. This estimate can be made instant by instant, in a runtime way. The advantage is that this value is available in every moment. This opportunity can be exploited in a path choice metric. When a source node has data to send, it needs a route to reach the destination node. The node can start a process to establish the route to destination, and by this process, a set of routes can be obtained. To select the most convenient path, the source, can use a metric. In the literature, the ETX (Expected Transmission Count) and ETT (Expected Transmission Time) metrics are very interesting [47]. ETX is based on the estimate of loss packet rate for a link. ETX, for link i, is defined by the following (see [47]):

$$ETX_i = \frac{1}{1 - P_i} \tag{2.56}$$

P_i is the packet loss probability for the link i, it can be expressed as:

$$P_i = 1 - (1 - P_{fi}) * (1 - P_{ri}) \tag{2.57}$$

P_{fi} and P_{ri} are the forward and reverse packet loss rates (for the link i) respectively. In this way, ETX_i is the expected packet transmissions number to send successfully a packet on link i. The real ETX problem is that does not take into account the different data rates of each link in a path. ETT resolves this problem, and considering a packet size equal to S_i and a link bandwidth equal to B_i, ETT_i (for link i) can be calculated as:

$$ETT_i = \frac{1}{1 - P_i} * \frac{S_i}{B_i} = ETX_i * \frac{S_i}{B_i} \tag{2.58}$$

Thus ETT_i represents the estimate time to successfully send a packet on the link i. For both metrics, to estimate the packet loss rate, the sending of probe packets on the link is essential. This probe packet introduce latency and overhead. At this point, come into play the IWPM model. The IWPM model can be introduced in the metric concept into two different ways.

- First way: the source node sends the probe packets and considers the only loss packets due to network congestion; in this way the node can evaluate the packet loss probability P_{ci} due to congestion state. Instead, IWPM model provides the packet loss probability P_{IWPMi} due to transmission channel condition. The ETX and ETT modified metrics are the following:

$$ETX_i^{'} = \frac{1}{1-(P_{ci}+P_{IWPMi})} \tag{2.59}$$

$$ETT_i^{'} = \frac{1}{1-(P_{ci}+P_{IWPMi})} * \frac{S_i}{B_i} \tag{2.60}$$

- Second way: we can avoid the probe packet sending, thus the overhead and latency are eliminated. The metrics can consider only the channel condition using only the IWPM estimate (the network congestion can be evaluated in other ways); the new relations are the following:

$$ETX_i^{''} = \frac{1}{1-P_{IWPMi}} \tag{2.61}$$

$$ETT_i^{''} = \frac{1}{1-P_{IWPMi}} * \frac{S_i}{B_i} \tag{2.62}$$

The previous two application cases are only two examples of the potentiality offered by the IWPM model, as we have explained in this chapter, the model can be used both as a generative model, to facilitate the software simulations, that as a model to support the QoS.

In conclusion, the performance of each Markov chain based model is good if a static scenario is considered, i.e. if the user always maintains the same speed value, but none of the chain based models can represent a dynamic scenario in which the user speed, or other variables, varies in a continuous way. This is because each model is represented by a matrix, and this can only reflect a specific scenario configuration, in fact, if the model is used to represent the channel behavior of a scenario with a variable user speed, its performances are very bad. To resolve this problem the IWPM model is proposed. It is possible to apply the IWPM model to a dynamic scenario in which, instant by instant, it is possible to know, depending on the circumstances of the scenario, what is the probability to receive a corrupted packet, in this way it is possible to foresee the loss and so action can be taken on certain parameters such as the packet size or the available QoS, in order to maximize the throughput of the system. The model is validated in a steady state condition but also in dynamic non-steady state; the excellent results obtained in both cases prove the model accuracy.

3

Call admission control in a mesh scenario

3.1 Introduction

The interest about distributed network architectures is arising. It allows to obtain more scalable network and in this context, also the interest of research around the IEEE 802.16 distributed mesh mode is growing. The mesh distributed mode supported by IEEE 802.16 protocol, with the capability to establish direct links between SSs (Subscriber Station) and with the wide coverage area and the promised bits rate, allows to create interesting scenarios. The protocol defines guidelines to realize request/grant process, but it does not define a distributed Call Admission Control (CAC) algorithm. An effective manage of call admission control process is essential to guarantee QoS constraints to admitted connections.

In distributed mesh mode, when a mesh node has an amount of new data to transfer to a destination node, it would require to the neighboring node, the instauration of a new connection. This last node has to decide whether to admit the new call, and obviously, how much bandwidth to be allocated to the new connection, for the service lifetime. The first is the admission decision, the second one is inherent the bandwidth to grant to the node for the admitted connection.

Both the decisions are inherent the bandwidth utilization in the network and influence the desired QoS level: the arrival of a new connection, can modify the allowed bandwidth to the existing connections, thus, all the QoS constraints must be reviewed. Therefore there is a "risk" in this choice, because admitting a new connection, we must accept the possibility to worse the provided QoS to the old connections. The first of the previous listed processes decision is called call admission control, and this decision influences the network bandwidth utilization for a long time, i.e. it is a long term decision. The second one, instead, is a short time decision and it defines the amount of bandwidth to grant to the requester node.

In this chapter we present a new CAC algorithm, referred to a mesh scenario which takes into account a set of three traffic classes with different

priority levels. The focus is to guarantee to higher priority flows the respect of QoS constraints defined in term of end-to-end delay. To make this the distributed CAC algorithm admits all the new calls in a greedy way, thus the lower priority flows can exploit the bandwidth availability until a set of higher priority calls claim to obtain bandwidth. The lower priority admitted calls are thus preempted to leave room for the new calls. The preemption process has a negative side, it can leave in data subframe little gaps which are not useful. The solution is the presence of a defragmentation process started by granter node.

In the rest of this chapter the our algorithm GCAD-CAC (Greedy Choice with Bandwidth Availability aware Defragmentation) and its performance evaluation are presented in detailed way.

3.2 Call Admission Control in WiMAX mesh networks: the state of the art

The study of the IEEE 802.16 technology is still an open issue. There are not many works that fill the gaps in the protocol, this is true for the mesh mode and even more for algorithms related to distributed mode. The same call admission topic in mesh mode is a bit neglected by the literature. Instead, some works treating distributed scheduler performance can be collected by literature, in particular [48] analyzes distributed scheduler performances and illustrates how dynamically to set the *xmt holdoff exponent* parameters. The optimization of mesh scheduling is described in [49] evaluating a combined centralized - distributed scheduling. Authors of [50] proposes only an improvement of distributed mesh scheduler. A proposal for a call admission control algorithm in distributed mode is the work [51]; in this last paper the concept of connection preemption with some limitations is presented; in [51] three traffic classes with assigned priorities are considered, the admission algorithm is based on the concept that all the bandwidth can be divided among the three classes, but in this way in a steady state, the advent of new data flows with higher priority are refused because this class consumed the bandwidth reserved to it; instead, new data flows with lower priority can be admitted. Also the scenario used to validate the proposed CAC algorithm is very simple, the maximum path length in scenario is two hop. In [52] the authors propose an end-to-end bandwidth reservation scheme with a CAC algorithm referred only to VoIP traffic. In [53] there is the proposal of a simple CAC algorithm. The authors consider a traffic differentiation using the priority field of unicast CID. The CAC algorithm is based on a threshold mechanism. The requests with higher priority, if there is sufficient free minislots, are always admitted, whereas the low priority requests are refused in case of congestion, which is verified with a bandwidth utilization computation. If bandwidth utilization is greater then fixed threshold then low priority requests are refused. Also the simulated scenario is too simple, each node is a neighbour of each other node

in the network. The paper [54] describes a CAC algorithm related to PMP mode; it is very interesting for the connection preemption concept that is introduced and the admission decision is based on traffic class and bandwidth utilization of each traffic class. Each traffic class has a bandwidth portion reserved to it and also can preempt the lower priority admitted calls.

Other interesting works treating the call admission control in PMP mode are [55] - [59] and in particular, although it considers the PMP mode, the work [60] is to be taken into account to enrich our knowledge, in fact, the authors of [60] apply the Games Theory ([61], [62]) to call admission issue.

The contribution of our work, can be considered important in the context of the research about 802.16 mesh distributed architecture, because, at the best of our knowledge, the literature presents lacks or few works about CAC in mesh distributed mode. Our intent is to present a distributed call admission control algorithm which takes into account three different traffic classes. The proposed GCAD (Greedy Choice with bandwidth aware Availabilities Defragmentation) algorithm presents two interesting processes: preemption and defragmentation process. Preemption occurs when there is a new call with higher priority, this call can preempt a call with lower priority. Preemption process can cause a fragmentation in data subframe, i.e. we can find, in data subframe, some very little unusable gaps of free minislots. Defragmentation process collects these gaps creating a continuous availability.

3.3 GCAD: A new Call admission control algorithm

We propose a CAC algorithm for an IEEE 802.16 distributed mesh network, each mesh node can support three different data traffic classes with *"1"*, *"2"* and *"3"* as priority values. The values *"1"* and *"3"* are the highest and the lowest priority values respectively. When a new source starts to transmit data, it has to individuate a path to send data to destination node. Subsequently, the mesh node, can submit a bandwidth request to next hop node. In the following of this chapter, we describe our proposal for the source and for the next hop node behavior. The first one is described in term of bandwidth estimation, or more properly minislots number estimation, and the last one in term of call admission control process. In the following we indicate source node as requester and the next hop node as granter. Obviously the next hop node in turn becomes a requester and so on.

3.3.1 Minislot number request estimation

Each node has a data queue, when a packet appears in the queue, the node creates a bandwidth request. The node can classify the queued packets using the priority field present in the unicast CID. The three traffic classes can have QoS constraints expressed in term of end-to-end delay, thus, the node has to estimate the amount of minislots request. Each data subframe is dived into

a fixed number of 256 minislots. In turn, considering an OFDM (Orthogonal Frequency Division Multiplexing) modulation, each minislot is characterized by a number of OFDM symbols with its efficiency. We define the following parameters:

- n: request minislots number;
- MS: OFDM symbols number for each minislot;
- p_{size}: packet size (bits);
- eff: efficiency of an OFDM symbol, expressed as number of data bits for each symbol;
- dl: delay constraint;
- d_{sym}: OFDM symbol duration (s);
- f: frame duration (s);
- h: path to destination hops count;

considering a packet with QoS constraints, we evaluate n value resolving the following equation:

$$(n * MS * d_{sym}) + \left(\frac{p_{size} - (n * MS * eff)}{n * MS * eff} \right) * f = \frac{dl}{h} \tag{3.1}$$

The first term of equation (3.1) indicates the delay contribution related to packet forwarding in n minislots, i.e. the packet is spread on n minislots. The allocated minislots can be insufficient to send the whole packet, thus it can be fragmented on a number of frames; the second term of first member takes into account the delay contribution due to eventually packet fragmentation. Finally, the second member of (3.1) says that the delay constraints must be respected for each hop of the path. Instead, request for packets belonging to best effort (BE) data traffic, without delay constraints, is conducted in the following way:

- t_i: arrival time of the first queued packet of BE traffic;
- t_f: arrival time of the last queued packet of BE traffic;
- n_{BE}: number of BE queued packets;
- p_mean: mean packet size of BE queued packets;
- R: estimated BE rate;

we estimate the request n with:

$$R = \frac{p_{mean} * (n_{BE} - 1)}{t_f - t_i} \tag{3.2}$$

$$n = R * \left(\frac{f}{MS * eff} \right) \tag{3.3}$$

Periodically, in both cases, nodes verify if the received grant is sufficient to transmit the queued packets, otherwise they make a new request.

3.3.2 Call Admission Control Algorithm

The proposed GCAD-CAC algorithm is described by the flow diagram depicted in figure 3.1. The parameters expressed in the flow diagram are defined as the following:

- B_A: minislots number available at arrival instant of a new request;
- B_A^P: minislots number collected by preemption;
- $B1_A^T$: total minislots number obtained after a preemption to admit a new request with priority equal to *"1"*;
- $B2_A^T$: total minislots number obtained after a preemption to admit a new request with priority equal to *"2"*;
- B_D: total minislots number which can be obtained by defragmentation process.

When a mesh node receives a new request, expressed as a number of requested minislots: R_n, it admits all kinds of requests if there is sufficient available bandwidth. This explains why the algorithm is defined *greedy*; many CAC algorithms define utilization constraints and refuse a new connection if its traffic class has achieved the utilization threshold. We instead, try to take advantage by the actual available minislots, also, trying to respect all QoS delay constraints. If a new request arrives with higher priority than previous admitted one and there are not sufficient available minislots, then admitted calls with lower priority can be preempted. Thus, before to preempt a connection, the granter evaluates the amount of minislots obtainable by preemption: B_A^P. If the total available minislots ($B1_A^T$ or $B2_A^T$ for request with priority equal to *"1"* and *"2"* respectively) is greater or equal to R_n, then the preemption of a previous admitted request c_i with:

$$Priority(c_i) < Priority(R_n) \tag{3.4}$$

is executed. An important condition to be considered is the following: $B1_A^T$ and $B2_A^T$ are evaluated considering only contiguous minislots.

For example, considering allocation scheme depicted in figure 3.2, a new request, with priority equal to *"1"*, can preempt the *"e"* and not *"d"* allocation because only *"e"* is contiguous with available minislots. After preemption the new request is admitted. The granter, in this case, advises the connection *"e"*, that is preempted. In order to advise the preempted connection, we send to the owner node a grant message with minislot range field equal to *"0"*, and the preempted connection can remake a new request to try to obtain eventually free minislots.

After preemption test, if a new request with priority equal to *"1"* or *"2"*, have not sufficient available minislots, the granter node can activate the defragmentation to collect fractioned available minislots in a whole availability.

In figure 3.3, it is possible to note the case in which there is advantage in defragmentation utilization. The rectangle without number represents free

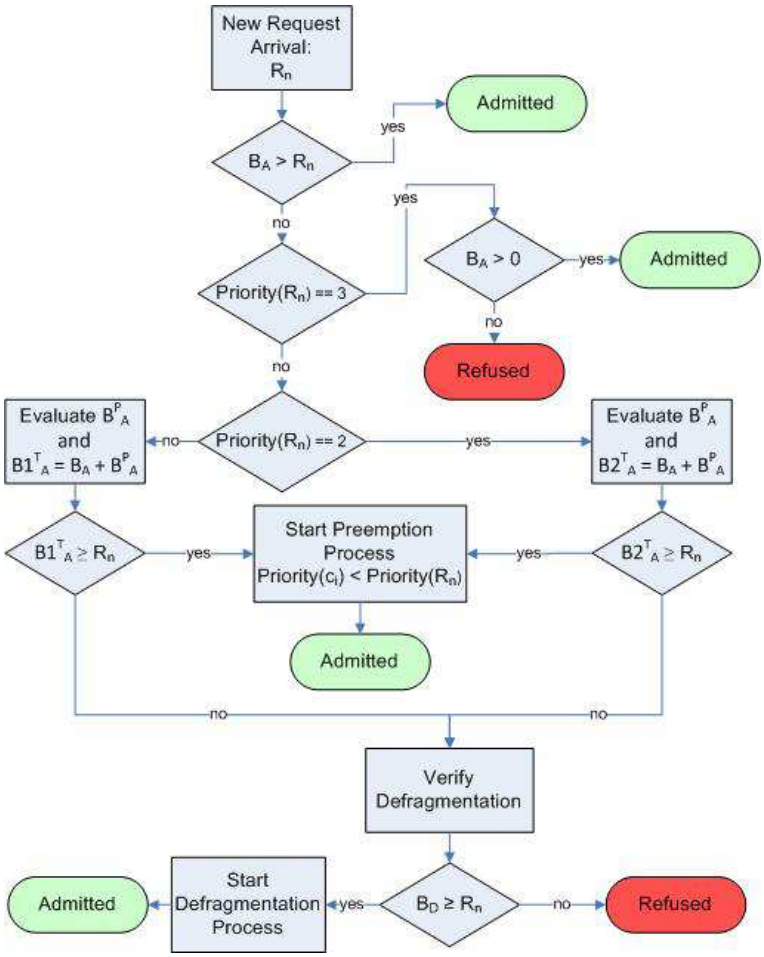

Fig. 3.1. Call admission control proposed algorithm

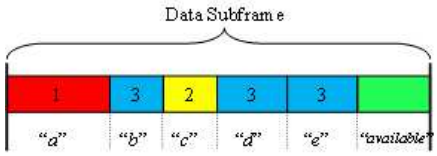

Fig. 3.2. Data subframe with minislot allocations

minislots. The case *(a)* represents the data subframe before a preemption due to arrival of a new request with priority equal to *"2"*; the data subframe state after preemption is represented in case *(b)*, the preemption causes the presence of a free minislot gap between two allocations; with defragmentation

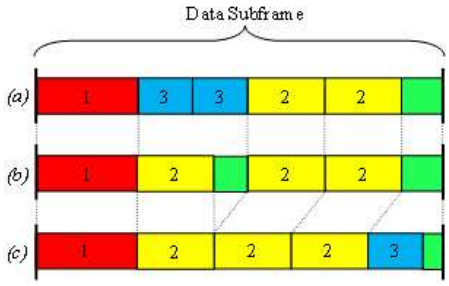

Fig. 3.3. Data subframe states: (a) before preemption; (b) after preemption and finally (c) after defragmentation process

two gaps are unified and a new request can be admitted, it is in case *(c)*. To realize defragmentation, the granter sends a grant message with range equal to *"0"* to all the interested nodes. In this way the granter node advises the defragmented connections owners to bargain for the new grants. The granter, obviously, in the "bargain" process, allocates minislots in a contiguous way, and admits a new request, in advanced minislots, only after re-allocation of the defragmented connections.

3.4 Simulation Scenario

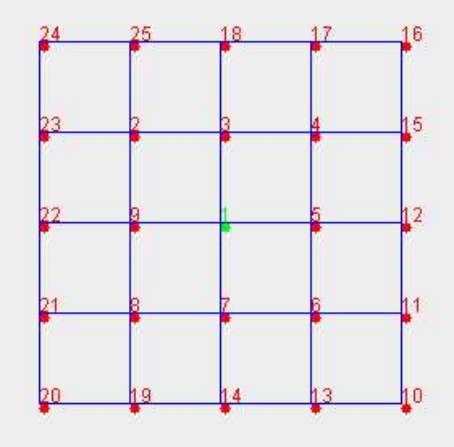

Fig. 3.4. Simulated scenario

To test the proposed algorithm, we design a network simulator for IEEE 802.16-2004 protocol in JAVA language. In simulator we implement our algorithm and also other two algorithms to make a performances comparison. In

figure 3.4 the mesh simulated scenario is depicted. We consider a mesh network with 25 nodes, one of these (the number 1) is the BS. The depicted lines represents the active links. The scenario is a square with area: $5km * 5km$. All the traffic is from mesh nodes to BS node.

Table 3.1. Simulation settings

SIMULATION SETTINGS		
PHY SETTINGS		
Modulation		OFDM, QPSK 1/2
BW(Channel Bandwidth)		25Mhz
NFFT		256
G		1/8
Frame length		20 ms
Symbol efficiency		184 bits
Coverage radius		500 m
MAC SETTINGS		
msh-ctrl-len		4
msh-dsch-num		4
msh-csch-data-fraction		0
scheduling-frame		1
data queue size		50
SOURCES SETTINGS		
number of sources		3 - 24
QoS delay constraints		
priority: 1		40 ms
priority: 2		80 ms
priority: 3		/
Sources rate (CBR)		
	packet size	packets/s
priority: 1	64 bytes	128
priority: 2	2500 bytes	25
priority: 3	2500 bytes	125
Simulation run duration		500 s
number of runs / configuration		10
confidence interval		95%

Table 3.1 summarizes all simulation settings. In each simulation the source nodes and the packets generation start instants are randomly selected. The presented algorithm is compared with other two algorithms individuated in the literature. The first is extracted by paper [53] and in the following of paper we indicate it as THR algorithm (THR because it is based on threshold mechanism). The second algorithm is a CAC algorithm for 802.16 PMP scenario and it is proposed in [54], it is very promising and we adapt it to a distributed mesh scenario; in the subsequent sections we refer to it as PMP

algorithm. In the final part of this section, THR and PMP algorithms are briefly introduced.

THR is a call admission control algorithm for 802.16 distributed mesh mode. Calls are classified into three different classes and the admission decision is based on few concepts:

- there is the presence of two checkpoints fixed along the available minislots: cp1 and cp2;
- there is a threshold value for bandwidth utilization;
- if the bandwidth utilization at checkpoint cp1 is less than threshold, all the calls are admitted without to distinguish between priorities, otherwise, to admit a low priority call, the node, searches a frame from checkpoint cp2, if there are sufficient availabilities, the request is admitted else refused.

PMP is a call admission control algorithm referred to 802.16 Point to MultiPoint mode. In call admission decision, the algorithm distinguishes between four different service classes: UGS (Unsolicited Grant Service), rtPS (real time Polling Service), nrtPS (not real time Polling Service) and BE (Best Effort). In our work, instead, three different traffic classes are considered and thus, to import in mesh mode, the PMP call admission control proposed in [54], we maps UGS, rtPS and BE in traffic classes with priority *"1"*, *"2"* and *"3"* respectively. Respecting the previous service mapping, the call admission decision is taken following these criterions:

- Advent of request with priority equal to *"1"*: B_1 is the bandwidth request with priority *"1"*. When a node receives the new request, it verifies if the remaining bandwidth is less than B_1 request. If the condition is verified, then the request is admitted, else the mesh node verifies if this condition can become true considering the preemption of previous admitted requests with priority less than *"1"*. If the condition, with the new bandwidth availabilities, becomes true then the request is admitted else it is refused.
- The mesh node receives a request with priority equal to *"2"*: B is the total bandwidth, B_2 is the request with priority *"2"* and R_{e1} is the bandwidth reserved to calls with priority *"1"*. If the bandwidth admitted to the previous requests with priority *"2"* plus B_2 is less than $B - R_{e1}$ and if the remaining bandwidth is not less than B_2, the request is admitted; otherwise if the first condition is true, we can calculate the remaining bandwidth plus the amount of bandwidth released by preempted connection with priority *"3"*, if it is not less than B_2, the request is admitted else it is refused.
- The request with priority *"3"* can use the remaining bandwidth, and can be preempted if it is necessary.

To built a comparison, we test the algorithms using an increasing sources number: from 3 to 24. It is equally divided between the three traffic classes. In this way, with a sources number equal to 24, we mean that the scenario

contains 8 sources with priority *"1"*, 8 sources with priority *"2"* and 8 with priority *"3"*.

3.5 Performance evaluations

To evaluate performances algorithms, we select a set of parameters and use it to make a comparison between the proposed GCAD, the PMP and THR algorithms. The performances parameters are the following:

- packet loss percentage: it is defined as the percentage of total packets generated by sources and not delivered to destinations. A packet can be lost because the data queue of a mesh node is full, or because a request is not admitted;
- throughput: the percentage of sent packet received at destination;
- average number of refused request: it takes into account the average number of requests which are not admitted;
- average end-to-end delay: it is the average time interval required by a packet to complete the path from source to destination;
- delay jitter: it is a variability measure of packet delay. The delay jitter is very important for real time application.

In the figures 3.5, 3.6 and 3.7 the algorithms behavior, in terms of packet loss percentage, are represented. The figure 3.5 considers the case of traffic class with priority value equal to *"1"*. It is the higher priority traffic class. GCAD presents the best performance and maintains the percentage, always under 5% value.

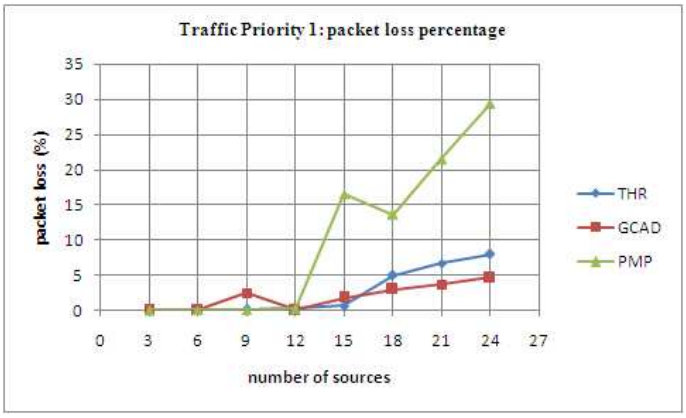

Fig. 3.5. Packet loss percentage of sources with priority equal to "1"

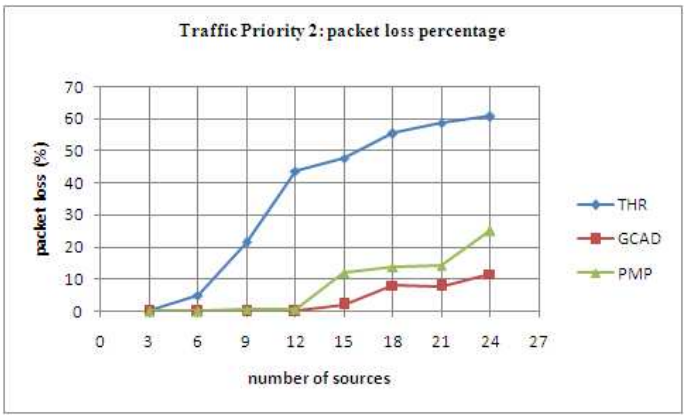

Fig. 3.6. Packet loss percentage of sources with priority equal to "2"

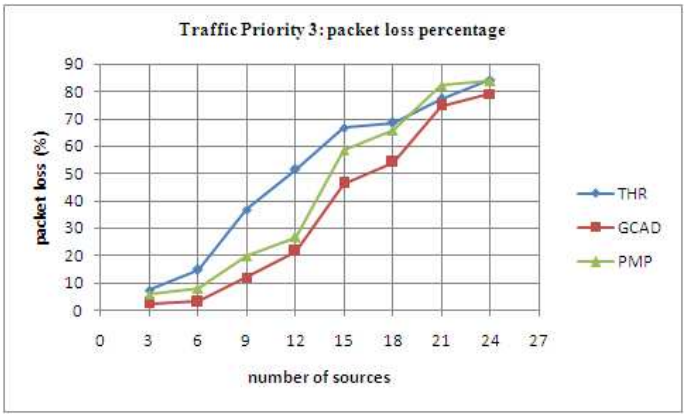

Fig. 3.7. Packet loss percentage of sources with priority equal to "3"

The worst case is obtained by PMP algorithm, in fact increasing the number of sources, the percentage of packet loss, tends to reach high values. The trend of GCAD, instead, grows slowly increasing the network congestion. Also observing the figures 3.6 and 3.7 GCAD shows the best trends. In figure 3.6 the worst case is related to THR algorithm, while in the figure 3.7 all the algorithms have a similar answer to increasing congestion. Considering the three cases, we can confirm what are the algorithm focus: THR tries to give more importance to priority *"1"*, neglecting priority *"2"* and *"3"*; PMP wants to put on a par the two more important priority traffic classes; GCAD has the same focus of PMP but it allows to reach the best performance due to the presence of defragmentation process. The defragmentation process gives to algorithm the capability to accept a higher number of requests and bandwidth amount. This is visible in figures 3.8, 3.9 and 3.10. In the figure 3.8 the

only algorithm which has refused calls is THR, instead the figure 3.9 shows that GCAD is able to obtain a higher number of requests of priority *"2"* and this is confirmed by packet loss depicted in figure 3.6. In figure 3.8, PMP did not refuse calls with higher priority, but it reaches high values of packet loss in congested network, this because PMP accepts all the requests but giving them little amounts of bandwidth. The PMP and GCAD behaviors depicted in figure 3.10 are similar.

In this way, evaluating the packet loss and the average refused calls, we can conclude that the introduction of defragmentation process, allow to manage the bandwidth in a more optimized way. The elimination of little availabilities gaps, give to granter, the possibility to create contiguous allocations, with the right size, to admit new calls.

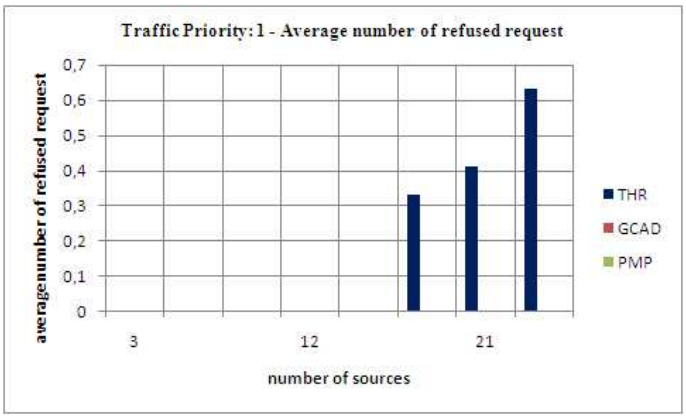

Fig. 3.8. Average number of refused request: sources with priority equal to "1"

Another way to see the capability, of each algorithm, to allow good results in terms of successfully transmitted packets, is to analyze the throughput performance. The throughput trends are depicted in figures 3.11, 3.12 and 3.13. The figure 3.11 is referred to sources with priority *"1"*; our algorithm obtains the best performance, also THR behavior is good, and this because it preserves a bandwidth portion to sources with higher priority. PMP performance instead, as depicted in figure 3.11, is characterized by a degradation due to bandwidth portion preserved for other kind of traffics. Also the figures 3.12 and 3.13, related to priority equal to *"2"* and *"3"* respectively, confirm the quality of our proposal. Another point in favor of GCAD algorithm is due to the greedy choice, in fact, if there is a sufficient number of minislots, it accepts each kind of request, and only in a second moment it starts the preemption process if and if it is necessary.

In figures 3.14, 3.15 and 3.16 the average end-to-end packet delays are depicted. The figure 3.14 considers the priority *"1"* case. Observing the depicted

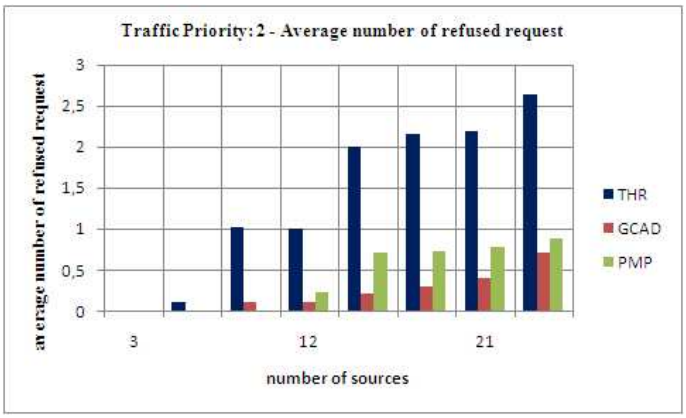

Fig. 3.9. Average number of refused request: sources with priority equal to "2"

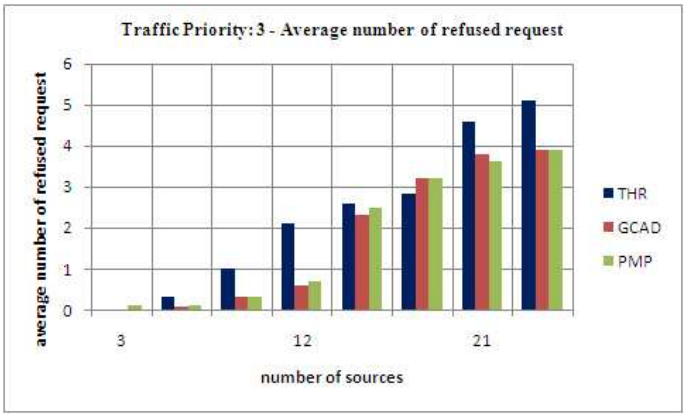

Fig. 3.10. Average number of refused request: sources with priority equal to "3"

trends, it is possible to see how the only algorithm, which respects the delay constraint, in each network condition, is the GCAD algorithm (there is the need to remember that the QoS constraint for data flow with priority value equal to *"1"* is an end-to-end delay value less than $0.04s$). PMP and THR do not respect the QoS delay constraint in scenario with 18, 21 and 24 sources.

Also in priority *"2"* case the THR algorithm overflows the delay threshold (in this case the end-to-end delay constraint is less than 0.08 s). Instead in priority *"3"* case, there are not quality thresholds. The GCAD algorithm does not present the best behavior in each case and this is due to the presence of defragmentation process. From one hand it allows the optimization of bandwidth management, and on the other hand it pays this with a not perfect delay behavior.

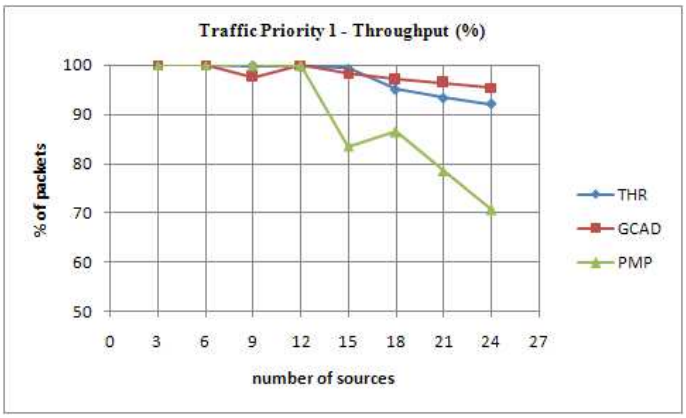

Fig. 3.11. Throughput of sources with priority: "1"

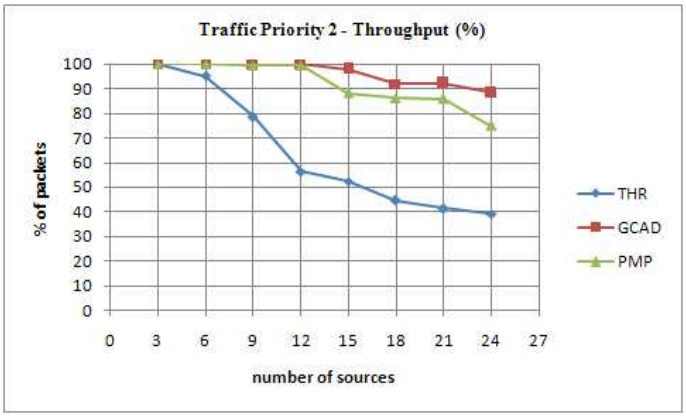

Fig. 3.12. Throughput of sources with priority: "2"

Finally in figures 3.17 and 3.18 we depict the jitter trends related to priority *"1"* and *"2"* cases.

The GCAD algorithm, in traffic with priority equal to *"1"*, is characterized by the best results, its jitter trend is regular and the values are not great also in congested network. This delay jitter characteristics is very important in real-time application. Instead, observing figure 3.18, we can see that delay jitter trend is more irregular, this surely is due to defragmentation process. In fact it, can introduce variable delays, because it causes a new bandwidth bargaining process of connections involved in defragmentation.

Summarizing we can conclude that in this chapter we present a new call admission control algorithm for 802.16 distributed mesh networks. The algorithm is characterized by an initial greedy choice, by a preemption and a defragmentation processes. The proposed algorithm is tested in a scenario

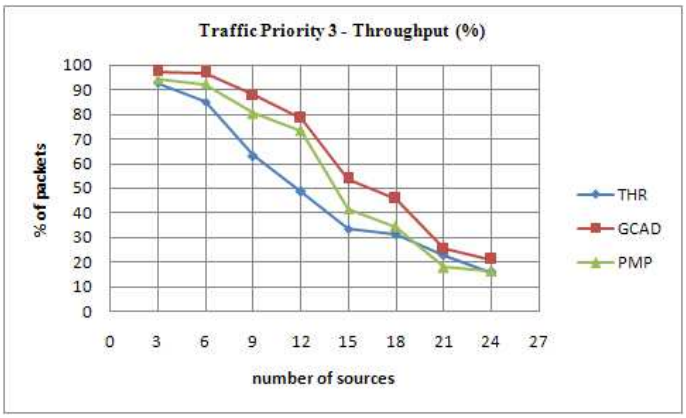

Fig. 3.13. Throughput of sources with priority: "3"

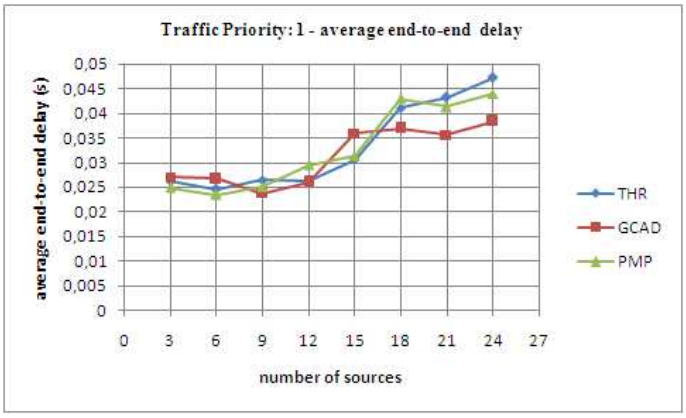

Fig. 3.14. Average end-to-end delay: sources with priority equal to "1"

of 25 mesh nodes with a max number of 24 sources. The performances of proposed GCAD algorithm are evaluated by throughput, average end-to-end delay, average delay jitter, number of refused requests and packet loss percentage. The GCAD performances are compared with other two CAC algorithms extracted by the literature. The GCAD algorithm presents the best performances reached through the presence of a defragmentation process. It allows an optimized management of minislots allocations.

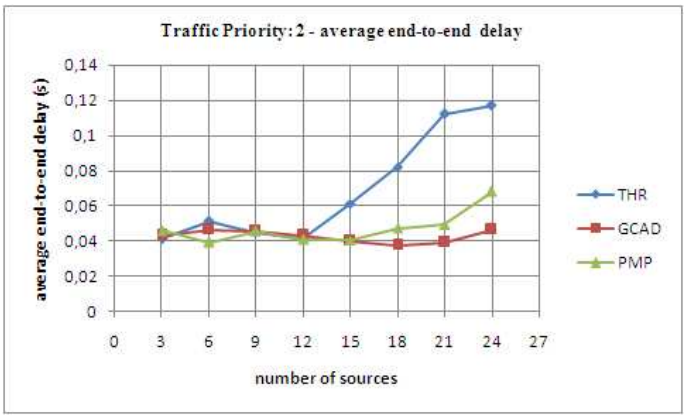

Fig. 3.15. Average end-to-end delay: sources with priority equal to "2"

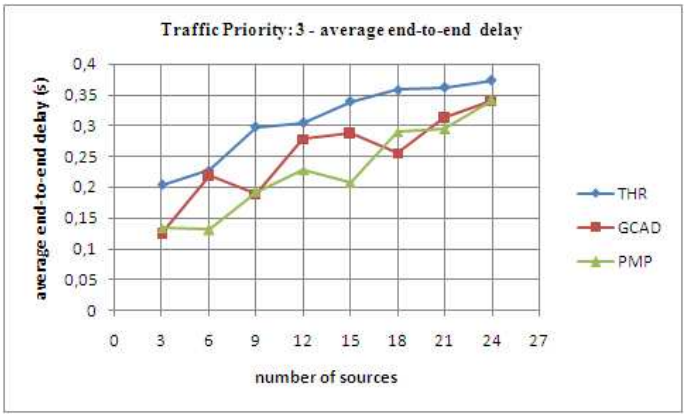

Fig. 3.16. Average end-to-end delay: sources with priority equal to "3"

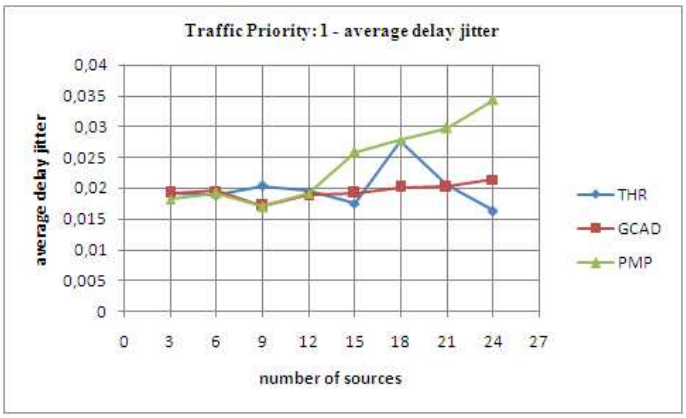

Fig. 3.17. Average delay jitter: sources with priority equal to "1"

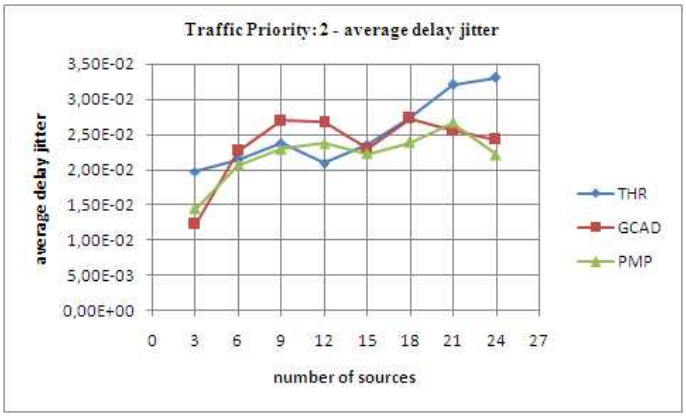

Fig. 3.18. Average delay jitter: sources with priority equal to "2"

4

A metric as routing support in a multi route mesh scenario

4.1 Introduction

The IEEE 802.16 WiMAX standard provides a set of mechanisms to create a mesh network, in order to join the mesh network a new node needs to listen the channel, waiting for a particular configuration message defined MSH-NCFG. When the new node, defined as *candidate node*, receives two MSH-NCFG messages from the same source, the *candidate node* selects the source node and elects it as the *sponsor node*. In this way, the *candidate node* can collect all the necessary information listening all the configuration messages sent from the *sponsor node*. The configuration messages contain both network parameters and information about neighboring nodes, thus the candidate can create a link with the *sponsor node* but also with the other neighboring nodes. The result of this process is the creation of a mesh network which can be managed in a centralized or distributed way.

At this point, in a generic mesh network, each node is able to know its neighborhood (neighboring nodes) and to communicate with them. Using MSH-NCFG messages it is aware of the presence of another set of nodes defined extended neighborhood (nodes which are two hops away from it).

According to the previous concepts, a node is not able "to see" beyond the extended neighborhood; in this way to individuate a route to reach a destination or in particular to reach the BS, that takes place the role of gateway to the *rest of the world*, there is the need of a routing algorithm.

Wireless mesh networks offer very attractive characteristics for building new wireless infrastructure, they provide a cost effective broadband connectivity for wireless terminals spread over a large area. In addition, the mesh network are more reliable and easier to deploy and maintain due to the elimination of a single point of failure and self-organization characteristic. Wireless mesh networking is a relatively new technology originating out of ad hoc networking research, as a consequence, there is still an ongoing effort to find routing protocols which perform best in large static or quasi-static wireless mesh networks.

To individuate a route in a IEEE 802.16 network it is possible to use a variety of routing protocols already tested in other technologies; an example of these is the AODV protocol. This routing protocol, as other protocols, is able to provide not only a unique route from a generic source node to a destination but it can be used to discover more than one existing path within the network. Faced with a series of alternatives comes the need to choose the best option. But of course, in order to classify an option as better than another, it is necessary to set an objective function that is defined as the goal which we want to achieve by choosing an option instead of another.

Considering a generic algorithm which identifies a number of routes within a network, our contribution has been to define a metric to support the choice of the route. A metric is a function which assigns a weight to each route discovered in a network; using the weight we are able to choose the route with the lesser or greater weight associated by the intended objective.

4.2 The state of the art

A first simple idea to individuate an interesting route could be the choice of the shortest path that is constituted by the smallest number of hops, but this sometimes does not lead to the better solution. This is due to the fact that the choice of route with the smallest hop count may lead to the creation of a route constituted by links with "destructive" behavior, i.e. by links with high values of PER (Packet Error Rate) associated to it. In practice, the selection of the shortest path does not take into account the qualitative analysis of the links constituting the path.

In the literature there are various metrics that are able to perform in good way the task assigned to them. The proposed routing protocols mainly consider interference, hop distance, path loss, network density and the number of channels to select either a minimum latency or a high throughput multihop route between a source and a destination.

A first example of a metric which can achieve better results than the minimum hop count is that defined in [63], in this paper the authors present a metric function based on concept of interference. Multiple access interference is a major limiting factor for wireless communication systems. Interference in wireless system is one of the most significant factor that limits the network capacity and scalability. The objective of this paper is to propose an efficient approach for increasing the utilization of WiMAX mesh through appropriate design of multi-hop routing and scheduling. As multiple-access interference is a major limiting factor for wireless communication systems, the authors adopt an interference-aware cross-layer design to increase the throughput of the wireless mesh network. To reach the focus to increase the network throughput, the authors propose a metric function defined blocking metric: *B(k)*. The blocking metric of a multihops route indicates the number of blocked/interfered nodes by all the intermediate nodes along the route from the source toward

the destination node *k*. The authors define the blocking value *b(n)* of a node *n*, as the number of blocking/interfered nodes when *n* is transmitting, consequently the blocking metric of a route is the summation of the blocking value of the nodes that transmit or forwards packets along the route. Two example of calculation of blocking metric for a route is illustrated in figures 4.1 and 4.2. Defining *B(k)* as:

$$B(k) = \sum_{n \in route} Ng(n) \tag{4.1}$$

where *Ng(n)* represents the set cardinality of the neighbors of node *n*. The route in figure 4.1 has a metric value:

$$B(k) = 2 + 4 + 3 + 4 = 13 \tag{4.2}$$

instead the route in figure 4.2 has a metric value:

$$B(k) = 2 + 4 + 5 + 4 = 15 \tag{4.3}$$

Following the previous concepts the best route to increase the throughput decreasing the interference, we must choose the route in figure 4.2.

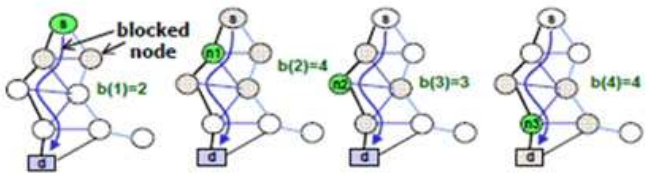

Fig. 4.1. Calculation of blocking metric for a route (case a)

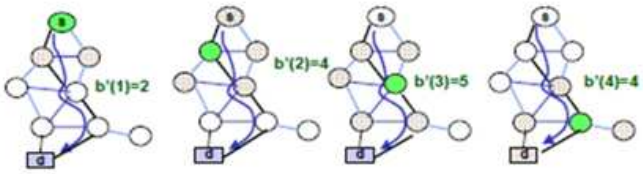

Fig. 4.2. Calculation of blocking metric for a route (case b)

In this metric interference based, even if a blocked node does not have any packet to send, it is considered for calculation. In this way the blocked node does not give the real situation of the interference in the network and

this because it is built exclusively on a topology based interference. An improvement to this metric is presented in [64]. To overcome the limitation of the previous metric, the authors of [64] take into account also the number of packets contained in the node queue. The interference based metric become the following: the blocking metric of the node n is defined equal to the number of node blocked by n multiplied to the number of packet queued in the node n. This metric can be expressed as:

$$B_{queue}(k) = \sum_{n \in route} Ng(n) * Q(n) \tag{4.4}$$

where *Q(n)* represent the queued packets in node n. In this way the metric become not only topology based interference aware but also queue load based interference aware. This metric allow to obtain interesting results but in turn it neglects other concepts as the link quality and the probability to lose a packet due to environment impairment effects.

In literature there are other metrics that take into account also the impairment effects in each link of the route. Two interesting metric are the ETX [65] and ETT [66] metrics, where the second one is an improvement of the first one.

The ETX (Expected Transmission Count) metric is related to the concept of probability to lose a packet during a transmission between two nodes. to use the ETX metric there is the need to estimate the loss packet rate for each link and in order to evaluate this parameter, each node can send, periodically a probe packet, thus evaluating the state of the received packet it is possible to realize an approximation of the link channel behavior. If P_{fi} and P_{ri} indicate the forward and reverse packet loss rates (for the link i) respectively, then the packet loss probability for the link i, it can be expressed as:

$$P_i = 1 - (1 - P_{fi}) * (1 - P_{ri}) \tag{4.5}$$

and finally the ETX_i for link i, is defined by:

$$ETX_i = \frac{1}{1 - P_i} \tag{4.6}$$

In this way, ETX_i is the expected packet transmissions number to send successfully a packet on link i. the ETX value for the route is obtainable by:

$$ETX = \sum_{i \in route} ETX_i \tag{4.7}$$

The real ETX problem is that does not take into account the different data rates supported by each link in a generic route.

To take into account the bandwidth of each link it is defined the ETT (Expected Transmission Time) and it represents the evaluation of provisioned time to receive at destination a packet in a correct way. Considering the following parameters:

- S: size of the transmitted packet;
- B_i: bandwidth available on link "i";
- P_i: packet loss probability of link "i";

the expected transmission time to receive a packet in a correct way related to link *"i"* is evaluated by:

$$ETT_i = \frac{1}{1 - p_i} * \frac{S}{B_i} \tag{4.8}$$

Instead the ETT related to the whole route is defined as:

$$ETT_{route} = \sum_{i \in route} ETT_i \tag{4.9}$$

Both the two metrics present a problem: as previously introduced, to evaluate the packet loss probability there is the need to send periodically a probe packet, the periodical sending of probe packets introduces an increasing overhead and this waste of bandwidth contributes to decrease the network throughput.

Other interesting works are presented in literature, for example the [67] is an interesting work and it is an overview of a set of routing protocol applicable to IEEE 802.16 technology. In work [68] instead is presented an improvement for the ETT metric and is designed to obtain good performance in IEEE 802.11 scenario. The [66] in addition to describing the ETT, introduce also another modified version of this metric called WCETT, realized to obtain high throughput in multiradio channel scenario. Other interesting ideas are presented in [69] where complex scenario of mobile node in a multi-hop relay network is considered; in [70] and [71] an interesting math elaboration and the analysis of 802.16 mesh scenario are respectively presented.

For others interesting works you can see the following: [72], [73] and [74].

The IEEE 802.16 protocol does not define a routing protocol or a path selection metric, the intent of protocol developers is to demand these challenges to higher protocol layers. The literature is rich of works related to routing protocol and we do not want to develop another generic routing protocol, but our focus is to design a route selection metric which can be successfully applied in a WiMAX scenario to guarantee the achieving of high throughput values.

4.3 DIM: A Delivering time based Interference Metric

The IEEE 802.16 standard protocol defines the mechanisms to build a mesh networks and to allow the communication among the mesh nodes. The basic criterion for a mesh node, in order to transmit toward a neighboring node is that no one node can transmit on its own initiative, including the BS node, without coordinating its transmission within its extended neighborhood; if

this criterion is not respected the resulting scenario is a network with an high number of collided transmissions. the coordination become an important issue but there is also another important concept to highlight: when a node is transmitting, the other nodes are in a silence state, in turn when the transmitting node stops the data transmission then another node can transmit its data and so on, in this way each node represents an interference for the other nodes. An observation related to the previous concepts is the following: in a mesh network, in order to obtain high throughput values, it is necessary to decrease the interference. Obviously we do not to decrease the number of neighboring nodes, but when a node has to select a route to a generic destination, we can favorite the choice of a route which interfere with the minimum number of nodes.

Our basic idea is to develop an interference based metric but with the adding of concepts related to quality of interfering links. The following are the ideas related to develop our metric:

- we want to realize high throughput values;
- a node *"n"* interfere with another node if and only if *"n"* has data to transmit;
- we weight the interference of a neighboring node *"n"* with the time necessary to deliver its queued data to the neighboring nodes;
- the time to dispose a data packet is evaluated by expected transmission time;
- the route with minimum interference must be chosen;
- selecting the route with the lesser interference means that we are going to interfere with the set of nodes that need less time to dispose their traffic.

The metric function obtained considering the basic principles defined here is called Delivering time based Interference Metric (DIM) and considering the following parameters we can illustrate the metric function:

- P: set of nodes belonging to a route;
- wp: weight of the route build on the set P;
- N_i: set of neighboring nodes of node *"i"*;
- Q_j: set of queued packets at node *"j"*;
- $Pk_{Qj}(z)$: number of packet in Q_j with node *"z"* as next hop node;
- $ETT(j,z)$: is the expected transmission time related to link (j,z);

$$wp = \sum_{\forall i \in P} \sum_{\forall j \in N_i} \sum_{\forall z \in N_j} \left(Pk_{Q_j}(z) * ETT(j,z) \right) \tag{4.10}$$

The outer summation is to consider all the nodes which compose the route; instead the second summation considers all the neighboring node of the generic node *"i"* which belongs to the route; the inner summation is to take into account all the *"z"* neighboring nodes of node *"j"*, in fact in this way we consider the right next hop destination for packet queued in the node *"j"*.

Following this metric function we evaluate for each packet of node *"j"* the right ETT value and to weigh the interference of *"j"* we make the sum of the ETT values calculated for each packet.

This metric can be considered as an evaluation of the interference metrics collected by literature and in the following of this chapter we test the behavior of DIM in a mesh scenario comparing its performance with other interference based metrics.

4.4 Simulation scenario

The proposed metric DIM is evaluated in a WiMAX scenario and a performance comparison with a set of interference based metrics is realized. To test the metrics we implement them in a WiMAX simulator developed in JAVA.

The simulator is constituted by three elements the MAC layer, the PHY layer and also the channel error model. The channel error model is used in order to take into account into the simulation the impairment effect of the channel and thus in order to make real the possibility that a packet can arrive to the receiver in a corrupted way. The presence of a channel error model contributes to make more realistic the simulation scenario.

To implement a channel error model we use the IWPM model presented in chapter 2 and it is interesting to note that the presence of this model can simplify the software simulation. In fact without the presence of the IWPM generative model we would be forced to implement all the impairment effect in the simulator and consequently, to evaluate their effect on the single packet, we would have to process the packet bit by bit. The result of this complex process is a long processing time for a single sent packet.

The presence of the IWPM allows to consider a packet as a single entity which can be processed in a single step. The simulated scenario is a square area of $5km * 5km$ and the blue lines represent the active links among the nodes. In the simulator, to evaluate the metric in a scenario which is as realistic as possible and to allow the creation of routes with different qualitative characteristics, a different channel error model for each link is introduced. In particular, we configure an IWPM generative model for each link, in this way each IWPM is able to extract two values:

- a mean value for the probability to receive a wrong packet;
- a standard deviation for the same probability value;

this pair of values is used to set a distribution and from this distribution, instant by instant and for each packet, a value for the packet loss probability for the particular link can be extracted. In figure 4.3 it is represented the simulated scenario with the described process for the link *(15, 16)*.

In order to evaluate the metrics we realize and implement in the simulator an on demand routing algorithm. When a set of packets is present in the queue of a node, the node starts the process of route discovery invoking the routing

Table 4.1. Simulation settings

SIMULATION SETTINGS		
PHY SETTINGS		
Modulation		OFDM, QPSK 1/2
BW(Channel Bandwidth)		25Mhz
NFFT		256
G		1/8
Frame length		20 ms
Symbol efficiency		184 bits
Coverage radius		500 m
MAC SETTINGS		
msh-ctrl-len		4
msh-dsch-num		4
msh-csch-data-fraction		0
scheduling-frame		1
data queue size		50
SOURCES SETTINGS		
number of sources		3 - 24
QoS delay constraints		
priority: 1		40 ms
priority: 2		80 ms
priority: 3		/
Sources rate (CBR)		
	packet size	packets/s
priority: 1	64 bytes	128
priority: 2	2500 bytes	25
priority: 3	2500 bytes	125
Simulation run duration		500 s
number of runs / configuration		10
confidence interval		95%

algorithm. The routing algorithm individuates and signals to the invoking node the presence of a set of usable routes to reach the particular required destination, finally the node can choose, among the set of discovered routes, the most promising one accordingly to the route selection metric.

When the node needs to make a new request for a new data flow, even to the same destination, it invokes again the routing algorithm and evaluate again the route selection metric. This because the network configuration can be changed from the previous evaluation instant. In order to calculate the metric function, each node evaluates the weights for the interference as explained in section 4.3 and to inform the neighboring nodes about its weights values, each node send them in the configuration messages: MSH-NCFG and MSH-DSCH. In this way the overhead introduced to evaluate the weights are very little and

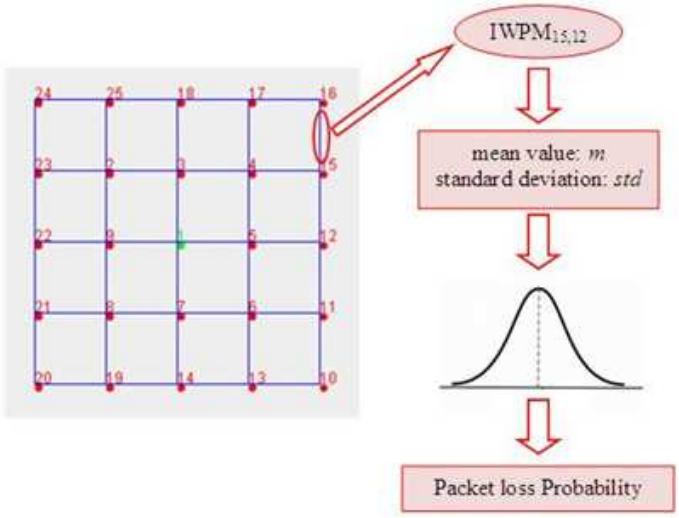

Fig. 4.3. Simulated scenario

corresponds only to a field introduced in the MSH-NCFG and MSH-DSCH messages.

In table 4.1 the simulation settings are summarized in particular it is possible to note the MAC parameters settings, the PHY settings and also the source settings. A set of three traffic class with different priority values and delay constraint are considered.

4.5 DIM Performance evaluations

In a first step, to evaluate the DIM performances, we compare DIM with other two metrics based on the interference and with the minimum hop count metric (MIN-HOP). The two metrics interference based are the metrics introduced in section 4.2 and analytically expressed by the equations (4.1) and (4.4), we identify these metric with B and Bqueue respectively. Both the metrics take into account the interference of the 1 hop neighborhood but as adding the Bqueue weights the interference with the number of packet queued in the nodes belonging to the route.

In the figures 4.4, 4.5 and 4.6 the percentage throughput behavior for the four compared metrics is illustrated. The percentage throughput represents the percentage of the transmitted packets which arrives at the receiver in a correct way; it is necessary to take in mind that a packets is not successfully received if it is deteriorated by channel impairment effects but also if the packet is deleted from data queue due to the network congestion. The figures 4.4, 4.5 and 4.6 consider the three traffic classes allowed in the simulated scenario.

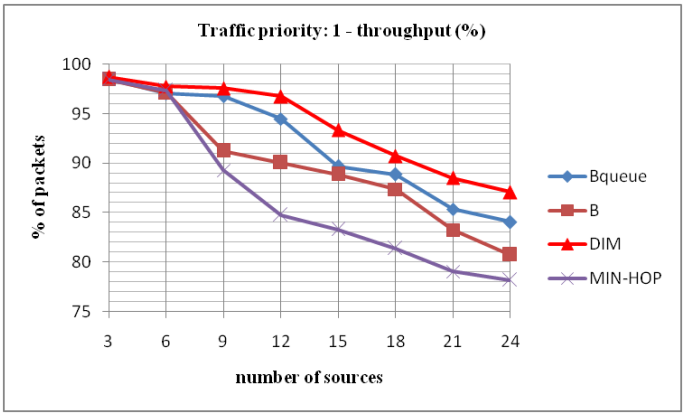

Fig. 4.4. Throughput for traffic classes with priority value equal to "1"

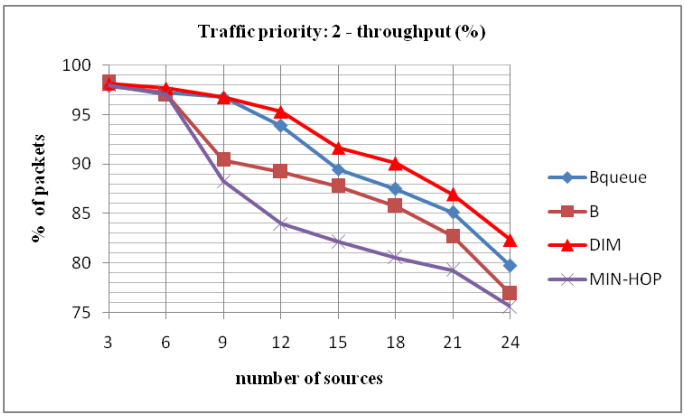

Fig. 4.5. Throughput for traffic classes with priority value equal to "2"

In particular the figure 4.4 represents the percentage throughput related to the traffic with priority equal to *"1"* and figure 4.5 instead is related to traffic with priority equal to *"2"*: in these two figures it is visible the best behavior of the proposed DIM, the worst case is related to MIN-HOP metric. The metrics performances decrease if we consider in the order the following metrics: DIM, Bqueue, B and MIN-HOP, this behavior is clear and it happen because the DIM metric is the only metric which considers the interference weighed with a value that take into account the real situation of the network; the Bqueue take into account the interference but it is weighed with the queued packets but neglects the link quality; B instead, consider only the interference based on the network topology; finally the MIN-HOP metric selects the shortest route without interference or link quality considerations. In this way following the path from DIM to MIN-HOP we loss the awareness of the real state of the

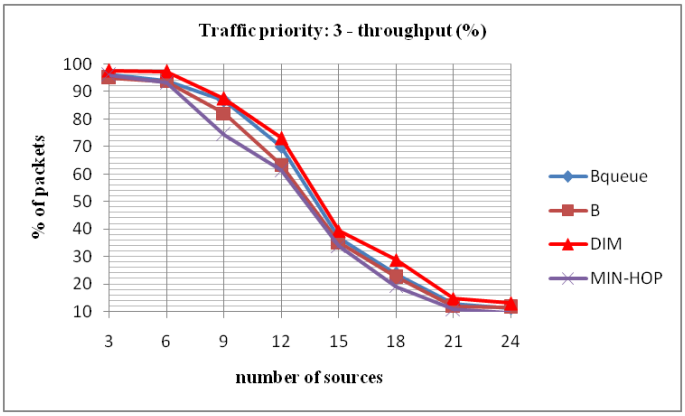

Fig. 4.6. Throughput for traffic classes with priority value equal to "3"

network. Observing the figure 4.6 the throughput for the traffic with priority equal to *"3"* is represented. Also in this case there is an advantage for the DIM but it is not very clear, this is due to the fact that this kind of traffic is neglected by the allocation and call admission control algorithms. There is not guarantee for the priority *"3"* traffic and this is visible also in metric behavior of figure 4.6.

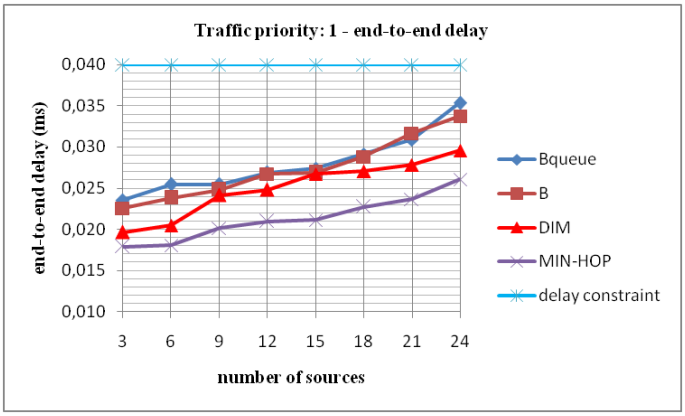

Fig. 4.7. End-to-end delay for traffic classes with priority value equal to "1"

In the figure 4.7, 4.8 and 4.9 the end-to-end delay for the four metrics and related to the three traffic classes are depicted. The performance results depicted in these figures is very interesting, in fact the best behavior is obtained by the MIN-HOP metric; this fact can be motivated in a very simple way, in fact the MIN-HOP metric selects always the shortest route and conse-

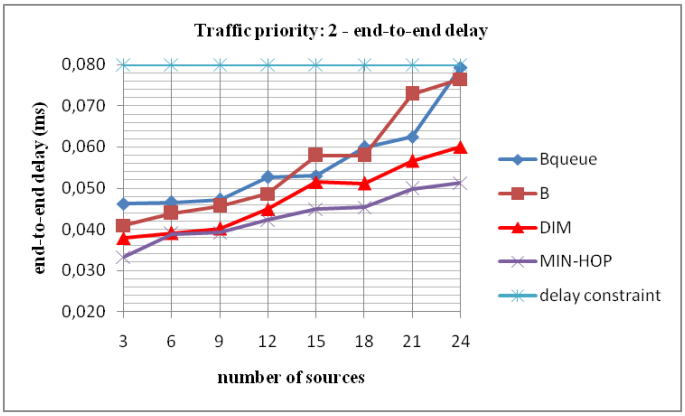

Fig. 4.8. End-to-end delay for traffic classes with priority value equal to "2"

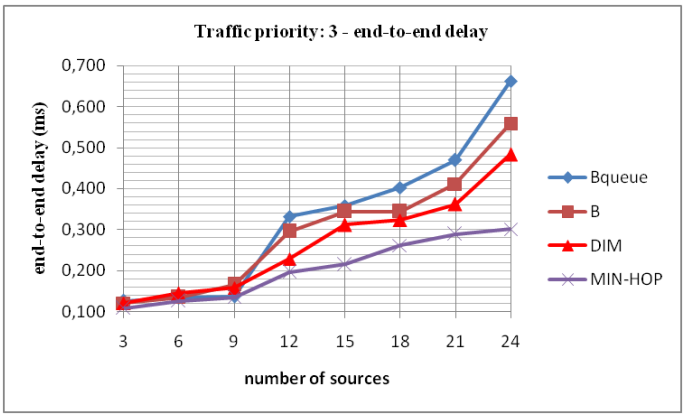

Fig. 4.9. End-to-end delay for traffic classes with priority value equal to "3"

quently even if this selected route allows a little throughput value, the packets are received at destination in the smallest end-to-end delay. The worst cases is related to Bqueue and B metrics which do not take into account neither the path length concepts nor the capacity of the mesh links. These results are visible in all the three figures. It is interesting to note that in figures 4.7 and 4.8 also the delay constraints for the two traffic classes are depicted and following these constraints we can say that all the metric, efficiently helped by allocation and call admission control algorithms, allow to respect the imposed constraints. Summarizing the introduced results we can conclude that the DIM metric allows to obtain the best performances in throughput focus and an acceptable behavior related to the delay constraints.

This results, obviously, is bound to the set of metrics selected to make the comparison. The DIM is compared with metric related to the interference

concept and what happens if we compare DIM with a metric not based on interference concepts? To reply to this question we have compared DIM with the ETT metric. ETT and DIM metrics are based on two different aspect of a network.

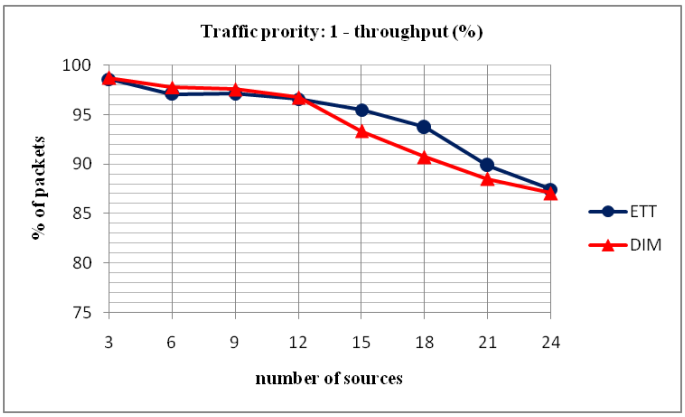

Fig. 4.10. Throughput for traffic classes with priority value equal to "1"

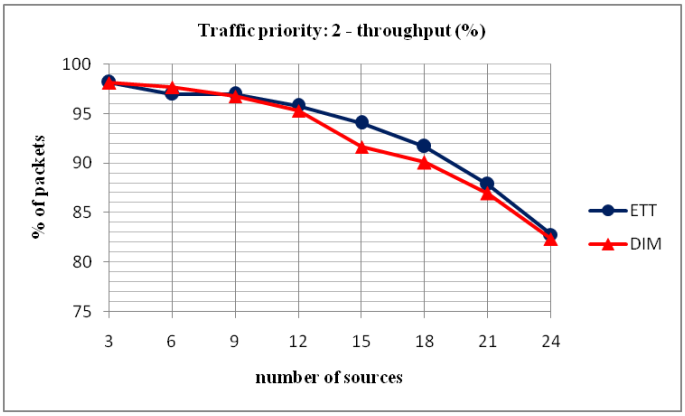

Fig. 4.11. Throughput for traffic classes with priority value equal to "2"

ETT considers the expected time to successfully transmit a packet on the selected route and in this way takes into account the quality of each link of the route using bandwidth, packet size and packet loss rate, in a practice way ETT estimate the quality of the route itself. The DIM metric instead, evaluate the interference of the route on the neighboring nodes; in order to evaluate the interference, it considers the quality of network links interfering

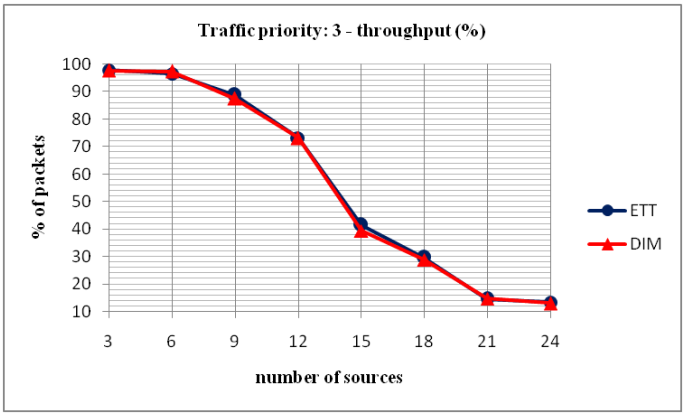

Fig. 4.12. Throughput for traffic classes with priority value equal to "3"

with the link belonging to the selected route, but DIM neglects to evaluate the quality of the route itself. The different concepts underlying the two metrics can represent an advantage for the ETT metric. In an ideal network, where the channel impairment effect can be neglected, probably the best behavior can be associated to the DIM metric, but as is visible in the figures 4.10 and 4.11 the best behavior is obtained by ETT metric. These figures with the figure 4.12 represent the percentage throughput for traffic classes with priority equal to *"1"*, *"2"* and *"3"* respectively. The throughput trends depicted for the two metrics in figure 4.12 are practically the same. Also the trends of end-to-end delay depicted in the figures 4.13, 4.14 and 4.15 are in favor of the ETT metric.

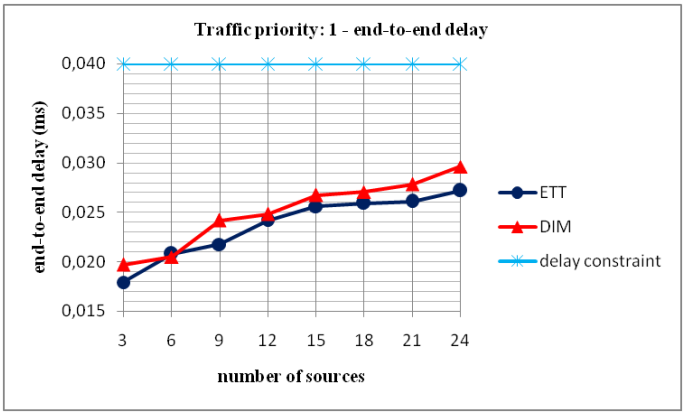

Fig. 4.13. End-to-end delay for traffic classes with priority value equal to "1"

Fig. 4.14. End-to-end delay for traffic classes with priority value equal to "2"

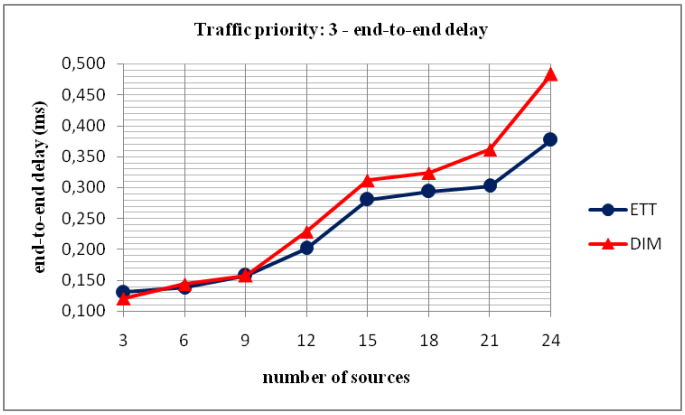

Fig. 4.15. End-to-end delay for traffic classes with priority value equal to "3"

4.6 DIEM: An improvement of DIM

The DIM metric, built on the concept of interference, allow to reach good throughput performance if compared with other interference based metrics. If compared with the ETT metric, the advantage of DIM is not true, because as we have explained, the ETT metric considers the real quality (in term of bandwidth and packet loss rate) of the selected route.

Observing the performance evaluation of DIM and ETT we found the idea to improve the DIM performance. The basic idea is to create a metric which sums the two basic concepts of the two metrics in order to exploit the advantage of each one.

The new metric is identified as DIEM (Delivering time Interference and ETT based Metric) and is defined by a set of equations. If we consider a mesh

node S which want to select a route to reach the destination node D, we define:

- $R_{S,D}$: the set of routes from S to D;
- wp_r: the weight of route r belonging to $R_{S,D}$ and evaluated with DIM metric using the equation (4.10);
- ETT_r: ETT value for route r belonging to $R_{S,D}$ and evaluated using the equation (4.9).

Using the previous parameters, we can compute the DEIM weight of the route r as:

$$wdiem_r = \frac{wp_r}{wp_{max}} + \frac{ETT_r}{ETT_{max}} \tag{4.11}$$

where wp_{max} and ETT_{max} are defined by the following:

$$wp_{max} = \max\{wp_r | r \in R_{S,D}\} \tag{4.12}$$

$$ETT_{max} = \max\{ETT_r | r \in R_{S,D}\} \tag{4.13}$$

In practice, the DIEM metric function is designed by a weighed sum of the two contributes of ETT and DIM metrics. The two contributes are normalized by the maximum value in order to make that each value of the sum can contribute to the final value with the same contribute. The first term of equation (4.11) takes into account the interference concept, instead the second term considers the quality of the route itself. This last term is neglected in DIM (quality in DIM is considered only for the neighboring nodes).

4.7 DIEM Performance evaluations

To evaluate the improvements obtained with DIEM metric, we make a comparison among ETT, DIM and DIEM metrics. The three metrics are compared in the same scenario considered for performance evaluation of section 4.5. Also the performance parameters are the same of section 4.5, in fact we consider percentage throughput and end-to-end delay.

The figures 4.16, 4.17 and 4.18 illustrate the percentage throughput related to traffic classes with priority equal to *"1"*,*"2"* and *"3"* respectively. In all these three figures the best behavior of DIEM metric appears very evident and clear. The greater percentage throughput related to DIEM metric is obtained thanks to the dual nature of the metric.

The improvements of DIEM are confirmed also in end-to-end delay represented in the figure 4.19 but not in the figures 4.20 and 4.21. The high performances of DIEM in terms of throughput are paid in end-to-end delay results. Overall, taking into account all the results and the fact that DIEM always respects the delay constraints imposed by applications, we can say that the best choice among a set of routes can be reached using the DIEM metric.

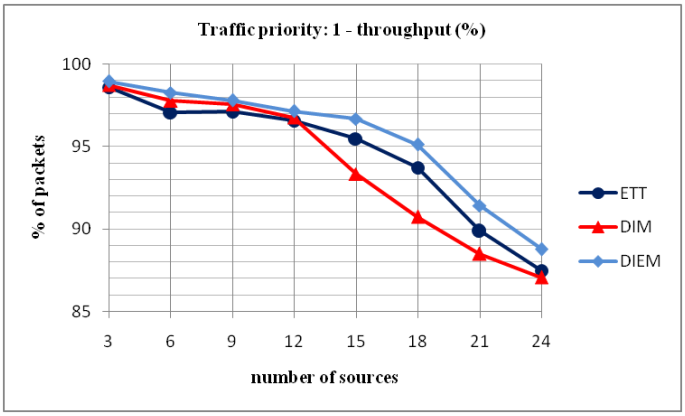

Fig. 4.16. Throughput for traffic classes with priority value equal to "1"

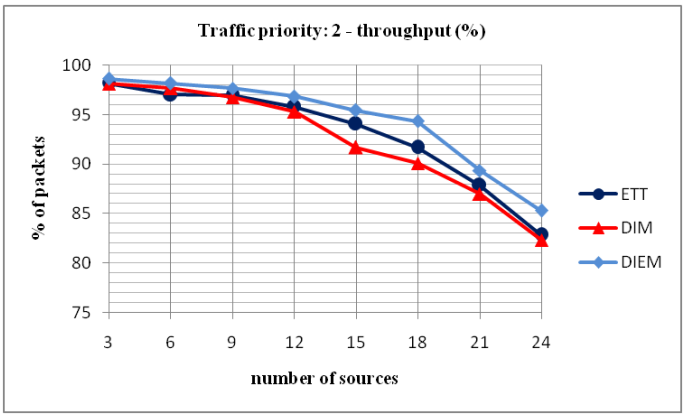

Fig. 4.17. Throughput for traffic classes with priority value equal to "2"

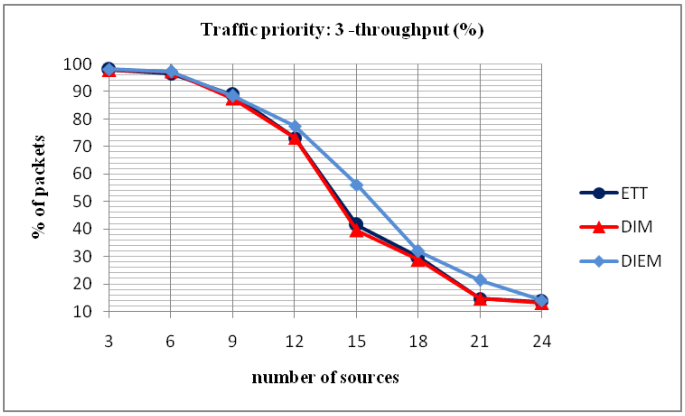

Fig. 4.18. Throughput for traffic classes with priority value equal to "3"

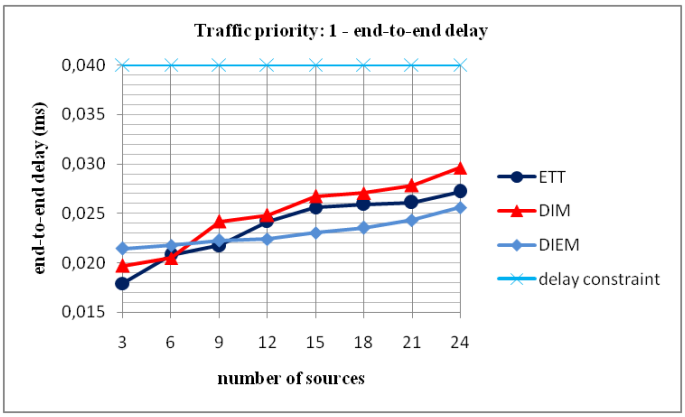

Fig. 4.19. End-to-end delay for traffic classes with priority value equal to "1"

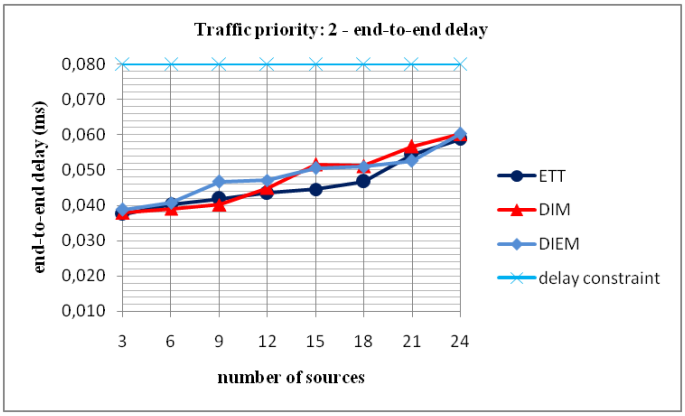

Fig. 4.20. End-to-end delay for traffic classes with priority value equal to "2"

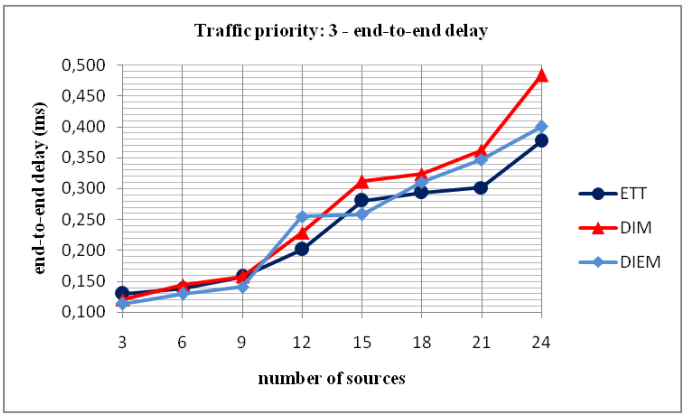

Fig. 4.21. End-to-end delay for traffic classes with priority value equal to "3"

5

A framework to support the quality of service

5.1 Introduction

In the previous chapters we have introduced a set of solutions to obtain a particular objective. In particular the chapter 2 presents a performance analysis of channel error models and also new models are introduced with the focus to create a generative model for the channel error behavior and to design a support to quality of service. In the third and fourth chapters instead we make analysis related to the MAC layer and in particular we have developed a call admission control algorithm and a metric function for the route selection. Both these last two solutions have the focus to support QoS and to guarantee high throughput values for the considered WiMAX mesh scenario.

However, the previous challenges are not been carried out in order to exist in isolation from each other, but the ultimate goal is to take all the developed solutions and ensure that these solutions can work together, with the aim of creating a cross-layer framework in which all solutions introduced can cooperate. The framework aims to support the quality of services.

Later in this chapter we will resume the solutions introduced so far, some of them will be further improved, others will be used in a different way than their initial presentation. Furthermore, new components will be presented. We would not take, as baseline of the goodness of our framework performances, any framework already presented in literature and this because would be difficult to implement a complex architecture in a manner faithful to the intentions of the authors; also to ensure that we do not affect the implementation with the presence of any of our additional mechanism it is an hard work. This obviously does not belittle our work, since we will attempt to show how the framework introduced is able to achieve better results than the case in which this framework is absent.

However, the literature presents a wide set of solution identifiable as QoS framework solutions for IEEE 802.16 architectures. An example of interesting works, for both PMP and mesh mode, are the following: [75] - [80].

5.2 QoS based traffic classification

The IEEE 802.16 protocol, operating in a PMP mode, provides a set of mechanisms to support the quality of services, one of this is the data flow mapping into four traffic classes. Each of these classes has a set of quality parameters associated to it, in this way, it is possible to classify each MAC PDU in the right traffic class, ensuring the compliance with the quality constraints associated to the data flow of belonging. Instead, in an IEEE 802.16 scenario operating in mesh mode, each mechanism defined in PMP mode ceases to exist. In the mesh mode the only claim made by protocol is that the QoS must be ensured packet by packet. As we have already introduced in the previous chapters, it is possible to replicate the PMP traffic class classification using the flags belonging to the header of the MAC PDU.

Also in the call admission control and metric simulation scenario we have considered a traffic classification, but in this case we want to create a more restrictive constraints for the traffic classes. In this way, the first element of the framework is the TT (Traffic Table). The elements stored in this table is summarized in table 5.1.

Table 5.1. Traffic Table

Priority value	Delay constraint (ms)
1	20
2	60
3	-

All the PDU can be mapped into three traffic classes and the framework is able to realize the classification using the flag of MAC header. We consider three traffic classes with admitted priority values equal to *"1"*, *"2"* or *"3"*. The priority value equal to *"1"* corresponds to the traffic with the highest priority, instead the priority value equal to *"3"* corresponds to the traffic with the lowest priority, finally the priority value equal to *"2"* correspond to a traffic class with a medium priority. We select as QoS constraint the end-to-end delay for each PDU. With end-to-end delay we consider, as an end-to-end delay, the time interval that elapses from the time instant in which the PDU appears in the queue of source node and the time instant in which it arrives in classifier of the destination node. The second column of table 5.1, thus, shows the higher bound for the end-to-end delay for the PDUs. In this case, to proof the framework quality, we introduce *"tighter"* quality constraints.

5.3 Call admission control and allocation algorithm

The second brick used to build the QoS framework, is a call admission control (CAC) algorithm. The CAC algorithm, used as admission maker when a mesh

node receive a request for a new data flow, is the GACD algorithm presented in chapter 3. This algorithm is inserted in the framework without introduce modification and this because it is developed to obtain high throughput values but also it is developed as a QoS-aware algorithm. In fact the GCAD algorithm, to admit a data flow take into consideration all the quality constraint defined by Traffic Table.

It is necessary to note that the GCAD algorithm contains also a very simple bandwidth allocation scheme. In fact when the new call is admitted, this allocation component merged in the admission algorithm, decide the amount of bandwidth to assign to the admitted call. To refresh this behavior we remember the following concepts:

- when GCAD receives a request for a data flow belonging to a traffic class with priority value equal to *"1"* or *"2"*, the GCAD, after various considerations and analysis, admits the new call if there is sufficient bandwidth to satisfy the request and consequently it grants to the new call the requested amount of bandwidth;
- when GCAD receives a request for data flow belonging to a traffic class with priority value equal to *"3"*, the GCAD algorithm verify the possible presence of the requested amount of bandwidth, if the amount requested is present then GCAD admits the new call and grants to it the request number of mini slots; otherwise, GCAD grants the residual available amount of bandwidth.

5.4 MSNEA: Mini Slot Number Estimation Algorithm

The MSNEA is a new component that allow to built an efficient framework. MSNEA is a Mini Slot Number Estimation Algorithm and its challenge is to determine the amount of bandwidth which a node needs. When a node, or for greater accuracy, the algorithm or agent designed with the task to take under observation the data queue, realizes that the data queue is not empty, there is the need to request bandwidth to send the data packets. In particular in the IEEE 802.16 mesh scenario the frame is divided into two part:

- control subframe;
- data subframe.

The data subframe is the only part of the frame used to transmit data packets and it is divided in a well defined number of mini slots. Consequently, when a node has to request bandwidth, has to evaluate the number of mini slots which is need.

The evaluation of this quantity can seem a simple concept but it is very important for two reasons:

- if the estimated number of mini slot is smaller than the number of which the node really needs then becomes difficult to ensure that there is not an

accumulation of data packets in the queue and also it is very hard to try to guarantee the respect of quality constraints;
- if the estimation number of mini slots is greater than the number of which the node really needs then there is a waste of bandwidth.

The previous concepts illustrate the importance of the presence of an efficient estimation algorithm. In the chapters 3 and 4 is used in the simulation scenario only a simple version of MSNEA. This older version is based only on a condition which allows the determination of requested mini slots number for the various traffic classes. The condition is expressed by the equation (3.1) for the traffic with priority value equal to *"1"* and *"2"* and by equation (3.3) for traffic with priority value equal to *"3"*.

In our framework we improve the old version of mini slots estimator and the new version is represented by the flow chart of figure 5.1. In addition to the flow chart it is necessary to provide a set of *"conditions"* which are used in it. Two of these are the previously mentioned conditions introduced in chapter 3 and for clearness we report them here. There is the need to define the following parameters:

- MS: OFDM symbols number for each minislot;
- p_{size}: packet size (bits);
- eff: efficiency of an OFDM symbol, expressed as number of data bits for each symbol;
- dl: delay constraint;
- d_{sym}: OFDM symbol duration (s);
- f: frame duration (s);
- h: path to destination hops count;
- t_i: arrival time of the first queued packet of BE traffic;
- t_f: arrival time of the last queued packet of BE traffic;
- n_{BE}: number of BE queued packets;
- p_{mean}: mean packet size of BE queued packets;
- R: estimated BE rate;

And with these parameters we can estimate the *nms* request (number of minislot) for traffic with priority equal to *"3"* using the equations:

$$R = \left(\frac{p_{mean} * (n_{BE} - 1)}{t_f - t_i} \right) \tag{5.1}$$

$$nms = R * \left(\frac{f}{MS * eff} \right) \tag{5.2}$$

and the *nms* request for traffic with priority equal to *"1"* or *"2"* resolving the following:

$$(nms * MS * d_{sym}) + \left(\frac{p_{size} - (n * MS * eff)}{n * MS * eff} \right) * f = \frac{dl}{h} \tag{5.3}$$

Now our intention is to explain the behavior of MSNEA following the flow chart depicted in figure 5.1 and also using equations (5.1) and (5.3) and the new equation:

$$\frac{dl}{h} - (t_{now} - t_{last}) \leq \left(\frac{total\_byte * 8 * f}{nms * MS * eff} \right) \tag{5.4}$$

which is defined using these parameters:

- t_{now}: time instant in which takes place the calculation;
- t_{last}: time instant corresponding at the arrive of the last queued packet;
- *total_byte*: total bytes present in queue and referred to the same traffic class.

The first term of equation (5.4) is indicated in flow chart as *tneed* and the second term as *tactual*. Also this equation allows to calculate the *nms* request for traffic with priority equal to *"1"* or *"2"*. The use of this equation and of the other will be explained below. In the flow chart we indicate the equation (5.2) with the term *condition (1)*; the equation (5.3) as *condition (2)* and the new introduced equation (5.4) as *condition (3)*.

The MSNEA is invoked by the node at the instant in which the node has a MSH-DSCH to send, in this way MSNEA can evaluate the possibility to make a new bandwidth request for an existing data flow or for a new data flow. The first step made by algorithm is the extraction of the first PDU by data queue, we indicate this PDU as PDU1. The PDU1 belongs to a traffic class with a priority indicated as PDU1.priority; the MSNEA verifies the presence of an existing pending request for bandwidth for data flow with this priority. If exists a pending request then the MSNEA searches in the queue the presence of other PDU with a different priority value. If the MSNEA finds this PDU then verify also for this priority the presence of pending requests. If the algorithm finds a pending request then repeat the process for the last priority value. When the MSNEA does not find pending request, scans the data queue searching all PDUs with the priority equal to PDU1.priority. During the queue scan, the algorithm evaluates a set of parameters and in particular :

- *Start_time*: the arrival time instant of the first queued PDU with priority equal to PDU1.priority;
- *End_time*: the arrival time instant of the last queued PDU with priority equal to PDU1.priority;
- *Total_byte*: the total amount of byte related to all the PDUs queued with priority equal to PDU1.priority.
- *Card_pack*: is the number of queued PDUs with priority equal to PDU1.priority;
- *Interval*: is defined as

$$Interval = End\_time - Start\_time \tag{5.5}$$

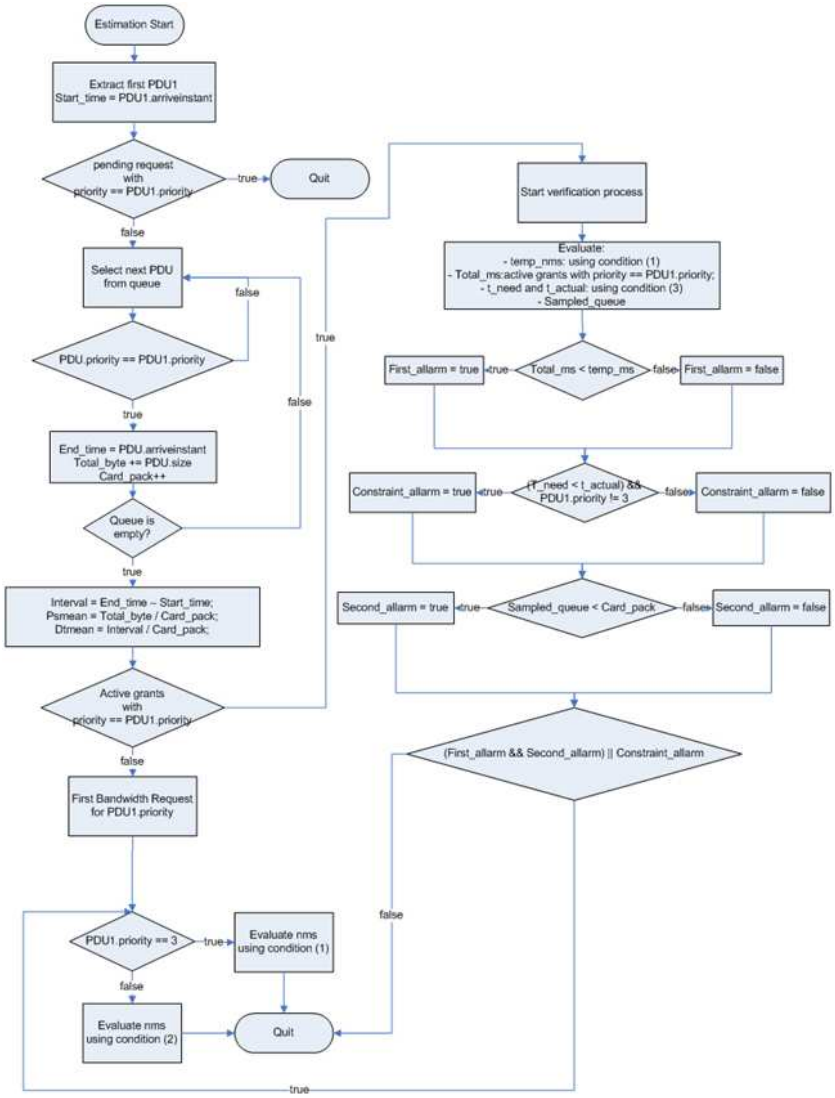

Fig. 5.1. Flow chart of MSNEA

It represents the time interval elapsed between the two arrival time instants of the first and last queued PDUs with priority equal to PDU1.priority.

- *PSmean*: is defined as

$$PSmean = \frac{total\_byte}{Card\_pack} \tag{5.6}$$

it represents an estimation of mean packet size.

- *Dtmean*: is defined as

$$Dtmean = \frac{Interval}{Card\_pack} \tag{5.7}$$

  it represents an estimation of time rate of queued PDU with priority equal to PDU1.priority.

At this point, if the MSNEA does not find already active grants for data flow with priority equal to PDU1.priority, it has to make the first request and it has to estimate the number of mini slots *(nms)* on the basis of PDU1.priority. If the PDU1.priority is equal to *"3"*, then the estimation is made by *condition (1)* i.e. by equation (5.2), instead if PDU1.priority is equal to *"1"* or *"2"*, then the estimation uses the *condition (2)*, i.e. the equation (5.3). Otherwise, if the node has an active grant for the same priority the MSNEA has to evaluate a set of condition to decide if it is necessary to make a new request. To support the decision, the algorithm evaluates the state of three Boolean variables: *First_alarm, Second_alarm* and *Constraint_alarm*. To assign a value to these variables the algorithm elaborates the following parameters:

- *temp_nms*: a first estimation of the number of slots that the node needs to deliver the queued PDUs with priority equal to PDU1.priority, it is make by *condition (1)*;
- *Total_ms*: is the number of mini slots that the node already has;
- *tneed*: it represents the time interval necessary to deliver to destination the last queued PDU, with priority equal to PDU1.priority, respecting its delay constraint;
- *tactual*: the node, using the mini slots previously granted to it for the PDUs with priority equal to PDU1.priority, has an available time interval to deliver the queued PDUs, this time interval is *tactual*; *tneed* and *tactual* are the first and the second term of equation (5.4) respectively;
- *Sampled_queue*: to verify if there is the need to request another amount of mini slots, the MSNEA sample the data queue length, the sampling takes place when the node receives a new grant; this parameter indicates the last sampled value.

The first variable that the MSNEA considers is the *First_alarm*, this variable is set with *"true"* if the *Total_ms* is smaller than the *temp_ms* and this means that the number of mini slots owned by the node is not sufficient to deliver all the queued PDUs with priority equal to PDU1.priority. This condition represents a first alarm for the MSNEA. The second evaluated variable is the *Constraint_alarm*, it is considered only for PDU with priority equal to *"1"* or *"2"* and is set to true if the mini slots previously granted to node it is not sufficient to guarantee the compliance with the delay constraint. The last variable is *Second_alarm* and it is set to *true* if comparing the actual value of queue length it is greater than the *Sampled_queue* parameter; this means that the previously granted mini slots are not suffient to guarantee the disposal

of queued PDUs, i.e. if there is not a new grant then there is a continuous accumulation of PDUs in the queue.

The last verification is useful to understand if it is necessary to make a new estimation for a new request. For MSNEA, if *First_alarm* and *Second_alarm* are both true or the *Constraint_alarm* is *true* then it is necessary to make a new request and the *nms* is evaluated using the *condition (2)* or *condition (1)* on the basis of PDU1.priority. it is interesting to note that if *First_alarm* and *Second_alarm* are both *false* then if *Constraint_alarm* is *true* then MSNEA establish to make a new request, this mean that the number of mini slots previously granted to the node are sufficient to dispose the queued PDUs but are not sufficient to guarantee the compliance with the delay constraint. The *Constraint_alarm* is the strongest condition to decide for a new request. It is necessary to clarify that for PDUs with priority equal to *"1"* or *"2"* the mini slot number estimation is made using the *condition (2)* and no the *condition (3)*, as we can see the *condition (3)* is used only to set the alarm variable, this because the estimation made by *condition (2)* is smaller than the value obtained by *condition (3)* and thus we obtain a conservative estimation. In this way we want to say that it is better to make a new estimation that probably will be insufficient and not to make a request that obtain as results a waste of bandwidth; to remediate to insufficient bandwidth there is the possibility to test again subsequently the needs of a new request.

5.5 PADIEM: Priority Aware Delivering time Interference and ETT based Metric

In order to support the route selection, in scenarios in which there is a set of multiple routes to reaches the destination node, we introduce a metric that is able to assign a weight to each route. The introduced metric function is an improvement of metric DIEM illustrated in chapter 4. The DIEM metric is designed with the goal to obtain high throughput values neglecting each concept of traffic classes, instead here we present the improved version of DIEM: PADIEM (Priority Aware Delivering time Interference and ETT based Metric) which takes into account also the QoS concepts weighing the queued packets with the weights associated to each traffic class. In this way is interesting to consider that:

- we weight the interference of a neighboring node *"n"* with the time necessary to deliver its queued data to the neighboring nodes;
- the time to dispose a data packet is evaluated by expected transmission time;
- we weight the necessary time to deliver a packet with the weight of traffic class of packet itself;
- the route is selected following two ideas: we want the path with lesser interference value and with the lesser ETT value;

- selecting the route with the lesser interference means that we are going to interfere with the set of nodes that needs less time to dispose their traffic and furthermore, this set of nodes has associated to it the traffic with lower priority.

The metric function can be expressed in analytic way using the following parameters:

- P: set of nodes belonging to a route;
- w_p: weight of the route build on the set P;
- N_i: set of neighboring nodes of node *"i"*;
- Q_j: set of queued packets at node *"j"*;
- $Pk_{Q_j}^{tp}(z)$: number of packet in Q_j with node *"z"* as next hop node and belonging to a traffic class with priority equal to *"tp"*;
- w_t(tp): it is the weight of traffic class with priority value equal to *"tp"*;
- ETT(j,z): is the expected transmission time related to link *(j,z)*;

$$w_p = \sum_{\forall i \in P} \sum_{\forall j \in N_i} \sum_{\forall z \in N_j} \left(Pk_{Q_j}^{tp}(z) * ETT(j,z) * w_t(tp) \right) \tag{5.8}$$

For more clearness, we repeat here some concepts introduced in the fourth chapter: the outer summation is to consider all the nodes which compose the route; instead the second summation considers all the neighboring node of the generic node *"i"* which belongs to the route; the inner summation is to take into account all the *"z"* neighboring nodes of node *"j"*, in fact in this way we consider the right next hop destination for packet queued in the node *"j"*. Following this metric function we evaluate for each packet of node *"j"* the right ETT value, in order to weigh the interference of *"j"* we make the sum of the ETT values calculated for each packet and each time value is weighed with a value associated with the traffic class of the packet itself. Finally the value associated by the PADIEM metric to a particular route can be obtained by the following equation:

$$wpadiem_p = \frac{wp_p}{wp_{max}} + \frac{ETT_p}{ETT_{max}} \tag{5.9}$$

where:

- $R_{S,D}$ is the set of routes from S to D;
- w_p is the weight of route p belonging to $R_{S,D}$ and evaluated with equation (5.8);
- ETT_p: ETT value for route p belonging to $R_{S,D}$ and evaluated using the equation (4.9).

where wp_{max} and ETT_{max} are defined by the following:

$$wp_{max} = \max\{w_p | p \in R_{S,D}\} \tag{5.10}$$

$$ETT_{max} = \max\{ETT_p | p \in R_{S,D}\} \tag{5.11}$$

At this point we want to explain the concepts related to this new metric using a simple example. In the figure 5.2 a simple mesh scenario is depicted, we consider a source node *"S"* which has to individuate an "interesting" route to reach the destination node *"D"*. Each nodes of the network has a couple of values associated to it:

- Q represents the number of data packets that are present in the queue of node;
- P represents the priority value associated to the queued packets.

For simplicity we consider:

- each node has only one type of traffic which must be delivered;
- an ideal channel behavior, i.e. we suppose that the packet error rate is equal to zero;
- the packet size and the link bandwidth is the same for each link.

Defining the previous simplifications, we want to put the attention only on the mean of introduction of the weight for the priority.

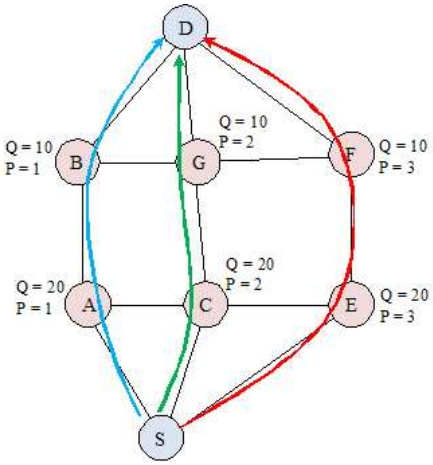

Fig. 5.2. Example of route selection

Consider for example the three route highlighted by the three colored arrows and for each path we can compute the w_p values using the equation (5.8). The w_p values, allow to calculate the first term of equation (5.9) and it is the contribute related to the interference; in our example we consider only this contribute to highlight the difference among the DIEM, DIM and PADIEM metrics. DIEM is an improvement of DIM and adds the evaluation

of the ETT value for the selected path, but both these two metrics use the same relation to evaluate the interference contribution. We indicate with *wpd* the interference contribute evaluated by DIM or DIEM metrics and they are used to make a comparison among the metrics. We use the following weights for the three traffic classes: 20, 10 and 1 for traffic priority values equal to *"1"*, *"2"* and *"3"* respectively. Now we present the three routes and the calculation of the interference due to the neighboring nodes and the queued packets:

- $Route(1) : S - A - B - D$
  - The node A interferes with the nodes C and D; interference contributes:
    $C : 20 * 10 = 200$;
    $D : 10 * 20 = 200$;
  - The node B interferes with A and G; interference contributes:
    $A : 20 * 20 = 400$;
    $G : 10 * 10 = 100$;

  $wp(1) = 200 + 200 + 400 + 100 = 900$;
  $wpd(1) = 20 + 10 + 20 + 10 = 60$;
  (obviously we do not consider the S/B coefficients because it is the same for each link).
- $Route(2) : S - C - G - D$
  - The node C interferes with the nodes A, G and E; interference contributes:
    $A : 20 * 20 = 400$;
    $G : 10 * 10 = 100$;
    $E : 20 * 1 = 20$;
  - The node G interferes with B, C and F; interference contributes:
    $B : 10 * 20 = 200$;
    $C : 20 * 10 = 200$;
    $F : 10 * 1 = 10$;

  $wp(2) = 400 + 100 + 20 + 200 + 200 + 10 = 930$;
  $wpd(2) = 20 + 10 + 20 + 10 + 20 + 10 = 90$;
- $Route(3) : S - E - F - D$
  - The node E interferes with the nodes C and F; interference contributes:
    $C : 20 * 10 = 200$
    $F : 10 * 1 = 10$
  - The node F interferes with the nodes G and E; interference contributes:
    $G : 10 * 10 = 100$
    $E : 20 * 1 = 20$

  $wp(3) = 200 + 10 + 100 + 20 = 330$
  $wpd(3) = 20 + 10 + 10 + 20 = 60$.

PADIEM introduces the priority aware concept for the queued packets and following the previous w_p calculations, the best choice is obtained selecting the *Route (3)* which presents the smallest value of w_p. Selecting the *Route (3)*, the new data flow will interfere only with a data flow with priority equal to *"2"* (queued packets in nodes C and G) and with the traffic of nodes belonging to the *Route (3)* itself (traffic of priority equal to *"3"*). Instead, the best choice based on DIEM or DIM metrics (obviously related only to interference evaluation) is, indifferently, the *Route (1)* or *(3)*. The advantage of PADIEM based choice is clear, in fact using the *Route (1)*, the new data flow will interfere with a data flow with priority equal to *"1"* (queued packets in nodes A and B) and with a data flow with priority equal to *"2"* (queued packets in node C and G). With this example, we have proved briefly and in a clear way the advantage of the new metric, the introduction of the priority aware concept, allows to reduce the interference with routes in which there are allocated data flows with priority value higher than the new data flow.

5.6 PSEA: Packet Size Estimation Algorithm

The Packet Size Estimation Algorithm (PSEA) algorithm is a new component that is not present in the simulation scenario of other chapters. It represents a useful mechanism designed to estimate the packet size value that has to be used at physical layer. The PSEA is strictly related to the physical layer of the protocol. We design this algorithm with the intent to increase the throughput of the network and to limit the PER (Packet Error Rate) of each mesh link.

In the chapter 4 we have introduced in the scenario simulation the presence of a channel error model, thus the presence of a generative model allow to consider the real characteristic of the channel: a packet transmitted on the channel, can be received in a correct or corrupted way and it is interesting to remember that a packet with a greater size value has a greater probability value that the packet can be received in a corrupted way. Using great values of packet size the throughput of the network can decrease but using little values of packet size can become very difficult try to guarantee the compliance with QoS constraints. The best choice for the packet size value is the right compromise between the two trends.

Using IWPM, presented in chapter 2, it is possible to design a simple algorithm to ensure compliance with QoS constraints, in this way IWPM represents the basis for the PSEA algorithm. In figure 5.3 the PSEA algorithm is presented by a flow chart. Each application can have its particular QoS constraints, and one of these constraints can be expressed in term of packet error rate (PER). The PER explains what is the percentage of bad packets that is allowed for the particular application and it can be considered as the admitted value for the probability to obtain a bad packet. For example a multimedia application allow a maximum PER value equal to 1%, this value can be seen in this way: the maximum probability to obtain a bad packet must

be equal to 0.01. The algorithm represented in figure 5.3 has a set of values as input. The input of algorithm are: the instant user speed (in our scenario we consider only fixed user, consequently the user speed is equal to zero), the actual packet size value and the QoS constraint P_a which is, as previously explained, the maximum admitted value for the probability to receive a bad packet. The first two values become the IWPM input and P_B will be the output. The subsequent step is the calculation of DC value that is evaluated by:

$$DC = P_a - P_B \tag{5.12}$$

If P_B is equal or minor than P_a then the constraint is respected and the algorithm is not useful, otherwise we must decrease this value. P_B is influenced by speed *(v)* and packet size *(ps)* and obviously, to obtain our goal, we can operate only on packet size value. DC is the exceeding value. Three values must be calculated and these are:

- dP is evaluated as:

$$dP = \left(\frac{P_{v,216} - P_{v,6}}{6} \right) \tag{5.13}$$

  the interval $(P_{v,216} - P_{v,6})$ represents the probability value range considering a fixed speed value equal to *"v"*, and *"6"* is the number of sub-interval of IWPM, thus dP is the probability step for a single sub-interval;
- Dn is a parameter computed as:

$$Dn = \frac{DC}{dP} \tag{5.14}$$

  and it represents the number of time that P_a must be decreased by dP entity.
- Dps can be calculated by the following:

$$Dps = Dn * Sps \tag{5.15}$$

Sps is the packet sub-interval size ($Sps = 35byte$) of IWPM, and Dps is the needed packet size variation to ensure compliance with QoS constraint. Thus the final packet value is:

$$ps = ps + Dps \tag{5.16}$$

The Dps value obviously is negative if P_B is greater than P_a and the constraint is not respected, thus finally the packet size will be decreased. Obviously this algorithm can be used also if, at starting point, the P_B value is smaller the P_a. In this case we are respecting the bound and indeed we can optimize the data transmission increasing the packet size until the limit condition is not reached: $P_B = P_a$. in this way the Dps value is positive.

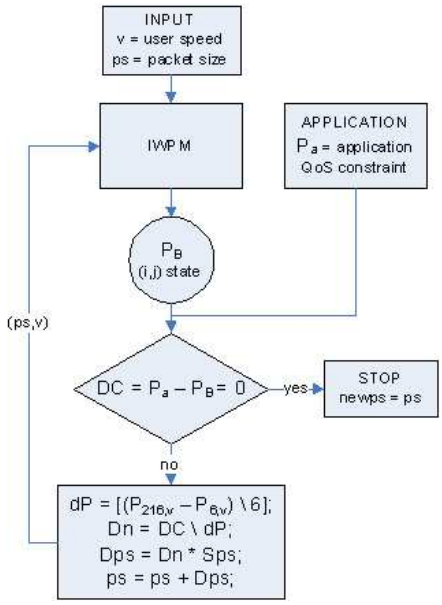

Fig. 5.3. PSEA flow chart

In this algorithm a simplification is applied, to calculate the packet size variation *(Dps)*, we hypothesize that the probability to have a bad packet, has a linear behavior as function of packet size. This is a simplification and the consequence is that the algorithm obtains the right value after a little number of iterations, i.e. the value of *"ps"* calculated by the algorithm does not respect the QoS constraint and thus there is need to make a new iteration of algorithm. The *"ps"* value converges to right value in a finite number of iterations, but the final value realize the condition: $P_B = P_a$ that is the optimal condition. For example considering: $P_a = 0.01$, $v = 78km/h$ and $ps = 200byte$ the algorithm calculates the right packet size value (at condition $P_B = P_a$) equal to 72 byte in 7 iterations. The PSEA algorithm as described here, highlights a particular aspect of our framework, in fact the presence of this algorithm which collaborates with other elements of MAC layer, means that the nature of framework is definable as cross-layer: considering each described element is increasingly afloat the cooperation between MAC and PHY layers of IEEE 802.16 protocol. In the subsequent section we will describe how the elements of framework work together.

5.7 Cross - Layer Framework Scheme

Brick by brick we have built the QoS oriented framework. In the figure 5.4, in order to illustrate the complete framework is made a representation of a

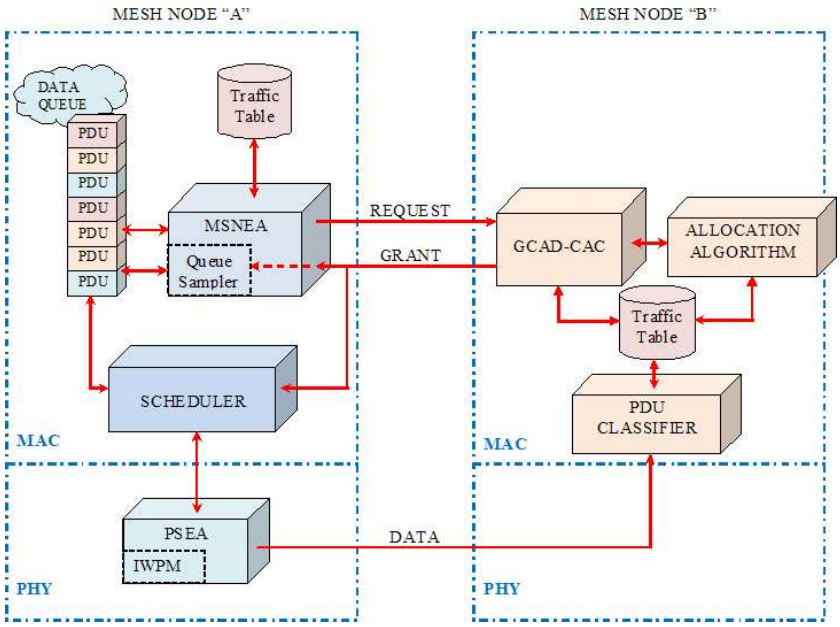

Fig. 5.4. Cooperation of framework elements

communication between two generic mesh nodes A and B. The two mesh nodes contain all the described elements but in particular, in the node A we have represented only the elements involved in the bandwidth request process, instead in the node B we can note the elements involved in the admission and allocation processes. Considering the mesh node A, which is the requester node and i.e. the node that make a new bandwidth request to the node B, we can see the MSNEA element. This element communicates with the Traffic Table and the Data Queue and also contains the Queue Sampler. The communication with the Traffic Table it is necessary to built the bandwidth request and to estimate the right value for the mini slots number estimation, in fact in section 5.4 we have introduced a set of conditions related to the particular traffic priority class. The task of Queue Sampler is to sample the queue length to understand if the actual amount of grants are sufficient to dispose the queued packet. When the mini slots number is estimated, the node A can send to the node B the bandwidth request and it wait for the grant. The node B receives the request and then it can evaluate the possibility to admit the new request and how much bandwidth grant to the node A. This task is performed by GCAD-CAC and Allocation algorithms; in order to complete their task, these two elements must have access to the Traffic Table, in fact the allocation and call admission control algorithms take into account in the traffic priority for the new request.

The node B, working in this way, is able to sent a grant to the node A; this grant can be positive or null, in the first case the new data flow is admitted instead in the last case the request is refused. The node A use the received grant to tell to the scheduler how to work. In this process participate also the PSEA element which decides the packet size value at PHY layer. Finally, in the node B it is possible to note the presence of the Classifier, its task is to classify the received packets, in this way, if the packet must be sent to another node, it is possible to start the process for a bandwidth request to reach the destination node, instead if the destination coincides with the node B itself then the packet is transferred to the upper protocol layer.

5.8 Performance Evaluations

Table 5.2. QOF and SOS characteristics

Characteristics	**QOF**	**SOS**
CAC	GCAD	GCAD
Request estimation	MSNEA	eq: 3.1 - 3.3
Route selection metric	PADIEM	DIEM
Packet size choice	PSEA	-
Traffic flow classification	TT	TT

To evaluate the framework performance, we have implemented all the algorithms described in this chapter in a IEEE 802.16 simulator realized with JAVA language. Our framework, which is indicated in the following as QoS Oriented Framework (QOF) (QoS Oriented Framework), is analyzed comparing its performance with the performance of an architecture built using as bricks all the old solutions presented in the previous chapters, this architecture is indicated as Set of Old Solutions (SOS). All the elements which characterize the two architectures are summarized in the table 5.2.

The simulation settings parameters are summarized in the table 5.3. In particular it is interesting to note the PSEA setting. For PSEA algorithm we set a PER bound for the traffic with priority equal to *"1"* and *"2"*, i.e. PSEA try to guarantee the PER value equal to 0.5% for traffic with priority equal to *"1"* and 2% for traffic with priority equal to *"2"*. A *"large"* bound is defined instead for the smallest priority traffic. To consider a realistic channel error behavior, we configure an IWPM generative model for each link, in this way each IWPM is able to extract two values:

- a mean value for the probability to receive a wrong packet;
- a standard deviation for the same probability value;

Table 5.3. Simulation settings

SIMULATION SETTINGS		
PHY SETTINGS		
Modulation		OFDM, QPSK 1/2
BW(Channel Bandwidth)		25Mhz
NFFT		256
G		1/8
Frame length		20 ms
Symbol efficiency		184 bits
Coverage radius		500 m
MAC SETTINGS		
msh-ctrl-len		4
msh-dsch-num		4
msh-csch-data-fraction		0
scheduling-frame		1
data queue size		50
PSEA SETTINGS		
		PER bound
priority: 1		0.5%
priority: 2		2%
priority: 3		-
SOURCES SETTINGS		
number of sources		3 - 24
QoS delay constraints		
priority: 1		20 ms
priority: 2		60 ms
priority: 3		-
Sources rate (CBR)		
	packet size	packets/s
priority: 1	64 bytes	128
priority: 2	2500 bytes	25
priority: 3	2500 bytes	125
Simulation run duration		500 s
number of runs / configuration		10
confidence interval		95%

this pair of values is used to set a distribution. A particular distribution is linked with a particular link, thus, from this distribution, instant by instant and for each packet, a value for the packet loss probability can be extracted.

In the figures 5.5, 5.6 and 5.7 the throughput trends, related to the 3 traffic classes, are depicted. In particular, in the figure 5.5 it is possible to observe the throughput of QOF and SOS for the traffic with priority equal to *"1"*. The QOF framework presents the best behavior, in fact increasing the number of sources in the scenario simulations, the throughput trend decreases very slowly, instead the SOS trend decrease in a rapid way. It is possible to

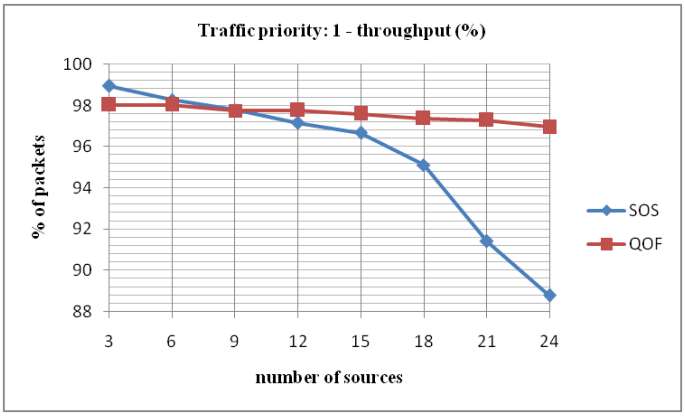

Fig. 5.5. Throughput for traffic classes with priority value equal to "1"

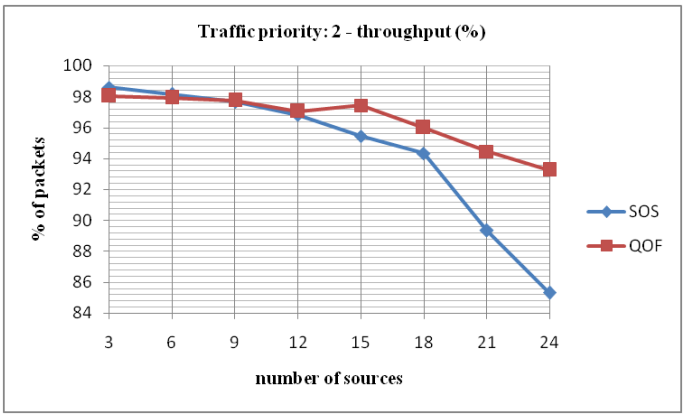

Fig. 5.6. Throughput for traffic classes with priority value equal to "2"

observe the same result in the figure 5.6 which is related to traffic class with priority equal to *"2"*. Instead, in figure 5.7 the two frameworks present approximately the same behavior and also they allow to reach the same percentage of successfully transmitted packets.

The best results of QOF framework can be explained by the presence of a set of improved mechanisms which cooperate together in a cross layer architecture in order to reach high values of throughput. In particular the presence of the PSEA, PADIEM and MSNEA allows to reach the best throughput results in a hard scenario characterized by a channel with a realistic behavior:

- PSEA allows to calculate the packet size at PHY layer to obtain a small value of PER, obviously decreasing the PER we can increase the percentage of successfully delivered packets. As expressed in table 5.3, the PSEA is set with values which try to advantage traffic with priority equal to *"1"*

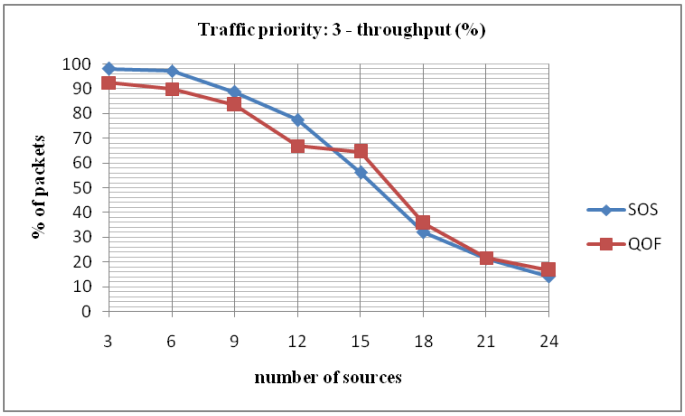

Fig. 5.7. Throughput for traffic classes with priority value equal to "3"

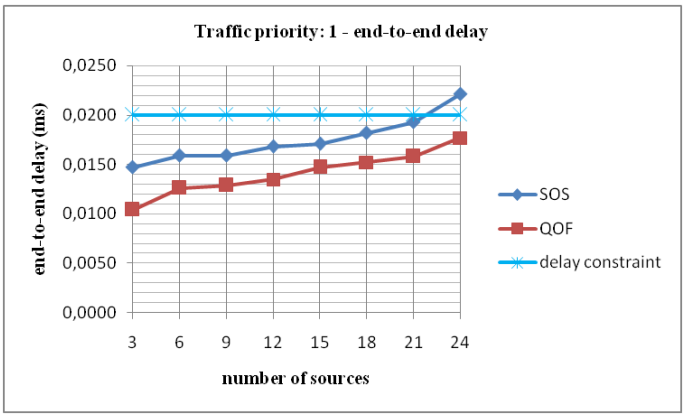

Fig. 5.8. End-to-end delay for traffic classes with priority value equal to "1"

or *"2"*, neglecting the traffic with the smallest priority and this is visible in the figure 5.7;

- PADIEM can choose an *"interesting"* route taking into account interference, link quality and traffic priority aware concepts;
- MSNEA allows to eliminate waste of bandwidth.

Interesting results can be seen also in the figures 5.8, 5.9 and 5.10. These figures, which illustrate the end-to-end delay for successfully transmitted packets, show the best performances of QOF framework. Also in this case all the framework elements contribute to obtain the depicted results and in particular the MSNEA allows to guarantee the compliance with the delay constraints. Summarizing, we can conclude that the set of solutions, each of which introduced in order to resolve a particular issue, involved in a QoS Oriented

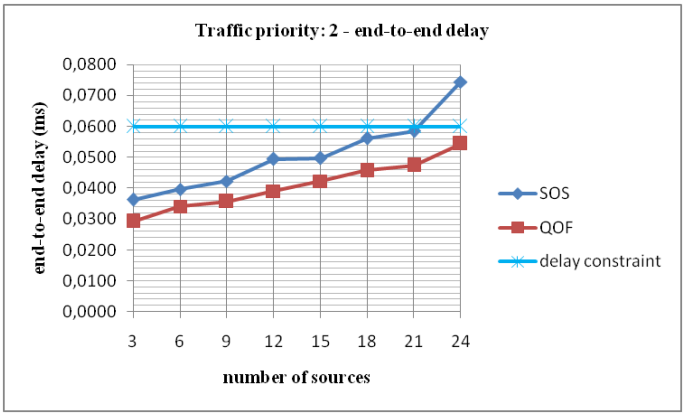

Fig. 5.9. End-to-end delay for traffic classes with priority value equal to "2"

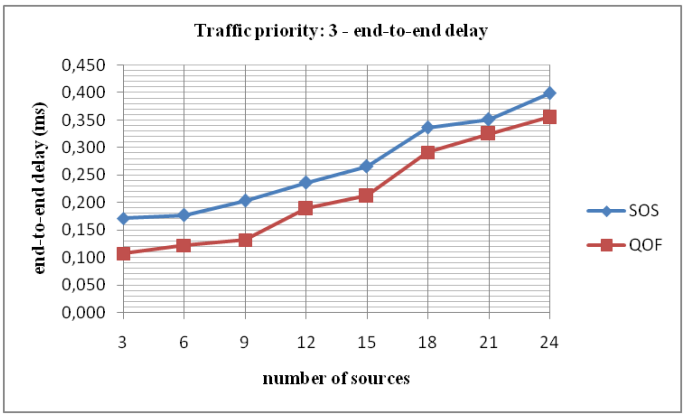

Fig. 5.10. End-to-end delay for traffic classes with priority value equal to "2"

Framework which work in a cross layer way, represents an interesting solution to create a network architecture that is able to guarantee excellent QoS levels.

Conclusions

The IEEE 802.16 protocol defines guidelines to provide wireless broadband services in a wide area. The protocol defines physical (PHY), medium access control (MAC) layer and also each management aspect; the first layer defines five air interfaces and the second one allows itself to be interfaced with IP (Internet Protocol) or ATM (Asynchronous Transfer Mode) upper layer protocol. In the last IEEE 802.16 version, precisely in IEEE 802.16e, also user mobility is contemplated. As previously stated, this protocol, commercially known as WiMAX (Worldwide Interoperability for Microwave Access), allows wireless multimedia services to be provided to a wide area. The installation of 802.16 wireless infrastructure, instead of creating a new fixed and wired infrastructure from scratch, brings to many benefits both economic and practical.

In order to contribute to develop of this new technology, in our work we treat a set of issues related to both PHY and MAC protocol layers. In particular, the first chapter is based on the channel behaviour analysis in a mobile WiMAX scenario. The analysis is conducted with a performance comparison between a set of Markov Chain based models collected by literature. The following models: MTA (Markov-based Trace Analysis), Gilbert - Elliot, FSM (Full-State Markov) and HMM (Hidden Markov Model) are designed using packet error traces (a sequence of *"1"* and *"0"*) obtained by a simulator that takes into account channel impairment effects such as path loss, Doppler effect and multipath fading. To compare the models performances, by each of them artificial traces are generated and then Entropy Normalized Kullback-Leibler distance, standard error and other statistical properties of random variable G (free error packets burst length) and B (corrupted packets burst length) of artificial traces are computed. The purpose of this work is not only to identify the model that best describes the channel error behavior in IEEE 802.16e scenario but also to create a new one. After models comparison, a hybrid model designed to achieve better performance in the artificial trace generation will be presented. This hybrid model presents the best performances and it is able to model both the B and G variables behavior and so the time variant

channel error behavior. With the Hybrid model we are be able to foresee the packet loss, and, thus, an action can be taken on certain parameters, such as the packet size or the available QoS, in order to maximize the throughput of the system. In literature there is a great diffusion of Markov chain based models to describe channel behavior, but no one of these is independent by scenario configurations, and also no one is tested with realistic dynamic scenario. As a consequence of this we have created a new model that can be used independently from the channel configuration.

The IWPM model, a new channel behavior model, is proposed, which presents these characteristics: the possibility of applying the model to a dynamic scenario in which, instant by instant, it is possible to know, depending on the circumstances of the scenario, what is the probability of having a bad packet. The IWPM model is presented in three different versions: as a function of one, two or three variables.

Subsequently our study are oriented on other issues related to a distributed mesh scenario. Thus in chapter three we have presented a new call admission control algorithm for 802.16 distributed mesh networks. The algorithm is characterized by an initial greedy choice and by preemption and defragmentation processes. The proposed algorithm is tested in a scenario of 25 mesh nodes with a max number of 24 sources. The performances of proposed GCAD algorithm are evaluated by throughput, average end-to-end delay and number of refused requests. The GCAD performances are compared with other two CAC algorithms. The GCAD algorithm presents the best performances reached through the presence of defragmentation process which allows an optimized management of minislots allocation.

A distributed wireless mesh network, probably represent the most promising and interesting architecture for a WiMAX network. It allows the direct connection among a set of subscriber stations and also it guarantee the scalability of the network. But the mesh distributed mode present also a set of fascinating challenges, the distributed call admission control issue treated in the chapter three is one of these, but another interesting one is the route selection. When a ode has to transmit data packets to a destination node, it need of a routing algorithm to individuate a route. The interesting fact is that a routing algorithm can individuate not only one route to reach the destination but a set of route; consequently there is the question: which route the source node choose? To select a route among a set of candidate routes can be used a function which assigns a weight to each route. The weight of route can be built considering different concept as the path length, the interference or the quality of links belonging to the route itself. In chapter four we present two metrics, the first is the DIM metric based on interference concepts and link quality. The DIM metric presents good results if compared with other interference metrics but does not present the best behavior if compared with the classical ETT metric. The DIEM metric is an improvement of DIM metric and eliminate the problem highlighted by DIM.

All the new solutions, designed in the first four chapters, allow to obtain very interesting results if compared with other existing solutions; furthermore all these elements can work together in order to allow, to a wireless distributed mesh network, to provide a set of service characterized by well defined QoS levels. All the developed solutions: the IWPM model, the call admission control algorithm, the route selection metric and other new mechanisms are collected in a QoS Oriented Framework (QOF) which is described in the chapter five. It represent an interesting example of cross layer framework which is able to support the QoS. The term cross-layer architecture refers to a specific architecture, which indicates a collaboration between two or more layers of protocol stack in order to achieve a common goal. In this case, the goal is to achieve certain levels of quality of service, so that customer expectations in terms of quality are met.

References

1. IEEE 802.16-2001. IEEE Standard for Local and metropolitan area networks, Part 16: Air Interface for Fixed Broadband Wireless Access Systems.
2. IEEE 802.16c-2002. IEEE Standard for Local and metropolitan area networks, Part 16: Air Interface for Fixed Broadband Wireless Access Systems - Amendment 1: detailed system profiles for 10-66 GHz.
3. IEEE 802.16a-2003. IEEE Standard for Local and metropolitan area networks, Part 16: Air Interface for Fixed Broadband Wireless Access Systems - amendment 2: medium access control modification and additional physical layer specifications for 2-11 GHz.
4. IEEE 802.16.2-2004. IEEE Recommended Practice for Local and metropolitan area networks, Coexistence of fixed broadband wireless access systems.
5. IEEE 802.16-2004. IEEE Standard for Local and metropolitan area networks, Part 16: Air Interface for Fixed Broadband Wireless Access Systems.
6. IEEE 802.16f-2005. IEEE Standard for Local and metropolitan area networks, Part 16: Air Interface for Fixed Broadband Wireless Access Systems - Amendment 1: Management Information Base.
7. IEEE 802.16e-2005. IEEE Standard for Local and metropolitan area networks, Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems - Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands and Corrigendum 1.
8. IEEE 802.16k-2007. IEEE Standard for Local and metropolitan area networks: Media Access Control (MAC) Bridges, Amendment 2: Bridging of IEEE 802.16.
9. IEEE 802.16g-2007. IEEE Standard for Local and metropolitan area networks, Part 16: Air Interface for Fixed Broadband Wireless Access Systems - Amendment 3:Management Plane Procedures and Services.
10. IEEE 802.16 Concormance01-2003. IEEE Standard for Conformance to IEEE 802.16, Part 1: Protocol Implementation Conformance Statement (PICS) Proforma for 10-66 GHz WirelessMan-SC air interface.

11. IEEE 802.16 Conformance02-2003. IEEE Standard for Conformance to IEEE 802.16, Part 2: Test Suite Structure and Test Purpose for 10-66 GHz wirelessMan-SC air interface.
12. IEEE 802.16 Conformance03-2004. IEEE Standard for Conformance to IEEE 802.16, Part 3: Radio Conformance Tests (RCT) for 10-66 GHz WirelessMAN-SC Air interface.
13. IEEE 802.16 Conformance04-2006. IEEE Standard for Conformance to IEEE 802.16, Part 4: Protocol Implementation Conformance Statement(PICS) Proforma for Frequencies below 11 GHz.
14. Jie Wu, "An extended dynamic source routing scheme in ad hoc wireless networks". Proceedings of the 35th Annual Hawaii International Conference on System Sciences, 2002. HICSS. 7-10 Jan 2002 Page(s):3832 - 3838.
15. Perkins C.E., Royer E.M., "Ad-hoc On-Demand Distance Vector Routing", Second IEEE Workshop on Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Publication Date: 25-26 Feb 1999, page(s): 90-100.
16. Perkins C.E., Bhagwat P., (1994) "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers", In Proceedings of ACM SIGCOMM Conference (SIGCOMM '94), pp. 234-244.
17. H. Shen Wang, N. Moayeri, "Finite-State Markov Channel-A Useful Model for Radio Communication Channels" IEEE Transactions on Vehicular Technology. Vol. 44, Issue: I, Pages: 163-171, February 1995.
18. J. P. Ebert, A. Willig, "A Gilbert-Elliot Bit Error Model and the Efficient Use in Packet Level Simulation". Tec. Report. Technical University Berlin. Telecommunication Networks Group.
19. A. Konrad, B. Y. Zhao, A. D. Joseph, R. Ludwig, "A Markov-Based Channel Model Algorithm for Wireless Networks". 4th Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems (MSWIM 2001). Pages 28-36, Rome, Italy, July 2001.
20. P. Bergamo, D. Maniezzo, A. Giovanardi, G. Mazzini, M. Zorzi, "An Improved Markov Chain Description for Fading Processes" University of Ferrara. IEEE ICC 2002. Vol. 3, Pages: 1347- 1351, August 2002.
21. A. Konrad, B. Y. Zhao, A. D. Joseph, "Determining model accuracy of network traces" Journal of Computer and System Sciences. Vol. 72, Issue: 7, Pages: 1156-1171, November 2006.
22. S. A. Khayam, H. Radha, "Markov based modeling of wireless local area networks". MSWiM'03, September 19, San Diego, California, USA. Pages: 100-107, 2003.
23. Q. Zhang and S.A. Kassam, "Finite-state Markov model for Rayleigh fading channels", IEEE Transaction on Communication. Vol. 47, Issue: 11, Pages: 1688-1692, 1999.
24. C. C. Tan, N. C. Beaulieu, "On First-Order Markov Modeling for the Rayleigh fading channel." IEEE Transactions on Communications. Vol. 48, Issue: 12, Pages: 2032-2040, December 2000.

25. R. Carruthers, N. C. Beaulieu, "On an Improved Markov Chain Model of the Rayleigh Fading Channel". Global Telecommunications Conference, 2007. GLOBECOM '07. IEEE 26-30 November 2007. Pages: 1529-1534, 2007.
26. R. Carruthers, N. C. Beaulieu, "A Quadrature Markov Chain Model of the Rayleigh Fading Channel". IEEE International Conference on Communications, 2008. ICC '08. 19-23 May 2008. Pages:1404-1409, 2008.
27. Geoffrey W.K. Colman, S. D. Blostein, N. C. Beaulieu, "An ARMA Multipath Fading Simulator". Wireless Personal Communications: Improving Capacity, Services and Reliability. Boston, MA: Kluwer, 1997.
28. X. Zhu, J. M. Kahn, "Markov Chain Model in Maximum-Likelihood Sequence Detection for Free-Space Optical Communication Through Atmospheric Turbulence Channels". IEEE Transactions on Communications. Vol. 51, Issue: 3, Pages: 509-516, March 2003.
29. C.-C. Chong, C.-M. Tan, D.I. Laurenson, S. McLaughlin, M.A. Beach, A.R. Nix, "A novel wideband dynamic directional indoor channel model based on a Markov process", IEEE Transactions on Wireless Communications. Vol. 4, Issue: 4, Pages:1539 - 1552, July 2005.
30. P. Kuczynski, A. Rigoll, W. H. Gerstacker, J. B. Huber, "Hidden Markov Modeling of Error Patterns and Soft Outputs for Simulation of Wideband CDMA Transmission Systems". AEU - International Journal of Electronics and Communications. Vol. 58, Issue: 4, Pages: 256-267, 2004.
31. O. S. Salih, C.-X. Wang, D. I. Laurenson, "Double embedded processes based hidden Markov models for binary digital wireless channels," Proc. IEEE ISWCS'08, Reykjavik, Iceland, 21-24 Oct. 2008. Pages: 219-223, 2008.
32. Cheng-Xiang Wang and Wen Xu, "A new class of generative models for burst error characterization in digital wireless channels," IEEE Transaction on Communications. Vol. 55, Issue: 3, Pages: 453-462, March 2007.
33. J. Garcia-Frias, P. M. Crespo, "Hidden Markov models for burst error characterization in indoor radio channels," IEEE Transaction on Vehicular Technology. Vol. 46, Issue: 6, Pages: 1006-1020, November 1997.
34. W. Zhu, J. Garcia-Frias, "Stochastic context-free grammars and hidden Markov models for modeling of bursty channels," IEEE Transaction on Vehicular Technology. Vol. 53, Issue: 3, Pages: 666-676, May 2004.
35. E. Costamagna, L. Favalli, P. Gamba, P. Savazzi, "Block-error probabilities for mobile radio channels derived from chaos equations," IEEE Communication Letters. Vol. 3, Issue: 3, Pages: 66-68, March 1999.
36. M. Zorzi, R. R. Rao, "Perspectives on the impact of error statistics on protocols for wireless networks", IEEE Personal Communications. Vol. 6, Issue: 10, Pages: 32-40, Oct. 1999.
37. P. Sadeghi, R. Kennedy, P. Rapajic, R. Shams, "Finite-State Markov Modeling of Fading Channels - A Survey of Principles and Applications", IEEE Signal Processing Magazine. Vol. 25, Issue: 5, Pages: 57-80, September 2008,.
38. S. A. Khayam, H. Radha, "On the impact of ignoring Markovian Channel Memory on the Analysis of Wireless System". IEEE International Conference on Communications (ICC '07), 24 - 28 June, Glasgow. Pages: 199-2004, 2007.

39. L. R. Rabiner, 1989. "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", Proceedings of the IEEE. Vol. 77, Issue: 2, Pages: 257-286.

40. G. Han, B. Marcus "Analyticity of Entropy Rate of Hidden Markov Chains". IEEE Transactions on Information theory. Vol. 52, Issue: 12, Pages: 5251-5266, December 2006.

41. "Propagation Prediction Models", COST 231 Final Report, Chapter 4, Pages: 134-148.

42. Rohde, Ulrich L., Jerry C. Whitaker, "Communications Receivers: Principles and Design", 3rd ed., McGraw-Hill, New York, N.Y., 2000. Chapter 17.9 - The Radio Channel.

43. C. Spillard, et al., "Mobile link propagation aspects, channel model and impairment mitigation techniques". CAP-D14-WP22-UOY-PUB-01, Report CAPANINA. Pages: 33-52, 29 April 2005.

44. Matlab, The Language of Technical computing. MathWorks Inc. 2004.

45. M. I. Rahman, S. S. Das, F. H.P. Fitzek, "OFDM Based WLAN System" Technical Report R-04-1002 - Center for TeleInFrastruktur (CTiF) Aalborg University.

46. NIST-SEMATECH Handbook of statistical method: www.itl.nist.gov / div898 / handbook / index.html.

47. Saad Biaz, Bing Qi, Yiming Ji. "Improving Expected Transmission Time Metric in Multi-rate Multi-hop Networks". IEEE CCNC 2008.

48. S. Y. Wang, C. C. Lin, H. W. Chu, T. W. Hsu, K. H. Fang, "Improving the Performance of Distributed Coordinated Scheduling in IEEE 802.16 Mesh Networks", IEEE Transactions on Vehicular Technology, Volume 57, Issue 4, July 2008 Page(s):2531 - 2547.

49. M. Guizani, P. Lin, S. M. Cheng, D. W. Huang, H. L. Fu, "Performance Evaluation for Minislot Allocation for Wireless Mesh Networks", IEEE Transactions on Vehicular Technology, Nov. 2008, Volume: 57, Issue: 6, On page(s): 3732-3745.

50. S. Y. Wang, C. C. Lin, K. H. Fang, "Improving the Data Scheduling Efficiency of the IEEE 802.16(d) Mesh Network", IEEE GLOBECOM 2008, Nov. 30 2008-Dec. 4 2008, New Orleans, LO.

51. T. C. Tsai, C. Y. Wang, "Routing and Admission Control in IEEE 802.16 Distributed Mesh Networks", IFIP International Conference on Wireless and Optical Communications Networks, 2007. WOCN '07. 2-4 July 2007, page(s): 1-5, Singapore, WOCN 2007.

52. C. Cicconnetti, V. Gardellin, L. Lenzini, E. Mingozzi, A. Erta, "End-to-End Bandwidth reservation in IEEE 802.16 mesh networks", IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems, 2007. MASS 2007, 8-11 Oct. 2007, page(s): 1-6. Pisa.

53. F. Liu, Z. Zeng, J. Tao, Q. Li, Z. Lin, "Achieving QoS for IEEE 802.16 in Mesh Mode", 8th International Conference on Computer Science and Informatics, Salt Lake City, USA.

54. F. Hou, P. H. Ho, X. Shen, "Performance Analysis of a Reservation Based Connection Admission Scheme in 802.16 networks", GLOBECOM 2006, San Francisco, CA.

55. Agrawal, D. P., Li, W., Wang, H. (2005). "Dynamic admission control and QoS for 802.16 Wireless MAN". Paper presented at the Wireless Telecommunications Symposium 2005.

56. Chang B. J., Chen, Y. L., Chou, C. M. "Adaptive hierarchical polling and cost-based call admission control in IEEE 802.16 WiMAX networks". Wireless Communications and Networking Conference, 2007.WCNC 2007, 11-15 March 2007, page(s): 1954-1958. Kowloon.

57. Chen, H. H., Qian, Y., Rong, B., (2007). "Adaptive power allocation and call admission control in multiservice WiMAX access networks". IEEE Wireless Communications, Volume: 14, Issue: 1, 14-19.

58. Chandra, S., Sahoo, A. (2007, June). "An efficient call admission control for IEEE 802.16 networks", 15th IEEE Workshop on Local and Metropolitan Area Networks, 2007. LANMAN 2007, 10-13 June 2007, page(s): 188-193. Princeton, NJ.

59. Hossain, E., Niyato, D. (2007). "Radio resource management games in wireless networks: an approach to bandwidth allocation and admission control for polling service in IEEE 802.16". IEEE Wireless Communications, Volume 14, Issue 1, 27-35.

60. Hossain, E., Niyato, D. (2007). "QoS-aware bandwidth allocation and admission control in IEEE 802.16 broadband wireless access network: A non-cooperative game theoretic approach". Computer Networks: The International Journal of Computer and Telecommunications Networking, 51(11), 3305-3321.

61. Osborne, M. J. (2003). "An introduction to Game Theory". Oxford University Press.

62. Drew Fudenberg, Jean Tirole, "Game theory". MIT Press, 1991.

63. Hung-Yu Wei, Ganguly, S., Izmailov, R., Haas, Z.J., "Interference-aware IEEE 802.16 WiMax mesh networks", IEEE 61st Vehicular Technology Conference, 2005. VTC 2005-Spring. Volume 5, 30 May-1 June 2005, Page(s):3102 - 3106.

64. F. Jin et al., Routing and Packet Scheduling for Throughput Maximization in IEEE 802.16 Mesh Networks, seen 15/04/2007 at http://www.seas.gwu.edu/~hchoi/publication/wireless/802.16mesh.pdf.

65. D. D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in Proceedings of the ACM International Conference on Mobile Computing and Networking (MOBICOM), Sep. 2003, pp. 134-146.

66. R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in Proceedings of the ACM International Conference on Mobile Computing and Networking (MOBICOM), Sep. 2004, pp. 114-128.

67. "Routing in 802.16 Mesh Networks: A Survey Paper". White Paper, Indian Institute of Technology Bombay. April 2007.

68. Biaz, S., Bing Qi, Yiming Ji, "Improving Expected Transmission Time Metric in Multi-Rate Multi-Hop Networks", 5th IEEE Consumer Communications and Networking Conference, 2008. CCNC 2008. 10-12 Jan. 2008 Page(s):533 - 537.

69. Bo Wang, Mutka, M., "Path selection for mobile stations in IEEE 802.16 multihop relay networks", 2008 International Symposium on a World of Wireless, Mobile and Multimedia Networks, 2008. WoWMoM 2008. 23-26 June 2008 Page(s):1 - 8.

70. Fanchun Jin, Arora, Amrinder Jinho Hwang, Hyeong-Ah Choi, "Routing and packet scheduling in WiMAX mesh networks" Fourth International Conference on Broadband Communications, Networks and Systems, 2007. BROADNETS 2007. 10-14 Sept. 2007 Page(s):574 - 582.

71. Ntlatlapa, N., "A Routing Metric and Algorithm for IEEE802.16 Mesh Networks", 2008 Third International Conference on Broadband Communications, Information Technology and Biomedical Applications, 23-26 Nov. 2008 Page(s):324 - 328.

72. Richard Draves, Jitendra Padhye, Brian Zill, "Comparison of Routing Metrics for Static Multi-Hop Wireless Networks", SIGCOMM04, Aug. 30Sept. 3, 2004, Portland, Oregon, USA. Copyright 2004 ACM.

73. J. Broch, D. Maltz, D. Johnson, Y.-C. Hu, J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols". MOBICOM, Oct. 1998.

74. Sheng-Shih Wang, Hua-Chiang Yin, Yi-Hsueh Tsai, Shiann-Tsong Sheu, "An Effective Path Selection Metric for IEEE 802.16-based Multi-hop Relay Networks", 12th IEEE Symposium on Computers and Communications, 2007. ISCC 2007. 1-4 July 2007 Page(s):1051 - 1056.

75. Yi-Ting Mai, Chun-Chuan Yang, Yu-Hsuan Lin, "Cross-Layer QoS Framework in the IEEE 802.16 Network" The 9th International Conference on Advanced Communication Technology, Volume 3, 12-14 Feb. 2007 Page(s):2090 - 2095.

76. Castrucci, M., Marchetti, I., Nardini, C., Ichimescu, A., Ciulli, N., Landi, G., Neves, P., "A Framework for Resource Control in WiMAX Networks", The 2007 International Conference on Next Generation Mobile Applications, Services and Technologies, 2007. NGMAST '07. 12-14 Sept. 2007 Page(s):316 - 321.

77. Niyato, D., Hossain, E., "A Hierarchical Model for Bandwidth Management and Admission Control in Integrated IEEE 802.16/802.11 Wireless Networks", Wireless Communications and Networking Conference, 2007.WCNC 2007. 11-15 March 2007 Page(s):3763 - 3767.

78. Niyato, D., Hossain, E., "A Radio Resource Management Framework for IEEE 802.16-Based OFDM/TDD Wireless Mesh Networks", 2006 IEEE International Conference on Communications, Volume 9, June 2006 Page(s):3911 - 3916.

79. Chun-Chuan Yang, Yi-Ting Mai, Liang-Chi Tsai, "Design of the QoS framework for the IEEE 802.16 mesh networks Export", International Journal of Communication Systems, Vol. 22, No. 12. (2009), pp. 1543-1562.

80. Bo Fu, Hejiao Huang, "A QoS Framework with Traffic Request in Wireless Mesh Network", Book Series Lecture Notes in Computer Science. Springer Berlin, Heidelberg, Volume 5682/2009, Pages 418-427.

List of Publications

International Conference

C1. De Rango F., Malfitano A., Marano S. "Wireless Channel Evaluation of IEEE 802.16e Protocol in HAP Architecture with Mobility Scenario under different Modulation Schemes". 13rd International Conference on Telecommunications (ICT 2006), Madeira Island, Portugal, May 9-12, 2006.

C2. De Rango F., Malfitano A., Marano S. "PER Evaluation for IEEE 802.16 - SC and 802.16e Protocol in HAP Architecture with User Mobility under Different Modulation Schemes". IEEE Global Telecommunications Conference (Globecom'06), 28Nov-2Dec., Alaska, 2006.

C3. De Rango F., Malfitano A., Marano S. "BER and PER Evaluation for IEEE 802.16e Protocol in HAP Architecture with User Mobility". Wireless Telecommunication Symposium (WTS 2006), Pomona, CA, USA, Apr.27-29, 2006.

C4. Malfitano A., De Rango F., Marano S. "Parametric Markov Chain Model in HAP Architecture with IEEE 802.16 Protocol". Int. Symposium on Wireless Personal Multimedia Communications, 3-10 Dec., Jaipur, India, 2007.

C5. De Rango F., Malfitano A., Marano S. "Markov Chain Based Models Comparison in IEEE 802.16e Scenario". WINSYS, July 26-29, Porto, Portugal, 2008,

C6. De Rango F., Malfitano A., Marano S. "Markov Chain Based Models Comparison and Hybrid Model Design in IEEE 802.16e Scenario". MILCOM, November 17-19, San Diego, CA, 2008.

C7. De Rango F., Malfitano A., Marano S. "Instant Weighed Probability Model to Guarantee QoS in IEEE 802.16e scenario". WCNC, 5-8 April, Budapest, Hungary, 2009.

C8. De Rango F., Malfitano A., Marano S. "Instant Weighed Probability Model for Time Variant Channel in IEEE 802.16e scenario". IWCMC 2009, 21-24 June, Leipzig, Germany.

C9. De Rango F., Malfitano A., Marano S. "Two Variable Instant Weighed Probability Model for Time Variant Channel in IEEE 802.16e scenario". IEEE Mobile WiMAX Symposium 2009. July 9-10, Napa Valley, California.

C10. De Rango F., Malfitano A., Marano S. "Bandwidth Availabilities Aware Defragmentation Based CAC Algorithm for IEEE 802.16 Distributed Mesh Networks". IEEE SPECTS 2009. July 13 -16, 2009, Istanbul, Turkey.

C11. De Rango F., Malfitano A., Marano S. "A New Call Admission Control Algorithm for IEEE 802.16 Distributed Mesh Networks". PIMRC2009, September 13-16, 2009. Tokyo, Japan.

International Journal

J1. De Rango F., Veltri F., Tropea M., Santamaria A.F., Fazio P., Malfitano A., Marano S., "Interdisciplinary issues for the management of next generation autonomic wireless systems: nature-inspired techniques and organic computing", to be published on International Journal Mobile Network Design and Innovation, 2008.

Chapters of Books

B1. De Rango F., Malfitano A., Marano S. "Cross -Layer end-to-end QoS Architecture: The Milestone of WiMAX" chapter of book: WiMAX Security and Quality of Service: An End-to-End Perspective, edited by Seok-Yee Tang, Peter Mller and Hamid Sharif. John Wiley and Sons.

B2. De Rango F., Malfitano A., Marano S. "Cross-Layer Architecture: The WiMAX point of view" chapter of book: Quality of Service Architecture for Wireless Networks: Performance Metrics and Management. Edited by Sasan Adibi. IGI Global.

DEIS - DIPARTIMENTO DI ELETTRONICA, INFORMATICA E SISTEMISTICA
Novembre 2009

Settore Scientifico Disciplinare: ING-INF/03