This thesis is entitled

A cross-layer mechanism for QoS improvements in VoIP over multi-rate WLAN networks

written by

Anna Sfairopoulou

directed by

Dr. Carlos Macian and Dr. Boris Bellalta

and supervised by

Dr. Miquel Oliver Riera

Has been approved by the
Department of Information and Communication Technologies
A thesis submitted in partial fulfillment of the requirements for the degree of
Doctor per la Universitat Pompeu Fabra

April 2008

To Carlos and Boris

Acknowledgements

First of all I want to trully thank my two thesis directors and dear friends, Carlos Macian and Boris Bellalta, for the continuous encouragement and guiding. Their dedication and time they devoted on this research made this thesis possible and has led to its final success. A big thanks also to Miquel Oliver for fully supporting me since the early stages of my PhD and showing his faith in me in various ways.

During these almost six years in the UPF, my colleagues in the 350 office have become my second family. Many thanks to all, especially to Jaume Barcelo, Cristina Cano and Laura Wong, who also reviewed parts of this thesis. Apart from their practical help, our "philosophical" discussions on life, the universe and everything during coffee breaks was a huge motivation for me. A very special thanks to my dear Edu for his unlimited support, his very useful comments as an "external reviewer" of the thesis, but mostly for his capacity to make me disconnect and smile always.

Finally, I want to thank my parents for fully supporting my decision to leave and follow this road. My mother for passing me some of her linguistic skills and my father for transmitting me his engineering curiosity on how things work. In some level, this thesis is a perfect combination of both.

Abstract

VoIP over WLANs has become a hot research topic during the past years due to the widespread deployment and ease-of-use of both technologies. Nevertheless, this widespread (and especially the commercial) deployment of VoIP over WLANs, cannot be ultimately successful unless most of the VoIP quality of service issues are solved efficiently.

The 802.11 standard, as most of the IP-related technologies, was not created having voice in mind and as such it brings a lot of new limitations for successfully deploying VoIP on top of it. The capacity of a wireless cell in terms of number of supported calls is still low compared to the promised data rates. Voice transmission over the wireless link under variable channel conditions can easily suffer from an increased packet error and loss ratio, with direct effect on its performance and quality. Multi-rate transmission is one of the key features of the IEEE 802.11 PHY/MAC specifications which allows each mobile node to select its physical layer parameters (modulation and channel coding) to optimize the bit transmission over the noise/fading-prone channel. These sporadic rate changes occur on the mobile nodes due to a reaction of the Link Adaptation algorithm of the 802.11 specification to a number of various factors (user movement, meteorological conditions, interferences etc). They have however a direct impact on the transmissions of all active calls and produce a general degradation of their quality of service (QoS) and a very hostile environment for VoIP.

In this thesis we propose a codec adaptation algorithm, which allows a cell-wide optimization of network resources and voice quality on multi-rate WLANs. It is based on the combined cross-layer feedback from Real-Time Control Protocol (RTCP) packets and the MAC layer and uses the Session Initiation Protocol (SIP) to notify the codec change to the users and re-negotiate the new codec without interrupting the call. The algorithm can be implemented both in a centralized and a distributed mode with very few modifications. To validate this proposal various simulations were performed and

different cases were studied, with the results showing an important capacity and quality increase over the standard case.

Additionally, a joint solution including this codec adaptation algorithm together with a Call Admission Control mechanism is proposed. The objective is to study the performance of applying codec adaptation both at the moment of rate changes (for solving the multi-rate effect on the QoS of the calls), as also adapting the codec of new incoming calls (for increasing the cell capacity). A number of decision policies are presented, which dictate various approaches on adapting the codec of the calls and can be therefore used according to the parameter we want to optimize: from blocking and dropping ratio to the maximum number of simultaneous calls or the obtained average QoS of the flows. At the same time, a new quality and quantity index for the VoIP flows is presented, named Q-factor, that can help tune this optimization process, combining the metrics used in the evaluation of it.

The study concludes with the implementation proposal and details of an Access Point optimized for VoIP service, which includes the modifications proposed in this thesis. The necessary extensions are presented for the new AP software and hardware architecture optimized for VoIP traffic, including a new SIP header and a Wireless QoS Multi-Rate Module which implements the complete solution.

Resumen

El uso de VoIP (voz sobre IP) a través de redes inalámbricas (WLANs) es un tema de investigación muy actual ya que el despliegue de ambas tecnologías durante los últimos años ha sido muy extenso. Aun así, este despliegue (y especialmente el comercial) no se puede considerar de todo un éxito si los temas de calidad del servicio no se resuelven primero de manera eficaz.

El estándar 802.11, como la mayoría de las tecnologías relacionadas con comunicación IP, no se ha creado teniendo en cuenta las necesidades del trafico sensible como el de voz. Por lo tanto el desarrollo de un servicio de VoIP a través de WLAN conlleva nuevas limitaciones y requerimientos. La capacidad de una celda inalámbrica en términos de numero de llamadas soportadas es aun muy baja si tenemos en cuenta las velocidades prometidas. La transmisión de voz a través de un enlace inalámbrico y con condiciones de canal variables puede sufrir fácilmente un incremento de errores y perdidas de paquetes, con efecto directo en la calidad de los flujos y el rendimiento del sistema. Una transmisión "multi-rate" (de velocidad variable) es una de las características de la especificación PHY/MAC del IEEE 802.11 que permite a cada estación seleccionar sus parámetros de nivel físico (modulación y codificación de canal) para optimizar la transmisión de bits sobre un canal con ruido y desvanecimientos. El algoritmo de adaptación de enlace (Link Adaptation) del estándar 802.11 es el responsable de esta función. Su reacción a factores como movilidad de usuarios, condiciones meteorológicas, interferencias, etc, provoca los cambios de rate en las estaciones móviles. Su impacto directo en la transmisión de *todas* las estaciones y las llamadas ya activas produce una degradación general de la calidad de servicio y un ambiente muy hostil para Voz sobre IP.

En esta tesis proponemos un algoritmo de adaptación de codecs que permite una optimización de los recursos de la red y de la calidad de voz en un entorno de redes inalámbricas multi-rate. Se basa en la combinación del feedback a diferentes niveles

(cross-layer), desde paquetes de Real-Time Control Protocol (RTCP) y nivel MAC hasta usando el protocolo de inicio de sesión SIP para notificar los cambios de codec a los usuarios y re-negociar el nuevo codec sin interrumpir la llamada. El algoritmo se puede implementar tanto en versión centralizada como distribuida con pocas modificaciones. La propuesta se ha validado usando varias simulaciones considerando diferentes casos y escenarios. Los resultados muestran un importante incremento tanto de capacidad como de calidad comparado con el caso estático.

Adicionalmente, proponemos una solución conjunta que incluye este mismo algoritmo de adaptación de codecs y un mecanismo de control de admisión de llamadas (Call Admission Control - CAC). El objetivo es estudiar el rendimiento de la celda cuando la adaptación de codecs se aplica tanto a llamadas activas (como respuesta y solución al efecto de multi-rate) o a llamadas nuevas (para aumentar la capacidad de la celda). Se presentan varias políticas de adaptación. Cada una de ellas implica diferentes maneras de adaptar los codecs de las llamadas y por lo tanto se pueden utilizar dependiendo del parámetro que queremos optimizar: desde probabilidad de bloqueo y de corte de llamada, hasta el numero máximo de llamadas simultaneas en la celda o su calidad promedia. Al mismo tiempo, proponemos un nuevo índice de calidad y capacidad, el *Q-Factor*, que puede ayudar en el ajuste del proceso de optimización. El estudio concluye con la propuesta de implementación de un Punto de Acceso optimizado para VoIP que incluye las varias modificaciones presentadas en esta tesis. Presentamos las extensiones necesarias para la arquitectura software y hardware del nuevo AP. Estas incluyen una nueva cabecera SIP y el modulo "Wireless QoS Multi-Rate Module" que implementa todas las partes de la nuestra propuesta.

Publications derived from this work

Based on the results obtained from this thesis, the following publications were made:

- Journals / Book Chapters

  1. A. Sfairopoulou, B. Bellalta and C. Macian, In IEEE Communications Letters, "How to tune VoIP codec selection in WLANs?", Accepted for publication (date of acceptance June 2008).
  2. J. Barcelo, B. Bellalta and A.Sfairopoulou, In ACM Mobile Computing and Communications Review, "Wireless Open Metropolitan Area Networks", Accepted for publication (date of acceptance June 2008).
  3. A. Sfairopoulou, C. Macian and B. Bellalta, On book "Wireless Multimedia: Quality of Service and Solutions", Chapter "Adaptive codec selection for VoIP in multi-rate WLANs". To be published by Idea Group Inc., June 2008.
  4. B. Bellalta, C. Cano, J. Barcelo, A. Sfairopoulou and M. Oliver, On book "Wireless Quality-of-Service: Techniques, Standards and Applications", Chapter "Policy-based QoS provision in WLAN Hotspots". To be published by Editorial: Auerbach Publications, Taylor&Francis Group, 2008.
  5. B. Bellalta, C. Macian, A. Sfairopoulou and C. Cano, In Lecture Notes in Computer Science 4712: 342-355, Springer Verlag, "Evaluation of Joint Admission Control and VoIP Codec Selection Policies in Generic Multirate Wireless Networks", September 2007.
  6. A. Sfairopoulou, C. Macian and B. Bellalta, In Lecture Notes in Computer Science 4606: 52-61, Springer Verlag, "VoIP Codec Adaptation Algorithm in Multirate 802.11 WLANs: Distributed vs. Centralized Performance Comparison", July 2007.

7. A. Sfairopoulou et al.; On book "Traffic and QoS Management in Wireless Multimedia Networks", Contributions in Chapter 2: "Packet Scheduling And Congestion Control", Ed. Springer Verlag, May-June 2008.

- In proceedings of National and International Conferences

1. J. Barcelo, B. Bellalta, A. Sfairopoulou, C. Cano, M. Oliver, "No Ack in IEEE 802.11e Single-Hop Ad-Hoc VoIP Networks", The 7th IFIP Annual Mediterranean Ad Hoc Networking Workshop, June 2008, Mallorca, Spain.
2. J. Barcelo, B. Bellalta, C. Macian, M. Oliver, A. Sfairopoulou, "Position Information for VoIP Emergency Calls", Web Information Systems and Technologies (WebIST'07), 3-6 March 2007, Barcelona, Spain.
3. A. Sfairopoulou, C. Macian and B. Bellalta, "QoS adaptation in SIP-based VoIP calls in multi-rate 802.11 environments", In IEEE International Symposium on Wireless Communication Systems (ISWCS'06), September 2006, Valencia, Spain
4. A. Sfairopoulou, C. Macian and M. Oliver, "Architecture of a VoIP Traffic Exchange Point", In 12th Open European Summer School (Eunice '06), September 2006, Stuttgart, Germany
5. C. Macian, A. Sfairopoulou, M. Oliver, "Arquitectura para un punto de intercambio de trafico de voz sobre IP", In XV Jornadas Telecom I+D 2005, November 2005, Madrid, Spain
6. J. Barcelo, A. Sfairopoulou, M. Oliver, J. Infante, C. Macian, "Arquitectura de gestion de un operador neutral Wi-Fi", In V Jornadas de Ingenieria Telematica JITEL 2005, September 2005, Vigo, Spain

- Workshops / Technical Documents

1. In the 8th Cost290 Project meeting, Technical Report TD07(018), "Dynamic measurement-based codec selection for VoIP in multirate IEEE 802.11 WLANs", A. Sfairopoulou, C. Macian and B. Bellalta, February 2007, Vienna, Austria.
2. In the 10th Cost290 Project meeting, Technical Report TD07(042), "Joint Admission Control and VoIP Codec Selection Policies in WLANs", A. Sfairopoulou, B. Bellalta and C. Macian, October 2007, Malaga, Spain.

- Pending to be approved

1. A patent including parts of this thesis is in evaluation process

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Motivation and problem statement

Voice over IP (VoIP) has attracted the interest of the research community from its very beginning. As users are starting to spend more time connected, various aspects of their traditional life are passing to the internet world, from radio and television, to entertainment, social communities and e-commerce. So, partially replacing or at least complementing the traditional telephony with internet telephony seems only inevitable. And although in the modem era this idea could seem utopia with the slow internet connections offered for most of the home access, with the increasing internet broadband access connections offered nowadays the quality of the VoIP services could get to be even comparable to the PSTN carrier-grade service.

Nevertheless, the strength of VoIP technology does not lie in barely trying to replace the traditional circuit switched telephony calls. The possibility to create new services combining voice with data and offering a real multimedia application environment to the user is huge. The convergence of data with voice applications is what in fact makes this technology of packetized voice so attractive. Additionally, these services are much easier to deploy than in the PSTN Intelligent Network (IN) system, more cost effective and follow the internet "open" philosophy where everyone can participate, create its own VoIP network and applications or even become a VoIP service provider. Session Initiation Protocol (SIP), a protocol for setting up, controlling and tearing down VoIP sessions, has played a very important role in this new VoIP service era, with the flexibility it provides in session management and its IP-oriented design. We closely examine this protocol in chapter 2. However, VoIP technology has a big road ahead in order to cope with the years of development that the PSTN telephony has

had. PSTN is a technology that can be considered mature and solid, and one of the few that can achieve almost 100% availability (in fact the promised availability of the PSTN network is of 99.999%, also known as "the five nines"). VoIP, just almost a decade old in the form that we know it today, can be still considered a new trend and as such many aspects of it are yet to be determined. Although standardization, security and regulatory aspects are constantly discussed by the research community, the main effort is in fact placed in voice quality issues. Most of the problems arise from the fact that IP-networks were not created having in mind the telephony service and its strict requirements and thus cannot cope with them effectively. Recommendation G.114 [66] of the International Telecommunications Union (ITU-T) indicates that the end-to-end delay has a great impact on the perceived quality of interactive VoIP conversations with a threshold maximum of 150 ms one-way (150 - 400 ms for international long-distance trunks). A similar indication is given for the packet loss threshold, with an upper bound of 3-5% roughly. Both of these conditions cannot be guaranteed by the best-effort Internet [52]. Therefore, many of the recent VoIP advancements were directed towards achieving a satisfactory level of Quality of Service (QoS), for example by providing more robust codification techniques that can accept higher packet losses or by consuming less resources using silence suppression combined with lower bitrate codecs.

Still, many QoS issues remain unsolved, especially with the appearance of new access networks that create new performance limitations for VoIP. Users are starting to be "always connected", with the various types of cellular and wireless networks available to them, from WLANs to GPRS/3G networks or the recent IP Multimedia Subsystem (IMS), and the new mobile devices that can connect to any of them from practically anywhere. And naturally, users demand to be able to use the same services independently of the access network. In particular the extension of WiFi hotspots (venues that offer WiFi access) in the last years has been growing fast, ranging from $14,752$ in 2002 [25] to approximately $142,320$ nowadays [42], meaning there has been an increase of more than a 900%, and it is expected to continue growing in the following years. The IEEE 802.11 technology is one of the most successful ones actually, replacing both home access, enterprise communication infrastructures and hotspot access in public places, like hotels, airports etc. Therefore, VoIP over 802.11 WLANs has become a hot research topic during the past years due to the widespread deployment, the ease-of-use of both technologies and the user interest on it.

However, both the capacity of a wireless cell in terms of number of supported calls as well as the quality of the voice transmission over the wireless link under different channel conditions are crucial for deciding whether this technology can be widely deployed and accepted for voice service. WLANs, as most of the IP networks, were initially oriented to data services and thus there are many inherited problems when it comes to offering VoIP services on top of this technology. These can vary from increased packet losses due

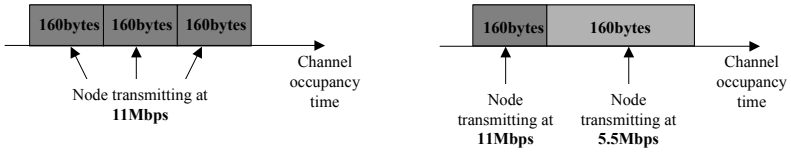


Figure 1.1: The effect of a rate change on the cell resources distribution

to channel errors and collisions to the fact that until recently no traffic differentiation was available and VoIP performance was suffering a fast degradation in the presence of TCP flows.

Most of these problems have been addressed already by the research community and a variety of solutions can be found. Additionally, the newer version of 802.11e IEEE standard has managed to solve some of them with the enhancements it includes, proposing for example traffic differentiation by using separate Access Categories for each traffic type. Of specific interest in this study however is the fact that the actual capacity of an 802.11 cell in terms of simultaneously active calls is surprisingly low taking into consideration the high data rate of 11Mbps to 54Mbps that the different standards of this technology can offer. Garg and Kappes [34][32] have studied in depth this effect and have put the basis for the WLAN voice capacity discussions. More recently Trad et al. [77] have performed similar studies for the newer 802.11e standard. The main problem identified was that the WLAN is characterized by an inefficient use of the channel due to the high MAC and PHY protocol overheads and the shared channel access method. Especially when the channel is shared between elastic (TCP) and inelastic (VoIP) traffic this problem increases. In his work [8] Bellalta evaluated the joint performance of VoIP and TCP using a model-based admission control. This discussion and a further analysis of the 802.11 problematics for VoIP will be the topic of chapter 3.

Due to this limited voice capacity, a careful planning of the number of calls accepted and under which conditions these will be accepted is mandatory in order to maximize the cell capacity and distribute efficiently the network resources among them. This is where Call Admission Control (CAC) methods enter, trying to determine the available cell capacity at each instant and accept or reject new calls accordingly. However, this task complicates even more when that capacity becomes variable due to network specific mechanisms. One of them is the Link Adaptation (LA) procedure of the 802.11 networks, a mechanism to adjust the physical transmission rate of a node according to the channel conditions it perceives and so as to optimize the bit transmission over the noise/fading-prone channel. The channel conditions in an 802.11 cell may vary due to a number of different circumstances, such as user movement, variations in meteorological

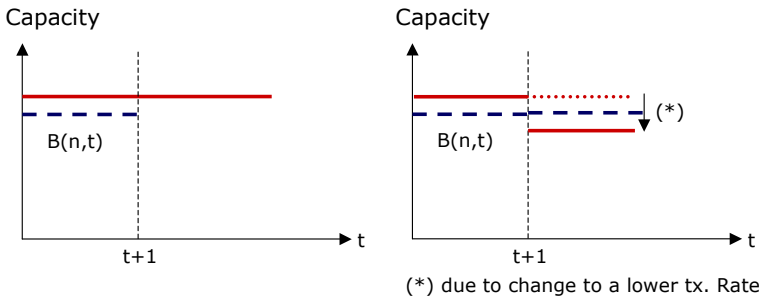


Figure 1.2: Capacity variation due to rate change leads to congestion

conditions, obstacle interference, etc. Heusse et al. [37] identified an anomaly provoked by the LA mechanisms and caused mainly by the 802.11 "fair" channel access mechanism. They observed that the rate change of one of the nodes of the cell provokes a general degradation on the transmission rate of all the other nodes. In other words, these sporadic rate changes occurring on the mobile nodes due to the link adaptation algorithm of the 802.11 specification, have an impact on the transmissions of all active nodes and produce a general degradation of the network performance. The main reason for this is that by reducing its transmission rate a node demands more cell resources (channel occupancy time) in order to transmit the same amount of data as before (see Figure 1.1). This increased demand can lead the system to a congestion state with a direct impact on all active calls. The CAC algorithms alone are not able to deal with this situation, since it affects calls *already admitted* to the network by the CAC. Figure 1.2 depicts this effect. While the bandwidth demand at instant t considering n active nodes ($B(n,t)$) is lower than the available cell capacity, after a transmission rate drop at instant $t+1$ this cell capacity is now lower than the bandwidth demand, which makes the situation unsustainable and leads to congestion state. The main causes of this performance anomaly are analyzed extensively in chapter 3 and compose the basis of our study.

In this thesis, we focus on the effect that these multi-rate WLAN networks have on the already sensitive VoIP traffic. The inherited limitations of WLANs and especially the capacity variation caused by their multi-rate characteristic can only introduce new constraints on meeting the VoIP QoS demands. We find that the QoS degradation provoked is unacceptable according to the ITU standards and the user experience, and we discuss the various solutions that can be found in literature. Our goal is to provide a solution destined to the specific characteristics and problematics of VoIP over capacity varying WLANs. As a result, an innovative cross-layer algorithm is proposed that both

monitors the QoS of the calls and adapts their voice codec when considered necessary. This cross-layer codec adaptation can use the advantage of combining information distributed between different layers in the 802.11 architecture and the power of the SIP mechanism for session renegotiation without interrupting or dropping any call.

1.2 Contributions and methodology

This thesis intends to provide an overview of the impairments observed on voice flows due to the multi-rate characteristic of 802.11, and discuss some of the solutions that have been proposed so far, subsequently presenting a codec adaptation solution. Main goals are:

1. To analyze the source of the various problems encountered on VoIP due to the limitations of the WLANs and discuss the different solutions found on the literature. Specifically and in more detail analyze the impairments caused by the multi-rate WLANs and discuss why the existing solutions are not sufficient to cope with it.

2. To introduce a solution based on a cross-layer codec adaptation algorithm, developed for the specific characteristics and problems of VoIP over multi-rate WLANs. Design and explain its architecture and sub-modules, the way it combines information from MAC layer with RTCP QoS feedback reports and how it uses SIP re-Invite method to negotiate a codec change when this is deemed necessary. To demonstrate it is able to satisfy the QoS needs of VoIP traffic and that it can prevent the dropping of the active calls. Finally to validate the results by using extensive simulations and testings under different scenarios.

3. To examine the benefits of a combination of this codec adaptation algorithm with a Call Admission Control method for better cell-wide resource optimization, introducing as an addition a number of guideline decision policies and providing a comparison between the performance of each.

4. Under the same scenario, to introduce and sketch a new voice quality and quantity index, the Q-Factor, which can provide a unification of the three important metrics of the evaluation procedure, the blocking rate, dropping rate and average MOS value, in the trade-off between the number of calls and the quality achieved by each one.

5. Finally, to review and discuss the characteristics, difficulties and possibilities of a real implementation of our proposal in an 802.11 Access Point. To design both software and hardware architecture of a new VoIP optimized AP including the QoS enhancements presented in this thesis.

In order to achieve these objectives we followed a methodology based on analysis, algorithm design and validation through simulations. First, an in-depth study of the problem based on the work found in the literature was performed, in order to identify the reasons and circumstances under which a capacity variable 802.11 channel provokes quality degradation on VoIP flows. Through the literature study we have identified a number of shortcomings and limitations of the available solutions. Our proposal of a cross-layer codec adaptation algorithm has been designed, including the basic functionalities of it, the necessary input information and interactions with each involved layer, the expected output and a complete state diagram of the algorithm. As a validation method for our proposal, and since the main contribution is a software based, cross-layer algorithm, the simulation method has been chosen. Extensive simulations were thus performed including a number of different tools:

- we used the well known NS-2 simulator [58] with some modifications to include the elements missing (a SIP patch obtained from National Institute of Standards and Technology (NIST) [57]). Using this tool, we were able to simulate at packet level the VoIP capacity of a 802.11 cell and evaluate the algorithm performance through metrics such as packet loss, delay and jitter, which are the most common ones used for VoIP QoS evaluation. These results are presented in chapter 4. The VoIP capacity obtained from our experiments under different codec and rate combinations, has been additionally validated and compared against the analytical model presented in [11]. The system behavior was shown to match quite precisely the expected results, as foreseen by the analytical model.

- in order to obtain flow level metrics, like the call blocking and dropping rate, the average number of simultaneous calls at the cell, the average number of accepted calls etc, it was not efficient to use the packet-level simulations provided by NS-2. Thus a new tool was needed to simulate our system at flow-level. For this we have created a C++ simulator using the COST simulation toolkit [18]. To model the variable voice capacity in this flow-level simulator, we have created a new analytical VoIP capacity index based on the proposal of Hole and Tobagi [39] and modified for our multirate/multicodec scenario, as explained in detail in chapter 5. Again, this new capacity indicator, was validated against the experiments performed in NS-2 and was proven to be precise.

Additionally a number of policies for Call Admission Control have been designed and implemented in the same flow-level simulator, together with the Q-Factor. An integration of the complete solution composed by the Codec Adaptation algorithm with the policy-based CAC mechanism has been tested. Based on the results obtained we have evaluated the proposed mechanism, proven to be highly beneficial for VoIP QoS.

1.3 Structure of this thesis

After introducing the problem and the objectives in this chapter, the remainder of the thesis is organized in two parts as follows:

Part I focuses on a background overview of the technologies and standards used in this thesis and the scenario definition.

In Chapter 2 we review the typical VoIP system's elements: from the codification of the human voice, to the signalling protocols to establish a session and the protocols involved in the media transmission over an IP network. We also review two of the most known VoIP QoS measuring methods nowadays, the Mean Opinion Score (MOS) and the E-model.

In Chapter 3 a short background on the IEEE 802.11 standard is given, including the recent 802.11e version of it. An emphasis is placed on the limitations of this technology and the problems that it imposes on VoIP transmissions, as also the main reasons behind them. Finally, a detailed introduction to the Link Adaptation mechanism of 802.11 networks and the basis of the multi-rate anomaly are also found.

Part II focuses on the main contributions of the thesis, the proposal, analysis and evaluation under different scenarios of the cross-layer codec adaptation algorithm.

In Chapter 4, after an extensive review of the state of the art work in VoIP over multi-rate networks, a solution based on a cross-layer codec adaptation algorithm is presented. The architecture design, the modules and the procedures composing this are analyzed, and two different implementation methods (distributed and centralized) are discussed. The performance of the algorithm is validated using extensive simulations covering both VoIP-only and VoIP-TCP (heterogeneous) traffic scenarios.

In Chapter 5, the cooperation of the voice codec adaptation algorithm with a Call Admission Control mechanism is presented. The adaptation procedure is based on a number of decision policies described here in detail. Additionally, the Q-Factor is presented, a new metric combining three of the evaluation metrics used in this study, namely the blocking and dropping probability and the average MOS value. This chapter ends with an analysis of the codec complexity and its effect on the codec adaptation procedure when the node's processing power is limited.

In Chapter 6, the focus is placed on designing a VoIP optimized Access Point, which includes all the enhancements proposed previously in the thesis. Both software and hardware architecture are reviewed and the guidelines for a testbed implementation are given. In addition an extension to the SIP protocol including a new header is proposed in order to address the issue of communication between a transparent SIP proxy and a mobile node.

Finally in Chapter 7, the main conclusions of this research study are outlined and some future work guidelines are given.

Part I

Background overview: VoIP over WLAN networks

CHAPTER 2

Voice over IP: From encoding to media transmission

2.1 What composes a VoIP system?

In order to better understand what the impact of a multi-rate 802.11 environment on a VoIP system can be, it is necessary to review the elements that compose such a system, and which among them are prone to interact with 802.11 in order to provoke alterations on the VoIP quality. According to Figure 2.1, the main elements of a VoIP system are [24]:

- Human voice: The foremost element of a VoIP system is human voice, or in general an audio source (music from a CD or stored speech, for example). Throughout this thesis, and since the main impact of 802.11 will be over real-time transmissions, as will be explained, an interactive speaker is assumed.
- A microphone and speakers: Or, in general, some device to capture human voice and transform it from a pressure wave to a continuous electric signal, which can be processed by a computer. Conversely, at the reception point the inverse procedure must take place, and hence some speakers or headsets are highly desirable.
- A sampling and encoding device: In order to adapt an analog signal to be transported over a packetized data network, a dual process must occur: First, the analog signal must be transformed into a train of discrete samples, so that one or more of these samples can be inserted on every data packet. Additionally, in order to limit the (theoretically, infinite) range of values that a voice sample can take (from a whisper to a shout), it is advisable to reduce the value range to a pre-specified set of equispaced or nonuniform values. This procedure will allow the system to set a fixed number of bits to encode all possible values of a sample.

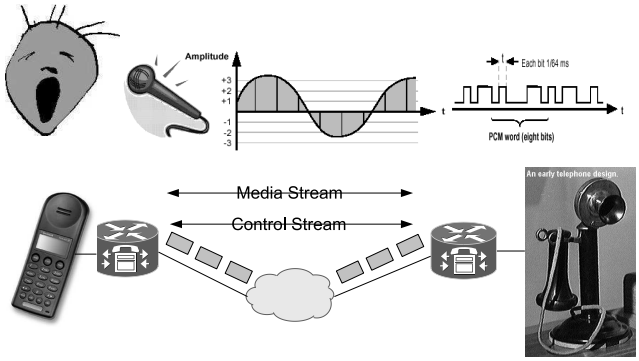


Figure 2.1: Main elements of a VoIP system

Second, a standardized and efficient way to digitally encode such values must be agreed upon (e.g., agreeing that 00000000 will mean -1 Volt and 11111111 will mean +1 Volt, with all values in between equally spaced). Additionally, the encoding can be designed to be robust in the case of data loss (e.g. by introducing some redundancy between samples), or to be especially efficient in terms of capacity usage (e.g. by only encoding the difference between two consecutive samples, thus saving some bits), or in general to try to maximize some desired property. The diversity of goals and procedures designed to encode and decode voice samples has given rise to a broad number of encoding-decoding algorithms, generally known as *codecs*, and which will be reviewed in section 2.1.1.

- A media transport protocol: If the encoded voice samples are to be transported over a data network, some protocol devised for this purpose has to be used. Since VoIP is an interactive, real-time application, the chosen protocol has to show good properties in the face of delay, jitter and loss. As will be explained in section 2.1.3, Real-time Transport Protocol (RTP) [70] is the protocol of choice for such a purpose in the Internet.

- A session management protocol: Beyond transmitting the encoded voice samples, there has to be a way for the communicating parties to negotiate which codec will be used, when the interchange of voice will start and end, to which port number the samples should be addressed, etc. Besides, a VoIP transmission can involve more than one voice/sound streams, as in the case of a multi-party conference or a multi-track sound recording. Hence, a protocol is needed that manages the set

up, negotiation and tear-down of media sessions among participating peers. This is the role of the Session Initiation Protocol (SIP) [69], which is the subject of section 2.1.2.

- An application: So far, the corresponding elements for the capture, sampling & encoding of voice, as well as its transmission over a data network have been mentioned. But to what purpose is voice being transmitted? Is it a conversation between two humans? Is it the broadcasting of a music videoclip? Is it part of a multimedia session á la Messenger, involving voice, webcam video, and chat? In general, all the elements explained up to now are merely *tools* that an application will use to provide a specific multimedia service over the Internet. As seen through the many examples mentioned above, the same set of tools serve to support a wide variety of different applications. The applications themselves are not the subject of this thesis, and will only be treated as examples of the usage of the other elements.

- Specialized network nodes: In the same way that the Internet uses routers and switches for its correct operation, for the correct operation of the above mentioned protocols and the associated communication architecture, a number of dedicated network nodes will be necessary. Such nodes will implement such functions as finding users by using their VoIP identifier, storing user preferences, redirecting calls to a voicemail, translating the codec used by a user to the one supported by another one, etc. Such nodes are not the focus of this thesis, but some reference to them will be necessary to highlight the characteristics of some of the proposed solutions. Hence, in section 2.1.2 a brief explanation of the main one, the proxy server, will be made for clarity. Additionally, a proposal for a specialized network element, a VoIP-enabled Access Point, implementing the solutions derived from this thesis will be also analyzed in chapter 6.

Of the many elements that compose a VoIP system, only some are involved in coping with the multi-rate phenomenon. As has already been pointed out, the codecs can be designed to optimize different parameters, such as capacity usage or robustness, or they can even adapt their behavior to the changing channel conditions. The particular characteristics of some of them will be reviewed in section 2.1.1.

Considering the protocols involved in a typical voice session, the media transport protocol (RTP) to begin with, was not designed to be adaptive in any way, so it can not be used in any adaptive solution. The session management protocol, SIP, does provide a mechanism for re-negotiating the characteristics of a call if network conditions change, but it was not intended for quick reaction and repeated usage, as would be the case in WLANs. Furthermore, it does not, by itself, record or notice any changes at the PHY/MAC layer, so it can not react to rate changes. The control protocol companion

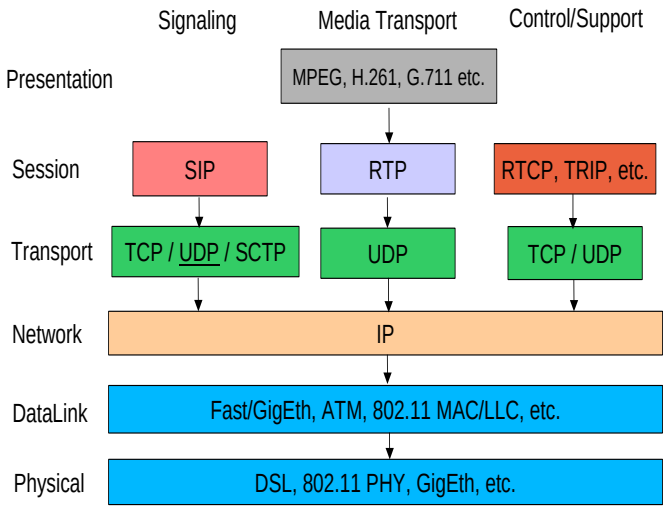


Figure 2.2: The VoIP protocol stack

of RTP on the other hand, the Real-Time Control Protocol (RTCP) since it continuously delivers quality feedback of the monitored session, provides an indirect way of detecting the effect that a PHY layer rate change has on it. As it is then, the only way of using SIP mechanisms to cope with the multi-rate phenomenon is by coupling it with PHY/MAC information and/or RTCP feedback in some way. This cross-layer solution, combining the information obtained from the different layers of the WLAN architecture, is the main subject of chapter 4.

Last, applications themselves can be designed to implement their own quality monitoring mechanisms at the application layer. They are then independent of the network and do not really take multi-rate changes into account; they simply measure any quality degradations due to any causes, and react. On the one hand, that allows to implement mechanisms that are valid for any situation and independent of any technology or specific effect. On the other, however, since they do not take into account the nature of the problem, but only its symptoms, it is much more difficult to implement an efficient response. For this kind of adaptive applications for example, an increased packet loss due to an error prone channel with bad signal to noise ratio (SNR) would be interpreted equally to an increased packet loss due to a varying channel occupancy because of a node's link rate change. However, the two situations are totally different and an efficient solution would better take into consideration the real source of the problem. To put it in other words, "one size does not necessarily fit all". Since such mechanisms are not specific for multi-rate environments, they will not be further considered in this study.

In the next sections, a brief introduction to the main elements relevant to the designed solutions of the specific multi-rate issue will be reviewed. Hence, codecs,

RTP/RTCP and SIP will be summarily presented and their main characteristics reviewed, before proceeding to the solutions themselves.

2.1.1 Voice Codecs

The transformation of analog speech into discrete binary-encoded samples amendable to transport through packet networks is a complex procedure, in which several degrees of liberty exist. As a consequence, a number of different algorithms have been devised. All of them have to perform the same rough steps:

1. Transform the continuous analog signal into a train of equally spaced, discrete samples. The sampling rate must conform to Nyquist's Theorem, which states that in order to be able to reconstruct the original analog signal, it must be sampled at a frequency equal to twice its bandwidth. Although human voice roughly comprises the band between 0 and 20.000 Hz, it is generally limited to the 0-3400 Hz band with the help of a low-pass filter, in order to save bandwidth. The typical sampling rate, slightly above Nyquist's minimum, is 8000 Hz.

2. The amplitude of the samples (the "volume") can take a broad range of values. Consequently, in order to be able to provide a discrete value to each one, which can then be codified in a binary word, a huge number of bits would have to be used. In order to limit the capacity needed to codify a sample, a set of limited, standardized values is chosen, and all intermediate values are rounded to those ones. As a consequence, a certain error is introduced, called the quantization noise. It is intuitively easy to see that this error is relatively more important for small amplitude values ("whispers") than for big ones ("shouts"), for the SNR is smaller in the first case, making the message difficult to understand. Hence, nonuniform quantization is used, by which the distance between two standard values is smaller (provoking a smaller quantization error) for smaller values of amplitude.

3. Last, the resulting sample values must be transmitted across a network. Not necessarily the sample value itself must be transmitted, other more sophisticated schemes can be used: For example, the difference between two samples could be coded, potentially reducing the number of bits needed. Or even the value of a sample could be used by the receiver to predict the next one, eliminating the need to send it altogether.

Obviously, the result of filtering the voice frequency band and limiting the number of bits used to codify each sample is to reduce the overall quality, and hence also the understandability, of the message. However, the effect is far from linear, so that "intelligent" ways of coding the samples can go a long way in reducing bandwidth

consumption without greatly degrading the quality. The different methods to code, predict or otherwise optimize the transmission and calculation of samples also introduce a degree of error.

In general, three codec families exist:

- **Waveform codecs**: These codecs simply sample, quantize and send the information, without further considerations. They are simple and provide very good quality, since they closely reproduce the original analog signal. Being so simple, they take low processing effort and hence do not introduce any additional delay into the system, which is an optimal characteristic for real-time communications. In exchange of this, they need fairly large bandwidth to provide good quality, and degrade rapidly otherwise. A well known example of a waveform codec is the ITU-T G.711 codec [64].

- **Source codecs (a.k.a. vocoders)**: The basis of these codecs is always a mathematical model of the speech generation process at the human voice tract. The model usually takes the form of a linear multi-parameter filter. By transmitting the adequate filter parameters, any sound can theoretically be reproduced. Furthermore, since the generation of human voice presents a fair amount of correlation among consecutive samples, it is possible to predict the next samples from previous ones, with a high degree of probability. Hence, combining sample prediction at the receiver with the sending of only the filter parameters, the overall bandwidth needed can be much reduced. The price to pay, however, is a synthetic-sounding voice, which is only fair, since it was synthetically generated. Additionally, these sophisticated algorithms need much more processing effort than waveform codecs and generally use several samples at once in order to operate and predict the next ones, so that the overall effect is introducing some additional delay, as well as necessitating more powerful (and hence more expensive) signal processors.

- **Hybrid codecs**: A mixture of both previous techniques: Hybrid codecs use a mathematical model of the voice tract, but use a number of different input vectors to compare the result with the original signal. This way, a more precise encoding can be found for every sample. In this case, not only the filter parameters are sent but also an indication of which of the standardized excitation vectors has been used in generating it. As could be expected, these codecs lie somewhat in between the previous two in terms of bandwidth usage and quality, and are widely used, like the G.728 [65] and G.729 [67], [68] codecs.

A particular case of the previous family are the variable-rate codecs. By slightly changing the characteristics of the algorithm, they can trade some additional bandwidth against better quality, or higher robustness in the face of packet loss or a noisy channel.

This type of codecs, although originally designed for GSM networks, can be particularly interesting for a 802.11 multi-rate scenario, since this presents frequent channel changes. Adaptive codecs can change their parameters as fast as every 20ms (per frame basis), and could present an automatic, user-independent way of coping with the multi-rate issue. Widely used actually in 3G cellular networks, their translation to the 802.11 world seems only logical, and some of the existing proposals will be reviewed later in chapter 4.

2.1.2 Session Management using SIP

A Voice over IP call consists mainly of two parts: signalling (a.k.a. call control) and media transmission. The signalling procedure is responsible for establishing the call, authenticating users, setting up the route, controlling the status of the call and terminating the session when the call is finished. The most used signalling protocols nowadays are the Session Initiation Protocol (SIP) [69] and the H.323 standard [35]. While H.323 is based on previous PSTN architectures, SIP was created following the guidelines of the HTTP protocol, with a request-response model, directly focused on the Internet architecture. As such, it re-uses as many existing Internet protocols and elements as possible, and where new items are needed, it tries to keep them at a minimum, and as simple as possible.

SIP [69] was originally designed by Henning Schulzrinne (Columbia University) and Mark Handley (UCL) back in 1996. Since then, it has been increasing in success and acceptance among the VoIP community and in November 2000 it was adopted as the 3GPP signalling protocol and a permanent element for the new IP Multimedia Subsystem (IMS) [15], [75].

The idea behind SIP is to provide a simple, lightweight means for creating and ending connections for real-time interactive communications over IP networks - mainly for voice, but also for video conferencing, chat, gaming or even application sharing. Hence, it is specifically focused on call control and nothing but call control: It does *not* implement or control QoS, mobility management, media synchronization and/or mixing (e.g. such as in a multi-conference), or in general, any kind of application-specific media processing. SIP is only a tool that applications can use to support more complex functionality, like implementing a video downloading service through the Internet. But SIP only provides the call control part, nothing more, nothing less.

SIP packets are generally called *messages*. Every SIP request messages carries a *method* indicating the request type, and the corresponding response carries a *status code* indicating the answer. The original SIP specification had only six methods, which gives an idea of its commitment to simplicity. There is a large number of status codes, built along the lines of HTTP: each consists of a 3-digit code, the first indicating the general kind of answer, and the last two giving more concrete information, such as: 183,

1 - Informational response, 83 - Session in progress; 400, 4 - Request failure, 00 - Bad request. The most common answer to any request, if everything worked properly, is 200 OK.

The six original request methods were:

- **INVITE**: Used for initiating a session. Its two most interesting characteristics for this study are:
  - That it carries the session description parameters in its body during the negotiation phase, such as which codecs the user can support, which media types, etc.
  - That sending further INVITEs after the session has been established, with new session parameters, serves to re-negotiate the session characteristics among the peers
- **ACK**: Confirms the session establishment, à la TCP. It can only be used with INVITE.
- **BYE**: Terminates a session.
- **CANCEL**: If an INVITE has been sent, but a response is still pending, CANCEL serves to cancel the pending INVITE.
- **OPTIONS**: It serves to enquire about a peer's capabilities prior to negotiating the session characteristics
- **REGISTER**: This method is used for binding a permanent address (i.e., a user's SIP identifier) to a current location (i.e., an IP address).

Of these six methods, mostly the INVITE is of relevance here[1]. In the body of this message, all the relevant session parameters are transported during the set up phase. Paramount among them are the codecs supported by every partner and the corresponding sampling rates, the media types (e.g. audio and/or video), and the IP addresses and port numbers to which media packets shall be addressed. The corresponding 200 OK response carries the callee's selection of codecs. Once this negotiation has taken place, the ACK finishes the set up and media packets can be interchanged. Figure 2.3 presents a schematic version of a basic SIP call.

If due to a change in the communication characteristics, such as the addition of a new media stream (e.g., a video stream on top of an existing audio stream - from audio-conference to video-conference), the session parameters should be re-negotiated, a new INVITE-OK-ACK cycle is started, without interrupting the existing media streams. It

[1]We will explain in chapter 6 a mechanism using the SUBSCRIBE, NOTIFY and OPTIONS methods of SIP for the communication between the SIP Proxy and the nodes.

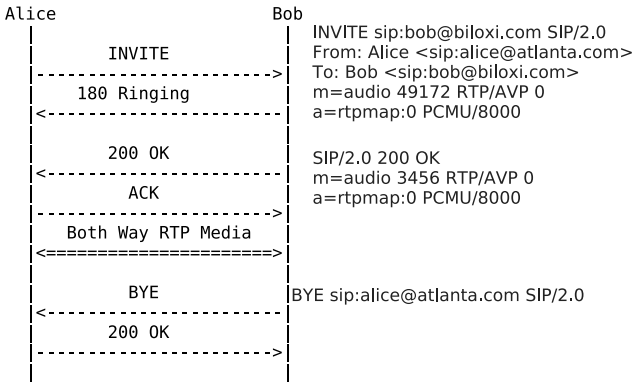


Figure 2.3: Basic SIP call

is this characteristic that can be used in the context of a multi-rate scenario to react to a rate change by negotiating a new codec, more suitable for the new conditions.

Although SIP is an end-to-end protocol, and hence does not mandate the usage of any network elements, any VoIP architecture beyond the trivial one needs at least one node to find and authenticate users and to route calls: The SIP proxy. In the most common VoIP architecture, the SIP proxy receives all signalling messages from both participating peers, which gives it a privileged overview of the session state. In some cases, the proxy can even control the interchange of the data packets themselves. As will be seen throughout this study, taking advantage of these characteristics, together with an adequate positioning of the SIP proxy co-located at the 802.11 access point can facilitate the implementation of very efficient solutions to the multi-rate issue.

Summarizing, SIP is a request/response session control protocol. The session characteristics are negotiated during set-up with the help of INVITE messages. Further INVITEs during the life of the session allow for updating the session parameters, including the codecs used, which can be used to alleviate the effects of a rate change in 802.11 environments.

2.1.3 Media transmission on the Internet: RTP/RTCP

The media transport protocol of choice for most multimedia applications in the Internet nowadays is undoubtedly the Real-time Transport Protocol, RTP. Originally specified in RFC 1889 (and updated in RFC 3550) [70], it was designed together with its companion protocol, the Real-Time Control Protocol, RTCP. The role of this second protocol was to provide feedback on communication quality to the users, so that they could react

accordingly. The most critical aspect of both protocols is that even if they provide extensive information on QoS, they themselves *do not* use that information in any way: They simply collect and transport it, and it is up to the controlling applications to implement the corresponding adaptation mechanisms. In other words, although they provide the means for detecting any potential problem of the monitored session (e.x. increased packet loss or a jitter delay higher than normal), they provide no solution to this problem whatsoever. That is one of the reasons why some multimedia architectures bypass RTCP and implement themselves their own mechanisms and even protocols, tailored to their exact needs.

The rationale behind defining a new transport protocol for real-time multimedia data over IP networks derives from the strict requirements in terms of QoS, that neither TCP, nor UDP could fulfill. VoIP traffic necessitates an end-to-end delay of no more than 150 ms for good voice quality, together with a strictly bounded jitter. The reason is that interactive voice requires the sampling of the signal at very short intervals, and their reproduction at the exact same rate. Hence, beyond the samples reaching the destination in a short period of time, to avoid the "walkie-talkie" effect, packets must arrive almost periodically. Furthermore, although low packet loss ratios are not highly problematic for VoIP communications (the user simply hears a short "click"), this ratio must still remain below the 5% mark for understandability. These strict requirements could be achieved neither by TCP nor by UDP for different reasons:

- TCP: In the case of TCP, mainly its use of retransmissions for reliability provokes an unspecified, unbounded delay in the reception of data. Furthermore, its phases (slow start, congestion avoidance, fast retransmit) further increase the variability of both delay and jitter. Hence, TCP is not recommendable for use in real-time communications.

- UDP: UDP does not present the above limitations, but is equally unsuited. Its purely best-effort nature brings with it a lack of mechanisms to detect packet loss, out-of-order packet delivery (Figure 2.4) and unbounded delay and jitter. Hence, it is also not suitable for periodic real-time packet transmission.

RTP brings the solution by complementing UDP's weaknesses while traveling on top of it. Basically, RTP encapsulates one or more voice samples and adds a sequence number and a timestamp to it. With the sequence number, data loss can be detected (although no retransmission will be requested, obviously) and these packets are skipped during retransmission. Furthermore, the sequence number also guarantees in-order playback. The timestamp permits adequate playback synchronization, even in the presence of packet loss, as well as delay and jitter calculation for QoS measurement purposes. It is precisely its simplicity, while covering UDP's main weaknesses, what has permitted RTP's widespread success.

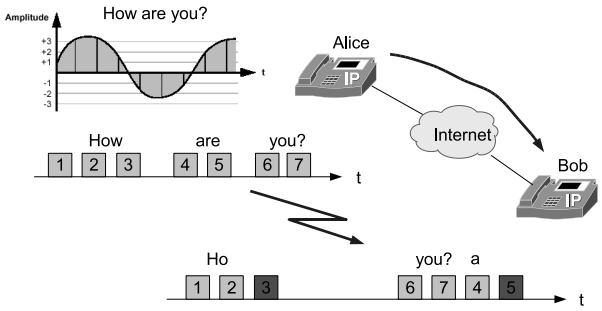


Figure 2.4: Network distortion on VoIP transmission

RTCP, on the other hand, provides extra information at the cost of more complexity. But since RTCP does not transport user data, it can accept non-real-time delay and processing times. RTCP packets contain so-called *reports*, which contain the actual session QoS information, from the point of view of both the sender and the receiver. This information includes:

- A timestamp, for clock synchronization purposes, as well as delay and jitter calculations
- The sender's and receiver's packet and octet count, for integrity checks
- The percentage as well as the absolute number of packets lost during the session
- The highest sequence number received, to help detect lost data
- The interarrival jitter

These RTCP packets or reports are periodically interchanged among the partners; however, in order not to waste too much capacity on them, their transmission frequency is calculated so that RTCP traffic does not exceed 5% of available capacity. With this information, as has been previously stated, both the sender and the receiver can obtain an accurate picture of the session's QoS state. However, it must be recalled that neither RTCP, nor RTP take any correcting measures, since this could take too much time and/or be inadequate for a specific application. Hence, it remains solely the application's responsibility to redress any quality degradation.

Codec	Bitrate (B)	Packet size (L)	MOS score	Ie(0% loss)
G.711	64 Kbps	160 B	4.1	0
G.726	32 Kbps	80 B	3.85	7
G.729A	8 Kbps	20 B	3.7	11
G.723.1	5.3 Kbps	20 B	3.6	15

Table 2.1: Codecs Parameters

A last question still arises: How must QoS for VoIP traffic be defined? Can the user's perception of "good" or "bad" quality be accurately mapped to technical parameters like delay, jitter or packet loss? This was proven an elusive goal, and the next section reviews the most common indicators used to bridge that gap.

2.2 Measuring quality for VoIP

Call quality can be measured using subjective testing or instrumental monitoring. In subjective testing methods, humans are asked to evaluate the quality of the service according to a standardized process and give a score, typically from 1 to 5 (MOS score). These kinds of tests are time consuming because many users have to be asked. That is why in the last few years more efforts have been focused on instrumental measurement tools, like the ITU-T E-Model. Both methods are described briefly here.

a) Mean Opinion Score - MOS

One of the most used tools for measuring voice quality is the Mean Opinion Score or MOS. Defined in ITU-T P800 standard [60], MOS is a tool based on subjective testing, where a number of users are asked to listen to a voice sample (corresponding to a particular codec) and give a score for the received media as they perceive it after the transmission. The MOS score, calculated from the average of the users' scores, ranges from 1 for an unacceptable call to 5 for an excellent call. A typical range for acceptable Voice over IP quality would be from 3.5 to 4.2.

The relationship between the MOS score and some of the most used codecs, can be seen on the codec Table 2.1 [22].

Although it may be the best known voice quality tool, MOS is difficult to implement since human intervention is necessary (although estimates of voice quality can be made by automatic test systems). That is why ITU-T proposed a few years ago the E-model.

b) E-Model

The E-Model is a planning tool for estimating the overall quality of a telephone network. ITU recommendation G.107 [28] introduced it with the objective to determine a quality rating that incorporated the "mouth-to-ear" characteristics of a speech path. The output of an E-model calculation is a single scalar, called R factor, derived from delays and equipment impairment factors. Once the R factor is obtained, it can be

mapped to an estimated MOS.

The R factor can be obtained through the following expression:

$$R = Ro - Is - Id - Ie + A \tag{2.1}$$

where Ro represents the basic signal-to-noise ratio (SNR), Is represents the combination of all impairments which occur simultaneously with the voice signal, Id represents the impairments caused by delay, Ie represents impairments caused by low bit rate codecs, the so-called "equipment impairment factor", and A is the advantage factor, that corresponds to the user allowance due to the convenience in using a given technology. This means that a telephone call quality is judged different by a user if the advantage of access and use of this technology can recompense for the lower quality (cellular phone users do not expect the same call quality as in PSTN calls).

Cole and Rosenbluth [23] give the following simplified expression for calculating the R-factor for VoIP, talking into account that many of the factors of the above expression can be simplified to a default number:

$$R = 94.2 - Id(Ta) - Ie(codec, loss) \tag{2.2}$$

where Id is a function of the absolute one-way delay (Ta) and Ie is, in short, a function of the used codec type and the packet loss rate. In Table 2.1, the Ie values of some of the standard codecs and considering 0% packet loss can be seen, as provided in Appendix I of [29]. For further details of the Ie/Id calculation refer to the work in [31].

After calculating the R factor, the equivalence between R and MOS can be determined as follows:

$$MOS = \begin{cases} 1 & R < 0 \\ 1 + 0.035R + 7.10^{-6}R(R-60)(100-R) & 0 < R < 100 \\ 4.5 & R > 100 \end{cases}$$

User Opinion	R Factor	MOS score
Very Satisfied	90 - 100	4.3 - 5.0
Satisfied	80 - 90	4.0 - 4.3
Some Users Satisfied	70 - 80	3.6 - 4.0
Many Users Dissatisfied	60 - 70	3.1 - 3.6
Nearly All Users Dissatisfied	50 - 60	2.6 - 3.1
Not Recommended	0 - 50	1.0 - 2.6

Table 2.2: Equivalence of R-factor and MOS score

This method of obtaining the MOS value through the R-Factor equivalence, has the advantage of permitting a real-time calculation of the actual instant quality of service

perceived by the user, as opposed to a fixed MOS value depending on the codec used and not taking into account factors that may vary during the call. Hence, the R-Factor will be used for obtaining the MOS in the simulation experiments presented later in this thesis, using the procedures described above and the loss/delay data gathered from the RTCP reports.

2.3 Conclusion

A VoIP system is composed by a number of different elements, starting from the human voice and ending to the actual codified signal and the protocols for its transmission over the network. From these elements, only some are of interested in this study and have been reviewed here briefly. We have introduced the most important protocols of a VoIP session, namely the SIP for session control and the RTP/RTCP for media transport and quality feedback. We have seen three basic codec categories and the characteristics of the ones used in the thesis. Finally, two of the most known quality of service measurement tools have been introduced, the Mean Opinion Score MOS and the E-Model specification, as also the relation between the two.

The basic VoIP elements are mostly independent of the technology used for the actual transmission of the voice packets. However, WLAN networks, which is the scenario under study in this thesis, introduce new challenges for deploying a successful VoIP services due to their nature and lack of QoS guarantees for the sensitive voice traffic. Thus in the next chapter we will review this technology and especially the de facto standard in the area of wireless networks, the IEEE 802.11, pointing out its limitations for achieving QoS for VoIP.

CHAPTER 3

VoIP over IEEE 802.11 WLANs

3.1 A hotspot scenario

A wireless cell is the coverage area provided by a single access point, meaning the geographical area where both the access point (AP) and the mobile stations (STAs) (also known as mobile nodes - MN) can communicate using the radio channel with an acceptable minimum quality. This quality can be usually measured in terms of Signal-to-Noise Ratio (SNR) or other derived metrics. In [1] this area is referred to as a Basic Service Set (BSS) while an Extended Service Set (ESS) contains multiple access points and their coverage areas. All or part of these coverage areas can overlap, so that a mobile station can select the access point to use. Typical scenarios with this configuration are found in public areas (like cafeterias, hotels, parks, airports), company buildings or home premises, where users can access the Internet from their notebooks or PDAs. In all these scenarios, the WLAN technology provides a certain grade of mobility and a broadband access to Internet at very low cost.

A wireless LAN can be deployed either in an infrastructure mode or a peer-to-peer (ad hoc) mode. In a typical enterprise environment, WLAN APs are deployed in an infrastructure mode such that all of them advertise the same Service Set Identifier (SSID) thus together forming an Extended Service Set. WLAN client stations need to associate with one of the APs to get connected to the network. In a peer-to-peer to mode, WLAN STAs can associate with each other and communicate between themselves without any AP support.

The scenario under study in this thesis is the one known as a Hotspot, composed by a single cell Access Point and a number of mobile stations connected to it in an infrastructure mode. The AP acts as a gateway providing the wireless nodes with

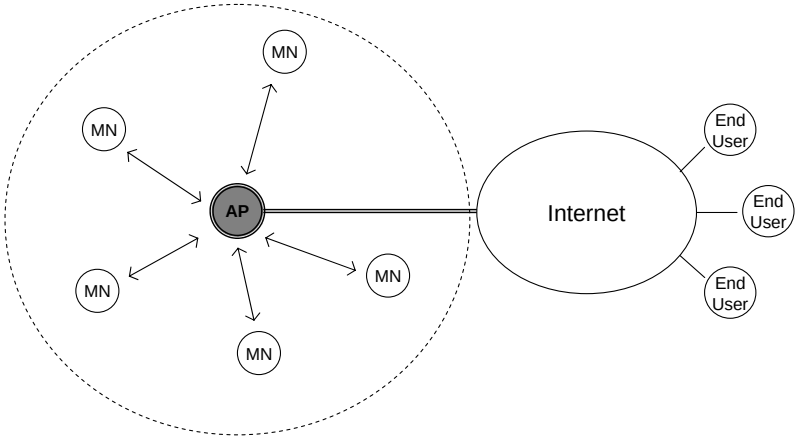


Figure 3.1: The Hotspot scenario

access to the fixed network, as depicted in figure 3.1. The STAs and the AP use the DSSS PHY specifications in the 2.4 GHz band. Ideal channel conditions are assumed, i.e., no packet is lost due to channel errors or the hidden terminal phenomenon. It is also assumed that the data rate used is the same in the uplink and downlink flows of the call, this is, the channel is assumed to be symmetric and the same SNR is observed from the AP and the STA.

In this chapter we describe briefly the IEEE 802.11 standard, the de facto standard for wireless access and thus for Voice over WLANs, emphasizing on the mechanisms that are of relevance to the rest of the study. We start with an overview of the common channel access functions, the DCF (Distributed Coordination Function) protocol found in the original 802.11 standard [1] and the EDCA (Enhanced Distributed Channel Access) introduced in the 802.11e standard [2] offering QoS enhancements. Additionally, a comprehensive analysis is performed of the challenges that VoIP services encounter over WLANs and in particular the reasons for the low VoIP capacity (in terms of maximum number of simultaneous active calls) in this kind of environments. We overview the Call Admission Control mechanism, necessary for addressing this limited capacity issue, and some of the most known CAC proposals found in the literature. We finally study one particular characteristic of the wireless networks, the Link Adaptation mechanism that leads to a capacity variable, multi-rate environment, and the effects of this in VoIP traffic, as this can be considered the basis of this thesis.

3.2 IEEE 802.11 background

The IEEE 802.11 group of standards [1] specifies a common medium access control (MAC) Layer, which provides a variety of functions that support the operation of 802.11-based wireless LANs. In general, the MAC Layer manages and maintains communications between 802.11 stations and the Access Point by coordinating access to a shared radio channel and utilizing protocols that enhance communications over a wireless medium. To control the nodes' access to the shared medium, 802.11 uses a CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) MAC protocol, which can be considered the wireless version of the wired well-known CSMA/CD (with Collision Detection). Moreover, mechanisms such as an ARQ (Automatic ReQuest) protocol and an adaptive PHY layer, which are able to adapt to the channel conditions, are introduced in the IEEE 802.11 protocol stack to mitigate the channel impairments.

3.2.1 The DCF MAC protocol

In IEEE 802.11, the data link layer (layer-2) is split in two sub-layers: the Link Layer Control (LLC) and the Medium Access Control (MAC). The LLC sub-layer basically implements a packet fragmentation function, which adapts the packets to the packet length required by the MAC/PHY layers, and the ARQ (Automatic ReQuest) protocol, which tries to mitigate the high channel-error rates by re-transmitting the erroneous packets. The MAC sub-layer on the other hand, governs the transmission attempts over the shared channel between the set of active stations and the Access Point.

Two functions can be found in the 802.11b standard, responsible for the coordination of the shared medium access: the Distributed Coordination Function (DCF) and the Point Coordination Function (PCF). PCF implements a polling mechanism that allocates and reserves channel resources for a specific station so that it can better provide QoS guarantees. Although it can be considered more appropriate for a traffic with strict QoS demands, like VoIP, it will not be considered in this study since its implementation in an AP and/or client cards is optional and as a result most of the commercially available 802.11b APs do not support PCF.

The DCF is based on the CSMA/CA MAC protocol. A station wishing to transmit has to first listen to the channel for a predetermined amount of time, called DIFS (Distributed Inter Frame Space), so as to check for any activity on the channel. If the channel is sensed "idle" then the station is permitted to transmit. If the channel is sensed as "busy" (i.e. some other station is transmitting) the station has to defer its transmission. In this case, the station must wait a random period of time before attempting to access the medium again. This ensures that multiple stations wanting to send data do not transmit at the same time. A random binary exponential back-off (BEB) algorithm is used for this purpose, designed to provide long-term fairness in

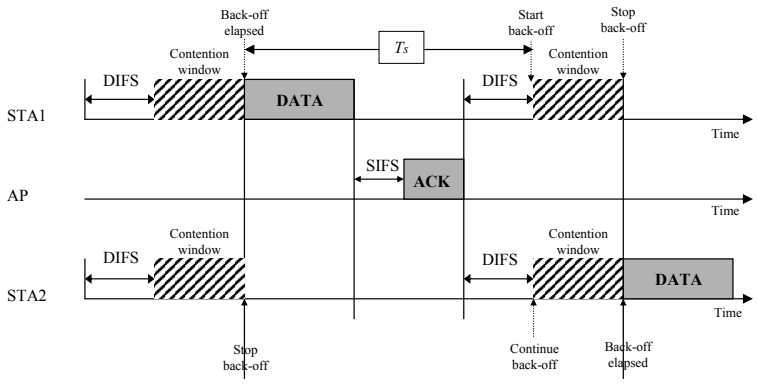


Figure 3.2: DCF Medium Access Control

terms of the service received by each node, distributing the channel capacity among all the active nodes, including the AP. The back-off timer significantly reduces the number of collisions and corresponding retransmissions, especially when the number of active users increases. For a clear and detailed explanation of the DCF protocol, including its performance analysis, refer to [12].

In Figure 3.2 the basic operation of the DCF MAC is depicted. Basically, to schedule a packet transmission, each STA has a counter which is decreased by one at each SLOT time (period of time that the channel is detected idle). Notice that the backoff counter (BEB counter) is not decreased if other STAs are transmitting as the channel is detected busy. When the counter reaches zero, the packet is transmitted over the channel. If all goes well, the packet will be received by the destination node, which answers with a level-2 ACK frame to confirm its correct reception after waiting for a Short Inter Frame Space (SIFS). However, if two or more nodes transmit at the same SLOT time, a collision occurs and the packet must be retransmitted until the maximum number of retransmissions is reached. The same treatment is done if the packet is received erroneously, as in both cases no ACK is transmitted (this is of crucial importance when rate adaptation mechanism based on monitoring the correct packet transmission are used, as there is no way to differentiate between a collision and channel errors). Therefore, there are several factors, which actually increase the time required to successfully transmit a packet: the time spent in back-off, the time spent in re-transmitting the packet after a collision or a channel error and the time required to transmit the corresponding ACK to each transmitted packet.

Setting T_s the time spent in transmitting a packet and the corresponding ACK, this will be

$$T_s(R^d) = \left(H + \frac{H_{mac} + L_{data}}{R^d}\right) + SIFS + \left(H + \frac{L_{ack}}{R^b}\right) + DIFS \qquad (3.1)$$

where H is the time to transmit all the PHY headers (preambles) at $R_{PHY} = 1$ Mbps, R^d is the data rate used from the set of data rates $\mathbb{R}^d$ and R^b is the basic rate used from the set of basic rates, $\mathbb{R}^b$. The set $\mathbb{R}^d$ of available data rates in 802.11b is $\mathbb{R}^d = \{1, 2, 5.5, 11\}$ Mbps. Notice the dependence of T_s with the packet length (L_{data}) itself but also with the transmission rate R^d used. For further details about this equation refer to [11].

3.2.2 QoS enhancements: EDCA

One of the most important problems of DCF is the failure to provide traffic differentiation, a key issue in order to guarantee QoS requirements, which makes difficult the coexistence between sensitive and best-effort flows.

The IEEE 802.11e [2] standard was released at the end of 2005 to fulfill the requirements of traffic differentiation and QoS provision in WLANs. The DCF was enhanced with the EDCA (Enhanced Distributed Coordination Access) which is able to satisfy the traffic differentiation requirement by classifying the packets in different categories, called Access Categories (ACs). Each AC has a different channel access priority by considering different MAC parameters. In Table 3.1 the MAC parameters of each queue are shown. For detailed explanation of these parameters and the role of each one in the EDCA mechanism see also [50] and [20].

AC	$AIFSN_j$	$TXOP_{limit}$ (ms)	$CW_{min,j}$	$CW_{max,j}$
0 (Background: BK)	7	0	31	1024
1 (Best effort: BE)	3	0	31	1024
2 (Video: VI)	2	6.016	15	31
3 (Voice: VO)	2	3.264	7	15

Table 3.1: Default EDCA Parameter Set values

The combination of these MAC parameters provides a higher priority to access the channel for the real-time traffic access category (low CW_{min}, high $TXOP_{limit}$ for the AC_VO queue). Conversely, the data access categories suffer from a low priority to access the channel (high CW_{min} values and high AIFS values for the AC_BK). Thus, EDCA is able to provide protection for the real-time traffic, increasing the number of possible VoIP calls in a WLAN hotspot, especially when the channel is shared between VoIP calls and best-effort (TCP) traffic. This example is depicted in Figure 3.3.

Furthermore, the IEEE 802.11e standard provides other mechanisms to make more efficient the channel transmission. For example the consideration of different ACK policies, with the new Block ACK (a single special ACK packet can acknowledge several frames) and the No ACK, which avoid the transmission of ACKs for services where the

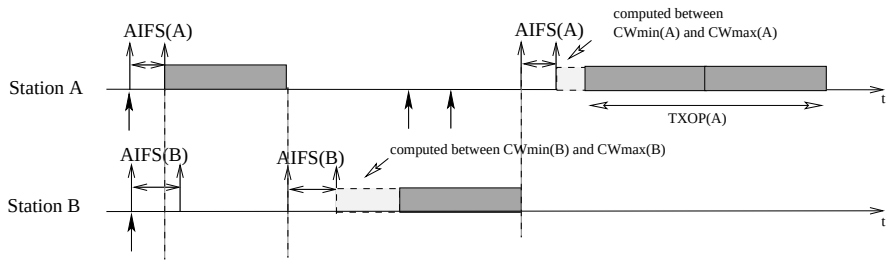


Figure 3.3: EDCA Medium Access scheme. In this example station A has higher priority than station B

information on if a packet has been received is unnecessary, since the packet will not be retransmitted (for example, in VoIP applications). These enhancements where not considered further in this study, however a discussion on the benefits of using No ACK for VoIP in ad-hoc networks can be found in [7].

The IEEE 802.11e standard also provides the basic interfaces and signaling mechanism to implement a Call Admission Control (CAC) mechanism on top of the EDCA. The CAC is responsible for the resources management, in other words to decide whether a new flow could be accepted or not. The decision is made before a voice call is established and is based on whether the required network resources are available to provide suitable QoS for the new call. However, the standard does not specify how to implement it and the policies to accept of reject a new flow, being one of the current open issues in the deployment of a successful VoIP service in WLANs, especially in heterogeneous traffic scenarios. As explained next, the capacity of an 802.11 cell in terms of VoIP calls is very limited due to a number of different factors, thus a CAC algorithm is necessary for the efficient distribution of the scarce network resources. After explaining the most common limitations of VoIP over WLANs we will review some of the existing proposals on CAC algorithms and see why, although necessary for resource management, these are not sufficient to solve the problems caused by the multi-rate capacity variations.

3.3 Voice Capacity limitations in 802.11

The WLAN capacity for VoIP calls has been a hot topic on research during the last years, as VoIP is expected to be one of the killer applications which boost the public WLAN use. The network capacity in terms of VoIP calls is defined as the maximum number of bi-directional calls that can be supported by one AP in a WLAN cell while maintaining acceptable QoS levels. When trying to determine an upper bound for this capacity, a first simplified calculation might lead us to the erroneous assumption that, given the 11Mbps maximum rate of 802.11b and the 128Kbps needed for the two flows of a bidirectional call using the G.711 codec, approximately 85 calls could

	PHY Rates (Mbps)			
VoIP Codec	11	5.5	2	1
G.711	12(11)	10(9)	6	3
G.726	13(12)	11(10)	7	5
G.729	14(12)	13	9	6
G.723.1	22(19)	19(17)	14(13)	10

Table 3.2: Maximum number of calls for each data rate

be supported. Additionally, one may think that this number could increase further by using a lower bitrate codec like the G.729, leading to a few hundreds of simultaneous calls possible. However, a number of factors arising from the specific characteristics and functioning of the 802.11 networks and its inherent inefficiency result into an actual number surprisingly much lower than this.

In [32] and [34] Garg and Kappes use both an experimental and an analytical method to calculate the upper bound of the number of VoIP calls in a cell. This upper bound turns out to be very low, with only 6 calls supported using a G.711 codec and a 10ms packetization interval (12 with a packetization interval of 20ms). A number of further studies like the ones by Hole and Tobagi [39] and Wang et al [80] validate these results.

As a reference to compute the VoIP capacity in WLANs, Hole and Tobagi [39] present an upper limit (but enough accurate, with +0/1 VoIP call error from the exact maximum number of calls) for the number of simultaneous calls in an infrastructure WLAN. To calculate this, they take into account the header overheads of each layer (RTP, UDP, IP, MAC and physical layer) as also the MAC protocol contention access mechanism, including the backoff procedure and the DIFS, SIFS and ACK transmission times. Their model is the subject of the section 5.2 since the proposed capacity index can be easily adapted for our multi-rate/multi-codec scenario and we will further use it for calculating the cell capacity in our experiments.

This upper limit is shown in Equation 3.2:

$$N = \left\lfloor \frac{1}{\frac{B}{L}[2T_s(R^d) + (T_{slot} \cdot \frac{CW_{min}}{2})]} \right\rfloor \tag{3.2}$$

where T_s was introduced in Equation 3.1, B/L is the rate of VoIP packets from the source and T_{slot} is the duration of an empty SLOT. The VoIP capacity provided by this upper limit for different VoIP codecs is shown in Table 3.2 (considering the IEEE 802.11b [1] MAC parameters). Note that ACK packets can be transmitted using either the basic rate (R^b) or the data rate (R^d). The values appearing on the table in parenthesis are obtained when ACK is transmitted using $R^b = 1$Mbps, while the ones outside are obtained using the data rate R^d, equal to the PHY rate of the node.

The obvious question arising from the analysis above is what is actually provoking this low VoIP capacity in 802.11 networks. Some factors for it, such as the downlink

starvation, the inefficiency due to large overheads, the simultaneous coexistence with TCP flows and the multi-rate channel are reviewed briefly next. However, the first three problems have been widely analyzed in previous research works, plus the EDCA QoS extensions can manage to solve some of them; thus, the emphasis of this thesis falls on the multi-rate channel effect, explained in detail in section 3.5

3.3.1 Downlink starvation

Considering the case of VoIP bi-directional traffic from several calls in a hotspot scenario, the AP serves as a concentration point and gateway to the wired network for all outgoing and incoming calls. While each node only sends a single VoIP stream (the one going to the wired network), the AP has to send out all streams coming from the wired in the wireless network. Thus, the AP is actually sending a 50% of the overall load of the network [9][32][61].

Due to the fair channel access of the 802.11 however, it is competing with the rest of the nodes in equal terms to access the common channel. Thus it tends to be saturated rapidly, acting as a bottleneck for all transmissions.

EDCA allows to mitigate this effect by using AIFS-1 values at the AP compared with the values of the STAs, so giving to it a higher priority to access the channel. See [17], where some elaborated EDCA parameters tuning algorithms which contribute to mitigate this effect are evaluated. We will not deal further with this problem in this thesis although the effects of it are considered at the justification of the results presented later.

3.3.2 Inefficiency due to large overheads

As the short VoIP packets traverse the various layers of the standard protocol stack, the header overhead is growing. A typical VoIP packet includes a payload between 10 and 30 bytes, depending on the codec and packetization interval used (for example, using the G.729 or the G.723.1 codec, although the payload can be larger using codecs such as G.711). The IP/UDP/RTP headers add a total of 40 bytes, so already the efficiency is decreased to 50%. However, the 802.11 MAC/PHY layers have an additional overhead due to the physical preamble, the MAC header, the backoff time, the ACK transmission time and the interframe times between the transmission of packets and ACKs. As a result, the overall efficiency can drop to lower than 3% [80]. A visual representation of this can be seen in Figure 3.4.

3.3.3 Coexistence with TCP flows

A TCP flow is characterized by trying to send packets continuously and increasing its throughput using all available channel bandwidth until all the associated data have been

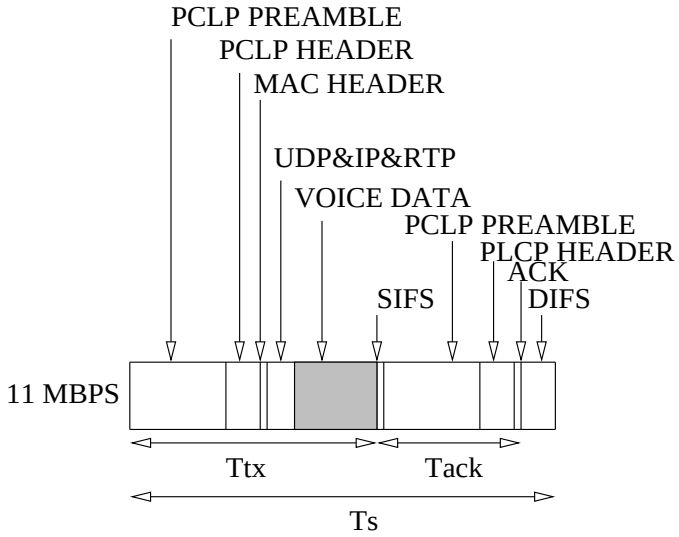


Figure 3.4: The voice payload size vs the total voice packet transmission time

transmitted. This behavior makes a node with an active TCP flow a fierce competitor for the channel resources as it will try to use all the shared bandwidth. In [10], Bellalta et al show how with only a few downlink or uplink TCP flows, the impact over the VoIP capacity is critical since with only $1-2$ uplink / downlink flows all VoIP calls are starved.

In order to provide better QoS for multimedia traffic in a network where elastic and inelastic traffic coexist, some solution based on service differentiation mechanisms is needed. This can be achieved using different priority queue management schemes and/or using different MAC parameters for different classes of traffic. Some reference works, although many more studies can be found in this area, are the studies presented in [26], [3] and [82], which all propose an enhancement in 802.11 DCF MAC to include service differentiation.

Based on the results of their work, the classification of the traffic in the four different Access Categories was standardized in 802.11e (Table 3.1). Thus EDCA mitigates the problem by providing traffic prioritization at each node: when a VoIP and TCP packets compete for access to the channel, the probability to gain the contention for the VoIP packets is higher. Therefore it provides a good solution to integrate data and VoIP traffic in the same Hotspot. Moreover, EDCA allows to modify its MAC parameters in running time, so adaptive solutions could be used in order to improve simultaneously both the VoIP protection and the TCP performance, like the adaptive EDCA solution presented in [17].

However, the protection that the Access Categories provide is not enough. As explained in [62], there is still a problem described as the “spill over” effect: when traffic is overloaded in one AC, performance on other ACs will also be affected. Although in this

thesis we focus mostly on a purely VoIP scenario, we will review briefly the problems arising from VoIP with TCP coexistence in chapter 4. We will see that a combination of our codec adaptations solution with some EDCA parameter tuning mechanism can be beneficial for alleviating the multi-rate effects in an heterogeneous traffic scenario.

3.3.4 Multi-rate channel

The 802.11 standard contemplates the possibility for the nodes to adapt their physical rate. To do this the nodes use a Link Adaptation (LA) algorithm, which chooses the most suitable of the available rates in order to maintain an acceptable packet error rate level. Data bits are transmitted with different modulation schemes to achieve the optimal channel capacity under a certain channel condition. In table 3.3 the available modulation schemes and physical rates in IEEE 802.11b are summarized. This mechanism can be very useful under time-varying channel conditions and error-prone channels, for reasons such as user mobility, path loss or interference among others. Link adaptation permits the AP to increase its transmission range so that users that are further away and/or perceive worse signal can still connect using lower data rates (more robust modulation schemes).

PHY rate (Mbps)	Code Length	Modulation	Symbol Rate	Bits/Symbol
1	11 (Barker Sequence)	BPSK	1 MSps	1
2	11 (Barker Sequence)	BPSK	1 MSps	2
5.5	8 (CCK)	QPSK	1.375 MSps	4
11	8 (CCK)	QPSK	1.375 MSps	8

Table 3.3: IEEE 802.11b Data Rate Specifications

However these rate changes provoke a variation in the perceived channel capacity not only for the node that adapts its transmission rate but for all the active nodes of the channel. In other words, the change of one of the nodes to a lower rate can also affect negatively the nodes transmitting in higher rates. This effect, also known as "multi-rate anomaly" was first analyzed by Heusse et al [37] and is the starting point of the study presented in this thesis, thus it will be next analyzed in more detail in section 3.5. But first, we will have a look on the mechanism most commonly used for managing this limited VoIP capacity in the WLANs: the Call Admission Control.

3.4 Call Admission Control proposals

We have seen that a number of WLAN-specific mechanisms limit significantly the available capacity of an 802.11 cell in terms of VoIP calls. Given this limited capacity for VoIP flows, Call Admission Control techniques are necessary. Nevertheless, only in the last 802.11 standard, the 802.11e, the CAC signaling was defined and still no specific

implementation was standardized, neither the specific policies for accepting or rejecting a new flow, giving space to proprietary solutions to develop.

Due to the arising necessity for an effective Call Admission Control in WLANs, various such algorithm proposals have been discussed in the past years and can be found in the literature, usually divided in two categories depending on the method they use for calculating the available cell capacity: measurement-based or model-based [30]. Measurement-based CACs take into account current network status based on actual measurements, like delay or throughput. Model-based (or calculation-based) CACs on the other hand compute specific metrics or criteria in order to forecast the status of the network once a new flow is accepted.

An example of the latter can be found in [62]. A prediction of the achievable throughput in the case that a new call is admitted is used in the model-based CAC proposed by Pong et al. [62]. Their model is based on the Bianchi model [12], EDCA parameters adaptation and the collision statistics for each flow.

There is an added difficulty in using CAC for VoIP in presence of data traffic, characterized by a greedy nature which leads VoIP flows to starvation. The task of finding a mathematical model, which captures the joint behavior of TCP flows in presence of VoIP calls complicates. The model based admission control presented by Bellalta in [10] is one of the few model-based CACs that include heterogeneous traffic, where the joint VoIP and data performance is evaluated.

Most of the CACs used for an heterogeneous traffic scenario are measurement-based, using some variation of a channel occupation index. In [83] the metric used as the control metric to decide whether to accept or not a new call is called *channel busyness ratio*, defined as the ratio of the time the channel is determined to be busy to the total time. In [33] Garg et. al. propose the use of a similar *Channel Utilization Estimation* (CUE) index, defined as *the fraction of time per time unit needed to transmit a flow over the network*. For each flow its CUE is estimated, then added to the actual CUE_{Total} of the cell and then it is compared to the cell's $CUE_{TotalMAX}$. If this is smaller then the new flow can be accepted. If not a decision policy is consulted to determine whether existing *data flows* can be curtailed so as for the new call to enter.

An extensive survey of the common CAC techniques in 802.11e can be found in [30].

The research on Call Admission Control methods is wide and much effort is placed on controlling voice QoS on a wireless cell and distribute the resources of it efficiently. However, none of these methods are actually dealing with the QoS of the voice flows once accepted at the network, or with the QoS degradation provoked by the capacity variable channel to the already *active* calls. The codec adaptation solution proposed in this thesis and explained in the next chapter, complements the work of CAC by dealing with the QoS of the calls once accepted in the cell. In fact, a combination of the two mechanisms is possible and will be discussed in chapter 5. But first, we will see how the

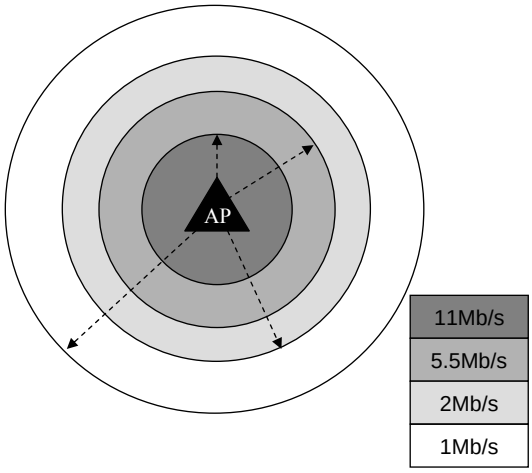


Figure 3.5: Transmission rate depending on distance from Access Point

link adaptation mechanism works and the reason for the variable VoIP capacity (i.e. the multi-rate anomaly) studied throughout this thesis.

3.5 Capacity variable channel due to the multi-rate mechanism

When a packet from the LLC/MAC layer is ready to be transmitted, it is sent to the PHY layer, where the bits are encoded and modulated to make possible their transmission over the wireless channel. Each MAC Protocol Data Unit (MPDU) is transmitted at a specific rate from the set of available rates $\mathbb{R}^d$, which are obtained by combining a modulation m and a channel codec rate c from the set of available modulations and coding rates for each STA and allowed at the BSS (see Table 3.3). The m modulation and c coding rate are announced in the PHY header (SIGNAL field), which is added to the MPDU to form the PHY Protocol Data Unit (PPDU). The PHY header, (PLCP preamble and the PLCP header) is always transmitted at the PHY rate, which is equal to $R_{PHY} = 1$ Mbps. Thus, the duration of that interval is equal to 192 μs. In the SIGNAL field of the PLCP header, the rate used for the MPDU transmission is indicated, which for the IEEE 802.11b comprise the rates $\mathbb{R}^d = \{1, 2, 5.5, 11\}$ Mbps[1].

The use of multi-rate PHY permits the increase of the WLAN's communications range Figure 3.5. Thus, the selection of a modulation and a codification rate satisfies

[1]Other considerations, such as the use of the short preamble, are included in the 2003 revision of the IEEE 802.11b standard and in the IEEE 802.11a and IEEE 802.11g PHY specifications.

the tradeoff between the coverage area and the data transmission rate (e.g. when there are bad channel conditions, the transmission rate is lower than in cases when the channel conditions are good). Note however that there are other factors that can provoke bad channel conditions apart from the distance from the AP, like meteorological conditions, interferences, etc. The mechanism that selects the proper rate based on information about the channel state is called the Link Adaptation (LA) algorithm.

In [48], the range achievable for each rate is shown. For example, considering the IEEE 802.11b DSSS specification, the 11 Mbps rate allows a range of about 50 meters from the AP and using the 1 Mbps rate the maximum distance from the AP is about 100 meters. Obviously, between 50 and 100 meters the other rates are scaled.

To select the transmission rate, the IEEE 802.11 standard does not specify a concrete LA algorithm, leaving this empty for proprietary solutions. However, the ARF (Auto-Rate Fallback) link adaptation mechanism [43] is the one most commonly used. ARF was the first link adaptation algorithm to be published. It defines that a user attempts to increase its transmission rate after a fixed number of successful transmissions at a given rate, or decreases the transmission rate to a lower one when a certain number of consecutive transmitted packets are detected erroneous by missing their respective ACKs. It is worth mentioning that a performance anomaly exists when consecutive packets are not received correctly due to the occurrence of consecutive collisions. This could cause an unnecessary change to a lower transmission rate, since the cause in this case is not the bad channel conditions. Although it is very improbable, the use of low CW_{min} values in the new EDCA standard for real-time traffic (for example, VoIP) makes this a non-negligible problem.

In terms of BSS performance, the use of multiple rates in the same cell introduces a resource management problem and leads to an interesting unexpected result: *STAs using low rates harm STAs using fast rates* [37]. This is due to the higher channel occupancies caused by the use of slow rates (as the application data packet length remains the same). A station with a lower physical rate consumes more channel resources (i.e. channel time) than a station with a faster rate in transmitting a fixed amount of data. Therefore, STAs transmitting packets at low rates have a higher channel occupancy than STAs transmitting at high rates, which reduce the number of packets that can be transmitted each second, increasing the packet losses and the packet transmission delay and jitter.

To give a simplified and understandable example, let us suppose that we have two STAs, both transmitting at a rate R. If we assume that both nodes use the same packet size L then the time each of them will occupy the shared channel will be $T \approx L/R$. If now one of the STAs starts transmitting at lower rate R' then the time that this will occupy the channel changes to $T' = L/R'$ where $T < T'$. Occupying the shared medium for a longer period of time means less available transmission time for the fast node,

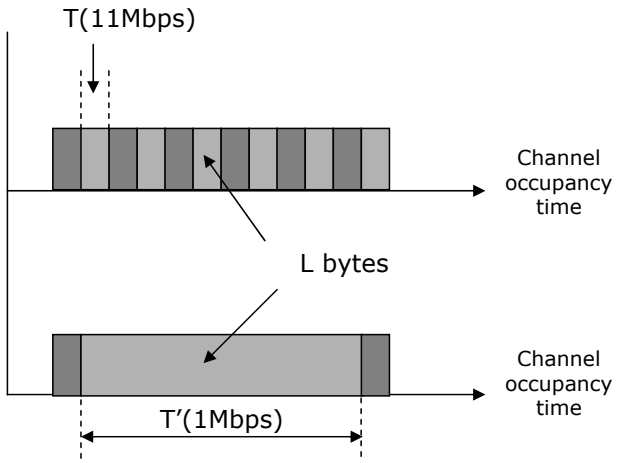


Figure 3.6: The multi-rate effect: channel occupancy when (a) both nodes use R rate and (b) when one node changes to rate R'

resulting in less number of total packets transmitted during the same period of time, even though the fast STA does not perceive any change in its channel conditions. Figure 3.6 presents this example; in the first case (a) both nodes transmit an equal number of packets each, while in the case (b) the slow node occupying the channel for longer affects the transmission of the fast node as well, which now transmits less packets at the same time.

In Table 3.2 it is assumed that all active calls use the same data rate. However, what would happen if some calls change their rate to a lower one? For example, choosing the G.711 codec, at a data rate of 11 Mbps, a maximum number of 11 calls can perform satisfactorily. However, if some of those active calls change to a lower rate, the maximum number of acceptable VoIP calls is reduced as the new system state will become unfeasible.

This is due to the higher relative bandwidth required by the calls which have changed to a lower rate, this is, the sum of all relative bandwidths will exceed the channel capacity and all calls start to suffer from congestion. The relative bandwidth is the real channel bandwidth required by a traffic flow. It is related to the VoIP codec and the instantaneous transmission rate used. VoIP calls using low transmission rates will require higher relative bandwidth values to transmit the same amount of voice data than calls using higher transmission rates.

Therefore, the VoIP capacity depends on the instantaneous set of rates used, fluctuating between the minimum number of active calls (at the lowest transmission rate)

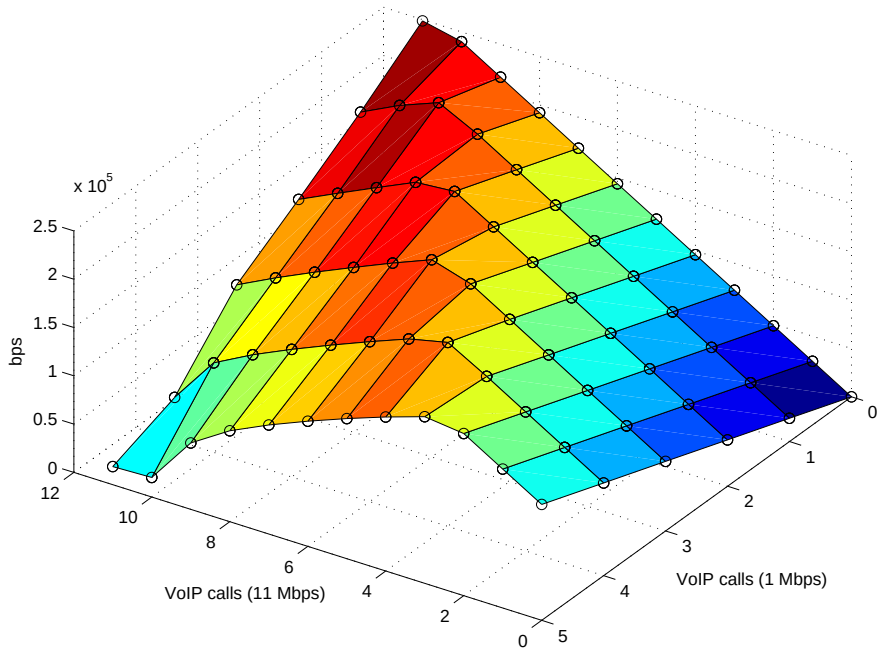


Figure 3.7: Maximum calls active at 11 Mbps / 1 Mbps rates

and the maximum capacity (at the highest transmission rate). To approximate the maximum number of calls when multiple rates are used simultaneously, we modified the upper limit shown in Equation 3.2 to:

$$\sum_{r=1}^{R} N_r \frac{B}{L} \left(2T_s(R^d(r)) + T_{slot} \cdot \frac{CW_{min}}{2} \right) \leq 1 \tag{3.3}$$

where N_r is the number of VoIP calls at rate r, B/L is the rate of VoIP packets from the source and $T_s(R^d)$ is the duration of a VoIP packet transmitted at rate R^d. This equation, modified for the multirate/multicodec scenario, is analyzed in detail in chapter 5. Equation 3.3 reduces to Equation 3.2 for a single transmission rate. For example, consider the possible combinations of VoIP calls using the G.711 codec and two rates: 11 Mbps and 1 Mbps. The feasible states (n_{11}, n_1) obtained using the analytical model found in [11] are depicted in Figure 3.7. Notice that 11 simultaneous active calls are possible if all of them use the 11 Mbps data rate. However, a single rate change of one of the active calls to the 1 Mbps data rate reduces the maximum number of calls to 9 calls at 11 Mbps and 1 at 1 Mbps. This leads to a capacity of one call less than before, so the available capacity now is not sufficient for all the active calls. Notice that when the system is unstable no call performs satisfactorily. Thus, to guarantee the system stability there are two options: i) drop the call which has changed to the lower rate or ii) drop one of the calls which continue using the data rate of 11 Mbps.

In the next chapter we give a third option to solve this problem: modify the codecs of some of the active calls in order to lower the total relative bandwidth required by the active calls and thus guarantee them a satisfactory performance.

Part II

A cross-layer algorithm for voice codec adaptation

CHAPTER 4

Codec Adaptation Algorithm

4.1 Coping with the multi-rate effect

Although the transmission of VoIP over WLAN suffers from the problems mentioned in Part I, at the same time wireless 802.11 networks are becoming more and more popular, both for home-use and for hotspots/public places, with their usage foreseen to grow during the following years.

The need for some new ideas and solutions has led the research community to study in more detail admission control techniques and capacity evaluation of such networks. One big step ahead was the evolution of the WLAN standards to the newer 802.11e, which can guarantee some minimum QoS for multimedia traffic offering different priorities for different service types, as analyzed previously. Nevertheless, few effort has been placed on the Link Adaptation mechanism which remains a problematic point for voice traffic having the effects mentioned in section 3.5 and thus still demands efficient solutions.

In this chapter we introduce a novel cross-layer algorithm for codec adaptation of VoIP flows in a multi-rate scenario. We first review the most relevant state of the art solutions: those that deal with the effects of the Link Adaptation algorithm on the quality of VoIP flows, and those that are in general trying to improve the quality of VoIP over WLANs based on adaptation of parameters, such as codec and packetization interval. We see what are the limitations of the existing solutions, why they are not applicable in our scenario and how our proposal differs and responds to these issues. We finally present in detail the codec adaptation algorithm and validate its efficiency against the standard non-adaptive case using simulation results.

4.2 State of the art

There have been different ideas on how to cope with the problems that the Link Adaptation mechanism of 802.11 causes on multimedia traffic. On one side we can find those that propose to change the base of the problem, the Link Adaptation mechanism itself, or, in other words, let the channel adapt its behavior according to the needs of the higher layers. This can be done by trying to avoid unnecessary rate changes or by choosing the new rate based on more elaborated criteria, different for each type of traffic.

An example of this idea is the implementation in [36]. The decision on *when* to change rate and *which* new rate to choose is usually made by a control algorithm, based on information about the current link conditions, normally in the form of statistics-based feedback. This type of information is very slow especially in situations where the user moves fast or the channel conditions change frequently, making the automatic rate control algorithm inefficient for multimedia traffic. A new control algorithm is therefore proposed, which by using signal-to-noise ratio (SNR) information and in combination with existing automatic rate control algorithms can achieve a more limited range of feasible settings and therefore less oscillation in rates, which is crucial for multimedia traffic.

In a similar way, in [44], Kim et al. propose a new service-based rate control algorithm as an addition to the already existing throughput-based and error-based methods. The service-based algorithm can select which of the two other algorithms is more appropriate according to the service type of transmission data, on a per packet basis. That is, choose the throughput-based algorithm for best-effort data, since it helps to preserve the throughput of best effort services and choose the error-based algorithm for real-time traffic, to enhance delay performance.

Another idea can be seen in [40], where the authors try to deal with the "false link adaptation" problem. In the case under study, the rate changes can be produced falsely as a response to congestion instead of wireless channel errors, the main target of link adaptation. This rate drop can then further produce more delays and therefore more congestion. Hence, what they propose is to classify the packet errors in two categories, due to congestion or due to wireless channel errors and react with link adaptation only when they fall into the second category.

In general these methods can optimize the link adaptation process but they normally need and/or propose changes on the core PHY/MAC layers of the 802.11 protocol. For this reason, on the opposite side there are those who propose the adaptation of the traffic parameters to the new rate, leaving the link adaptation process untouched. When talking about voice traffic parameters this most of the times is translated to adapting the codec and/or the packetization interval of the voice stream. Experimental studies like [39], [81] and [13] have pointed out the important role of these two parameters on increasing the total capacity of a wireless cell. Based on this fact many works on how

to best solve the link adaptation problem have turned their focus on adjusting these parameters.

This is for example the basic idea behind the work of McGovern et al. [53]. They begin by evaluating the effect of the link adaptation on the network; if the rate change has not provoked an overloaded system then no action needs to be taken, else the voice codec of the node who has suffered the rate change is adapted accordingly so as to restore the system to its previous, not overloaded state. To choose the new codec they use a metric called Channel Occupancy Time, which represents a fraction of the channel time per unit time required by a full-duplex VoIP call for a given codec and rate. Their adaptive solution is the one presenting most similarities to ours. However, it is less flexible since only the node that suffered the rate change is adapted, which as we will see is in cases not sufficient.

Another way to choose when to change codec is based on the estimation of the channel congestion state proposed by Trad et al. [78]. When the channel is detected congested (based on RTCP packet loss and delay feedback information) a central element performs the adaptation of codecs using common transcoding methods for all calls entering the wireless cell. The main assumption is that the AP is what acts as a bottleneck and thus a solution is needed on the border gateway at the entrance of the network. Their proposal is dealing with network congestion rather than the multi-rate anomaly. The new codec is chosen according to Round Trip Time information obtained from the RTCP receiver reports.

Some more general works, on adjusting media transmission in IP networks without specifically trying to solve the WLAN multi-rate problem, can be also found in the literature and serve as a good background reference work. In [51] a framework is proposed to monitor the QoS of the voice calls over IP networks with no QoS guarantees (like a typical WLAN can be considered to be as well) and then adjust accordingly different relevant parameters, like interleaving, playout buffer, redundancy and number of packets per frame. Barberis in [6] also adjusts the transmission rate of the voice sources according to an estimation of the network conditions in terms of delays and losses.

The work presented by Chen et al. [19] uses the same basic idea (adjusting codec and packetization interval) in order to solve capacity problems mainly occurring due to the handoff procedure. By lowering the quality of some of the calls in the network it can adjust the distribution of resources among existing calls and permit others to enter. In fact this is an example of how a codec adaptation algorithm can work together with a Call Admission Control technique in order to provide a more efficient use of the network resources. This is the subject of the next part of our study presented in chapter 5.

Many other solutions can be found in literature based on the same background. In [33] Garg and Kappes propose to reserve some of the network capacity so as to be able to absorb a data rate change for some of the connections. Clearly this implementation

can be very wasteful of network resources, especially if the reservation is such so as to deal with rate drops from 11Mbps to 1Mbps. A more simple solution would be to drop the problematic call [54]. Although this way the problem is solved very fast, dropping the call has a negative effect on the user's opinion of the network QoS.

An extension of the previous methods is to focus on multi-rate codecs, like the GSM Adaptive Multi-Rate (AMR) speech codec [27]. AMR has been developed by ETSI, adopted by the 3rd Generation Partnership Project (3GPP) and widely used in GSM/3G networks. It has eight different encoding modes corresponding to eight source bit rates ranging from 4.75 kpbs to 12.2 kbps. The codec is adaptive in the sense that it can switch its bit rate in a frame basis depending upon channel and network conditions. The philosophy behind this is to lower the codec rate as a reaction to channel conditions when the interference increases, enabling more error correction to be applied to guarantee a good speech quality (MOS) for voice calls [63].

For example, when the channel errors increase, the source bit rate is lowered so that a larger portion of the bit rate is used for channel error protection using a Forward Error Correction (FEC) mechanism. However the total (gross) bit rate is in fact constant [49]. Thus it would not be sufficient to use these codecs in the multi-rate WLANs case, since the channel would be still occupied during the same time by the slow node and provoking a similar effect on the voice quality of the others. A capacity increase is possible in GSM networks, increasing the interference among the nodes and using at the same time a more robust codec (lower rate AMR with higher channel error protection). However, this is not the case for the multi-rate WLANs.

An interesting fact is that very few bibliography exists actually on using this codec on 802.11 networks. Servetti et al. in [72] have proposed the use of the narrowband AMR for speech transmission over 802.11 WLAN networks, to change the codec rate by using shorter or larger voice packets (i.e. lower or higher bitrate), according to the channel conditions. The results obtained show a big improvement comparing to a constant bit rate approach. However, in the scenario under test they refer to bad channel conditions to the ones perceived individually on each STA's channel caused by factors such as noise, fading and interference. In fact their scenario consists of only two nodes and an AP, so no consideration of cases where the channel capacity is fully used and the users affect on each other is considered. This differs from the multi-rate channel effect, where the behavior of other nodes (which change from *fast* to *slow* rates) is what impacts over the performance of the rest, without any channel change occurring on the fast nodes. It is quite probable then that, using the AMR codec under a multi-rate scenario, *all* mobile nodes would see the erroneous channel and change to a lower codec at the same time, so reducing the overall MOS more than necessary. So, to maximize the total MOS of the cell other solutions would be more adequate.

Summarizing, we have identified through the literature study a number of limitations

that we address in this thesis:

1. There are very few works focused directly on multi-rate WLAN scenarios. Our work fills this gap by providing an exhaustive study on the specified anomaly caused when one node lowers its transmission rate and this affects directly all the other active nodes, without them perceiving bad channel conditions in any other way.

2. Although there are solutions that are similar to our proposal (i.e. codec adaptation), they are still less adaptive, suggesting for example a change of codec for *all* active calls, just *one* call or directly proposing to *drop the slow call*. Our solution permits higher flexibility by choosing which and how many calls should change codec through the constant real-time monitorization of the cell's conditions and avoiding any call drop if possible. Additionally, our mechanism is the first of a cross-layer type, using valuable information from the different layers (MAC and RTCP) for its decision-making and taking advantage of the possibilities that the SIP protocol provides for codec renegotiation.

3. Our proposal does not suppose any modifications of the 802.11 standard. The information and protocols we use for the monitorization and the adaptation procedures (feedback from RTCP, MAC and SIP as will be explained later) already exist in VoIP over WLAN scenarios and can be obtained in a simple manner.

4. We have seen many studies on Call Admission Control mechanisms in chapter 3 which try to manage efficiently the limited VoIP capacity of 802.11 WLANs and increase if possible the acceptance of *new* calls. On the other hand we have seen here a number of solutions focusing on maintaining or increasing the QoS of already *active* calls by changing some of their parameters. Nevertheless, very scarce bibliography exists on a combined solution that can address both new and active calls, trying to increase the cell capacity and maintain at the same time QoS at satisfactory levels for the accepted calls. Our solution is one of the few that combine these two different areas and propose a holistic approach for both new and active calls.

5. Finally, to the best of our knowledge there is no real implementation design of any of these solutions. We have tried to address this issue as well by providing a solution that can be easily implemented with minimum software and hardware modifications, and no special server needed. In chapter 6 we will present a first software and hardware design of how the different modules described here can be integrated and work together in a real AP.

The approach of the multi-rate problem presented next in detail follows the line of adapting the media transmission to the new physical link rate. A key idea behind this

solution is its ability to help maintain the QoS of active calls, without interrupting or dropping them. Between its main advantages is that of being a simple solution, that can be adapted to work in both centralized and distributed scenarios (installed in the AP or at the wireless STAs accordingly), with no new special server needed or modifications to the AP in the distributed case.

Note also that, while the main interest until now has fallen on different Call Admission Control (CAC) techniques and on preventing congestion by restricting the incorporation of troublesome new calls *before* they join the network, the codec adaptation algorithm is focused on the calls *already* accepted at the network and how they can recover fast and without interruptions from a change on the network conditions. As a matter of fact, the two techniques can be combined and work together in order to increase the performance of the network. However, since they are focusing on different problems, they can also act independently from each other. It is therefore believed that the codec adaptation algorithm can provide the part missing from the admission control mechanism. A proposal with different policies combining the two techniques will be explained in detail in the next chapter.

4.3 The codec adaptation approach

Following the above reasoning, one of the most effective solutions of the multi-rate problem on VoIP calls is to adapt the codecs of some of the active voice flows to the new cell conditions. The basic structure of this proposal can be seen in Figure 4.1 and the entire detailed algorithm flow chart in Figure 4.4.

Our solution is based on the simple concept that different codecs have different bandwidth needs without at the same time having an equivalent gap in the QoS they offer (in terms of MOS). Thus, in order to lower the total relative bandwidth demands of the VoIP calls, a lower bitrate codec can be used in some of them, which at the same time can preserve the QoS at acceptable levels. Giving an example, while using the G.729 codec the bandwidth needs (8Kbps) decrease almost 8 times compared with using the G.711 codec (64Kbps), the quality offered by the first codec is not 8 times worse, since G.729 codec results to a MOS of 3.92, while the MOS when using G.711 is 4.1, both considered as satisfactory MOS results.

Consider a multi-rate hotspot scenario, with a number of VoIP calls active. When a STA changes its rate to a lower one, this as we have analyzed will most probably have a direct effect in the perceived QoS of all the VoIP flows in the cell. Therefore, we need to *monitor* the system so as to capture these rate changes and/or the QoS variations on the STAs. In case that a problem is detected, a change on the codec of some of the active calls would help to lower the congestion level of the cell and recover the QoS levels. However, we need a method for evaluating the new cell conditions and these

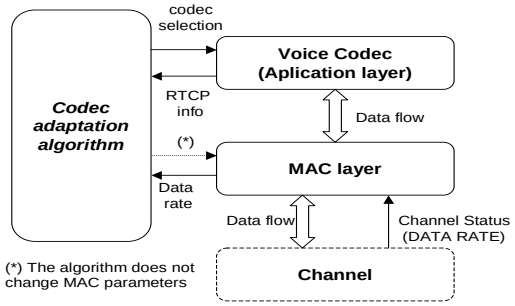


Figure 4.1: Information flow of the codec adaptation solution

QoS fluctuation, making sure that did not occur due to spurious error. We also need to decide how many calls and to which codec should change so as to reverse them. This is were an *adaptation* procedure enters. Finally, a method to instruct the necessary codec changes to the affected STAs is needed, which will also monitor the system recovery and whether the problem is solved or additional actions are needed. Thus, the codec adaptation should finalize with a *recovery* phase.

Therefore, the basic Codec Adaptation algorithm is composed of three main phases:

1. the *monitoring phase*, including the MAC monitoring function for information on the link adaptation changes and the RTCP filtering function, for real-time feedback on the quality of the VoIP flows, such as delay, jitter and packet loss

2. the *adaptation phase*, where all the calculations and decisions on codec changes are made, and

3. the *recovery phase*, where any codec change decided previously upon is negotiated through SIP messages without interrupting the call and the results of the change are measured.

Each of these phases is analyzed next in detail.

4.3.1 Monitoring

The monitoring phase is a constant feedback gathering procedure, focusing particularly in two types of feedback: link rate changes and quality of service alarm signals.

The information over rate changes arriving from the MAC layer is more immediate and can allow a faster reaction. Almost instantly after the rate change signal, a codec adaptation can be performed for the node that lowered its transmission rate[1](see Figure 4.2). Therefore, this method can be considered as a proactive reaction; although there

[1]Note here that this procedure can work equally well both for rate increase and decrease, but since

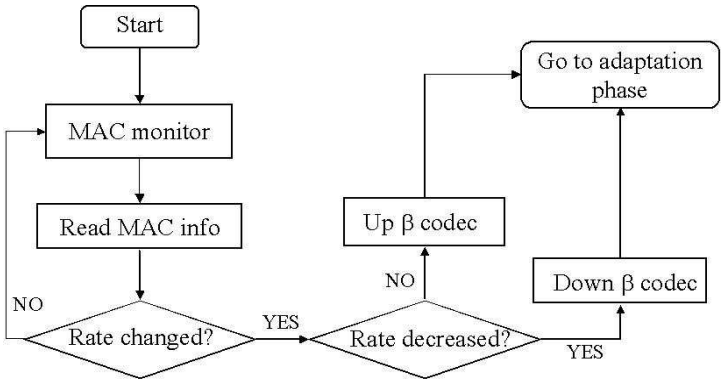


Figure 4.2: MAC monitor process

is no quality decrease observed yet, knowing the negative effects of a rate change to the whole cell, a codec change is performed for the now slow node trying to prevent the predicted quality decrease. In the following adaptation phase it can then be determined if the codec drop was helpful enough or if there is a need to adjust the codecs further. Apart from achieving a faster response, this method is also very important so as to maintain the fairness of the algorithm. In other words, the node that suffers the rate change is the first to change codec and therefore is the first to suffer the consequences of its rate change, while the rest of the nodes may avoid even noticing any alarm and thus the need to adapt to it. Additionally, as we have proved in [74], we achieve better results (i.e. less calls are needed to change codec and the recovery is faster) when we change the codec of the slow calls against changing the fast ones.

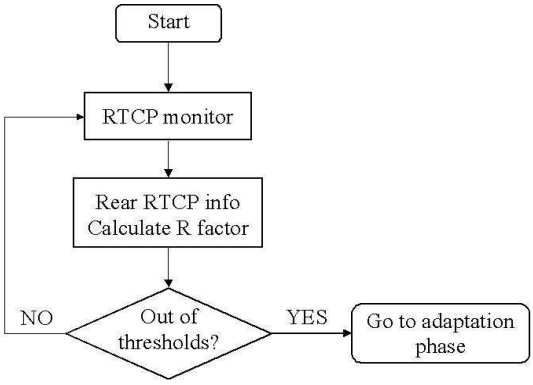


Figure 4.3: RTCP monitor process

However, the second type of feedback is also necessary, since not always the codec change of just the slow node is enough. If this is the case then this will be detected from

the rate decrease situation is more critical the analysis here focuses only on this case. In the case of a rate increase a codec increase would be proposed similarly to the decrease procedure, for better utilization of the network resources.

the RTCP packet filtering. In a VoIP session, when seeking quality of service metrics such as end-to-end delay, jitter or packet loss ratio, RTCP is the key. RTCP packets arrive to each of the nodes involved in a call session periodically, at a fixed interval time, usually set to 5 seconds [70] (it is also possible to lower this interval time in order to minimize the feedback delay when needed, as we will explain later in the adaptation phase).

Processing the information included in these packets the above mentioned QoS metrics can be obtained, necessary for calculating the R-factor of each flow using the E-model voice quality measuring tool as explained in section 2.2. We have chosen to use the E-Model because it can be calculated in real-time using the RTCP reports. This way we can obtain a dynamic QoS measurement during the call and not a static value associated to the codec a priori like the MOS is. The equivalence of this factor to the most known MOS can provide a first QoS metric for the system and define a decision threshold; if R-factor value falls at any moment below 70 (equivalent to MOS 3.6) the quality of the voice flow is not satisfactory (see the R-Factor to MOS Table 2.2) and the algorithm triggers the adaptation phase for a new codec selection. This threshold, as also the thresholds used in the next phases of the algorithm, can be furthered tuned using more exhaustive simulations. The ones used in this study are the typical values recommended by the ITU-T for acceptable voice quality.

In the distributed implementation of the algorithm presented here, the first call to react and enter the adaptation phase will be the one who first notices the QoS degradation, having the measured metrics fallen below the thresholds. Most of the times all nodes will perceive this quality decrease almost instantly after a rate change and simultaneously, therefore some backoff mechanism is needed so as to avoid a simultaneous reaction and codec change from all, as we will see in the adaptation phase next. Nevertheless, and especially in the case of a centralized implementation where the AP can control the order of the calls to change, we can use some more sophisticated policy in order to choose the next call to change codec. These policies will be reviewed in chapter 5.

4.3.2 Adaptation

The adaptation phase is in fact a two-step procedure: evaluation and decision. The idea is to check if the alarm situation that initiated adaptation was due to a spurious error or if it still continues, and if so evaluate the extent of the problem and decide what to do in response. The duration of the adaptation phase must be as short as possible since it is in fact what determines the critical reaction time, between the alarm signals and the codec change. Therefore it must be limited by an adaptation timer. Furthermore, this timer is not fixed but randomly chosen so as to also avoid that all nodes react and change codecs simultaneously, even if all of them perceive the alarm

signals at the same instant. This way, the change of codec for more calls than necessary can be avoided and the system is given some time to recover after every codec change. Thus the adaptation timer serves a double purpose: limit the adaptation phase duration and serve as a backoff to limit the simultaneous reaction of the nodes.

In order to evaluate the current conditions efficiently, more than 1 feedback sample is necessary. This can be obtain using again the RTCP packets data, only that this time and since the duration of this phase is limited, it is recommended to use an extension of the standard RTCP protocol, which allows to modify the interval between the transmission of two successive RTCP packets. This extension was proposed and evaluated by Ott in [59] and is referred to as "immediate RTCP feedback". While in the RTP standard a minimum of 5 seconds interval is defined, using this extension the interval can be set to be lower (i.e. RTCP reports are sent more frequently), according to the needs of each scenario.

Although it may appear that the use of more (fast) RTCP packets would overload the system and thus further increase the problem, this is not true. An estimation of the total overhead that these packets would introduce in the network can be roughly calculated as follows: Based on a RTCP packet size of 90 bytes and setting the frequency of fast RTCP transmit at 1 second, the overhead provoked by the control traffic is very small compared with the data traffic of a VoIP call using $G.711$ codec (with packets of 160 bytes every 20 ms), with the control traffic occupying only around 1.3% of the total traffic generated per user, way less than the allowed 5% for control traffic according to the RTP/RTCP standard.

The frequency of the fast RTCP transmission in combination with the adaptation timer is set so as to allow the receiving of a feedback sample big enough for an efficient evaluation, while trying to minimize the total delay of the process at the same time. That means that if, for example, the RTCP interval is set to 1 sec, then the timer should be set to a random number higher than 3 sec, so as to have a minimum of $N = 3$ fast RTCP packets. At the same time, this leads to a lower bound of minimum 3 seconds for the delay of the evaluation phase[2].

When the timer expires and using the information collected during this time, the algorithm should decide which codec is the most appropriate for the node evaluated under the current conditions. To do so, it calculates the average of the R-factor during the adaptation timer (and from there the MOS score equivalent), as also the average of packet loss and delay, critical voice quality metrics. The codec adaptation algorithm compares the average values as well as the current values of the parameters after the timeout with a set of thresholds, chosen using the common values of permitted QoS parameters for an acceptable voice transmission ($delay < 150ms$, $packet\ loss < 3\%$, $R > 70$). From the result of this comparison and the codec used until now in the

[2]The numbers used here are indicative and can be further adjusted through simulations.

transmission, the node can choose a new codec, using the following procedure (see also Algorithm 1):

a) if the average value of the parameter is out of threshold then check its current value; if the current value is also out of threshold propose a codec of α steps lower in the codec ranking else propose a β steps lower codec. This way, even if the average performance of the call was not satisfactory during the fast RTCP monitoring time, if some other call has meanwhile changed codec and the system is beginning to recover (so the current value is above the threshold) then the call will suffer a smaller codec drop.

b) if both the average and the current value of the parameter are above the thresholds then there is no need to change codec.

This check is performed for each one of the three parameters (delay, loss, R) used in the evaluation and an average of the proposed codec of the three is chosen. Note that $\alpha > \beta$, with $\alpha = 2$ and $\beta = 1$ in the simulations, and that the codecs are ordered based on their bit rates, as shown in Table 2.1.

Algorithm 1 Adaptation phase

```
while timer ≠ 0 do
  fastRTCP monitoring
  paramNow ⇐ current values of delay, loss, R
  for param ⇐ delay, loss, R do
    paramAvg ⇐ calculate average of parameter
  end for
  timer ⇐ timer − 1
end while
for param ⇐ delay, loss, R do
  if paramAvg > paramThreshold then
    if paramNow > paramThreshold then
      change(param) ⇐ α
    else
      change(param) ⇐ β
    end if
  else
    change(param) ⇐ 0 {No drop}
  end if
end for
changeTotal ⇐ Σ(change(param))/3
newCodec ⇐ drop(currentCodec, changeTotal)
```

4.3.3 Recovery

So far, the algorithm has analyzed the situation and taking into consideration all the feedback from the lower layers it has decided the most suitable codec to meet the needs

of the current network conditions. Here, at the recovery phase, is where the negotiations for the new codec agreement are performed at the application layer. This can be easily done using SIP, the signaling protocol for control of the call session parameters. More specifically, the SIP re-INVITE method is used, with a structure almost identical to the initial INVITE message and with the only difference of the new codec proposal in the SDP audio codec negotiation field.

Hence, during the recovery phase the wireless node is responsible for issuing and sending a SIP re-Invite message to the other end, so as to re-negotiate the new codec. If the other end accepts, then the call continues normally, otherwise the call is dropped. If the codec chosen as the most appropriate is lower than the lowest codec that a node can support, the easiest approach would be to drop the slow call in order for the others to continue with no problem. However, and depending of whether talking about a centralized or distributed implementation, there are other solutions in order to "save" the call. One idea, applicable to the distributed implementation case, would be for the call to continue as it is during some stand-by time, without any codec change; If during this time some other node changes codec and the quality metrics show that the problem is solved then the call can continue successfully, otherwise the call will be then dropped. On the same line, for the centralized implementation, the AP could choose another call for codec adaptation if the call originally chosen cannot change any further. These adjustments can vary highly, as they depend on the specific needs of the implemented scenario in each case and on the trade off between capacity and quality/fast reaction and recovery. The details of the different variations of this solution are not considered here, but some decision policies that can be applied in these cases are analyzed later in chapter 5.

After the negotiations are over, the algorithm returns to the adaptation phase and continues to monitor the system using the fast RTCP messages and evaluate its performance after the change; if the parameters are higher than the upper thresholds then it can return to the normal monitoring phase, else it needs to perform another codec change until reaching acceptable QoS levels.

4.4 Implementation Issues

a) Delay

An important parameter of any of the solutions focusing on VoIP traffic is the delay that they suppose and if this delay affects the call and is noticeable by the user. The delay during a call, translated into interruptions of the speech flow, should not be more than some hundred milliseconds so as not to be practically noticed by the user. When higher than this, the user will start to notice there is a communication problem. If this problem is solved fast enough from a human's perspective (in less than a few seconds),

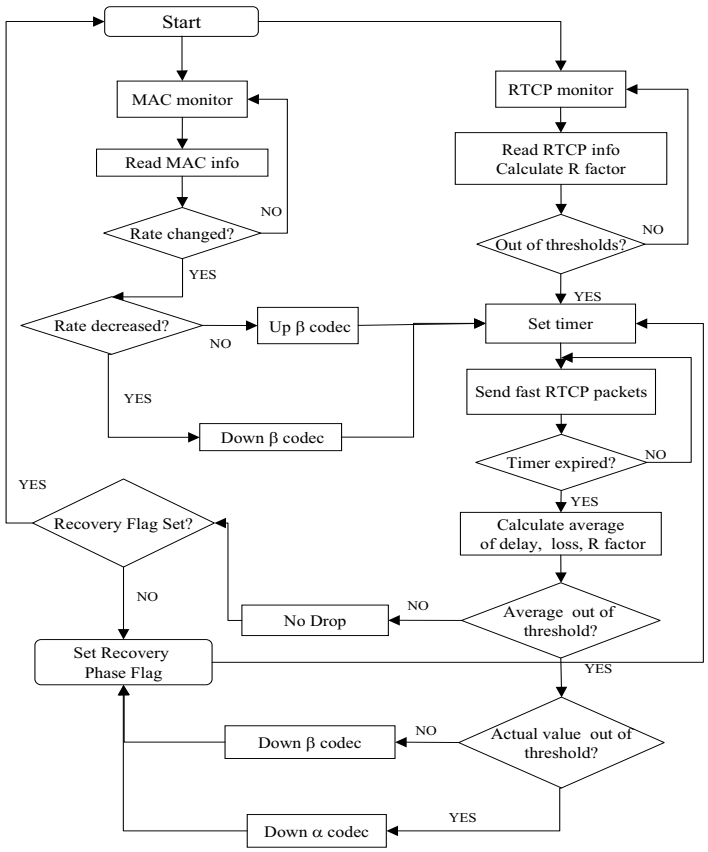


Figure 4.4: Algorithm Flow Chart

then it can be considered as an incidental interruption and will not affect much the rest of the communication. Otherwise, the user will most probably end up hanging up and terminating the call.

Therefore, the delay of the process mentioned above - and of any similar solution - is an important consideration when it comes to implementation. Although this delay is not as critical as in other environments, since the call *is not interrupted* during the process, it is essential to be able to recover from the network changes and their negative effects on voice quality as fast as possible in order to avoid the user hanging up.

The total delay, from the moment the algorithm receives the first alarm signals until the moment that the system recovers, can be represented by the following equation:

$$Delay_{Total} = D_{adapt} + D_{SIP} + D_{proc} \tag{4.1}$$

where D_{adapt} is the duration of the adaptation phase, D_{SIP} the time needed for the SIP re-Invite / OK / ACK messages for the codec renegotiation procedure and D_{proc} the general processing time of the algorithm. While D_{SIP}, being 1.5 times of the Round

Trip Time, depends on variable factors, like the transmission delay over the Internet when working on a wired to wireless scenario, it is usually in the order of milliseconds, as is the D_{proc}. So both these delays are negligible in comparison with the duration of the adaptation phase, and in extension the RTCP interval delay, which is what really increases the total process delay to the order of seconds. Remember that the adaptation random timer is also set depending on the frequency of RTCP packet arrival, which was the reason for choosing the fast RTCP extension in the first place.

With the adaptation timer normally set to a value between 3 and 5 seconds, the delay should not be more than a few seconds (e.g. less than 5), which is acceptable from the point of view of human perception and taking into account that during this time the call is not dropped or interrupted. From the simulations below and choosing a frequency of 1 second for RTCP interval, it can be seen that the delay is in fact not more than $3 - 4$ seconds. Even better results on reaction time can be obtained setting this interval to a lower value, although with the counterpart of overloading the cell with more control traffic.

b) "Ping-pong" effect

Another interesting consideration is the possibility of a user changing its transmission rate very frequently, due to being for example at the border limit between two different rate areas. This may cause what is known as "the ping-pong" effect, that is changing physical rate up and down very frequently. The issue arise when the use of the codec adaptation algorithm can provoke a similar frequent codec change, which can be even worse from the user's perception than continuing throughout the entire call with a lower bitrate codec. Even worse, what happens if the duration of the adaptation timer is higher than the frequency of rate changes?

For this reason, the solution presented here is focused on alleviating the multi-rate effect when this provokes a heavy drop in the perceived quality of the call rather than reuse the spare capacity obtained by a change to higher rate. When a user changes to a higher rate, no quality degradation is noticed and so there is no need for him to change codec. In fact, the only drawback of this implementation is that the released cell capacity is not efficiently used but from the users point of view the call quality is maintained. Although in the algorithm the possibility was contemplated of changing to a higher codec when a change to higher rate occurred, this option can be simply "switched off" in an environment with high user mobility and frequent rate changes.

Another idea would be to keep track of the codec changes of the same user and avoid a codec change in a very short interval from the previous one. This can be easily implemented with a backoff timer, or by setting an upper and lower threshold of measured SNR so that users that are at the border areas would be excluded from the codec adaptation when this is to higher codec. Although this solution is not further studied here it could be an interesting area for further enhancements to the original

codec adaptation algorithm.

4.5 Distributed vs Centralized Architecture

One of the main advantages of the algorithm presented above is its flexibility when it comes to implementation. Since it is entirely based on the feedback from packets already existing and circulating in the network, like RTCP reports and MAC layer information, it does not need -in its distributed implementation- any specific modifications on the MAC layer, the Access Point or any of the nodes involved in the cellular architecture. In addition to this, it can be implemented both in a distributed and in a centralized mode, with minimum changes between the two versions and permitting higher flexibility depending on the specific characteristics of the working scenario.

The basic difference between the two implementations is the location of the core adaptation algorithm. In the distributed scenario, the algorithm is implemented on each node and each node is made responsible for monitoring and adapting its own state. When a rate change is noticed on the MAC layer or based on the RTCP information arriving, the node is the one to determine whether or not to change codec. On the contrary, in the centralized case, the AP is in charge of monitoring all calls, the transmission rate of each flow and the codec used by each client. When a call passes from fast to slow then the AP can determine which and how many calls must change codec so as to reach network stability again, based on the RTCP information exchange between the clients. This is the implementation chosen for the rest of the study, where a centralized control joint with Call Admission Control mechanism is proposed.

Therefore, the complexity and processing work is higher in the centralized version since the AP has to intercept all the RTCP packets in their way from one end to the other and calculate the parameters needed for the threshold comparison for all calls. Then the AP decides according to the adaptation procedure and the policy used which calls to change and to which codec, giving more weight and priority in changing the slow calls first. It must then inform the nodes that there is the need to change codec and therefore suggest to issue a SIP re-Invite message to re-negotiate the codec with the other end. This can be more complicated than in the distributed version, although some ideas on the interfacing issues of the AP with the nodes and an implementation proposal architecture are given in chapter 6.

However, the centralized view of the problem as a whole gives much better results and more possibilities of achieving an optimal codec combination among the nodes. As it has been proved in [74], there is no need for all calls to change codec at the same time, and changing slow-rate calls gives better results than changing fast-rate calls. This is due to slow-calls being the ones actually causing the problem, as seen in the problem statement. This priority on slow over fast calls is not possible in the distributed

implementation, where each node will decide the action to be taken depending on its own limited view of the system, and thus the call to detect first the QoS decrease will be the first to react, without knowing if there are other calls in the cell and the state (codec/rate) of each one. Additionally, since the control of the adaptation phase timer is not centralized, more nodes can coincide and change simultaneously codec, while in the centralized implementation the AP can set this backoff accordingly between each codec change, which permits that less number of calls will have to change. Nevertheless, while the distributed approach may not be the globally optimal solution, it is easy to implement and it distributes the processing load of the algorithm.

Simulation results show that there is an improvement in the performance of the algorithm when used in its centralized version; less calls are changing codec, the packet loss percentage is almost zero and the overall MOS achieved is higher than in the distributed implementation. These results will be reviewed in the following section.

4.6 Performance results I : Distributed vs Centralized implementation

In order to test the performance of the codec adaptation solution explained above, extensive simulations were performed using the network simulator tool NS-2 [58], with the enhancement of a SIP patch to include the standard SIP agents (*Proxy*, *UserAgent*, and *DNS*) and perform the basic SIP operations. The patch was obtained from National Institute of Standards and Technology (NIST) [57] and we have adapted it for our $ns-2$ scenario with the goal of controlling the codec of each call while the call is in progress. The description of the testing scenario as also the performance results are provided next.

4.6.1 Scenario description

The scenario considered is a hotspot multi-rate scenario, where the network is composed by one 802.11e [2] basic service set (BSS) with 9 wireless nodes and one Access Point connected to the wired network, equal to the hotspot scenario explained in chapter 3. The values of the parameters set used can be found in Table 4.1. A total number of 9 calls (18 unidirectional flows) are considered active during the example scenario used through the simulations, with all of them established between one wired and one wireless client, while the Access Point is also acting as a Proxy Server. All nodes start by using 11 Mbps data rate (fast-rate calls) and at predefined instants ($t = 95sec$ and $t = 105sec$) two flows change to 1 Mbps data rate (slow-rate calls), while at instant $t = 145sec$ one of the slow nodes changes to higher rate again. These changes were chosen in order to simplify the process, show clearly the effect of the multi-rate channel and the performance of the algorithm. There is no mobility consideration for the nodes since,

Parameter	Value	Parameter	Value
R_{data}	$\{11,\ 5.5,\ 2,\ 1\}$ $Mbps$	R_{basic}	$\{11,\ 5.5,\ 2,\ 1\}$ $Mbps$
R_{phy}	$\{1\}$ $Mbps$	–	
DIFS	50 μs	CW_{min}	32
SIFS	10 μs	CW_{max}	1024
SLOT (σ)	20 μs	m	5
EIFS	364 μs	ACK	112 $bits$ @ R_{basic}
RTS	160 $bits$ @ R_{basic}	CTS	112 $bits$ @ R_{basic}
MAC payload	$[0, 18496]$ $bits$ @ R_{basic}	–	
MAC header	240 $bits$ @ R_{data}	MAC FCS	32 $bits$ @ R_{data}
PLCP preamble	144 $bits$ @ R_{phy}	PLCP header	48 $bits$ @ R_{phy}
Retry Limit (R)	$R_S = 4$, $R_L = 7$	K (Queue length)	20 $packets$

Table 4.1: System parameters of the IEEE 802.11b specification [1]

in general, wireless hotspots are characterized by low mobility and the rate changes can be provoked by factors other than mobility, such as interference due to an obstacle or a change in weather conditions.

The calls are considered to start with the G.711 codec, have the same duration and change when needed to one of the lower bitrate codecs seen in Table 2.1. The fast RTCP transmission interval is one of the tunable parameters when using the extended RTCP version [59] and is set at $\delta = 1$ seconds, in order to minimize the reaction time of the algorithm[3]. It is assumed that all users support all codecs and there is no other traffic or other interferences in the wireless network. Each STA has a queue length of $K = 50$ packets.

4.6.2 Analysis

As explained previously in chapter 3.5, in a scenario with 9 calls, all of them using G.711 codec, when 2 calls drop to 1 Mbps rate the new state is not feasible. This can be translated in degraded voice quality metrics such as high delays and high packet loss ratio. The effect can be observed in the example presented here.

a) No codec adaptation algorithm

When there is no codec control algorithm implemented in the network, observe the effect when two nodes start transmitting at a lower rate changing from 11 Mbps to 1 Mbps (at instants $t = 95sec$ and $t = 105sec$ on Figures 4.5, 4.6, 4.7, 4.8). As the authors in [53] mention *"the congestion in 802.11 is not gradual; the system has the a tendency to transition from an uncongested state delivering good performance to a congested state delivering very poor performance with the addition of little extra traffic"*. Due to this characteristic of the 802.11 networks, the observed packet loss percentage during the

[3] an even better reaction time can be obtained setting this interval to a lower value, e.g. 500 ms, since as we have seen the delay of the adaptation phase depends mostly on this interval time

simulation increases almost instantly after the rate change happens to values reaching 90%. In fact this result can be translated as a call drop since almost all packets are lost during a big part of the call. Moreover, the packet delay reaches very high values (of approx. 1 sec), as the queue length of the AP becomes saturated (Figure 4.6). The congestion of the system, both in terms of loss and delay, is much more obvious in the AP, since it aggregates the traffic of *all* calls, which is the reason of the big difference observed on the results between uplink and downlink (as explained in section 3.3.1). In this case the AP acts as a bottleneck dropping queued packets and provoking a significant increase in packet loss ratio and delay. The same saturation can be also observed in the very low throughput obtained in Figure 4.8 and the low quality perceived by the user in terms of MOS in Figure 4.7. This MOS, as calculated in real-time using the E-model, drops to values as low as 1, meaning communication breakdown according to the MOS standard definition.

The situation is corrected only when one of the two nodes that previously dropped to a lower rate changes again to a higher rate (11 Mbps) at simulation instant $t = 145sec$. After this point, a decrease on delay and packet loss is observed, although they still remain higher than the desired for a correct VoIP transmission, with delay above 100ms and packet loss percentage of 10% in the downlink.

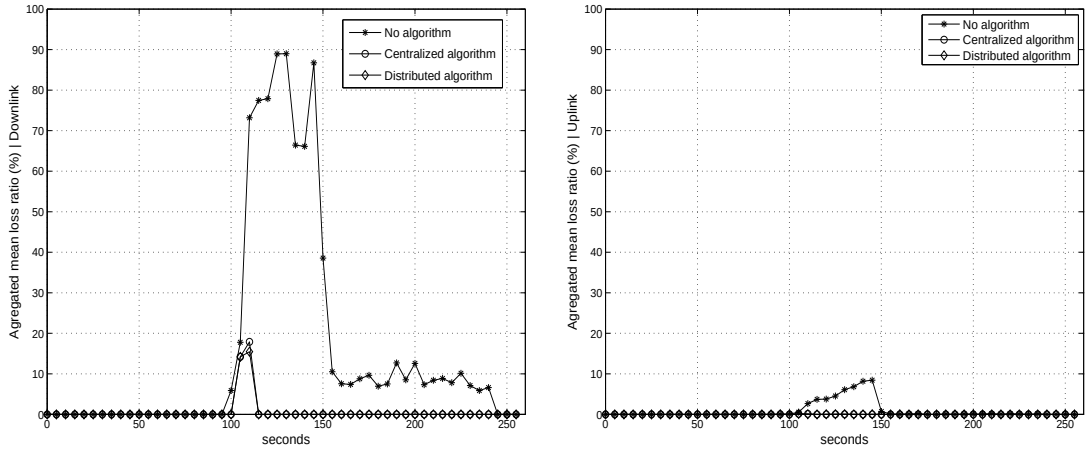
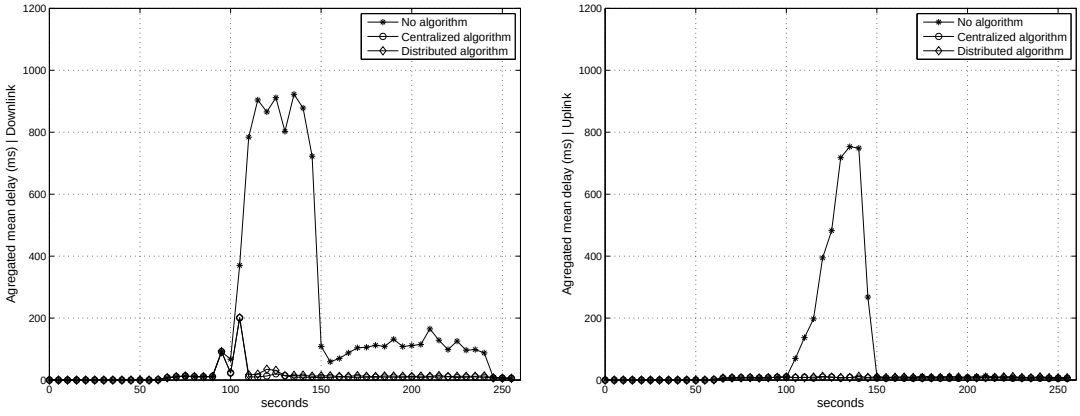


Figure 4.5: Average aggregated packet loss percentage of VoIP flows in (a) Downlink (b) Uplink (rate changes to lower at 95 sec and 105 sec, rate change to higher at 145 sec)

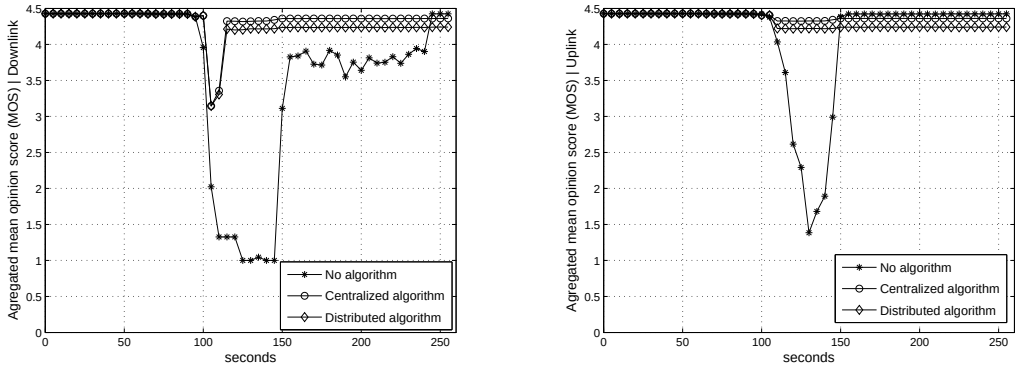
b) Distributed implementation of the codec adaptation algorithm

With the implementation of the codec adaptation algorithm in either of its two modalities (centralized and distributed), and since the codec of some of the calls is adjusted, the congestion level of the AP is significantly reduced, and as a result the



(a) (b)

Figure 4.6: Average aggregated packet delay of VoIP flows in (a) Downlink (b) Uplink (rate changes to lower at 95 sec and 105 sec, rate change to higher at 145 sec)

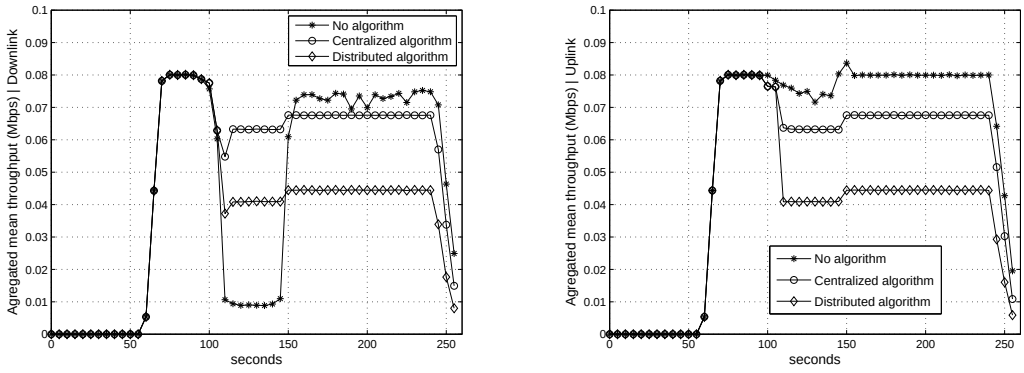


(a) (b)

Figure 4.7: Average aggregated MOS obtained for VoIP flows in (a) Downlink (b) Uplink (rate changes to lower at 95 sec and 105 sec, rate change to higher at 145 sec)

effects of the multi-rate are barely noticed by the users. Looking at the distributed implementation to begin with, almost instantly as the rate changes happen at $t = 95sec$ and $t = 105sec$ of the simulation, the algorithm takes action by first changing proactively the codec of the nodes that suffered the rate drop. The MAC information arrives as soon as the rate change takes place and the proactive codec change is very fast.

However, this change is not enough and the RTCP packets announce that the QoS alarm situation continues. The nodes that receive this alarm (in fact this could be *all* nodes) enter the adaptation phase and after the adaptation timer expires they decide whether to change or not the codec they use. Indeed, the throughput values of the Figure 4.8, indicate that only some nodes had finally changed codec; the average aggregated throughput is about 40Kbps (with overheads included) which is higher than the result



(a) (b)

Figure 4.8: Average aggregated throughput of VoIP flows in (a) Downlink (b) Uplink (rate changes to lower at 95 sec and 105 sec, rate change to higher at 145 sec)

would be if all nodes had changed, according to the bitrates used by each codec on Table 2.1. This is because, as the codec change of other nodes lowers the congestion levels bit-by-bit, after the adaptation timer the node may encounter the problem already solved.

So while the total throughput may be lower than before the rate changes, since some calls now use codecs that require less bandwidth, the system is no longer saturated. This can be verified in the packet loss and delay figures, where there is just one peak of high loss percentage reaching 20% and high delay of around 200 ms at the moment of the transmission rate changes and are corrected efficiently in less than $4sec$. The effect of the codec change observed in the packet loss ratio (Figure 4.5) agrees with the one expected. When using a lower bitrate codec, the offered load on the queue decreases and therefore less packet losses due to buffer overflow in the Access Point are observed. The results of MOS come to justify that the user perceived quality is maintained at very high levels, with only an instant drop at the moment of the rate change and until the nodes start reacting.

c) Centralized implementation of the codec adaptation algorithm

Even more impressive are the results of the centralized implementation. At the moment the MAC monitoring receives the rate change signal, it lowers by one the codec of the affected nodes. Along with this proactive codec change, only one more codec change of a fast node's call was in fact needed during simulation, provoked from the RTCP monitoring, and the system recovers without noticing any of the negative effects mentioned above. Again, as it can be observed from the packet loss and delay figures, the peaks of high loss percentage and delay at the moment of the transmission rate change are corrected very fast. During the rest of the time packet loss is practically 0

and delay is no more than a few milliseconds. The average MOS value, indicating the user perceived quality is maintained in very high values around 4.3, as can be seen in Figure 4.7, with only an instant drop at the moment of the rate change and until the nodes start reacting. This shows a huge gain compared to the MOS with value less than 1.5 achieved when no algorithm is present.

By having an overall control on the timing and the order of the codec changes and prioritize more efficiently the change of the slow calls before the fast ones, it results that the total number of calls that need to change codec is lower than on the distributed implementation. Observe the total throughput obtained (Figure 4.8) that is higher than in the distributed mode, which is translated in more calls transmitting with higher bit rate codec. This is because since the slow calls are the ones blocking the others, it is more efficient to lower more the codec of these calls, apart from also being the fairest solution. Again, both delay and packet loss results adjust to the expected performance as in the distributed implementation and even slightly better.

The results presented here were compared against an analytical model presented in [11]. The system behavior was shown to match quite precisely the expected results, as foreseen by the analytical model.

4.7 Performance Results II : heterogeneous traffic (VoIP with TCP)

Up until now, we have considered and tested the codec adaptation solution only in a purely VoIP scenario, where no other traffic but voice existed. However, in a real 802.11 cell, it is to be expected the coexistence of both elastic (TCP) and inelastic (VoIP) traffic at the same time. For this reason, we examine here in more detail how the algorithm reacts under these mixed traffic conditions.

4.7.1 Scenario description

The results below are obtained using the same NS-2 simulator. We use a BSS with one AP and we vary the number of nodes with elastic and inelastic traffic to examine two cases with different congestion levels. All VoIP flows start around instant 50 of the simulation while the TCP flows start earlier, from the beginning of the simulation. TCP flows are all uplink (from the node to the AP), so that the node's rate change will be more visible. In all cases, at instance 185 sec of the simulation, two of the nodes carrying voice traffic change their transmission rate from 11 Mbps to 1 Mbps. Later, at instance 285 sec of the simulation, 2 more nodes carrying elastic traffic (FTP) change their transmission rate from 11 Mbps to 1 Mbps.

In addition to codec adaptation when the rate changes occur on VoIP nodes, we have added an EDCA parameter adaptation mechanism for the case that the rate changes

occur on nodes hosting TCP flows, since codec adaptation can be only applied on VoIP flows. The EDCA tuning mechanism used here is a very simple one since the optimization of these parameters is out of the scope of this thesis. The tuning procedure considered during the simulations consists on modifying the default parameters of the Access Category 1 (Best Effort) of table 3.1, with new values of $AIFS = 15$ and $CW_{min} = 256$ when a rate change occurs on a node hosting TCP traffic. These modifications permit to restrict even more the TCP traffic in order to protect better the VoIP flows. The chosen new values where obtained from the study in [71], where the use of them were proved to provide efficient protection for VoIP against TCP flows in a WLAN scenario.

We study four different combinations of these algorithms in each of the two congestion cases. These are:

1. Use only of the default EDCA mechanism with static MAC parameters and no codec adaptation.

2. Use an adaptive EDCA mechanism where the parameters of the TCP Access Category change at the moment of TCP node rate change (285 sec) and no codec adaptation.

3. Use the Codec Adaptation algorithm under the default static EDCA parameters.

4. Use the Codec Adaptation algorithm together with the adaptive EDCA mechanism.

4.7.2 Analysis

In all the figures presented below we will focus on two stages of the simulation experiment: after the first VoIP rate changes (185sec of simulation) and after the TCP rate changes (285sec of simulation). In the first stage EDCA adaptation is not applied, either is activated or not, since it applies only on TCP flows' rate changes.

Case 1: 5 VoIP calls and 4 TCP flows

Voice : In this case, the cell is not fully using its capacity (is non-saturated) so the first rate changes affecting the two voice flows are not particularly noticed by the rest of the flows and do not provoke any great impact on the total performance of the voice or TCP flows. We can observe in Figure 4.9 how the MOS values remain high and there is no delay increase (Figure 4.10). In fact the Codec Adaptation is not even activated since all the QoS metrics remain in good levels, far below the set thresholds. However, in the second stage of the experiment (after the TCP rate changes), when using a static EDCA mechanism there are some peaks of increased delay that provoke a partial decrease of the obtained MOS. Nevertheless, the delay does not exceed the set

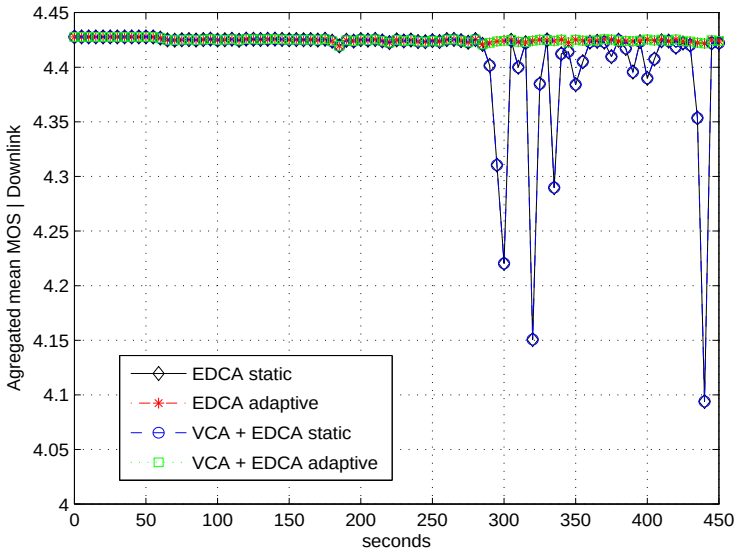


Figure 4.9: VoIP Mean Opinion Score - MOS (5 VoIP calls + 4 TCP flows)

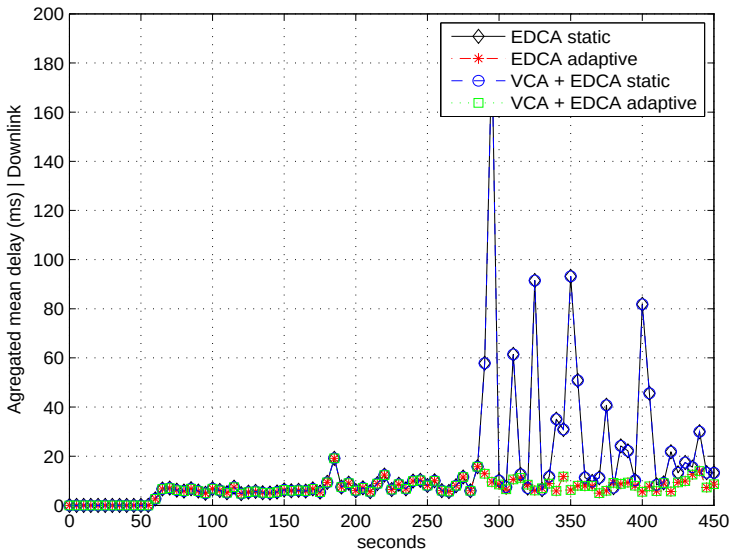


Figure 4.10: VoIP Delay (5 VoIP calls + 4 TCP flows)

threshold of 200 ms and so the Codec Adaptation mechanism is again not activated. This delay increase is avoided if using adaptive EDCA.

TCP : Although one may think that adapting EDCA parameters will lower the total throughput obtained by the TCP flows by further constraining them this is not actually true. Solving the congestion that the rate changes cause, the total performance in

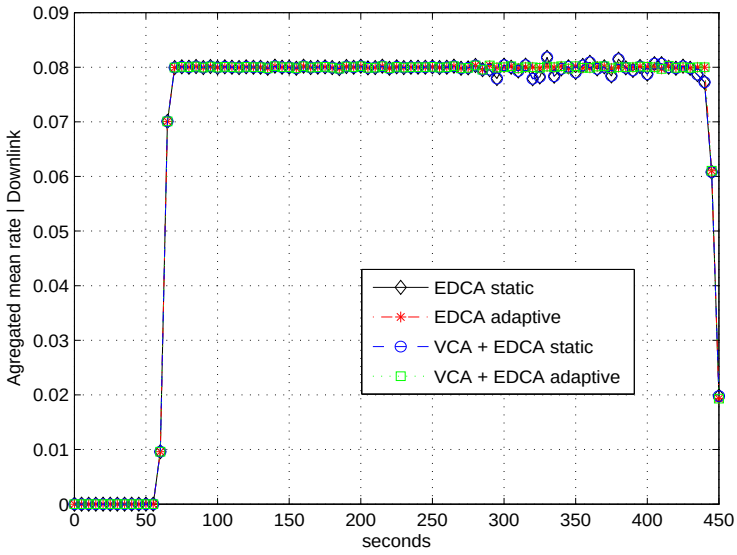


Figure 4.11: VoIP throughput (5 VoIP calls + 4 TCP flows)

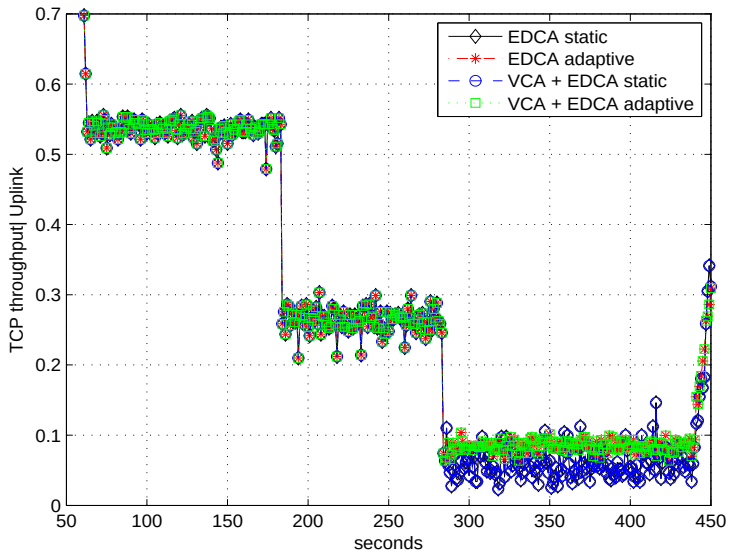


Figure 4.12: TCP throughput - Uplink (5 VoIP calls + 4 TCP flows)

average is actually higher than when using static EDCA. Since the AP is not saturated, downlink ACKs can be also transmitted faster and so the total TCP throughput is higher.

Case 2: 8 VoIP calls and 2 TCP flows

Voice : This case is considered in order to examine the performance of the different

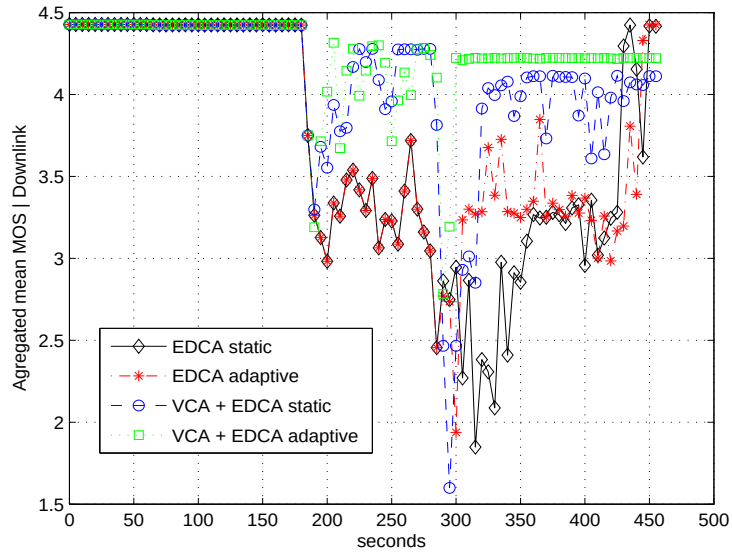


Figure 4.13: VoIP Mean Opinion Score - MOS (8 VoIP calls + 2 TCP flows)

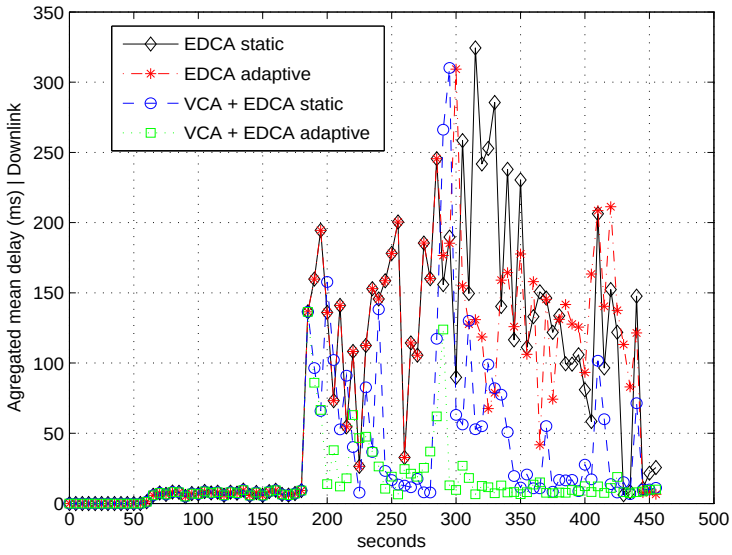


Figure 4.14: VoIP Delay (8 VoIP calls + 2 TCP flows)

mechanisms in a nearly congested cell. Lets first have a look at the results of the delay and MOS of VoIP flows (Figures 4.14 and 4.13). The results show clearly that the combination of Codec Adaptation and adaptive EDCA gives the best performance after both voice and TCP rate changes. Delay is not only maintained low but the variation (jitter) is also very low (very few peaks) and the equivalent MOS is the highest obtained.

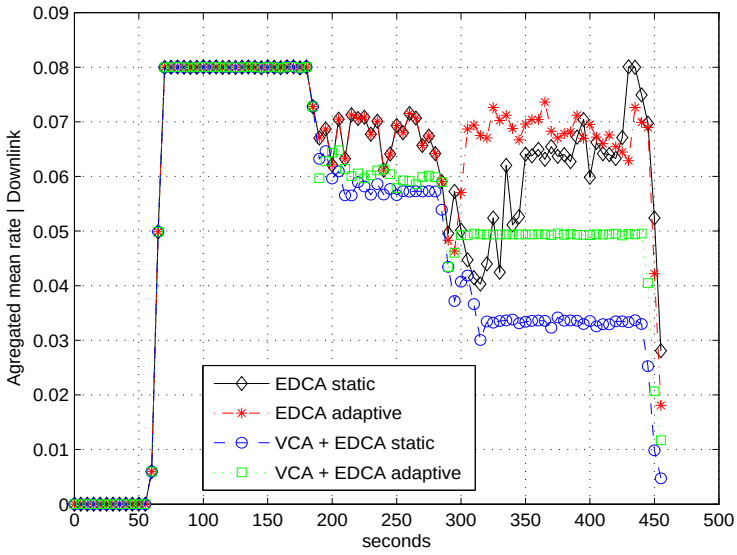


Figure 4.15: VoIP throughput (8 VoIP calls + 2 TCP flows)

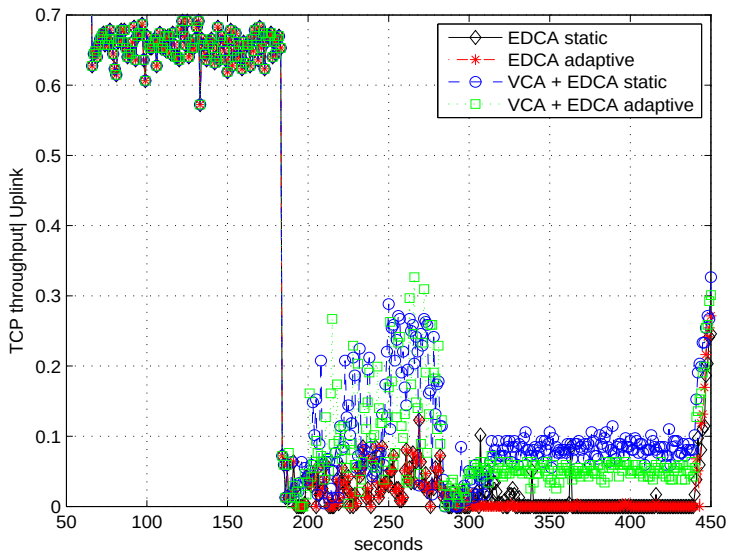
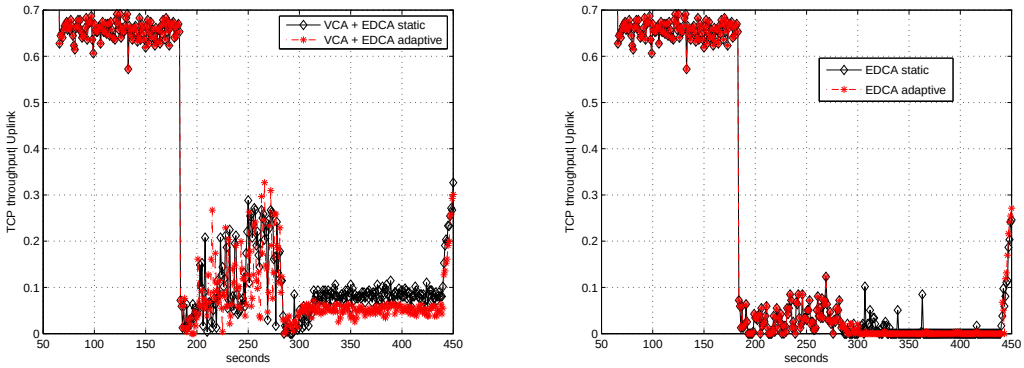


Figure 4.16: TCP throughput - Uplink (8 VoIP calls + 2 TCP flows)

This is because using a combination of both solutions the system can react equally well to both voice rate changes (using the codec adaptation) and TCP rate changes (using the EDCA adaptation). The throughput may be lower than when not using codec adaptation but it is more stable and this is normal since the nodes are now using a lower bitrate codec. In any case this fact does not affect much the obtained QoS as



(a) With Codec adaptation

(b) Without Codec adaptation

Figure 4.17: TCP throughput - Uplink (8 VoIP calls + 2 TCP flows)

seen in the MOS figure. Observe also how the aggregated VoIP throughput (Figure 4.15), although lower since some calls are now using lower bitrate codecs, is very stable, meaning a "smooth" VoIP transmission. This result contrasts the ones obtained when Codec Adaptation is not used.

The second important observation is the performance of the codec selection when used with static EDCA. In this case, the reaction after the voice rate change is similar to before, however after the TCP rate change the codec adaptation alone cannot completely solve the congestion provoked by the TCP flows. We can observe how the average voice throughput becomes very low, clear sign that a large number of calls are changing codecs, yet still the delay is not at the low levels obtained before. This means that most of the codec changes were unnecessary and the EDCA parameters adaptation could have solved the congestion faster and more efficiently. The "greedy" nature of TCP tends to occupy all the bandwidth that the codec adaptation releases and therefore the congestion remains.

In the case that we are not using Codec Adaptation, both EDCA static and EDCA adaptive are unable to react to the first voice rate changes. With the addition of the further TCP rate changes which only increase the congestion levels of the AP, voice performance is unacceptable with MOS values falling below 3.

TCP : One thing that we must notice is the reaction of the TCP throughput to the voice rate changes. The average throughput obtained is almost 70% lower (less than 0.1 Mbps) and this occurs even before the EDCA adaptation is applied (first stage of simulation)(Figure 4.16). This is caused by the high congestion levels on the AP; although TCP traffic is not symmetrical, the delay on the transmission of the response ACKs in the downlink direction affects the throughput of the uplink direction as well.

However, when codec adaptation is active, the extra capacity released by the voice codec changes can be used by the TCP flows and so their throughput is higher than when

not using codec adaptation. The same differentiation is noticed again after the TCP rate changes. With no codec adaptation the TCP average throughput is almost zero due to the high congestion levels of the AP. On the other hand, when using Codec Adaptation the TCP throughput is higher. In Figure 4.17 the two cases of applying or not codec adaptation are separated and the effects can be appreciated better. Additionally, there is an obvious difference between the case of using an adaptive EDCA mechanism or a static one, being the former more restrictive to TCP flows than the latter and leading to lower total throughput, in order to offer an additional protection to the VoIP flows. A more "relaxed" EDCA adaptation mechanism could be used, with a different set of parameters. Nevertheless, the optimization of these parameters is not the subject of this thesis.

4.8 Conclusions

In a purely VoIP scenario, both implementations of the codec adaptation algorithm presented here (centralized and distributed) give satisfactory performance, since no calls are being dropped, there is a fast reaction and correction of the quality degradation and there are minimal packet losses with high average MOS obtained. Comparing the two implementations, it becomes clear that the centralized gives better results as expected, since the Access Point has an overall control of the nodes and the codec they use and provides a more efficient combination of codecs. On the other hand, this means more processing effort for the Access Point and the results on the distributed method are quite satisfactory and give an interesting and almost equally effective alternative to the centralized version.

In the case of a mixed VoIP and TCP traffic scenario, we have explained how neither a Codec Adaptation solution alone nor an adaptive EDCA mechanism can react adequately to both cases of rate changes (VoIP and TCP). A combination of the two would provide the best solution, especially in higher traffic load scenarios. Additionally, we have seen that both mechanisms can cooperate in a joint solution, without one affecting the performance of the other.

CHAPTER 5

Enhancing call admission control with voice codec optimization

5.1 Benefits of cooperation between Call Admission Control and Codec Adaptation

One of the early detected problems when using VoIP services over 802.11 networks was the limited capacity they provided in terms of number of simultaneous calls, and this despite the increase of available bandwidth with the appearance of the different newer versions. This lack of capacity was caused among other things due to the high MAC layer overheads, the contention for the channel between the mobile nodes and the Access Point (uplink/downlink unfairness), and the capacity variations due to the multi-rate channel, as already seen in detail in chapter 3.

Given this limited capacity for VoIP flows, Call Admission Control techniques are necessary. The purpose of a CAC mechanism is to control the actual load of the network and either not accept (block) new call request or drop existing ones when the cell is overloaded (and thus cannot guarantee the necessary QoS). Various methods, analyzed in section 3.4, were proposed, all of them using some kind of capacity metric to determine the actual state of the cell.

However, although there is plenty of previous work focusing on Call Admission Control for new, incoming calls, and others focusing on modifying call parameters of active calls, like packetization interval and codec, in order to correct quality variations (with the most important of them already reviewed in chapter 4), there is very few literature on a combined use of the two techniques which would try to both maximize capacity and quality performance. The work of Chen et al. [19], which deals with the capacity variations due to handoff procedures and uses a codec rate and packetization adjustment to solve it, follows an approach similar to this line. However it is focused

on a totally different scenario, since the capacity variation is not due to the multi-rate effect on active calls but due to new calls arriving from another cell, which can be also controlled by the CAC. Nasser et al. [55] propose a Bandwidth Adaptation Algorithm which is activated when a new call arrives in an overloaded cell, to reduce its congestion levels and accept the new call. However, this work is focused in cellular networks rather than in 802.11 WLANs and again only dealing with new calls rather than a combined solution including already active ones. The idea of Garg and Kappes [33] presented earlier with the CUE index has also similarities to the work presented here. A decision policy is also used in our study, however it is the voice traffic that is adapted instead of the data traffic and in our case the same procedure can be applied as well for active calls in case of rate changes apart from new calls.

We believe that a combined implementation of the two methods, where the codec control is also performed for new calls in co-ordination with the CAC, can permit more flexibility and a significant increase on the overall cell performance.

In this chapter, this combined call admission control scheme and dynamic VoIP codec rate selection module is presented. The CAC is not only able to block (incoming) or drop (active) VoIP calls, but it also includes a decision policy to select the most adequate VoIP codec rate in order to avoid blocking/dropping of calls by adapting them to the current cell situation. A number of adaptation policies are proposed and evaluated, which try to optimize the network usage according to different criteria: call quality, number of simultaneous active calls, fairness, etc. These are simple yet representative policies, which provide a basic insight into the intrinsic trade-offs found in the most common adaptation policies, when applied to multi-rate environments. Results focus on the relation between the various metrics of interest for designing a successful VoIP service: offered load, blocking / dropping probabilities, voice quality and simultaneous number of active calls.

We must point out that although our work tries to provide an enhanced CAC mechanism, we are not actually proposing a new CAC mechanism from scratch. The codec adaptation enhancement could be applied with practically any of the CAC algorithm mentioned in the previous sections with small modifications.

5.2 VoIP capacity in a multi-rate/multi-codec scenario

In order to calculate the voice capacity of the channel at flow level, we have used the capacity index proposed by Hole and Tobagi in [39] as a reference upper bound for the number of maximum accepted calls N in a 802.11 cell. However, since our study focuses on a multi-rate WLAN and nodes can use any codec chosen from a set of available codecs, we have modified this first index to produce a new multi-rate/multi-codec capacity index to use in our simulations. We have seen briefly in chapter 3 these

T_{VOICE}	PLCP Preamble & header	192.0us
	MAC Header + FCS	20.4us
	IP/UDP/RTP header	29.1us
	Voice Data	(Voice octets x 8 /11)us
T_{ACK}	PLCP Preamble & header	192.0us
	ACK Frame	10.2us

Table 5.1: Component Times of T_{VOICE} and T_{ACK} for 11Mbps data rate

two indexes, and we will describe them here in detail.

a) Upper bound index of call capacity of an 802.11 network, by Hole and Tobagi

In their work in [39], Hole and Tobagi have provided a mathematical upper limit of an 802.11 network capacity of VoIP calls. The study follows the assumptions that there are no collisions, no link errors and that all packets arrive at the playout buffer before their respective playout deadline. In this scenario the only constrain for admitting or not a new call is the available throughput. Assuming that all calls use the same codec and the same link rate, then the upper bound of the value N is given by the equation:

$$N = \left\lfloor \frac{1}{\frac{B}{L}[2 \cdot T_s(R^d) + (T_{slot} \cdot \frac{CW_{min}}{2})]} \right\rfloor \tag{5.1}$$

where T_s is the time required to transmit a VoIP packet (see Equation 3.1 of chapter 3), B/L is the transmission packet rate for the VoIP codec used and T_{slot} is the duration of an empty SLOT. The default parameters used assuming a data rate of 11Mbps are given in table 5.1. The default values for CW_{min}, $SIFS$, T_{SLOT} and $DIFS$ are respectively 31, 10us, 20us and 50us.

b) Modified multi-rate/multi-codec capacity index

Considering the existence of multiple transmission rates and VoIP codecs in our scenario, an extension of the capacity index proposed by Hole et al. [39] is presented. Using the same argument and assumptions as before, a state is feasible if it satisfies the condition:

$$\sum_{r \,\epsilon\, \mathcal{R}} \sum_{c \,\epsilon\, \mathcal{C}} n(c,r) \frac{B(c)}{L(c)} \left(2T_s(c,r) + T_{SLOT} \frac{CW_{min}}{2} \right) \leq 1 \tag{5.2}$$

where $n(c,r)$ is the number of calls using codec c and rate r, $T_s(c,r)$ is the time required to transmit a VoIP packet using codec c and rate r, and $B(c)/L(c)$ is the transmission packet rate for VoIP codec c.

In order to validate the accuracy of this upgraded index, a comparison was performed for a number of different test cases against the results obtained from simulations using NS-2 [58]. In each case, given a fixed number of calls using a codec C_a and transmitting at rate R_a, the maximum number of calls using codec C_b and rate R_b accepted in the

cell without provoking congestion has been calculated both by the MR/MC index and experimentally by the simulator. The results give a 100% match between the two.

We will use this modified multi-rate/multi-codec index as our voice capacity indicator for the simulation experiments in flow-level described later. An ideal CAC mechanism is used only to guarantee the network stability (i.e that there are no more calls active than the maximum cell capacity as provided by this index).

5.3 Increasing cell efficiency using codec adaptation

Although typical CAC mechanisms can reserve a portion of the available capacity to new calls arriving to the system (if enough capacity exists to accept new calls), they are unable to react to the sudden rate changes occurring while a call is still active and to the variations on the total capacity that this rate change implies. Once a call is accepted on the cell, the only available way to react to congestion occurring afterwards is by dropping the call that caused it. If on the other hand there was some capacity increase (possibly released by a node due to its rate change to a higher one), this capacity remains unused until a new call arrives.

In the combined CAC/VCA (Call Admission Control with Voice Codec Adaptation) mechanism proposed here, a VoIP codec selection module is co-located with the admission control. It suggests the VoIP codec to be used on each case following a given policy. For example, a valid policy would be "always use the G.711 VoIP codec, independently of the considered rate, for it gives the best voice quality". The policies used in this study are introduced in the next section. The idea behind this enhancement is to give to the Admission Control mechanism a new tool which could permit to free some used capacity in order to lower the congestion level of the cell. This same tool could be used both in the case of new incoming calls which find the cell saturated as also for calls suffering from the multi-rate effect described above.

Releasing capacity is possible with a codec change of one or more of the active calls on the network. To give a simplified example, let us consider the case of an 802.11 BSS using all its capacity, where N calls are active, all using G.711 codec at 64Kbps and all transmitting at 11Mbps data rate. The upper bound of the value of N as calculated using the original index (equation 5.1) would then be $N = 8$. Consider now a new call request arriving with the same characteristics (codec and data rate). Assuming 8 calls already active at the cell, the new call cannot be accepted and the request must be rejected (blocked). However, if this last call arrives using the G.729 codec at 8Kbps instead of the G.711 (or if the codec selection module forces the use of this lower rate codec), then the new state as calculated using the modified equation 5.2 would now be feasible and the new call could be accepted.

Another case would be if this new call could not change codec for some reason

(e.g. not supporting the proposed codec) but one of the other active calls on the cell could lower its codec to the G.729 codec, then the same result would have been obtained. A similar situation would come up under any rate change. This is the basis of the suggested method of implementing the codec adaptation in co-operation with the admission control, on the entrance of new calls as an addition to reacting on rate changes.

A block scheme of the considered admission control can be seen in Figure 5.1. The combined CAC/VCA is requested to act in two cases, referred to here as "capacity variation states":

1. when a new call request arrives, with rate λ (in $calls/second$)

2. when a rate change occurs, with rate γ (in $changes/(call \times second)$)

The case of the capacity variation due to a rate change is the one extensively analyzed in the previous chapter and can be controlled with the VCA solution.

In the case of a new call request, however, VCA cannot react in the way defined before since no congestion and no quality degradation is yet visible (before accepting the new call). Thus a prediction of the capacity usage after the entrance of the new call is calculated using the cell state information available (number of active calls, rate and codec used by each one, etc.). If the result capacity is higher than the actual system capacity, then the new call cannot be accepted as it is. Therefore, a decision policy is consulted and the codec adaptation mechanism is invoked to perform the necessary codec changes that will permit the admittance of the new call. Notice that in this case the codec adaptation procedure is not exactly the one described in the previous chapter, since no RTCP or MAC information exist for new call requests. At the same time the active calls do not present (yet) any symptoms of congestion so there is no way for the algorithm to measure and choose which call should change codec according to the thresholds of performance metrics such as packet losses and delay. Therefore, the exact way in which these codec changes are performed is dictated by the chosen decision policy, explained in more detail in the next section.

Note though that these changes are not actually carried through unless the result codec combination, as calculated first theoretically using a capacity index such as the MR/MC index, is positive, i.e. unless the new call can be accommodated in the cell under the new codec combination. If this is not the case (i.e. if no codec re-arrangement can release the necessary capacity), any useless codec change is avoided and the capacity variation state is reversed by dropping or blocking the problematic call.

Special attention has to be placed in not confusing the CAC decisions and control at flow level with the codec adaptation procedure with its monitorization at packet level. The two algorithms (CAC and VCA) should work in parallel and in close cooperation but each on a different area. The VCA algorithm from one hand, with the monitoring

and adaptation procedure described already in detail in chapter 4, should be followed at packet level to provide a real-time instantaneous vision of the QoS of the active flows, perform the codec change through SIP re-Invite and monitor the recovery of the system. The CAC mechanism on the other hand, should be the responsible of controlling the available instantaneous cell resources and block or permit the entrance of new calls accordingly.

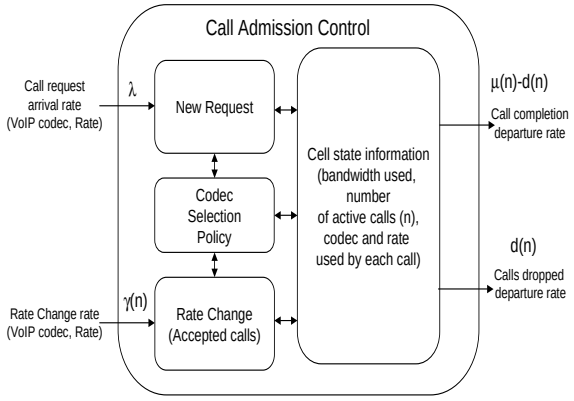


Figure 5.1: Call Admission Control scheme with VoIP codec adaptation

5.4 Decision policies

What is actually moving the combined CAC/VCA module proposed here is the decision policies behind it. A decision policy determines the action to be taken under each situation, depending on the criterion to be maximized. For example, if the main objective is to obtain a maximum average MOS value ($E[MOS]$), the policy would indicate that all calls should start with the highest codec and the ones who cannot meet this criteria will be dropped. This is in fact one of the policies described in more detail below, the policy P0-H. There is a clear trade-off between the different criteria, especially between the achieved quality and the quantity of accepted calls. In the policies described below the optimization of this trade-off is considered, as also the optimization of other relevant criteria like blocking and dropping probability, resources usage, signalling overhead etc. Nevertheless, the policies described here are not the only ones available; since they are simple on their structure, they can provide the basis from where various other combinations can be also considered to achieve new more complex ones, with different optimization results.

The decision policies can be applied both for new incoming calls as also for congestion provoked due to the multi-rate effect. Additionally, a combination of the above cases is also possible, using one policy for new calls and a different one for the already active

ones. Note that, the main objective of the decision policies is to decide how many calls should change codec (none, one, many) and in which order, depending on the metric that we want to optimize.

In order to categorize better the proposed policies we have divided them in three main groups: the ones that do not involve codec adaptation, known as Standard CAC or Non-adaptive policies, the ones that only perform adaptation on one call, named here Simple-Adaptive policies and the ones that adapt the codecs of more than one call, known as Multi-Adaptive policies. Each category with its specific characteristics and the policies that it includes is explained next.

5.4.1 Non-Adaptive Policies

This category includes standard, fixed-codec policies, where the calls continue through all their duration with the same codec they begin with. No codec adaptation is performed and when any congestion occurs during the capacity variation states the drop/block solution is applied. Three policies are considered depending on which codec each call chooses to start with. The first two ("start with highest codec" and "start with lowest codec") are used mainly for comparison and for having a highest/lowest theoretical value for the MOS factor and for the active calls number. However, normally each client can choose to start with any of the available codecs and so the third policy represents this general random codec case.

The fixed codec policies offer the simplest and fastest solution, and thus are the ones most commonly used until now. But on the other hand, this is also the least flexible solution and for a network where cell conditions change fast, not adapting to them implies high blocking and dropping probability or low quality of service offered. Additionally, there is no possibility for capacity re-use and efficient use of the network resources when the total available cell capacity increases by a rate change upwards.

- P0-L: Start with lowest codec. All calls enter the network with the lowest codec of the set $\mathcal{C}$. This permits a higher number of accepted and active calls (less blocking and dropping ratio) but with the counterpart of the lowest offered QoS ($E[MOS]$ achieved).

- P0-H : Start with highest codec. All calls start with the highest codec of the set $\mathcal{C}$. Obviously, this is the policy with the highest blocking and dropping ratio and the less number of simultaneous active calls. On the other hand it is one of the simplest policies to implement and provides an upper bound for the achieved average QoS of the VoIP flows.

- P0: Start with random codec. Using this policy, calls can start with any codec from the set $\mathcal{C}$.

Thus, the reaction of these policies to any congestion situation is to block (or drop) the problematic call, be it a new incoming call or an active call changing rate.

5.4.2 Simple-Adaptive Policies

In the Simple-Adaptive policies category, the only calls that change their codec are the ones that actually created the capacity variation state. This includes both new calls trying to enter and finding the cell capacity already fully used, as also nodes that change to a lower data rate in the middle of a call provoking a decrease on the total cell capacity. These policies can be considered the fairest ones, since the only call affected by a codec change is the one that created the problem. However, they have less degrees of freedom when it comes to codec adaptation since the set of calls that can change codec is very limited.

- P1: Codec-Rate Pairing. A specific VoIP codec is assigned to each transmission rate, so as to equalize the capacity consumption across transmission rates, i.e.: $B_i(v_a, R_a) = B_i(v_b, R_b), \forall a, b$, where a and b belong to the set of codec-rate pairs. Therefore, a STA using transmission rate R_a will use the VoIP codec v_a. Either for an incoming call or for a rate-changing call, there is only one possible combination of codec-rate and if the node cannot support this combination, the call is blocked/dropped. Notice that this is a "fair" policy as all STAs have a voice quality proportional to their transmission rate.

- P2: Change to any lower. Similar to the above policy, with slightly more options since the set of available codecs to choose from is bigger. The idea is to admit a call if it accepts to enter/continue using a codec equal to or lower than the one assigned for its rate according to the previously explained codec-rate pairing. This policy allows each VoIP call to proceed with a worse voice quality but reducing its blocking and dropping probability, as it is able to check if lower rate codecs would be better suited to the network conditions. This would allow, for example, an incoming call to be accepted in an otherwise saturated network, by accepting to use an "unfairly" low rate codec for its transmission rate.

5.4.3 Multi-Adaptive Policies

Using the policies of this category, any number of calls can change codec in order to solve the congestion. It is the most complex solution but also the one that gives better results in terms of dropping and blocking ratio, especially under heavy load conditions. The basic structure of the three policies included in this category is the same, the only difference lies on the metric they use in order to chose the next call to change. Note that the first call to pass from the adaptation process is always the one that provoked the capacity variation state, since adapting the codec of the slow call performs better

than adapting the codec of the fast ones [74]. However, there are different criteria that can be used to select which of the other calls have to suffer a VoIP codec degradation despite experiencing higher transmission rates (in the case that the codec change of the slow one did not solve the congestion). The criteria used in this study are an index based on the codec-rate combination, the "age" of the call and the randomness.

- P3-A: Change by Age: The duration of the call is taken into account as a criterion of which call will change codec next. The calls that have been active for less time ("young" calls) will change before the ones that have been active for longer ("old" calls). By using this parameter, the calls that are more probable closer to finish ("old" calls) will not be disturbed with a codec change unless really necessary, while the young calls which have higher probability to continue for longer time under this situation, will be the first to adapt to it.

- P4-I: Change by Index: A codec-rate combination index is an index of how appropriate the codec used is compared to the actual transmission rate. Based on the same concept as the policy P1, there is one codec-pair combination considered ideal for capacity usage and all the other combinations can be higher or lower compared to this value, higher meaning using more capacity than they should (e.g. high codec with low rate) and lower meaning the opposite (e.g. low codec with high rate). All calls are ordered according to their codec-pair index and the one with the highest value will be the one to change first.

- P5-R: Change randomly: Finally, a random change method is also considered for the sake of comparison. It is expected to give similar results with the above techniques. In this method, the next call to be changed is chosen randomly between the set of active calls on the cell.

5.5 Performance results

5.5.1 Scenario description

A scenario which consists of an IEEE 802.11b/e hotspot that includes a QoS-enabled AP (QAP) and a number of QoS-enabled mobile stations (QSTA) is considered. Only VoIP traffic is present in the cell, with call requests arriving with rate λ following a Poisson process, and with call duration following an exponential distribution with mean equal to $1/\mu = 240$ seconds, so that the traffic load A varies from 1 to 50 Erlangs. The frequency of rate changes per second and mobile node is $\gamma = 1/\tau$, where τ is the inter-arrival time between calls, which follows an exponential distribution.The available set of rates is $\mathcal{R}$ = {11, 5.5, 2, 1} Mbps. Nodes can support a random set of codecs, subset of $\mathcal{C}$ = {G.711, G.726, G.729, G.723.1}. A C++ simulator has been implemented based on the

COST simulation toolkit [18]. The reason for using this simulator instead of the NS-2 is the need to have control of the voice calls at *flow level* in order to obtain flow level metrics (e.g. blocking and dropping rate) and apply Admission Control policies, option that the NS-2 simulator does not include. The hotspot capacity is forecasted using the multi-rate/multi-codec index (equation 5.2).

When a rate change occurs, the new rate is randomly chosen from the set $\mathcal{R}$ following a uniform distribution, being this new rate either higher or lower than the current. Note here that we focus mainly on the case of a rate decrease since it is the most critical for the user. Although all policies could be applied and react in an equivalent manner in case of a rate increase, we have chosen to apply only the simply adaptive policy P1 in the increase case, policy which only changes the codec of the call that suffers the rate increase. The motivation of this decision was to be able to use the available extra capacity and obtain a better QoS for the voice flow in question without however bothering the other users with unnecessary codec adaptation, since their QoS has not been affected.

5.5.2 Result Analysis

Three combinations of the CAC with the codec adaptation algorithm were tested: a) apply codec adaptation joint with CAC for new calls and drop directly any call causing congestion at rate changes, b) apply codec adaptation only during rate changes and use simple CAC for new calls (block any that cannot fit), and c) apply codec adaptation both with CAC for new calls and at any rate change causing congestion. The results of these simulations are presented here.

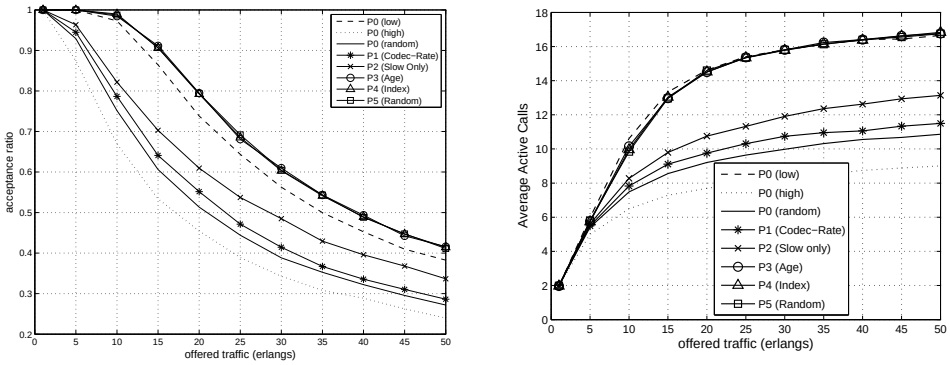
a) Applying Codec Adaptation on new calls only

In Figure 5.2.a observe how the acceptance ratio is significantly increased with the use of adaptive policies, especially the multi-adaptive ones (P3-P5). Even when a new call arriving finds the cell capacity already fully used, a codec change of this call and/or some of the others permits its entrance to the cell, liberating some extra capacity. In particular for the cases of low traffic load (A lower than $10 Erlangs$) the acceptance ratio with the multi-adaptive policies is almost 100%.

The same result can be observed in the high average number of active calls (Figure 5.2.b) and the low blocking ratio (Figure 5.3.a). However, dropping ratio (Figure 5.3.b) remains high, and especially using simple-adaptive policies is almost as high as the upper bound provided by the P0-High policy, where all calls start with the highest codec. Since no action is taken to relieve the multi-rate effects calls are directly dropped when the new situation after the rate change is unsustainable.

Finally we can observe at the average MOS ($E[MOS]$) figure 5.3.c that with the increase in the traffic load (for values of A higher than $25 Erlangs$) more calls are asked

to enter using the lowest possible codec (G.723.1 in our experiments), which leads to a total MOS value of around 3.6. Therefore, for high traffic load, multi-adaptive policies tend to give an average quality similar to the P0-L policy, since most of the calls are obliged to enter using the lowest available codec. On the other hand however, the dropping ratio remains much higher than the one given by P0-L. The reason for this is that there are still calls starting with a higher codec if they do not support the lowest one and they are later dropped, while using P0-L these calls will have been blocked from the beginning. All considered, it can be concluded that in this case of high traffic load and restricting codec adaptation only to new calls it is generally better to use P0-L. If on the other hand the traffic load is lower, multi-adaptive policies give the best overall performance.



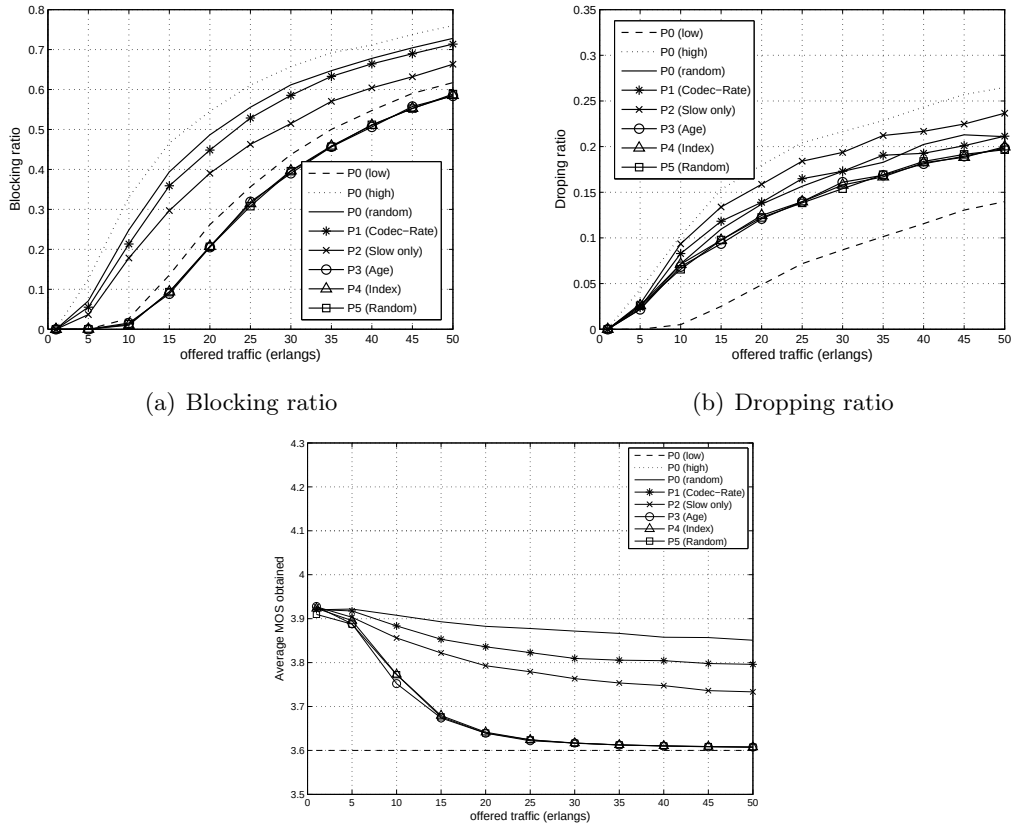
(a) Acceptance Ratio (b) Average number of active calls

Figure 5.2: Codec adaptation on new calls only (I)

b) Applying Codec Adaptation on rate changes only

In Figures 5.4 and 5.5.a we can observe how the acceptance ratio and the number of average active calls remain small (what is equivalent to a high blocking ratio) even when using the adaptive policies, since these (and thus codec adaptation) are not applied in the entrance of new calls and so there is no possibility for using the capacity more efficiently. As an example, compared to the previous case and for a traffic load of $10 Erlangs$, acceptance ratio is now at 70% for all adaptive policies, much lower than the almost 100% achieved earlier with the multi-adaptive ones. The highest number of simultaneously active calls, achieved for the highest traffic load case, is only 12 calls for the adaptive policies, significantly lower than the 17 that we obtained in the previous experiment.

On the other hand though, the dropping ratio (Figure 5.5.b) is significantly decreased, becoming almost zero when using multi-adaptive policies. This means that, although the capacity is still quite limited and less calls manage to enter the cell, once they are accepted at the network they can at least continue and finish without being



(a) Blocking ratio

(b) Dropping ratio

(c) Average Mean Opinion Score

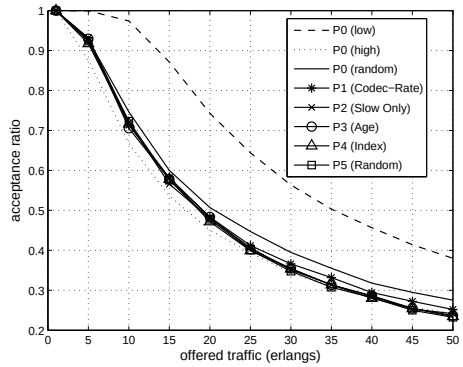
Figure 5.3: Codec adaptation on new calls only (II)

interrupted and dropped and with an acceptable QoS guaranteed. See in Figure 5.5.c how the average MOS is maintained higher than before, even at the highest traffic load of 50 Erlangs.

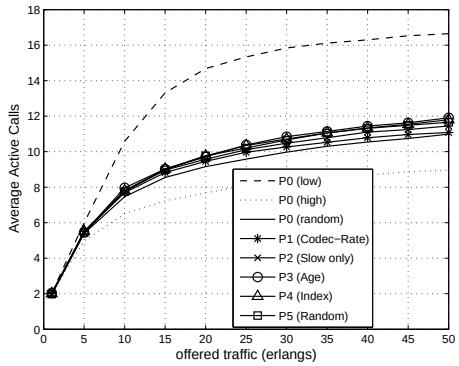
Compared to the non-adaptive policies, although policy P0-L offers the highest capacity in terms of number of simultaneous calls, an expected result since using the lowest codec, the average MOS provided is the lowest possible. The exact opposite case is achieved using P0-H. A fair trade-off between number of calls, dropping and $E[MOS]$ is achieved using the multi-adaptive methods.

c) Applying Codec Adaptation both on new calls and on rate changes

When applying codec adaptation both on the entrance of new calls and on rate changes, we can obtain a complete solution, which decreases both blocking and dropping ratio (Figures 5.7.a and 5.7.b). More calls are accepted in the network and less are dropped interrupted before finishing. The high number of active calls obtained however, comes at the expense of a lower codec for most of them. Again, there is a trade-off between capacity and quality, since the MOS when using the multi-adaptive policies

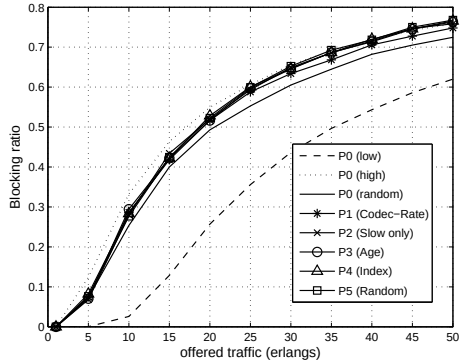


(a) Acceptance Ratio

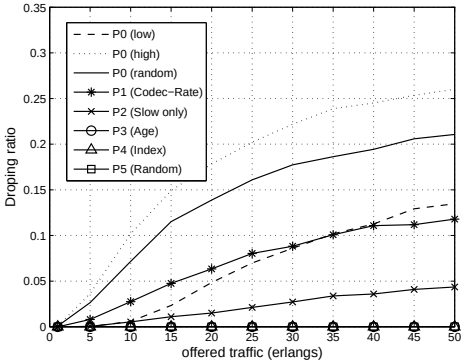


(b) Average number of active calls

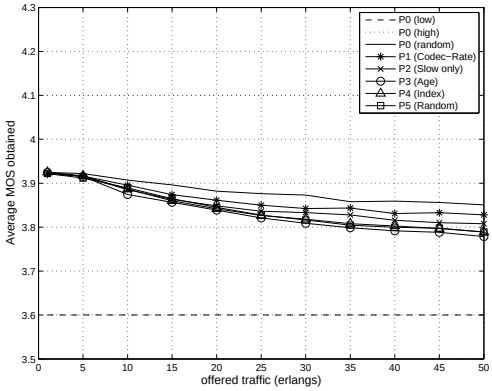
Figure 5.4: Codec adaptation on rate changes only (I)



(a) Blocking ratio



(b) Dropping ratio



(c) Average Mean Opinion Score

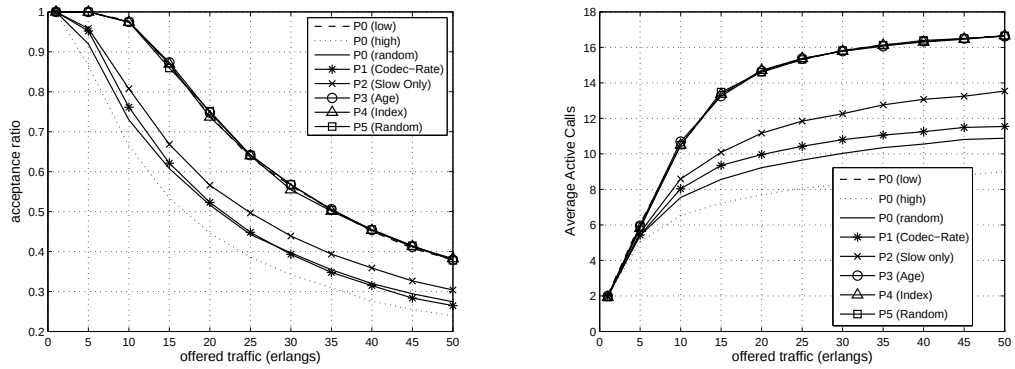
Figure 5.5: Codec adaptation on rate changes only (II)

now tends to values similar to policy P0-L, the policy showing the lowest MOS, as the calls are changing to the lowest supported codec as the traffic load increases (Figure 5.7.c).

Comparing them to non-adaptive policies, multi-adaptive policies obtain almost exactly the same performance as the P0-L, meaning the highest acceptance and lowest dropping ratio. On the other hand, the MOS obtained is quite higher, especially in the lower traffic load cases.

Simple-adaptive policies on the other hand, approach mostly the results obtained from the P0-R, that is the case where calls enter with the random codec and no adaptation is performed. Their flexibility is limited since they can only adapt one call and so their performance is only slightly better than this of P0-R. However, it is on the dropping ratio (Figure 5.7.b) where the effect of applying codec adaptation, even to one call, is appreciated. The obtained dropping ratio is very small, equal to the one that the multi-adaptive policies can give. And at the same time, since less calls change codec, a higher average MOS value is obtained (Figure 5.7.c).

In all cases above, we observed no significant difference between the three variations of the multi-adaptive policies (i.e. the order of the calls to change codec). In single-adaptive policies, policy P2 performs slightly better than P1 in most of the cases, due to its increased flexibility in the decision of the new codec.



(a) Acceptance Ratio

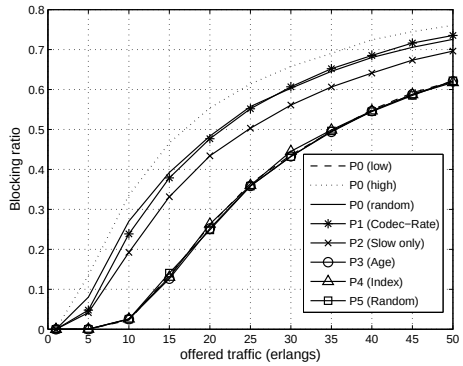
(b) Average number of active calls

Figure 5.6: Codec adaptation on both new calls and rate changes (I)

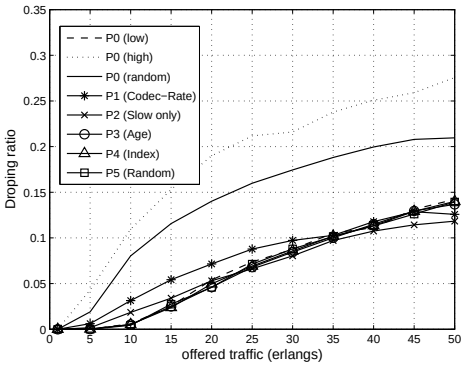
5.5.3 Conclusions

From the study above we see how a combination of the codec adaptation algorithm with a Call Admission Control mechanism is not only possible but also beneficial for the QoS of the active calls and also for increasing the number of accepted calls and the channel efficiency. We have evaluated the performance of the algorithm under three different scenarios and we have examined a number of decision policies that can tune the codec selection procedure.

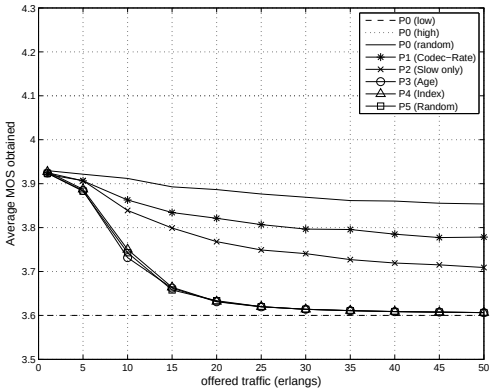
The main observation that we can extract from the results is that we need to take a number of different metrics into account when deciding which policy performs best.



(a) Blocking ratio



(b) Dropping ratio



(c) Average Mean Opinion Score

Figure 5.7: Codec adaptation on both new calls and rate changes (II)

Although some policies minimize dropping probability, they have a direct impact on the average MOS of the calls, and viceversa. So whether to choose one policy over the other depends highly on the parameter that we want to optimize. The trade-off between capacity and quality is evident also here and not always maximizing the capacity is the main scope of a network, especially when dealing with a sensitive VoIP network.

In the next section we introduce a parameter that can combine these two otherwise contradictory factors, quantity and quality, to one single factor which could provide some information on how the cell is performing from both points of view.

5.6 The Q-Factor, a new quality and quantity metric

We have seen previously that making a choice over which decision policy is the best in absolute terms is a difficult task, since achieving a high value in all of the system performance metrics under study is quite complicated. There are policies that maximize the number of calls but the average QoS as expressed by the $E[MOS]$ value is low compared to other policies. There are others that result in very small blocking and

dropping ratio but with the drawback of many codec changes which can be annoying to the user.

Although the usual trade-off between quality and quantity is to be expected here as well, in order to provide some kind of qualification for the policies under study and decide where and how to use Codec Adaptation depending on the traffic load, a new Grade of Service (GoS) factor which combines the above metrics is needed. This "Quality and Quantity" factor, called Q-Factor or $\overline{Q}$ (complementary, as the goal is to minimize it), can provide a combination of blocking (B) and dropping rate (D) with the average normalized MOS (MOS_n) in the hotspot: the first two serving as quantitative metrics of the system's capacity to accept new calls and finalize correctly the accepted ones, while the third one as a speech quality metric. In the next sections we will present briefly this new metric, how the Q-Factor can be calculated and how it can be used in order to tune the codec adaptation process.

5.6.1 Calculating the Q-Factor

Call Admission Control in cellular networks is designed trying to guarantee both grade of service at call level and quality of service at packet level [21]. Unfortunately there is not one single metric that can combine them both and provide an overall view of the performance of a CAC mechanism and consecutively of the cell capacity itself. The Q-Factor is a system performance metric that tries to respond to this demand by combining both dropping and blocking feedback as GoS parameters at call level with the packet loss and delay metrics representing the QoS at packet level, that can be found included in the E-Model/MOS calculation for the VoIP case.

Grade of Service is the probability of a call to be blocked, either entering the cell as a new request or as a handoff call from another cell. In [4] GoS is calculated as:

$$GoS = \alpha \cdot P_h + P_n \tag{5.3}$$

where P_h is the handoff failure probability and P_n the new call blocking probability, while $\alpha = 10$ is a weight variable to represent the penalization for dropping a handoff call relative to blocking a new call.

In the scenario used in this thesis, no handoff calls are considered and the dropping ratio refers only to the already active calls in the cell, so the above GoS can be modified by simply replacing P_h with this dropping ratio. Considering this change as also the incorporation of the MOS value, the Q-factor $\overline{Q}$ is calculated as:

$$\overline{Q} = \beta(1 - MOS_n) \cdot (\alpha \cdot D + B) \tag{5.4}$$

where B and D are the blocking and dropping probabilities and MOS_n is the normalized MOS between 0 and 1 and computed from $MOS_n = E[MOS]/MOS_{max}$ with

$MOS_{max} = 5$. Notice that $E[MOS]$ can be computed easily in real time (for example, using the E-model) from the feedback of the voice packets transmission. Here, β is simply a scaling parameter to maintain the $\overline{Q}$ similar to those values provided by the usual GoS definition in cellular networks [4]. For the simulation experiments following we set β to a value equal to 10. As a general rule, users have a stronger reaction against call dropping than against lower call quality, therefore we give the highest weight to dropping rate and the lowest to MOS. Further studying in this area is ongoing and will be discussed in the Future work, chapter 7.

5.6.2 Tuning the codec adaptation based on Q-Factor

Now, lets consider again the experiment started earlier in this chapter (in section 5.5), and again the three different cases of applying codec adaptation only on new calls (case (a)),only on rate changes (case (b)) or on both (case (c)). This example will help illustrate how the Q-Factor is calculated and how it can be used to extract conclusions about the policies.

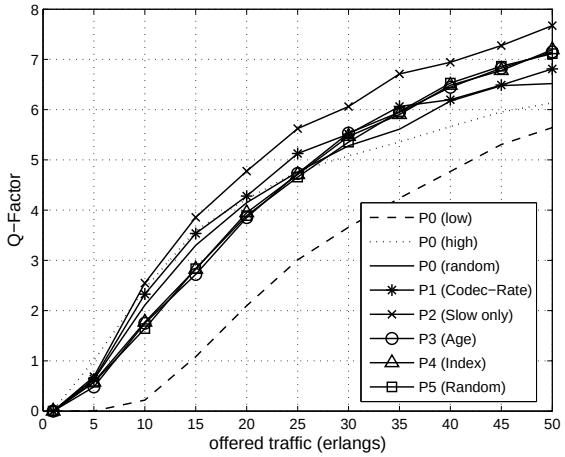
The values of blocking, dropping and $E[MOS]$ obtained from the previous experiment were combined here following the equation 5.4 in order to calculate the equivalent Q-Factor value, plotted in Figure 5.8.

The main conclusions that can be drawn from the Q-Factor results are:

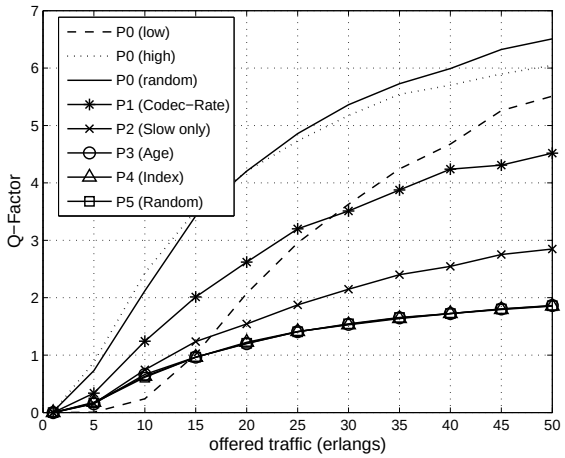
- a solution based on multi-adaptive policies is the general recommended option for achieving the best trade-off in quality and quantity. The policies of this category obtain one of the lowest $\overline{Q}$ value in all of the studied cases. The only policy that outperforms multi-adaptive is the fixed P0-L policy and only in case (a), due to the lower dropping probability of it.

- the highest weight in the $\overline{Q}$ calculation is given in the dropping rate parameter as explained previously. In cases (a) and (c) an increased number of calls are admitted in the cell using low bitrate codecs, since we apply codec adaptation on the entrance for new calls. However, many of them are actually dropped afterwards before finishing and especially when the traffic load increases. This can be depicted by the high $\overline{Q}$ values of these cases compared to case (b).

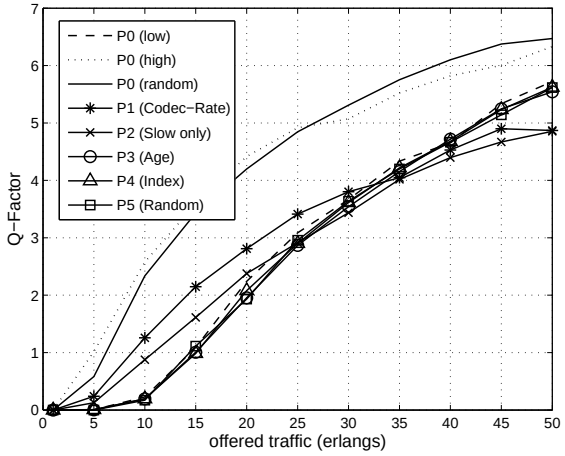
- based on this last fact, apply codec adaptation on rate changes only and normal CAC for new calls (case b)) is the most efficient strategy for increasing quality and capacity, especially when used with multi-adaptive policies. Notice that in cases (a) and (c) for high traffic load, a high value of $\overline{Q}$, higher than 5.5, is calculated for all policies. However in case (b), a lower overall value of $\overline{Q}$ is obtained from all policies, and even during the highest traffic load offer (50 Erlangs), there are policies that result to a $\overline{Q}$ value lower than 2 (the best performing, multi-adaptive). So apart from comparing the different policies between them under



(a) Codec adaptation on new calls only



(b) Codec adaptation on rate changes only



(c) Codec adaptation on both new calls and rate changes

Figure 5.8: Q-Factor ($\overline{Q}$)

the same scenario, the Q-Factor can be also used as an indicator of *when* to use codec selection. In this example, the results show that using the VCA algorithm on rate changes achieves the lowest Q-Factor value (less that 2 even at high traffic) thus gives the best overall performance.

Extending this last observation and as a general conclusion, a cell under low traffic conditions (in our case $A \leq 5$ Erlangs) has the flexibility to permit codec modifications, so as to try to accept more new calls (overall capacity increase) without heavily impacting the dropping and blocking probabilities: Hence the low $\overline{Q}$ values both for case (b) and case (c) in low traffic load. However, under heavier load conditions, there is less margin for adaptation and so the best solution to keep all three metrics in check is to change codec only as a response to rate changes (case (b)), and to focus in guaranteeing a low dropping probability for the already accepted calls.

We have seen that by using the $\overline{Q}$, we can get valuable conclusions for the performance of the system, the strategy to follow (i.e. when to apply codec adaptation) and the policy that give the best trade-off between quality and capacity. The most important contribution of the $\overline{Q}$-Factor is that we can obtain all this information by using just one metric. We will see more on how this factor can be used next, after first introducing another important parameter in the codec adaptation procedure: the codec complexity.

5.7 Codec complexity and its impact on Codec Adaptation

Looking at the results until here, where all the policies are compared against each other using the various performance metrics, it can be concluded that in most of the cases the multi-adaptive policies give the best performance all metrics considered. However some may argue that the policy P0-L (accept all calls using lowest available codec) performs equally well, especially in terms of blocking and dropping ratio. Even if the achieved MOS is actually the lowest one (even for low traffic load), the voice quality offered would still be in acceptable levels and this would simplify highly the codec selection process, avoiding all the annoying codec changes

Nonetheless, this is only partially true. This study would not be complete without considering another important effect that a codec change might entail, intentionally omitted until now for simplicity: the codec complexity. Some codec compression techniques require more processing power than others. The complexity of a speech-coding algorithm dictates the computational effort required and the memory requirements. This most of the times can be also reflected in the battery consumption of a mobile device. Thus complexity is an important cost factor for implementing a codec and as a

rule of thumb increases with decreasing codec bitrate, meaning that the lowest bitrate codec is normally the one with the highest complexity.

Codec complexity refers to the amount of memory and processing time needed to decode each frame of compressed data. Processing time is measured in term of Millions of Instructions Per Second (MIPS), and memory is the amount of RAM needed to process a frame. For example, G.711 is estimated to require less than 1 MIPS and 1 byte of memory, in contrast to G.723.1, which requires almost 18 MIPS of processing power and 2 Kbytes of memory [73]. In Table 5.2 the complexity of the codecs used in this thesis in terms of processing time is shown. The values are approximated and normalized with respect to G.711 codec, using a relative scale where G.711 is 1 and G.723.1 is 25 and taking as a guide the values found in [38] and [41] among others.

Codec	Complexity $\zeta(c)$ (MIPS)
G.711	1
G.726	10
G.729A	15
G.723.1	25

Table 5.2: Codec complexity $\zeta(c)$ in MIPS

Taking this into account and to complete the study including all the parameters, we will present next the effect that the codec complexity and the node's processor capacity have in the decision process.

5.7.1 Limiting node processor capacity

In the previous results there has been no concern for the codec complexity and the possibility that a node may not support all available codecs due to its limited processor capacity was not taken into account. We study here the case that the processor capacity is the one that dictates the set of available codecs $\mathcal{C}$ (chosen randomly in the earlier experiments of chapter 5). We will see there is a difference observed in the behavior of the codec adaptation algorithm and the policies under study: lower bitrate codecs come with a higher complexity, therefore are not always supported by the nodes.

In order to simplify the figures, only three policies where used in this study: P0-Low, where all nodes use the lowest bitrate codec (i.e. G.723.1 *if supported by the node's processor*), P0-High, where all nodes use the highest bitrate codec (i.e. G.711, always supported due to its low processing needs) and the P5-Random, the multi-adaptive policy that chooses the next call to change codec in a random manner. This selection was done in order to compare the results of one of the best performing policies (P5-Random - the performance of the rest of the multi-adaptive policies P3 and P4 is almost identical to this as already seen) against the fixed, non-adaptive policies that can serve as a representation of the two extreme cases. The objective here is not to compare again

all policies against each other but to see how the codec complexity and the processor capacity can affect on their performance.

The same scenario described in section 5.5 was also used here. In addition, the computational cost of the codecs has been considered. Each node has been randomly assigned a maximum processing power ζ_m with a uniform distribution between 1 and 40. Choosing an upper bound of 40 means that a significant portion of the terminals can not implement all codecs due to lack of processing resources, a situation still common nowadays. However, the effect of using a different upper bound will be also studied in the following section.

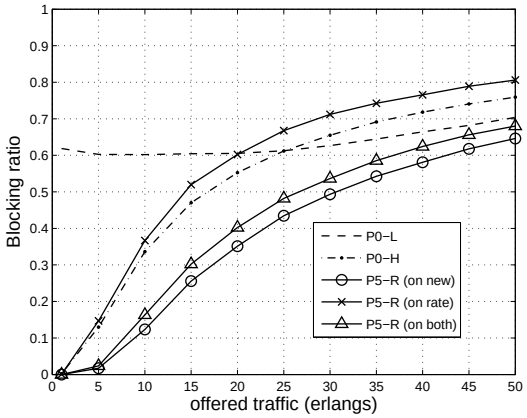


Figure 5.9: Blocking ratio

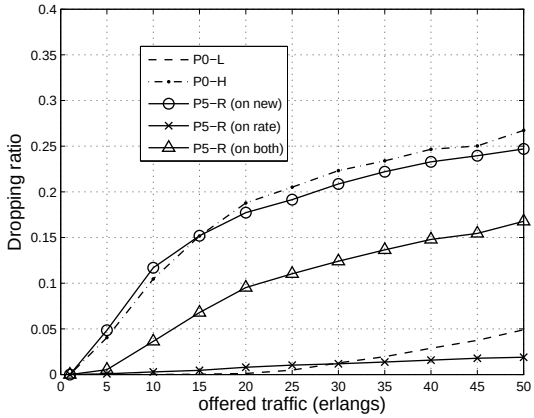


Figure 5.10: Dropping ratio

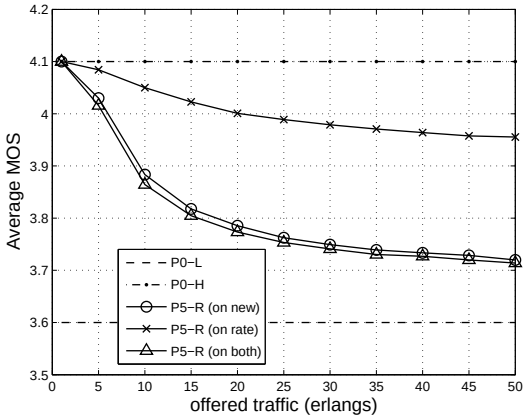


Figure 5.11: Average MOS

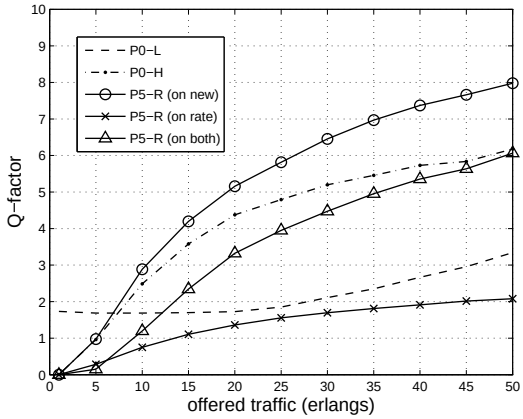


Figure 5.12: $\overline{Q}$-factor

The main differences observed compared to the previous results can be summarized to the following:

- The blocking probability of policy P0-L increases significantly. Stations with low processing power are blocked when trying to start a new call, since the codec that they are asked to use (G.723.1 in our experiments, in the general case the lowest

bitrate codec) has the highest complexity and thus it is not supported by their processor.

- The blocking and mainly the dropping probability of multi-adaptive policies under all scenarios are also slightly increased, since now there are less codec options for the nodes participating in the adaptation process. However this variation is small compared to the one noticed with the fixed, low-codec policy P0-L.

- due to the previous facts, policy P0-L is not the optimal solution now, for none of the traffic load cases, as depicted in Figure 5.12. Using multi-adaptive policies under rate changes, remains the solution that gives the best overall results and trade-off between all measured factors (lowest $\overline{Q}$).

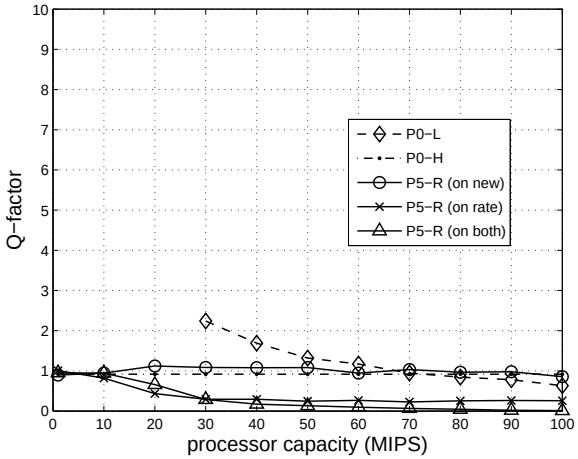
Nevertheless, the same general conclusion can be obtained with regard to the previous experiments where processing power was not considered. Again, for very low traffic load ($A \leq 5$ Erlangs) applying codec selection on both new and changed calls using multi-adaptive policies is the case that provides the best (lowest) $\overline{Q}$ value. See that the values of the $\overline{Q}$ for the two cases (*P5-R on rate* and *P5-R on both*) are very similar and equally low. Nevertheless, this changes completely as the cell load increases. Then, applying codec selection only on rate changes is the optimal solution, taking into account all parameters.

5.7.2 Q-Factor vs node processor capacity

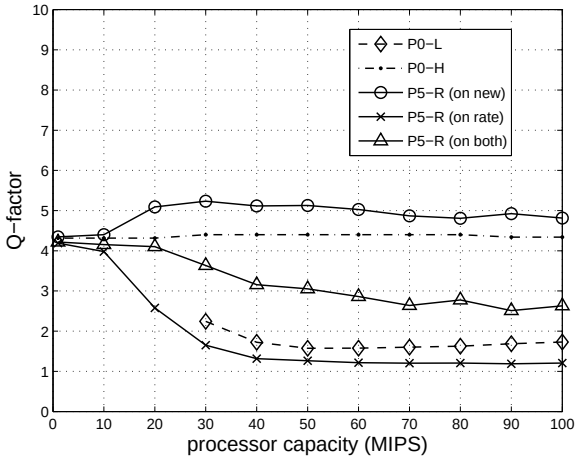
The experiments above were performed with a top processing power of 40MIPS. It is expected however that technological evolution will ameliorate this situation in the next years even for the simplest terminals. In this section we will examine the effect that different processor capacities have on the Q-factor and on the policies applied of our combined CAC with Codec Adaptation solution. We will see that the effect of increasing the node's processor capacity is simply to increase the effectiveness of the codec adaptation process.

We study three different cases of traffic load: low (5 Erlangs), medium (20 Erlangs) and high (40 Erlangs). Then in each case we compare the Q-Factor performance against the upper limit of processor's capacity, ζ_{max} varying from 1 and 100 MIPS. Each node is again assigned randomly a maximum processor capacity following a uniform distribution between 1 and ζ_{max}.

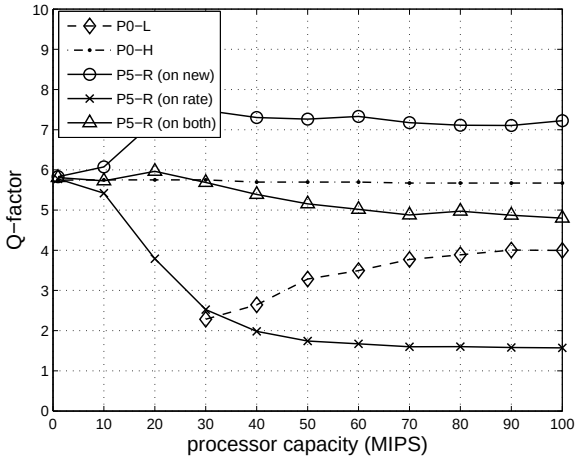
The results obtained are similar to before. Policy P0-L, for low processor capacity ($\zeta_{max} \leq 25$), has the effect of blocking *all* new call requests, since these are demanded to use a codec they cannot support (in our experiments, the G.723.1 codec with complexity $\zeta = 25$) and so they do not even start the call. Thus, we observe in Figures 5.13.a, 5.13.b and 5.13.c that for low processor's capacity values P0-L is not applicable and can be considered the worst choice in this case.



(a) Traffic load of 5 Erlangs



(b) Traffic load of 20 Erlangs



(c) Traffic load of 40 Erlangs

Figure 5.13: Q-Factor vs node processor capacity for different traffic loads

We can also observe again that for low traffic (Figure 5.13.a) we have more flexibility in applying codec adaptation either on rate changes only or both for new calls and on rate changes. Both options give an excellent (very low, almost zero) $\overline{Q}$ value, especially when the nodes' processor capability increase to higher than the 40MIPS that we have used in the previous section.

As traffic load increases, the difference between the scenarios under which codec adaptation is applied becomes more obvious. Thus, for high traffic load (Figure 5.13.c) applying codec adaptation only on rate changes is the obvious recommended strategy. As we have already explained the network is more congested and we have less flexibility for codec changes. Especially in this case of higher traffic it is more evident that increasing the node's processor capacity leads to a better and more efficient utilization of the codec adaptation mechanism. See in Figure 5.13.c how $\overline{Q}$ starts with a high value of around 6 but becomes less that 2 for the P5-R policy applied on rate changes only, even in this high load situation.

Thus, although the efficiency in general of the proposed codec adaptation solution compared to the standard case does not change, codec complexity is an interesting factor to consider. Applying codec adaptation as a reaction to rate changes remains the optimal solution, under any traffic load and limitation of processor capacity.

5.8 Conclusions

We have studied here the possibility of combining the voice codec adaptation mechanism with Call Admission Control and have concluded that a CAC/VCA cooperation would lead to an efficient resource usage, especially when used under low traffic load conditions. Using voice codec adaptation not only to mitigate the multi-rate effects but to increase the number of accepted calls in the cell, we can obtain a capacity increase.

A number of decision policies where also presented that can indicate when and how to use codec adaptation depending on the codec changes that we want to have. They can be categorized in three subgroups, non-adaptive, simple adaptive and multi-adaptive policies. From the comparison results between them we have obtained that using multi-adaptive policies in both new and rate changes when traffic is low or in rate changes for high traffic are the combinations that give the best performance.

In order to combine the three different metrics that where taken into account in this evaluation (blocking rate, dropping rate and MOS) the Q-Factor was presented. By re-uniting all three important metrics in one new GoS parameter, $\overline{Q}$, a simple yet effective tool for optimizing the adaptation process is provided.

Finally, the importance of the cost complexity in the performance of the VCA algorithm was analyzed and tested for a number of policies. We see that the limited processor capacity of the nodes can decrease somehow the effectiveness of the codec

adaptation, especially in high traffic load conditions. However, using codec adaptation still outperforms the standard non-adaptive case and with the expected terminal evolution, the increase in processor capacity will only increase the efficiency of our proposal.

CHAPTER 6

From thesis to praxis : QoS Extensions for an 802.11 Access Point Architecture optimized for VoIP

6.1 Introduction

In the previous chapters we have examined a Codec Adaptation solution, which can efficiently solve the effects of multi-rate 802.11 networks by using first, cross-layer RTCP and MAC feedback to measure the voice QoS degradation and then, the SIP re-Invite process to renegotiate a new codec for some of the active calls. We have examined the two variations of this algorithm (centralized and distributed). We focus mostly in the centralized implementation of this, installed in an AP, since it is more efficient providing a cell-wide codec optimization and also, it can easier cooperate with a CAC mechanism to provide the integrated CAC/VCA solution presented in the previous chapter.

Although the basic algorithm structure has been explained in detail, various implementation issues, like the SIP communication between the Access Point and the nodes, have been left open. In this chapter we complete the theoretical study with the elements missing for an actual implementation design in a real AP. We review a number of issues arising at the moment of integrating all elements and we see how these elements, from MAC and RTCP monitor to the SIP codec re-negotiation and the policy-based CAC can be efficiently combined and work together.

Integrating all the different modules that we have discussed until now, an architecture similar to the one shown in Figure 6.1 would be needed. The figure depicts the new module proposed, the Wireless QoS Multi-rate Module, which will be discussed in detail in section 6.5. Introducing it shortly here, this new module includes the basic algorithm pieces presented in previous chapters:

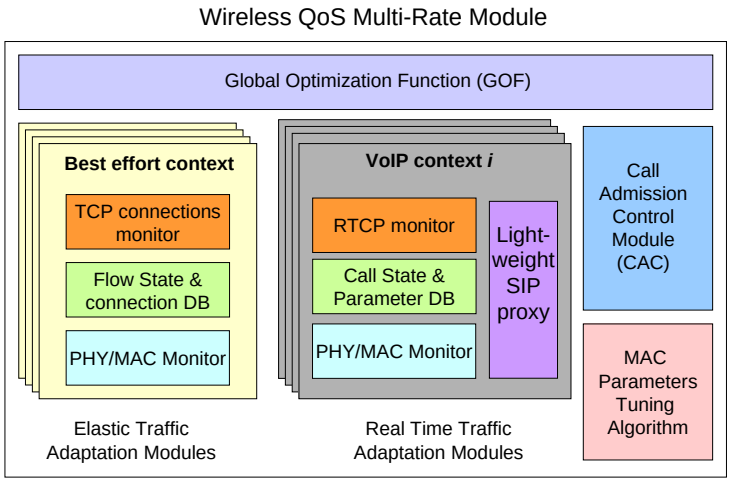


Figure 6.1: Wireless QoS Multi-rate Module architecture

- the codec adaptation algorithm monitor modules, both MAC and RTCP, for each of the active calls in the cell (each call is represented by its own *context*, equivalent to a thread or process of the algorithm)
- a policy-based CAC module, responsible for distributing the available cell resources among the incoming new calls
- a SIP proxy responsible for handling the codec renegotiation based on SIP (discussed in detail later in this chapter)
- a Global Optimization Function (GOF) that performs a centralized control of the process, configured according to the chosen decision policy (also discussed in next sections)
- an elastic or best-effort adaptation module, responsible for providing a solution on the rate changes occurring on TCP nodes, like the tuning of EDCA parameters seen in chapter 4 (this sub-module is out of the scope of this thesis)
- a Call State & Parameter DB, where all the relevant information on the parameters of active calls, the actual nodes' rate etc, necessary for the correct functioning of the module, will be saved

Since most of the functionalities of these elements have been already discussed in detail in previous chapters, in this chapter we will focus on the three important issues left to determine and which can be summarized in the following:

1. The need for a SIP Proxy. As the last and most important part of the codec adaptation mechanism, the actual codec re-negotiation, is based on a SIP re-

Invite message, it is only natural that a SIP proxy will be needed to initiate and participate in this SIP conversation with the nodes. In section 6.3 we examine the possible implementations of it, joint with the Access Point or separated from it, as also the reason for trying to implement this proxy in a manner as transparent as possible for the nodes.

2. Triggering a SIP re-Invite from the Proxy to the node. Although SIP protocol offers many possibilities, it does not however contemplate how a SIP conversation can be initiated from a Proxy to a node, if the latter is not already participating in a SIP session with the former (there is no SIP session established between them in advance). The transparent character of the SIP proxy used in our scenario increases the difficulty of this hidden proxy initiated conversation. In order to address this issue a header extension to the SIP protocol is proposed, the Wireless_MRQE SIP header, presented in section 6.4.

3. Joining all the necessary modules in a single Access Point with all the modifications needed. We name this new module that includes all the logic and control of the codec adaptation process a Wireless Multi-rate Module (WiMM). The details of this module are described in section 6.5

6.2 Background overview

The general traits of a typical AP architecture can be found in [46] and [45]. Mainly the software and hardware architecture is shown there, although some aspects of the functional architecture are also explored. Furthermore, diverse AP manufacturers provide similar information (although more sparsely) about their own products, like the TeamF1 [76]. Many modern AP designs, such as Linksys [47] and SMC [56], provide some VoIP support for WLAN scenarios. To the best of our knowledge however, no architectural optimization for the QoS support of VoIP or heterogeneous VoIP-TCP traffic over WLANs, beyond basic EDCA and/or CAC mechanisms, has been attempted so far, neither in academia, nor in commercial products. The main contribution of this chapter lies precisely in filling that gap, by providing a feasible AP architecture combining CAC and dynamic codec selection for the overall cell resource optimization in a capacity variable channel and considering heterogeneous traffic. The corresponding signaling mechanisms, based on extensions of the SIP protocol, to support those optimizations are also presented.

However, the most demanding problem remains the interchanging of SIP messages between a node and a SIP Proxy, so that the latter can instruct the former to issue a re-Invite message. The problem lies mainly on the fact that this conversation must be initiated by the Proxy, with the additional drawback that the node does not even know the existence of the Proxy (transparent/hidden proxy) as we will see in section 6.3.

The importance of the SIP protocol has been examined in chapter 2. SIP is the mechanism of choice for multimedia session signaling in the Internet. However, in its original intent to remain simple and centered exclusively on call control, it did not adequately cover sophisticated multimedia service management beyond pure one-to-one direct voice communication. Its successive modifications have increased its applicability, even though a strong commitment to remaining as simple as possible continues. Of particular relevance to this study is the fact that no method existed so that a SIP Proxy could start a dialog with a SIP terminal if a session was not already established between them. Hence, transporting policy information from the Proxy to the terminal was problematic.

The issue of interchanging information on session policies between a SIP terminal and a policy decision point has been addressed by Camarillo et al. [16] in the IP Multimedia Subsystem (IMS) [75] context. Their solution is based on the use of the SUBSCRIBE/NOTIFY SIP method to create the necessary dialog between both entities. Still, in their case, it is the terminal who must initiate the dialog, which presupposes a) prior knowledge of the Proxy by the terminal and b) that the terminal registers to the Proxy. As we will discuss later, this is not applicable to our scenario, in which the Proxy tries to remain transparent as long as possible. Nevertheless, the solution proposed here borrows heavily from this one, though with crucial modifications for the WLAN, non-IMS scenario.

6.3 The SIP Proxy

6.3.1 Location of the Proxy

In a WLAN scenario as the one considered in this thesis, the necessary QoS information is distributed between the terminals, the AP and the SIP Proxies involved in the calls (see also Table 6.1). For the correct execution of the Codec Adaptation algorithm, the feedback needed to perform a VoIP-friendly cell-wide optimization of network resources can be summed to the following:

1. A notification from the MAC layer, informing of a physical channel rate change (for every active communication, VoIP or other): This information can be obtained directly from the protocol stack at every node, or centrally at the AP, since it acts as border element of all flows / sessions going towards the rest of the network[1].

2. The actual call parameters (for every active call), meaning the used codec and

[1]This would not apply, if direct intra-cell calls, without AP mediation, would be allowed in ad-hoc manner. However, intra-cell calls are rare since the two end-users would be very close to each other, and hence this work centers on calls that originate or terminate beyond the cell boundary. Notice that it is not necessary that they go to the wired network for the AP to notice all MAC layer changes, only that the AP is involved in all communications.

QoS Information	Access Point(AP)	802.11 station (STA)	SIP Proxy
RTCP QoS feedback	X	X	X
SDP codec capabilities	-	X	X
SDP actual call parameters	-	X	X
MAC rate changes	X	X	-
SIP re-Invite	-	X	X
CAC policies	X	-	-

Table 6.1: Location of QoS information

packetization interval: This information is contained in the SDP messages transported in the INVITE, OK and ACK SIP messages during call negotiation and setup. An element with capabilities of SIP filtering, like a SIP Proxy, could obtain this information.

3. The VoIP codec capabilities of every node: Basically, a list of the codecs and media types supported. Only the nodes themselves have access to this information, which is rarely made public during a communication, unless an OPTIONS message explicitly requests it. A SIP proxy can use this method and obtain the capabilities list of each node transported in the SDP part of the SIP message.

4. Updated and periodic information on the instantaneous call quality (for every active call): As stated before, the RTCP reports transport this information between the communicating parties. Apart from the parties involved in the session, the standard contemplates the possibility of a third-party control entity to intercept these messages. We call this entity a RTCP monitor and can be easily included either on the Access Point or in a SIP Proxy.

5. The Call Admission Control policy used: this is normally implemented in the Access Point, although the option of a central network element for a multi-AP optimization implementing these policies can be also considered.

Since most of the information needed is divided between the AP and the SIP proxy, it is only logical to include the SIP proxy functionality to the Access Point and implement in this the joint CAC/VCA mechanism presented in previous chapters. However, the option of placing the control entity separated from the AP will be also discussed later in the future guidelines. This second option would imply that the SIP proxy, together with the RTCP monitor and the CAC/VCA Policies module would be implemented in a central control point, outside the wireless cell and which would have to communicate with a lightweight version of the Access Point, in order to exchange the necessary information between the two units. Even though this solution would permit an inter-

cell control and optimization of resources, it would be accompanied with an increased signaling traffic.

For this reason, in this study the placement of the control unit in the Access Point is chosen. Combining everything, a new Wireless QoS Multi-Rate Module (WiMM) is presented here, which includes all the necessary functions to perform the codec adaptation mechanism together with a CAC algorithm and a SIP Proxy implementation. However, before entering to the architectural details, there are two problems left to be solved: how to maintain the SIP Proxy transparent to the node and how to still be able to trigger a SIP re-INVITE from this hidden proxy.

6.3.2 The need for transparency

It is pretty straightforward to recognize that, since information transported in SIP messages is involved in the decision process, a SIP proxy (or another entity with similar functionality) should participate in it. If the SIP proxy lies in the signaling path of the call, because it is for example the outbound proxy for the user, or because it has placed a *Record-Route* header in the initial INVITE, it is then a simple process to gather the SDP and RTCP information, since the proxy will receive all signaling messages (and hence, SDP and RTCP info) coming from or going to the nodes.

However, not all VoIP operators would agree in adding an external proxy to the VoIP session. In the scenario described here, the 802.11 network acts as simple connectivity provider, and should remain as transparent as possible to the STAs at the service level. Obligating every wireless station to register with a specific SIP proxy controlled by a WLAN provider greatly complicates the configuration of the stations and could prove to be simply unfeasible, since most VoIP operators mandate the registration with a proprietary proxy of their own. If an alternative option is chosen, namely to intercept the SIP messages and add a *Record-Route* to it, no such restrictions apply in many cases, although certain commercial providers would still not accept it. In any case, it certainly breaks the transparency principle strived for here.

A less intrusive alternative consists in intercepting all SIP and RTCP messages in some central point of the architecture, record the information transported therein and relay the messages *unaltered* to their destinations[2]. In this way, the STAs do not need to alter their default behavior and/or be made aware of the proxy, and hence will remain compliant with all VoIP operators' requirements. This approach of the "hidden" proxy is the approach chosen here. Since in our implementation we consider a SIP proxy co-located with the Access Point, we can assume that it lies in the path of all the SIP

[2]The use of encryption in the signaling messages would certainly highly complicate this behavior. Nevertheless, if this would be the case, the communication between clients of different VoIP providers, where more than one SIP proxy is involved, would simply be impossible. The history of the PSTN evolution has shown that this rarely is a tenable alternative. Notice that encryption of the data packets does not affect the mechanism in any way and is fully acceptable

messages so intercepting them and saving the information that they include is practically feasible.

This alternative can make the SIP proxy used in the WiMM module transparent to the end users. However, the problem remains of how this hidden proxy can mandate a SIP re-Invite (i.e. start a SIP conversation) with the STAs without them registering to it. Based on the solution proposed by Camarillo et al. in [16] and the OPTIONS/SUBSCRIBE/NOTIFY methods, we propose a new Wireless Multi-Rate QoS header, extension to the SIP protocol, described next. Using this header, a Proxy can announce its existence to the nodes and offer the new QoS service, allowing the nodes to subscribe to this service voluntarily, while maintaining its transparency.

6.4 The Wireless_MRQE SIP header

The proposal made here is to use the OPTIONS SIP message with a new header, the *Wireless_MRQE* (Wireless Multi-Rate QoS Extensions) Header, for the purpose of a proxy-initiated communication with the SIP client. The OPTIONS message is one of the few that can be sent prior to establishing a communication, and hence can also be sent from the proxy (acting as a SIP User Agent Client - UAC) to the STA, even if the STA did not know in advance the existence of this proxy. As described in detail below, the OPTIONS message will serve a double role: First, it will serve to request the full set of media capabilities from the STA, which is a critical information for the optimization process. But second, it will also serve to trigger a SUBSCRIBE from the STA to the new *Event:MRQE* at the hidden proxy (see SIP Message "SUBSCRIBE").

SIP Message OPTIONS
OPTIONS sip:STA1@providerA.com
From:Hidden_AP@wifinet.com
To:STA1@providerA.com
Contact:Hidden_AP@132.119.28.3
Wireless_MRQE:Enable

SIP Message OK
200 OK sip:STA1@providerA.com
From:Hidden_AP@wifinet.com
To:STA1@providerA.com
Contact:STA1@132.119.28.11
Wireless_MRQE:Enabled

SIP Message SUBSCRIBE

SUBSCRIBE sip:Hidden_AP@wifinet.com
From:STA1@providerA.com
To:Hidden_AP@wifinet.com
Event:MRQE
Accept:Application/pidf+xml
Expires:600 Content-Type:Application/pidf+xml
Content-Length:XXX

$PIDFdoc$

SIP Message NOTIFY
NOTIFY sip:STA1@providerA.com
From:Hidden_AP@wifinet.com
To:STA1@providerA.com
Event:MRQE
Accept:Application/pidf+xml
Expires:600 Content-Type:Application/pidf+xml
Content-Length:XXX

$< XML : MRQE >$
$< COMMAND >$
$reInvite$
$< /COMMAND >$
$< SDP >$
$ip = 132.119.28.11$
$port = 32000$
$m = audio234$
$c = 4PCMA8000$
$< /SDP >$
$< /XML :: MRQE >$

The proposed *Wireless_MRQE* Header has four possible values: *Enable, Enabled, Disable, Disabled*. The imperatives are used in the requests from proxy to station, while the past participles are used in the responses from the station to the proxy, and serve to acknowledge that the station is able and willing to participate in the VoIP optimization process. Upon receiving the OPTIONS message with this header, the station shall respond with the corresponding OK including the *Wireless_MRQE:Enabled* header value, and immediately thereafter the station shall send a SUBSCRIBE message to the proxy (at the address stated in the *Contact* header field) for the *Event:MRQE*,

and including an empty PIDF document. From this point on, the proxy is free to contact the STA at any instant via NOTIFY messages (even in the absence of further SUBSCRIBEs), for the duration of the subscription. Such NOTIFYs also encapsulate PIDF documents of discretionary length and content, according to [69], and which will be used in this framework for the interchange of commands (like triggering a re-INVITE) and call information (e.g., a new SDP body with different codec) between the hidden proxy and the STAs.

The specific commands to be conveyed are:

- Trigger a re-INVITE: In the case that a call needs to update its call parameters (i.e. change codec) in order to adapt to a change in the cell QoS conditions, the AP will NOTIFY the STA, including an XML body (see SIP Message "NOTIFY") with the command *re-Invite* and an SDP body with the chosen parameters.

- Trigger fast RTCP: When a rate change is detected, or the regular RTCP packets indicate a degradation in the call quality, the algorithm foresees the sending of extra RTCP packets at a higher rate than usual, in order to quickly obtain additional feedback on the evolution of the QoS situation (see chapter 4). However, this mechanism, like the rest of the algorithm, is not implemented by default in every node. Hence, whenever such behavior must be triggered, the AP sends a NOTIFY with the command *fastRTCP*, followed by the frequency at which these fast RTCP reports must be sent, R_{rtcp}, as well as how many such "fast" RTCP packets should be sent, N_{rtcp}. In this way, the AP can adapt to the duration and severity of the situation, as well as to the load in the cell, by modifying both parameters.

- Trigger a BYE: If during the adaptation process one of the corresponding call partners cannot accept the new codec proposal, another call could be tried to be adapted, according to the decision policy chosen. However, there may be cases where the AP, in order to avoid an unacceptable QoS degradation, must drop the problematic call. In this case it can send a NOTIFY with the command *BYE*, triggering the tear down of the call. This implies an unwanted call termination for the user, but ensures the maintenance of the call quality for the rest of the users.

Note that, the initial SUBSCRIBE and NOTIFY messages, since they are sent without any need to update the call state (for a call has not yet been established), contain only and empty body.

Provided that the STA understands the new *Wireless_MRQE* header and reacts with the corresponding SUBSCRIBE, the bidirectional communication between STA and AP is solved, even in the absence of active calls between the two. This implies, however,

that the proposed QoS mechanism is *not* fully transparent for the participating STAs, which need to implement the MRQE extensions (header and event). This, nevertheless, is a very modest requirement, since it does not necessitate any new functionality from the STAs (the whole optimization computation takes place at the central instance), but only a new case of an existing procedure (i.e. new SIP header) and the willingness to trigger existing behaviors upon request from the proxy.

Not all STAs, quite naturally, will implement the MRQE extensions from day one. This does not represent a problem for the proposed solution: Basically, non MRQE-aware stations will ignore the new unknown header (as defined in the SIP standard), and the optimization will proceed with only the collaborating STAs. This would be a similar situation like the one reviewed in chapter 5, occurring when due to the station's processing power limitations, a limited set of codecs are supported by it and thus it cannot participate in the codec adaptation. The results are still very positive, provided that a majority of stations participate in the process. Hence, this solution presents a very smooth migration path, fully compatible with existing SIP implementations, and which provides incrementally better results, as more and more STAs implement the proposed extensions.

A further issue is how to find out which STAs from the ones using the cell for connectivity purposes (e.g. to surf the Internet) will also use the VoIP service, since only those STAs will participate in the optimization process. According to this centralized proposal, since the central instance will intercept all SIP messages coming from, or going to the cell, the emission of a REGISTER message from a STA, or the sending or receiving of an INVITE (in case that the STA was already registered with its operator's proxy before joining the cell), is proof that the VoIP service is in use. Hence, as soon as those messages have been detected, the corresponding OPTIONS will be sent, requesting the STA to join the MRQE procedure[3].

An example of a typical call flow including these new headers and enhancements is presented in section 6.6 and in Figure 6.6. However, in order to better understand this example, we will first have a look at the complete Wireless QoS Multi-rate Module that integrates all the above mentioned elements.

6.5 Architecture of an AP optimized for VoIP over wireless

Having explained all the SIP logic and message interchange between the station and the AP with the hidden proxy, it is time to analyze how al the modules are joined together in the AP implementing the Wireless QoS Multi-Rate Module (WiMM) and in

[3]This procedure can be simplified in the newer 802.11e standard where the nodes are classified in different Access Categories depending on their traffic type. However this more generic approach is chosen here.

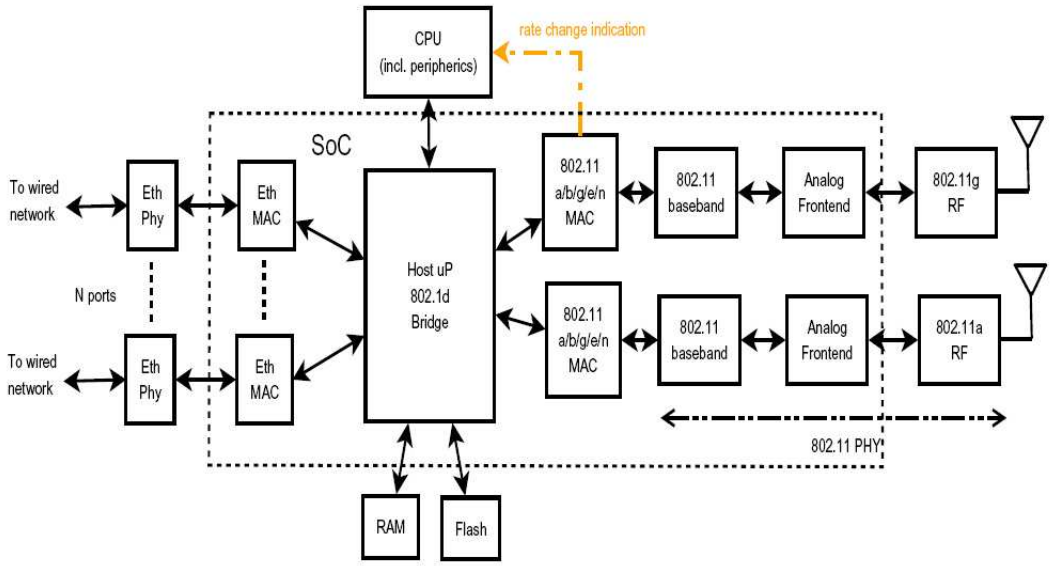


Figure 6.2: AP hardware block architecture

which manner this new module is inserted and inter-operates with the rest of the AP's modules.

Wi-Fi Access Points are, in essence, quite simple network nodes, both in their hardware as well as in their software architecture, although the latter is certainly more feature-rich than the former. A typical AP, used in many multi-AP hotspots, is essentially a layer 2 bridge, including both wireless and wired ports (see Figure 6.2). Typically, the wired ports (between 1 and 4) are (Fast-) Ethernet ones. The wireless ports are currently an 802.11g port, for the communication with the STAs in the cell, and an 802.11a port, to serve as wireless uplink towards the operator's backbone or to build a wireless backbone in mesh networks. Alternatively, one of the wired ports can serve as uplink. The wired and wireless ports are interconnected through an 802.1d bridge, which performs the necessary frame format adaptation between the protocols, as well as switching frames between ports. Such a bridge may use external memory as additional storage. This whole set of blocks (Ethernet MAC, 802.1d bridge, 802.11 multi-variant MAC and most of the 802.11 PHY) can nowadays be integrated in a single System-on-a-Chip (SoC), which results in very compact and cheap APs. Only the 802.11 RF blocks and the Ethernet PHY, as well as the peripheral memory, are usually located in separate and highly specific chips. The antennae, due to their high form factor, are usually set apart from the main board.

From the perspective of the VoIP optimization mechanism proposed here, it should be noticed that no alterations whatsoever at the hardware level are necessary to support it except one: It is necessary that the WLAN MAC layer informs the software stack of

any rate changes affecting it. This is actually not a real novelty, for most 802.11 drivers already have this kind of information available. Hence, basically no alteration of the hardware architecture is necessary to support the proposed mechanism, and it could then be implemented on top of any commercially available AP.

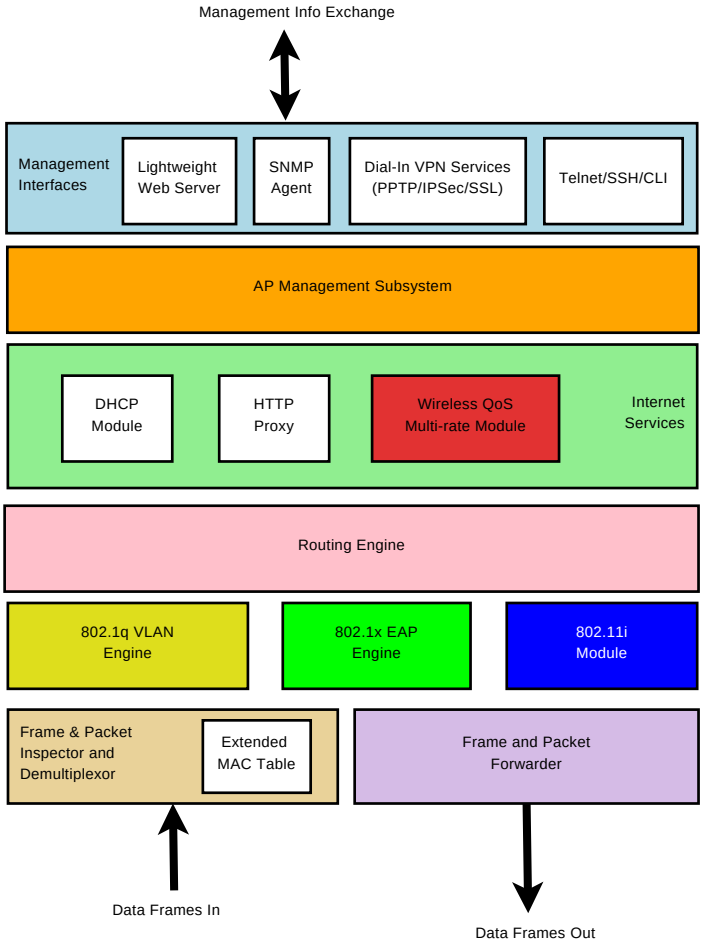


Figure 6.3: AP functional software architecture

Figure 6.3 presents a somehow idealized vision of the functional software architecture in a high-end AP. The functions of the different modules can vary from layer 2 functions, like MAC frame handling and MAC addresses learning, to layer 3 functions, like IP routing, or more advanced layer services that can help the process of the STAs association with the AP (e.g. DHCP server). Of particular interest here are two blocks: The Frame & Packet Inspector and Demultiplexor (Inspector in short) and the only truly new block in the software architecture, the Wireless QoS Multi-rate Module (WiMM). Their functionality will be shortly reviewed next.

6.5.1 Wireless QoS Multi-rate Module

The core novelty of the AP software architecture is the WiMM. Its schematic description was already presented in Figure 6.1. As can be drawn from the figure, it consists of a software process, called a *Context*, for every active call in the cell. Nowadays, for the 802.11g standard that amounts to some 20 contexts at most, since that is the maximum capacity of a VoIP cell. The overall function of a Context is to implement the VoIP optimization mechanism for a given call. The global cell-wide optimization policy, however, is implemented in the Global Optimization Function module (GOF). The GOF coordinates all Contexts, setting the values of the parameters used in the codec adaptation algorithm as mentioned in chapter 4 ($\alpha, \beta, \gamma, R_{rtcp}, N_{rtcp}$, etc) and the QoS thresholds for every individual call, so as to optimize both the number and the quality of simultaneous calls in the cell in the occurrence of a rate change for any of them. The GOF also implements the chosen decision policy, like the ones described in chapter 5, although a different, more sophisticated one can be also used. Furthermore, the GOF does not only coordinate the Contexts, but also the CAC, by providing it with the updated cell resource usage after every rate change. Conversely, the CAC informs the GOF of any new call setup request and its corresponding profile. The GOF then calculates which actions can be taken to leave room for the call, and triggers them in the Contexts.

The different WiMM modules are:

- *Call State & Parameter DB*: Its mission is to keep an updated record of the call parameters in use (codecs, media types, ports, etc.), the media capabilities of the STAs participating in the call (obtained from the OPTIONS message as explained before), as well as the parameters of the optimization algorithm (mainly $\alpha, \beta, \gamma, R_{rtcp}, N_{rtcp}$ and the different timers and QoS thresholds used). It also keeps track of the codec transition matrix pre-processed by the optimization algorithm, taking into account the media capabilities of the STA. Simplifying, this matrix is a list of the codecs supported by each node and ordered according to their bitrate.

- *MAC Monitor*: In charge of watching over any physical rate changes that directly affect the corresponding call. Its basic function was explained in detail in chapter 4. In summary, if the physical channel changed towards a lower physical channel rate, the MAC Monitor requests from the SIP Proxy a decrease of β steps in the codec used, so that a more bandwidth efficient codec is used. To that end, the SIP Proxy will consult the local Call State & Parameter DB, check the codec transition matrix for that call and build a new SDP message to be sent in a NOTIFY. The same mechanism is used for a change to a higher codec due to a rate increase and in order not to leave unnecessarily bandwidth unused, although this option was not studied extensively in this thesis. The goal is to always achieve the maximum

number of active calls with the maximum possible quality.

- *RTCP Monitor*: Responsible for interpreting the RTCP packets sent and received by the corresponding mobile nodes, and updating the QoS estimation, both instantaneous and long-term, of the call. Following the adaptation algorithm described previously (chapter 4), if the QoS of the call breaks certain thresholds (controlled by the GOF and stored at the local database), it triggers the renegotiation of the call codec. To that end, a signal is sent to the SIP Proxy, responsible for the generation of all SIP messages at the AP.

- *SIP Proxy*: Represents the transparent proxy referred to in previous sections. It performs basically two main functions, as depicted in Figure 6.4. First, it receives all the SIP messages interchanged by the corresponding STAs, and updates the local database with the active call state and the node's capabilities. Second, it reacts to the requests made by the MAC and RTCP Monitors by building the adequate SDP bodies and sending the corresponding NOTIFYs that will trigger the renegotiation of call parameters or the fastRTCP transmission. It represents an extremely reduced version of a full SIP Proxy, since most of the standard proxy functionalities are not needed. This also helps to keep the overall resource usage at the AP moderate.

- *CAC*: In charge of deciding if a new call setup request can be accepted or not, depending on the instantaneous resource usage at the cell. It can also accompany a call setup denial with a suggestion of a new call profile (i.e. a new codec), which would fit the current cell state. Such a suggestion is provided by the GOF, after performing a codec recalculation and rearrangement of the existing calls, if deemed adequate.

- *Global Optimization Function*: As explained, it coordinates all other modules according to an overall optimization policy. This being one of the fundamental success factors for the overall strategy, new algorithms are currently being tested in the framework of the same overall mechanism. Although here we present just the first notions of this function, the main functionalities of GOF is to implement the decision policy, set the thresholds used in the codec adaptation procedure and exchange information with the other modules so as to trigger the adequate actions in each case.

The WiMM is the only new module in a standard AP software architecture. Hence, it could be provided as an add-on to existing products, or included in future software releases, without having to severely impact the existing development, which presents a very convenient migration path for the adoption of the presented mechanism.

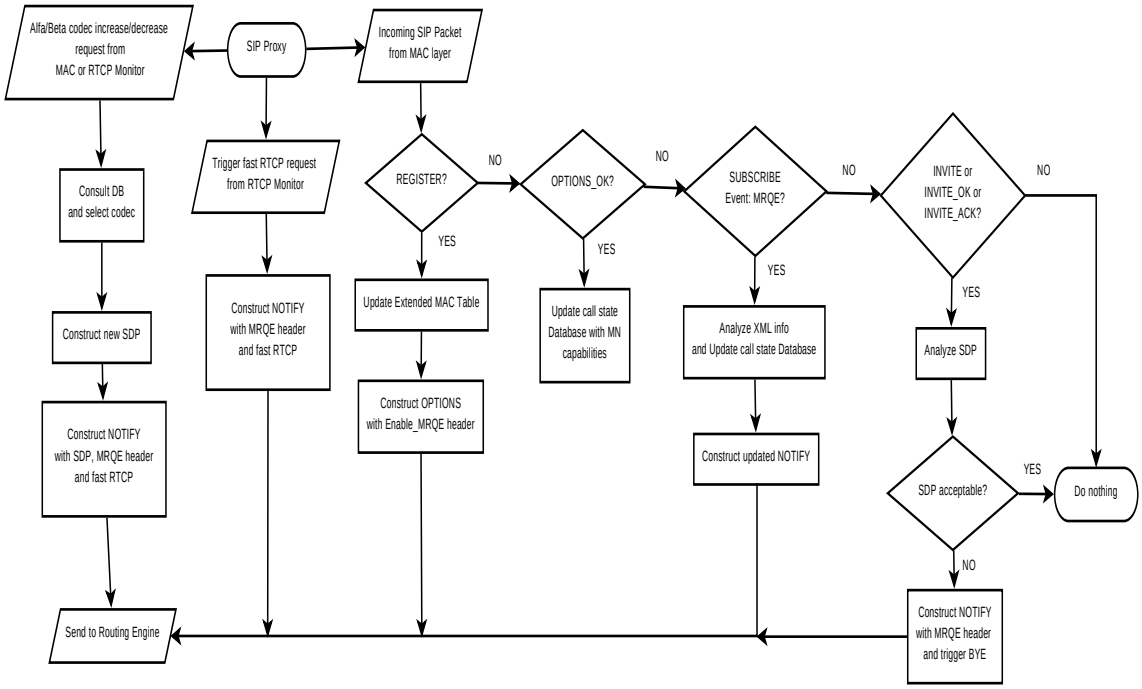


Figure 6.4: Context processing of SIP messages at the hidden SIP Proxy

6.5.2 Frame & Packet Inspector and Demultiplexor

The frame filtering functionality, needed for the filtering of SIP and RTCP messages in our study, is already present in most APs for MAC address learning purposes. Additionally it is necessary for detecting the initial broadcast join and DHCP requests from first time STAs. This pre-existing frame filtering functionality must be extended with a packet filtering functionality for the SIP/RTCP packets, which increases the computational burden of the AP. However, RTCP and SIP packets can be easily identified by the protocol field in the IP header, which allows for a very simple identification procedure.

Regular APs also store all MAC and IP addresses of the STAs that have joined the cell, in order to perform the MAC address learning process and avoid assigning duplicated IP addresses. This functionality must be now expanded to include the Context information for active calls. This practically means to extend the MAC table already available in all APs, so as to include a *Context-ID* field. If upon detecting an incoming SIP message such a field is empty for the corresponding IP address, then the Context had not yet been initialized and a new one must be created. An indication thereof will be passed to the WiMM along with a copy of the SIP message. Otherwise, the message is directly sent to the stored Context at the WiMM. Conversely, when the WiMM generates a new SIP message to be sent to one of the STAs (e.g. a NOTIFY with a *re-Invite* command), it is passed to the Routing Engine and then forwarded in the usual manner.

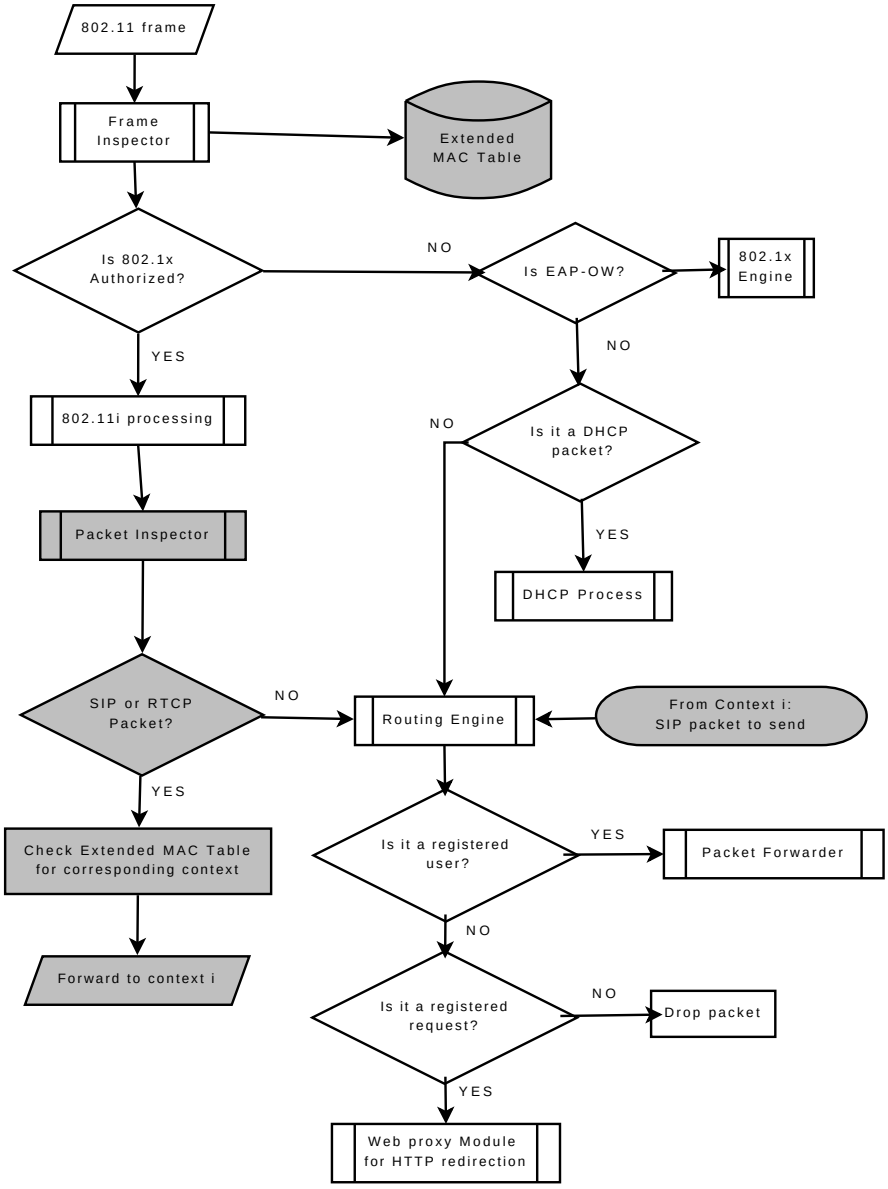


Figure 6.5: Basic MAC frame processing at the AP

A complete description of the flow of an incoming frame through the lower layers of the software stack at the AP, including the initial authentication procedure for a STA joining the AP is presented schematically in Figure 6.5. The new steps necessary for the VoIP optimization procedure have been highlighted.

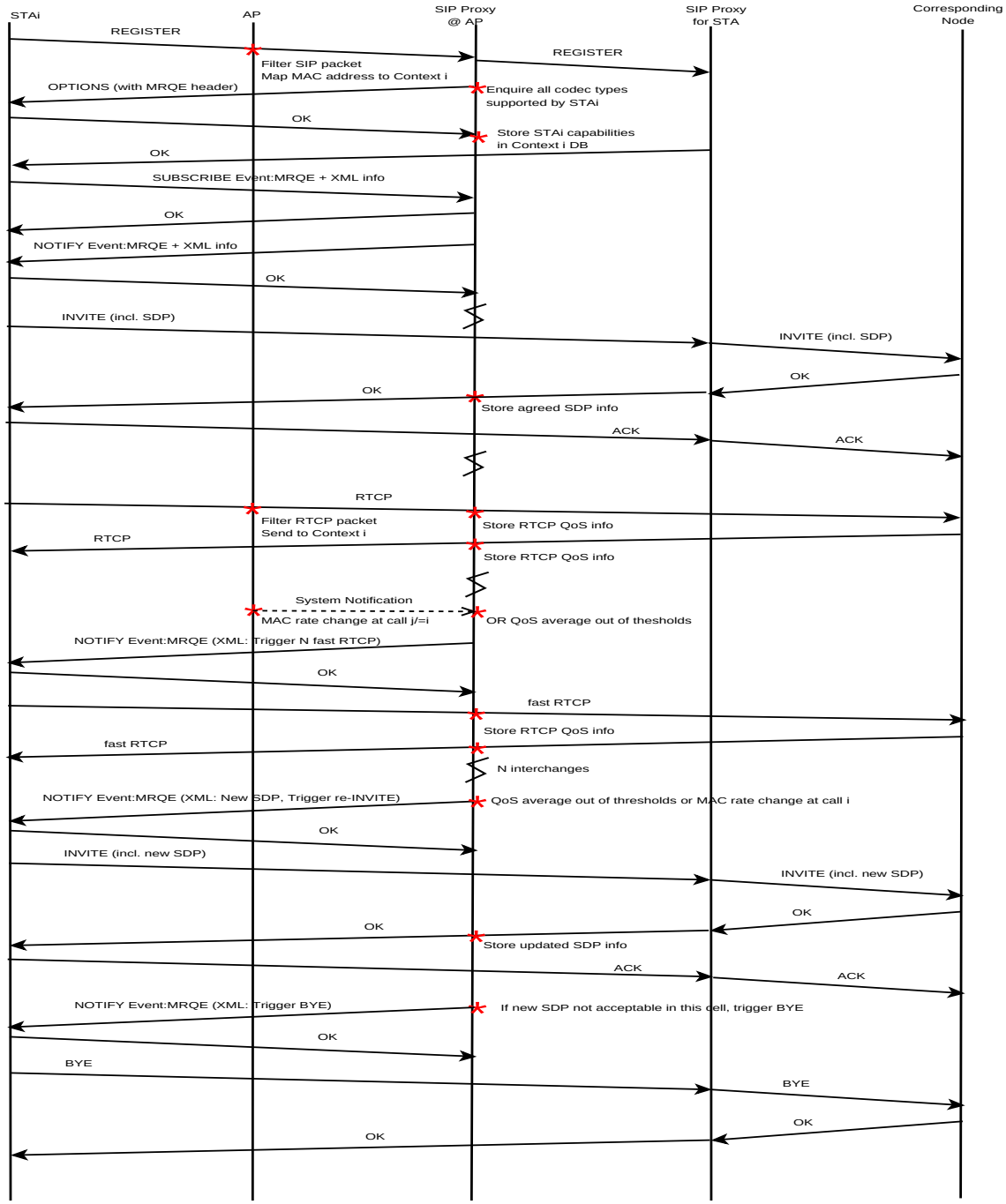


Figure 6.6: Exemplary MRQE optimization call flow.

6.6 MRQE optimization procedure : An example Call Flow

The whole MRQE optimization process and its signaling can be followed in Figure 6.6 and will be briefly reviewed here.

a) Node Registration and initial configuration

It is assumed that the call is started by one of the mobile nodes in the cell, STA_i, and that it had not yet registered with its proprietary SIP Registrar/Proxy (denoted "SIP proxy for STA" in Figure 6.6). Since all messages pass through the AP, and the hidden SIP proxy is co-located with it, upon detecting the SIP REGISTER message, the AP (more precisely, its MAC Frame & Packet Inspector and Demultiplexor function) filters it and checks if the MAC and IP addresses contained therein correspond to an already active VoIP terminal. The matching between MAC and IP addresses and software contexts in the AP architecture is kept at the Extended MAC Table, which is an extension of the traditional MAC address learning table of any AP. If the terminal was already in the table, a timestamp is updated and the REGISTER diverted to the corresponding context process at the WiMM. Otherwise, the terminal is included in the Extended MAC table, and the packet is equally diverted to the WiMM, with an indication that a new context must be created. As explained before, incoming calls and/or outgoing calls post-registering would proceed analogously, except that they would be detected by the incoming or outgoing INVITE instead of REGISTER.

Upon reaching the corresponding context at the WiMM, the SIP proxy triggers the sending of an OPTIONS message. As explained before, the role is twofold: On the one side, in the corresponding answer from the STA, the whole set of media capabilities will be communicated. With that information, the WiMM can build a table of possible codec transitions and their characteristics, which will serve for the optimization process. On the other hand, the OPTIONS contains the *Wireless_MRQE* Header, with the value *Enable*. If the STA, upon receiving this message, is able to participate in the VoIP optimization process, it will respond with a *200 OK* message by including the value *Enabled* in the same Header. The WiMM can then include the STA in its optimization procedure. Otherwise, it will be stored at the $Context_i$ database (labeled "Call State and Parameter DB" at Figure 6.1) as "non compliant", and excluded thereof. It should be reminded that supporting the OPTIONS message is mandatory as by the SIP standard, and hence is included in all commercial implementations. Unknown or unsupported headers, as is also stated in the standard, must be simply ignored, but do not cause any malfunction: The message is simply interpreted by processing the known headers.

After interpreting and responding to the *Wireless_MRQE* Header,the STA proceeds by sending an initial SUBSCRIBE for the *MRQE* Event, thereby opening the possibility for the hidden SIP proxy to communicate with it via subsequent NOTIFYs.

b) Call initiation and rate change

After some period of time, the STA_i will participate in a call. To that end, it will send an INVITE with its chosen set of parameters, as in Figure 6.6, or it will receive an INVITE to which it will respond with its chosen SDP parameters. In any case,

the AP will filter all further REGISTER, INVITE and OPTIONS messages, as well as their responses, in order to find out the agreed SDP parameters, which will be recorded in the $Context_i$ database. These data will serve to inform the optimization algorithm as to which is the current call state, in order to choose an adequate new state, if an adaptation is needed due to changing QoS conditions.

Furthermore, the Frame & Packet Inspector and Demultiplexor module of the AP architecture will also filter and divert all RTCP packets corresponding to STA_i to $Context_i$. The RTCP Monitor at $Context_i$ will snoop those packets to update the instantaneous as well as the evolution of the QoS state of the call, which will also be stored at the local Call State and Parameter Database.

At some point, the AP may notice a change in the physical rate of another active VoIP call, $j \neq i$. In this case, the response would be for the AP (using the SIP Proxy) to NOTIFY STA_i that N_{rtcp} fast RTCP packets must be sent at rate R_{rtcp}, in order to collect additional information about the situation. If in this interval the situation improves, it can be deduced that only a spurious degradation occurred. Then, there is not need to perform any codec adaptation and the call proceeds unaltered.

If, however, after these N_{rtcp} packets the situation continues or deteriorates, then immediate action is needed. Should that be the case, the Global Optimization Function at the WiMM would calculate a new cell-wide optimized constellation of codecs (by using the chosen decision policy), and would trigger the sending, by the impacted Contexts, like $Context_i$ in this case, of a NOTIFY with the chosen SDP parameters, as well as the *reInvite* command. If this new SDP parameter set is accepted in the responding 200 OK from the corresponding node, $Context_i$ will update its local database with the new call state information. If, on the contrary, the SDP is rejected by the corresponding node, the hidden SIP proxy at the AP will NOTIFY the tear down of the call by means of the *bye* command.

To summarize, an example of the complete WiMM operation and transactions between the sub-modules of it has been presented here. From the stations point of view, this VoIP optimization process implies very few changes on its actual VoIP session procedure, since all protocols included in the adaptation procedure (like RTCP and SIP) are already present in any VoIP over WLAN session. The only real change for the node is to subscribe to the new MRQE service and be willing to accept indications from the WiMM, like the Fast RTCP triggering and the codec re-negotiation. Since this process is proven to increase the QoS of the VoIP session under a multi-rate WLAN, the additional effort of the station is minimal compared to the gain from it. Note though, that the implementation proposed here is only a first design of a VoIP optimized AP and future work on this will help identify open issues and improve the solution offered.

6.7 Conclusions

This chapter has addressed the implementation issues arising from the solutions proposed throughout this thesis.

The need for a SIP Proxy was discussed, co-located with the AP but at the same time transparent to the nodes as much as possible. Furthermore, the necessary signaling extensions to support the mechanism of codec re-negotiation were presented, which take the form of a novel usage of the OPTIONS, SUBSCRIBE and NOTIFY SIP messages. Additionally, a new SIP header field, the *Wireless_MRQE* Header, and a new SIP Event, the *MRQE* event, have been defined and their usage explained. The new module that incorporates all the necessary elements for the codec adaptation procedure was finally described, named the Wireless QoS Multi-Rate Module.

We see that the design and implementation of the new AP including the CAC/VCA solution is quite simple and there are few modifications of a standard AP needed, mostly in the software part which makes it even more feasible. We have also explained that even if the new SIP extension is not supported by all STAs from day one, this will not present a particular problem, neither for the STA (it will ignore the unknown header and be excluded from the procedure) nor for the optimization procedure (provided that a majority of stations do support the extension). Although here only the basic design guidelines were given, this can serve as a good starting point for studying further the complications and the details arising from a real implementation. Part of the future work discussed in the next chapter will be dedicated in actually implementing this solution in a prototype Access Point, task already initiated and in progress actually.

CHAPTER 7

Conclusions

7.1 Lessons Learned

The aim of this thesis was to study the impact that the capacity variable (multi-rate) WLANs have on the QoS performance of VoIP flows. Having this in mind, the main effort has been focused on providing a solution that could maintain the QoS performance in acceptable levels and contribute to minimize the dropping ratio of the flows in case of a capacity variation provoked by a channel rate decrease.

The most important contribution of the thesis is the cross-layer algorithm for QoS monitoring and codec adaptation on multi-rate wireless networks. This algorithm provides an integrated environment where the QoS variations are constantly measured and a codec adaptation mechanism addresses them fast and efficiently. The proposed codec adaptation algorithm is composed by three phases: a) *monitoring*, using both MAC-layer for rate changes notifications and RTCP packets for QoS metrics; b) *adaptation*, for choosing the new more suitable codec; and c) *recovery*, for the new codec re-negotiation using SIP. To the best of our knowledge this is the first algorithm of a cross-layer character designed specifically for capacity variable channels and using the SIP protocol to maintain VoIP quality in satisfactory levels.

Two implementation modes of the algorithm have been studied: *centralized*, installed in the Access Point and *distributed*, installed in each mobile station. A performance comparison between the two shows that the centralized version can provide a better cell-wide control of the chosen codec constellation, leading to less codec changes. However, the distributed implementation is easier to implement and alleviates the computational burden from the AP, which makes it an interesting alternative.

As analyzed in chapter 4, this proposed algorithm, in either of its two modes (cen-

tralized, distributed), helps maintaining the measured QoS (using the E-Model and MOS metrics) in high levels and prevents calls from dropping. The codec adaptation procedure can react efficiently to the rate changes and compared with the standard non-adaptive solution can lower significantly both delay and packet losses of active flows, maintaining them both below the thresholds for acceptable voice QoS set by ITU-T. Additionally, this solution is highly flexible since there is no need for *all* active calls to change codec. Especially using the centralized version and giving priority to slow calls first, the number of codec changes is minimized.

Apart from measuring the efficiency of the codec adaptation solution, it was also necessary to provide a fast solution, so as to respond to the QoS degradation in an immediate manner. Thus an important test metric is the total delay of the adaptation process. Although the call is not interrupted during this process, this delay must be kept as low as possible. In order to minimize it, we have analyzed an extension of the RTCP protocol, the fastRTCP, which permits to set the interval between two consequent RTCP reports to a time lower that the default 5 seconds defined in the standard. Using this extension, we have managed to maintain the delay of the codec adaptation algorithm in acceptable levels from the user's perspective.

Moreover, the algorithm can also work in parallel with a Call Admission Control mechanism, in order to increase the limited WLAN capacity and distribute efficiently the resources among the incoming and active calls. The trade-off between the increased number of accepted calls using this method and the lower average QoS as perceived by the users (using the MOS/E-Model quality measurements) has been also discussed. A number of different decision policies has been examined in chapter 5, each one focusing on the optimization of different metrics. Comparing them all, we have concluded that the combined use of codec adaptation on new calls as well as a response to rate changes can be highly beneficial for the system, depending however on the traffic load on it. Especially the use of multi-adaptive policies, which allows more flexibility in changing voice codecs of various calls, can result in a low blocking and dropping probability as well as a very satisfactory MOS value. For low traffic load, applying the codec adaptation procedure both for new incoming calls and for rate changes is the option providing the best overall results. However, when the system is highly loaded, applying codec adaptation only as a rate change reaction is the recommended solution (trying to save active calls from dropping rather that permit the entrance of new).

A new metric named Q-Factor, which can address the trade-off between number of active calls and quality levels, has been proposed, designed and evaluated under the multi-rate WLAN scenario. The first notions of its potential use were given, remaining however an open area currently under further study as we explain later.

Other issues, like the role of codec complexity on deciding the best codec distribution in the cell, have been also analyzed in depth. Summarizing the main conclusions as

discussed in chapter 5, codec complexity plays an important role in the choice of the decision policy and the codec adaptation process, since in the most common case, lower rate codecs imply higher processor capacity need, something that the mobile nodes may not be able to support adequately. We have proven that our solution is applicable even when the processing capacity of the nodes is limited, however its efficiency increases when the codec set that can be used in the adaptation procedure is wider.

Although the major part of the study was focused on a pure-VoIP scenario, the behavior of the cross-layer codec adaptation algorithm in presence of elastic flows has been also considered in chapter 4. We found that applying codec adaptation alone is not sufficient, since it cannot address the effects of the rate changes occurring on TCP nodes. Thus a combination with some other mechanism applicable to TCP flows is essential. In this study a very simple EDCA parameter tuning mechanism was used, although other more sophisticated solutions can be also used. Nevertheless, results clearly show that although codec adaptation alone is not enough, it is however crucial under a multi-rate scenario for the performance of both VoIP and TCP flows; it can alleviate the effects on VoIP flows and consequently lower the total congestion levels of the cell.

Last but not least the proposal and design of a VoIP-enabled Access Point is described in chapter 6, which incorporates all the QoS enhancements discussed in this thesis. Various implementation issues have been identified and addressed efficiently. One of the basic considerations for our solutions was to find a way for the control entity to communicate with the mobile station so as to dictate algorithm-specific actions, such as the sending of the SIP re-Invite for the new call parameter negotiation, or the triggering of fast RTCP messages. In this line, a novel use of the SUBSCRIBE/NOTIFY/OPTIONS SIP messages as also a new SIP extension have been proposed in order to initiate a conversation from a transparent SIP proxy to the mobile nodes.

Finally, the Wireless QoS Multi-Rate Module has been presented. It brings together all the pieces of this study and can be easily included in any commercial Access Point. Both the software and hardware architecture have been reviewed, although the proposed modifications are mostly software based.

7.2 Open issues and future guidelines

Having a first approach on how the proposed solution could work, using the simulation results presented throughout the thesis and the design of its implementation on a real AP, seems that a deployment of this line can be highly beneficial for VoIP over multi-rate WLANs, with relative small additional cost. A number of avenues for further research remain open though, mentioned here briefly.

The codec adaptation algorithm proposed in this thesis is based on the standard

voice codecs, the ones most commonly found in the literature and that are most usually implemented in any user device. In fact, the codec adaptation is not codec dependent and can be implemented using any of the available codecs. However, the research on voice codecs is advancing, with newer methods for more efficient voice codification coming out every day. Among all these, we have already mentioned the Adaptive MultiRate (AMR) codecs. The solution offered by this family of codecs is similar to the one proposed by the Codec Adaptation Algorithm, although with the limitations for our scenario analyzed briefly in chapter 4. Additionally, these codecs were created for GSM networks and no extensive study of their use under multi-rate wireless networks can be found in the literature. The effect of using these codecs, as also other popular ones like the iLBC [5] or the SPEEX [79] could be an interesting area for further study.

The new metric that combines quality and quantity indicators, the Q-Factor explained in chapter 6, has been studied here briefly and only the first notions of its design and use were explained. There are many possibilities arising from a similar metric which should be studied in depth. Integrating metrics that cover both quantitative and qualitative results of the system, facilitates the decision-making in situations where this kind of trade-off is common. As an example, a complete cell-wide optimization can be designed using this factor as a guide for the decisions on how, when and where to apply codec adaptation.

Additionally, a new research line would be that of extending the algorithm to include multimedia traffic other than VoIP and their parameter adaptation. Multimedia traffic have similar QoS restriction as VoIP and suffer equally from the capacity variable WLAN networks. An extended study in this area is necessary in order to determine the differences that this traffic has compared to VoIP. Then, an equivalent codec adaptation solution could be planned to solve similar QoS degradation issues in a WLAN hotspot.

We have explained the benefits of using a centralized control entity, named the Wireless QoS Multi-Rate Module co-located with the Access Point in the 802.11 cell. A prototypical implementation of the whole mechanism is currently underway, which will help to calculate more precisely the additional cost (in terms of complexity, delay, computational power and storage) incurred by the mechanism.

However, the possibility of placing this control entity separately from the AP in a central point of the network outside the WLAN cell can be also discussed. The current trend towards "lightweight" APs, in which most complex computations are performed at a central element, called a *concentrator*, invites to rethink the current distribution of functions. Mainly, the AP architecture would have to be revisited and the WiMM relocated at the concentrator. Furthermore, a new signaling protocol between AP and concentrator would be needed, or modifications to existing protocols, like LWAPP [14], would have to be designed.

The main advantage of this last proposal is that by placing the WiMM module

beyond a single AP, inter-cell resource optimization is perfectly possible. This would allow to instruct certain wireless stations to join alternative APs in order to better distribute the VoIP load and optimize both the number and quality of active calls. This would be analogous to the procedure that is followed in cellular telephone networks. The alternative between performing codec adaptation or instructing the station to join a different cell should be studied further.

As a result, the most important and most immediate future extension of our proposal has to do with the scenario used. In this thesis a single-cell hotspot scenario has been considered, with the nodes connected in an infrastructure mode to one Access Point acting as a gateway to the wired network. This is a simple but typical hotspot architecture and many real hotspot implementations follow this scenario and can benefit from it. Nevertheless, with the further deployment of mesh networks, the next big step from here would be to study an inter-cell codec optimization process. This, apart from the lightweight AP implementation issues mentioned above, could include modifications to the core algorithm so that two additional possibilities could be contemplated:

- when the code adaptation inside the cell is not possible for any reason (p.e. fixed policy used at this cell, low processing power of nodes etc.) the option of handing over the call to another cell instead of dropping it could be used. A study of load balancing and choosing the most suitable cell among all can be included and it is not trivial. This procedure is similar to the one used actually in the mobile cellular networks.

- even if adapting the codecs inside the cell is possible, the same option of passing the call to another cell with higher available capacity and/or significantly lower load could be used, in order to maintain a higher quality of service for all calls. Again the study of associating the node to various Access Points and choosing the most adequate is a whole new different subject. Nevertheless the cross-layer algorithm presented here could be adapted and used in cooperation with these load balancing mechanisms.

In order to perform all the inter-cell calculations, the architecture of the lightweight AP and the concentrator as mentioned above would be necessary. The lightweight APs can include minimum functionalities, only the necessary for the nodes' association procedures to the cel. All the intelligence of the network, including CAC, codec adaptation, policies and rest of the optimization decisions reviewed here would lie on the centralized control module. This can minimize significantly the deployment costs of a large multi-cell network.

All in all, it seems an intriguing and promising avenue for research.

List of Acronyms

AC Access Category

ACK Acknowledgement

AIFS Arbitrary Inter Frame Space

AP Access Point

ARF Auto Rate Fallback

BSS Basic Service Set

CAC Call Admission Control

CSMA/CA Carrier Sense Multiple Access with Collision Avoidance

CW Contention Window

DCF Distributed Coordination Function

DIFS DCF Inter Frame Space

EDCA Enhanced Distributed Common Access

FEC Forward Error Correction

GoF Grade of Service

IEEE Institute of Electrical and Electronic Engineering

IMS IP Multimedia Subsystem

ITU International Telecommunication Union

LA Link Adaptation

MAC Medium Access Control

MIPS Million Instructions Per Second

MOS Mean Opinion Score

PHY Physical Layer

PSTN Public Switched Telephone Network

QoS Quality of Service

RTP Real-Time Protocol

RTCP Real-Time Control Protocol

SDP Session Description Protocol

SIFS Short Inter Frame Space

SIP Session Initiation Protocol

SNR Signal to Noise Ratio

STA Wireless Station

TCP Transport Control Protocol

TXOP Transmission Opportunity

UDP User Datagram Protocol

VCA Voice Codec Adaptation

VoIP Voice over Internet Protocol

WiMM Wireless Multi-rate Module

WLAN Wireless Local Area Networks

Bibliography

[1] IEEE Std 802.11. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *ANSI/IEEE Std 802.11*, 1999.

[2] IEEE Std 802.11e. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications; Amendment: Medium Access Control(MAC) Quality of Service Enhancements. *IEEE Std 802.11e*, 2005.

[3] I. Aad, C. Castelluccia, and R.A. Inria. Differentiation mechanisms for IEEE 802.11. *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 1, 2001.

[4] S.A. AlQahtani and A.S. Mahmoud. Dynamic radio resource allocation for 3G and beyond mobile wireless networks. *Computer Communications*, 30(1):41–51, 2006.

[5] S. Andersen, A. Duric, H. Astrom, R. Hagen, W. Kleijn, and J. Linden. RFC 3951: Internet Low Bit Rate Codec (iLBC). *Internet Engineering Task Force*, 2001.

[6] A. Barberis, C. Casetti, JC De Martin, and M. Meo. A simulation study of adaptive voice communications on IP networks. *Computer Communications*, 24(9):757–767, 2001.

[7] J. Barceló, B. Bellalta, A. Sfairopoulou, C. Cano, and M. Oliver. No Ack in IEEE 802.11e Single-Hop Ad-Hoc VoIP Networks. *The 7th IFIP Annual Mediterranean Ad Hoc Networking Workshop*, 1, 2008.

[8] B. Bellalta. *Flow-level QoS guarantees in IEEE 802.11e EDCA-based WLANs.* PhD Thesis, Universitat Pompeu Fabra, 2007.

[9] B. Bellalta and M. Meo. Call Admission Control in WLANs. *In the book Resource, Mobility and Security Management in Wireless Networks and Mobile Communications, Auerbach Publications, CRC Press, USA*, November 2006.

[10] B. Bellalta, M. Meo, and M. Oliver. VoIP Call Admission Control in WLANs in Presence of Elastic Traffic. *Journal of Communications Software and Systems*, January 2007.

[11] B. Bellalta, M. Oliver, M. Meo, and M. Guerrero. A Simple Model of the IEEE 802.11 MAC Protocol with Heterogeneous Traffic Flows. In *IEEE Eurocon 2005, Belgrade, Serbia and Montenegro*, November 2005.

[12] G. Bianchi. Performance Analysis of the IEEE 802.11 Distributed Coordination Function. *IEEE Journal on Selected Areas in Communications, Vol. 18, No. 3*, March 2000.

[13] L. Cai, Y. Xiao, X.S. Shen, and J.W. Mark. VoIP over WLAN: Voice capacity, admission control, QoS, and MAC. *Int. J. Commun. Syst*, 19:491–508, 2006.

[14] P. Calhoun, B. OHara, S. Kelly, R. Suri, D. Funato, and M. Vakulenko. Light Weight Access Point Protocol (LWAPP). *draft-ohara-capwap-lwapp-02, April*, 2005.

[15] G. Camarillo and M.A. García-Martín. *The 3G IP Multimedia Subsystem (IMS): Merging the Internet and the Cellular Worlds*. John Wiley and Sons, 2004.

[16] G. Camarillo, T. Kauppinen, M. Kuparinen, I.M. Ivars, and E. Res. Towards an innovation oriented IP multimedia subsystem [IP Multimedia Systems (IMS) Infrastructure and Services]. *Communications Magazine, IEEE*, 45(3):130–136, 2007.

[17] C. Cano, B. Bellalta, and M. Oliver. A Simple Model of the IEEE 802.11 MAC Protocol with Heterogeneous Traffic Flows. In *IEEE Personal, Indoor and Mobile Radio Communications (PIMRC) 2007, Athens, Greece*, September 2007.

[18] G. Chen. Component Oriented Simulation Toolkit. *http://www.cs.rpi.edu/∼cheng3/*, 2004.

[19] J.J. Chen, L. Lee, and Y.C. Tseng. Integrating SIP and IEEE 802.11e to support handoff and multi-grade QoS for VoIP applications. In *Proceedings of the 2nd ACM international workshop on Quality of service & security for wireless and mobile networks*, pages 67–74. ACM Press New York, NY, USA, 2006.

[20] S. Choi, J. del Prado, N. Sai Shankar, and S. Mangold. IEEE 802.11e contention-based channel access (EDCF) performance evaluation. *ICC '03. IEEE International Conference on Communications, Seattle, US*, May 2003.

[21] S. Chung and J. You. Call Admission Control in CDMA Cellular Networks with Grade of Service and Quality of Service Dimensioning. *WSEAS Transactions on Communications*, 5(12):2182, 2006.

[22] ID 14069 Cisco Technical Document. Understanding Codecs: Complexity, Hardware Support, MOS, and Negotiation.

[23] R.G. Cole and J.H. Rosenbluth. Voice over IP performance monitoring. *ACM SIGCOMM Computer Communication Review*, 31(2):9, 2001.

[24] D.E. Collins. *Carrier Grade Voice Over IP*. McGraw-Hill Professional, 2003.

[25] Gartner Dataquest. Gartner Says Simplistic Focus on Hot Spot Profits Misguided, Rationales for Growth Are More Complex. *http://www.gartner.com*, 2003.

[26] D.J. Deng and R.S. Chang. A priority scheme for IEEE 802.11 DCF access method. *IEICE Transactions on Communications*, 82:96–102, 1999.

[27] GSM ETSI. 06.90. *Digital cellular telecommunications system (Phase 2+) adaptive multi-rate (AMR) speech transcoding*, 1998.

[28] ITU-T Rec. G.107. The E-model, a computational model for use in transmission planning. 2000.

[29] ITU-T Rec. G.113. Transmission impairments due to speech processing. 2001.

[30] D. Gao, J. Cai, and K.N. Ngan. Admission control in IEEE 802.11 e wireless LANs. *Network, IEEE*, 19(4):6–13, 2005.

[31] MT Gardner, VS Frost, and DW Petr. Using optimization to achieve efficient quality of service in voice over IP networks. *Performance, Computing, and Communications Conference, 2003. Conference Proceedings of the 2003 IEEE International*, pages 475–480, 2003.

[32] S. Garg and M. Kappes. An experimental study of throughput for UDP and VoIP traffic in IEEE 802.11 b networks. *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*, 3, 2003.

[33] S. Garg and M. Kappes. Admission Control for VoIP Traffic in IEEE 802.11 Networks. *IEEE GLOBECOM 2003, San Francisco, USA*, December 2003.

[34] S. Garg and M. Kappes. Can I add a VoIP call? *IEEE International Conference on Communications (ICC'03), Anchorage, Alaska, USA*, May 2003.

[35] ITU-T Rec. H.323. Packet-based multimedia communications systems. 1998.

[36] I. Haratcherev, J. Taal, K. Langendoen, R. Lagendijk, and H. Sips. Automatic IEEE 802.11 rate control for streaming applications. *Wireless Communications and Mobile Computing*, 5(4):421–437, 2005.

[37] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda. Performance Anomaly of 802.11b. *IEEE INFOCOM 2003, San Francisco, USA*, April 2003.

[38] Y. Hiwasaki, H. Ohmuro, T. Mori, S. Kurihara, and A. Kataoka. A G. 711 Embedded Wideband Speech Coding for VoIP Conferences. *IEICE Transactions on Information and Systems*, 89(9):2542–2552, 2006.

[39] D. P. Hole and F. A. Tobagi. Capacity of an IEEE 802.11b Wireless LAN supporting VoIP. *IEEE International Conference on Communications (ICC'04), Paris, France*, June 2004.

[40] C.W. Huang, A. Chindapol, J.A. Ritcey, and J.N. Hwang. Link Layer Packet Loss Classification for Link Adaptation in WLAN. *Information Sciences and Systems, 2006 40th Annual Conference on*, pages 603–608, 2006.

[41] inAccess Networks. Packet voice codecs. World Wide Web electronic publication, 2005.

[42] Jiwire. WiFi Hotspot Finder. *http://www.jiwire.com/search-hotspot-locations.htm*, 2007.

[43] A. Kamerman and L. Monteban. WaveLAN®-II: A high-performance wireless LAN for the unlicensed band: Wireless. *Bell Labs technical journal*, 2(3):118–133, 1997.

[44] J-O. Kim, H. Tode, and K. Murakami. Service-based rate adaptation architecture for IEEE 802.11 e QoS networks. *Global Telecommunications Conference, 2005. GLOBECOM'05. IEEE*, 2005.

[45] M. Kuorilehto, M. Hannikainen, M. Niemi, and T. Hamalainen. Implementation of wireless LAN access point with quality of service support. *IEEE IECON 02, 28th Annual Conference of the Industrial Electronics Society*, 3, 2002.

[46] R.G. Lee, C.C. Lai, and C.C. Hsiao. Implementation of a Cable Access Point (CAP) Device for Ubiquitous Network Applications. *Proceedings of the 4th Annual Communication Networks and Services Research Conference (CNSR'06)-Volume 00*, pages 235–242, 2006.

[47] Linksys. Wireless-N Access Point with Power over Ethernet. World Wide Web electronic publication.

[48] J.S. Liu and C.H.R. Lin. A Relay-Based MAC Protocol for Multi-Rate and Multi-Range Infrastructure Wireless LANs. *Wireless Personal Communications*, 34(1):7–28, 2005.

[49] T. Lundberg, P. de Bruin, S. Bruhn, S. Hakansson, and S. Craig. Adaptive Thresholds for AMR Codec Mode Selection. *Vehicular Technology Conference, 2005. VTC 2005-Spring. 2005 IEEE 61st*, 4, 2005.

[50] S. Mangold, S. Choi, GR Hiertz, O. Klein, B. Walke, P. Res, and G. Aachen. Analysis of IEEE 802.11 e for QoS support in wireless LANs. *Wireless Communications, IEEE [see also IEEE Personal Communications]*, 10(6):40–50, 2003.

[51] M. Manousos, S. Apostolacos, I. Grammatikakis, D. Mexis, D. Kagklis, and E. Sykas. Voice-quality monitoring and control for VoIP. *Internet Computing, IEEE*, 9(4):35–42, 2005.

[52] A.P. Markopoulou, F.A. Tobagi, and M.J. Karam. Assessment of VoIP quality over Internet backbones. *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 1, 2002.

[53] P. McGovern, S. Murphy, and L. Murphy. Addressing the Link Adaptation Problem for VoWLAN using codec adaptation. *Global Telecommunications Conference, 2006. GLOBECOM'06. IEEE*, 2006.

[54] P. McGovern, S. Murphy, and L. Murphy. Protection Against Link Adaptation for VoWLAN. *Proc. 15th IST Mobile and Wireless Communications Summit, 2006*, 2006.

[55] N. Nasser. Adaptability enhanced framework for provisioning connection-level QoS in multimedia wireless networks. *Wireless and Optical Communications Networks, 2005. WOCN 2005. Second IFIP International Conference on*, pages 275–279, 2005.

[56] SMC Networks. SMCWEBT-G EZ Connect g Wireless Ethernet Bridge. World Wide Web electronic publication.

[57] ns 2 SIP patch. National Institute of Standards and Technology (NIST). *http://www.isi.edu/noesaquestlink*, 2005.

[58] NS2. Network Simulator, release 2.28. *http://www.isi.edu/nsnam/ns/*, February 2005.

[59] J. Ott, S. Wenger, N. Sato, C. Burmeister, and J. Rey. RFC 4585: Extended RTP Profile for Real-Time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF). Technical report, 2006.

[60] ITU-T Rec. P.800. Methods for subjective determination of transmission quality. 1996.

[61] S. Pilosof, R. Ramjee, D. Raz, Y. Shavitt, and P. Sinha. Understanding TCP fairness over wireless LAN. *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE*, 2.

[62] D. Pong and T. Moors. Call admission control for IEEE 802.11 contention access mechanism. *Global Telecommunications Conference, 2003. GLOBECOM'03. IEEE*, 1, 2003.

[63] Z. Qiao, L. Sun, N. Heilemann, and E. Ifeachor. A new method for VoIP quality of service control use combined adaptive sender rate and priority marking. *Communications, 2004 IEEE International Conference on*, 3, 2004.

[64] ITU-T Rec. G.711: Pulse code modulation (PCM) of voice frequencies, 1988.

[65] ITU-T Rec. G.728: Coding of Speech at 16 kbit/s using Low-Delay Code Excited Linear Prediction, 1994.

[66] ITU-T Rec. G.114: One-Way Transmission Time. 1996.

[67] ITU-T Rec. G.729: Coding of Speech at 8 kbit/s using Conjugate Structure Algebraic Code Excited Linear Prediction (CS-ACELP), 1996.

[68] ITU-T Rec. Reduced Complexity 8 kbit/s CS-ACELP Speech Codec, 1996.

[69] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. RFC3261: SIP: Session Initiation Protocol. *Internet Engineering Task Force*, 2002.

[70] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RFC3550: RTP: A Transport Protocol for Real-Time Applications. *Internet Engineering Task Force*, 2003.

[71] P. Serrano. Estrategias de configuración de redes wlan ieee 802.11e edca. *Tesi Doctoral, Departamento de Ingeniería Telemática, Universidad Carlos III de Madrid*, September 2006.

[72] A. Servetti and J.C. De Martin. Adaptive interactive speech transmission over 802.11 wireless LANs. *Proc. IEEE Int. Workshop on DSP in mobile and Vehicular Systems*, 2003.

[73] Ali K. Setoodehnia, Hong Li, Kamal Shahrabi, and Mojtaba Shariat. Voice quality of service in cable ip network. *International Journal of Modern Engineering*, 5(1), Fall2004.

[74] A. Sfairopoulou, C. Macian, and B. Bellalta. QoS adaptation in SIP-based VoIP calls in multi-rate 802.11 environments. *ISWCS 2006, Valencia,*, September 2006.

[75] IP Multimedia Subsystem. Stage 2, 3GPP TS 23.228, 2005.

[76] TeamF1. Managed Access Point Solutions (MAPS) datasheet. World Wide Web electronic publication.

[77] A. Trad, F. Munir, and H. Afifi. Capacity Evaluation of VoIP in IEEE 802.11e WLAN Environment. *Consumer Communications and Networking Conference (CCNC), 2006, 3rd IEEE*, 2, 2006.

[78] A. Trad, Q. Ni, and H. Afifi. Adaptive VoIP Transmission over Heterogeneous Wired/Wireless Networks. *Lecture Notes in Computer Science*, pages 25–36, 2004.

[79] J.M. Valin. Speex: A Free Codec For Free Speech.

[80] W. Wang, S. Chang Liew, and V. O. K. Li. Solutions to Performance Problems in VoIP Over a 802.11 Wireless LAN. *IEEE Transactions on Vehicular Technology, vol. 54, No. 1*, January 2005.

[81] P.Y. Wu, Y.C. Tseng, and H. Lee. Design of QoS and Admission Control for VoIP Services over IEEE 802.11e WLANs. *National Computer Symposium*, 2005.

[82] Y. Xiao. A simple and effective priority scheme for IEEE 802.11. *Communications Letters, IEEE*, 7(2):70–72, 2003.

[83] H. Zhai, J. Wang, and Y. Fang. Providing statistical QoS guarantee for voice over IP in the IEEE 802. 11 wireless LANs. *IEEE Wireless Communications Magazine*, 13(1):36–43, 2006.