POLYTECHNIC UNIVERSITY OF VALENCIA

DEPARTMENT OF COMPUTER ENGINEERING

Analysis and design of efficient techniques for video transmission in IEEE 802.11 wireless ad hoc networks

Thesis submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Informatics

Carlos Miguel Tavares de Araújo Cesariny Calafate

Ph.D. advisors:
Dr. Pietro Manzoni
Dr. Manuel Pérez Malumbres

Valencia, April 2006

To my wife, Mónica,
and my daughter, Lara.

Acknowledgments

Buddhists say that it's not so important what lies at the end of the path; it is by walking through that path that one builds wisdom and strengthens the spirit. Nevertheless, how pleasant it is to finally reach the end of such a long road! Obviously, this four-year-long journey was not made in isolation, being the result of a cooperative learning process where my advisors, my colleagues and several researchers worldwide also made an important contribution.

I begin this acknowledgment round by thanking the never-ending flow of ideas, energy and efforts of my two Ph.D. advisors: Dr. Pietro Manzoni and Dr. Manuel Pérez Malumbres. Their contributions and their clear-mindedness were only surpassed by their patience towards me.

I could never forget the help and support of other members of our research group, which made me feel at home. I thank Dr. Juan Carlos Cano for his trust and friendship; also, I thank Prof. Miguel Mateo for always being so kind, quickly fixing any hardware burn-outs caused by my never-ending simulation runs. I would also like to thank the remaining members of our research group: José, Miguel, Román, Julio, Lourdes and Alberto. You all made me feel perfectly integrated!

Special thanks go to my friends which joined me at coffee breaks: Arnoldo, Carlino, Danilo, David, Eric, Guillermo, Ingrid, Johann, Jordi, Luis, Pepe and Tito. Your company throughout these years was an escape to routine and everyday problems, helping me to endure work without complaining.

I finally thank all the members of my family, both the elder and the younger ones. Thanks to my dear wife Mónica for always being so supportive and comprehensive. You and Lara are my joy for living! Thanks also to my parents for accompanying me in my growth as an individual, teaching me ethical and moral principles that will prevail for all my life.

Thanks to you all...

Abstract

Wireless mobile ad hoc networks, also known as MANETs, are composed by independent mobile stations that communicate without requiring any sort of infrastructure for support. These networks are characterized by variable bandwidth values and frequent path breaks, which are due to channel noise, interference between stations and mobility. Such factors require significant adaptation capabilites at different levels of the protocol suites employed, enabling stations to quickly respond to fast-changing network conditions. Research on the most adequate protocols for the physical, MAC and routing layers is still on-going, though some basic consensus has already been reached and several testbeds have been setup around the world.

To deploy real-time multimedia services, namely voice and video, on top of such an unreliable network environment is a very challenging task. In this thesis we propose to achieve that goal starting from currently available Wi-Fi technology, and gradually finding the most adequate enhancements to each protocol layer of interest; we then combine these enhancements until we achieve a complete QoS framework for ad hoc networks. By using currently available technology we assure that the proposal of this thesis has an inherent high-level of applicability on real life environments.

Since our working field focuses on video transmission over wireless ad hoc networks, we will show how it is possible to support several QoS-constrained video streams in MANET environments characterized by moderate to high mobility levels, and by a significant amount of best effort traffic.

Resumen

Las redes inalámbricas móviles ad hoc, también conocidas como MANETs, están compuestas de estaciones móviles independientes que comunican entre sí sin necesitar de ningún tipo de infraestructura de soporte. Estas redes se caracterizan por tener un ancho de banda variable, así como por pérdidas frecuentes de ruta que se pueden deber al ruido del canal inalámbrico, a la interferencia entre estaciones móviles o a la movilidad. Dichos factores requieren una capacidad de adaptación importante a diferentes niveles de las pilas de protocolos empleadas, permitiendo a las estaciones responder rápidamente a cambios bruscos de las condiciones de la red. A pesar de que aún se estén realizando investigaciones para buscar los protocolos más adecuados para las capas física, MAC y de encaminamiento, sí se ha logrado un nivel básico de consenso, lo que permitió que por todo el mundo se haya empezado ya a montar entornos de prueba. Ofrecer servicios multimedia, tales como voz y vídeo en un entorno de red con tan poca fiabilidad es un desafío importante. En esta tesis nos proponemos alcanzar ese objetivo partiendo de la tecnología Wi-Fi de que disponemos actualmente, encontrando de forma gradual las mejoras más importantes para cada capa de protocolos que nos sea de interés; al final combinamos esos elementos para lograr una solución que ofrezca QoS en redes ad hoc. Al utilizar la tecnología que disponemos actualmente nos aseguramos que la propuesta de esta tesis tenga, de forma inherente, un alto grado de aplicabilidad en entornos reales. Una vez que nuestro campo de trabajo enfoca la transmisión de vídeo en redes inalámbricas ad hoc, demostraremos cómo es posible soportar varios flujos de vídeo con requisitos de QoS en entornos MANET caracterizados por altos niveles de movilidad, además de una cantidad significativa de tráfico del tipo best effort.

Resum

Les xarxes inalàmbriques mòbils ad hoc, també conegudes com MANETs, estan composades de estacions mòbils independents que es comuniquen entre sí sense necessitar de cap tipus de infraestructura de suport. Aquestes xarxes es caracteritzen per tindre un ample de banda variable, així com per pèrdues freqüents de ruta que es poden deure al soroll del canal inalàmbric, a la interferència entre estacions mòbils o a la mobilitat. Aquests factors requereixen una capacitat de adaptació importat a diferents nivells de les piles de protocols empleats, permitent a les estacions respondre ràpidament a cambis bruscs en les condicions de la xarxa. Malgrat que encara estiguen realitzant-se investigacions per procurar els protocols mes adequats per les capes física, MAC y de encaminament, sí s'ha aconseguit un nivell bàsic de consens, el que va permetre que per tot el món s'haixca començat ja a muntar entorns de proba.

Oferir serveis multimèdia tals com veu i vídeo en un entorn de xarxa amb tan poca fiabilitat es un desafiament important. En aquesta tesis ens proposem aplegar a aquest objectiu partint de la tecnologia Wi-Fi de que disposem actualment, trobant de forma gradual les millores mes importants per a cada capa de protocols que ens siga de interès; a la fi combinarem eixos elements per aconseguir una solució que ofereix QoS en xarxes ad hoc. Al utilitzar la tecnologia que disposem actualment ens assegurem que la proposta de esta tesis tinga, de forma inherent, un alt grau de aplicabilitat en entorns reals.

Una vegada que el nostre camp de treball enfoca la transmissió de vídeo en xarxes inalàmbriques ad hoc, demostrarem com es possible suportar diferents fluxos de vídeo amb requeriments de QoS en entorns MANET caracteritzats per alts nivells de mobilitat, així com de una quantitat significativa de tràfic del tipus best effort.

Contents

List of Algorithms

List of Figures

List of Tables

Motivation, objectives and organization of the thesis

Motivation

The massive deployment of devices with wireless capabilities is a recent phenomena. Nevertheless, during the next few years this trend is expected to become even more pronounced.

Most of the wireless networks available nowadays are infrastructure-based. However, users may not always want to communicate using an infrastructure due to security, costs, or bandwidth constraints. Also, there are many situations where no infrastructure is available; examples are mountain, jungle, forest and desert areas, as well as space and sea. In other cases the available infrastructure may fail, forcing users to find alternate means of communication; this includes disaster and war areas. So, the development of a technology that sustains reliable wireless ad hoc networking seems both logical and necessary.

Real-time multimedia communication between peers is another 21st century phenomena growing at a steady pace. Aided by the recent developments in terms of video compression technology, audio-visual conversations are the next-to-come standard for human communication.

When trying to combine wireless ad hoc networks with real-time multimedia streams we find that there is a long road ahead before an actual system can be deployed offering the desired reliability and QoS support.

Objectives of the thesis

In 2002, when the first steps of this thesis were taken, wireless mobile ad hoc networks based on the IEEE 802.11 technology were scarce and limited to a few laboratory testbeds. When analyzing the performance of these networks with the technology available we obtained very poor results due to low bandwidth, intermittent connectivity and lack of any QoS support. Though these preliminary results were discouraging, the systematic approach followed for detecting and solving the different problems found, along with the standardization of improved wireless technologies, has made the initial objectives set for the thesis attainable in a reasonable period.

The main contribution of this thesis is to propose a system for reliable video transmission in wireless ad hoc networks. This system consists of a global solution achieved with different building blocks, each targeting a distinct protocol layer. So, the overall functionality of the system results of combining these different building blocks, making the enhancements proposed for the different protocol layers work as an harmoniuos whole.

Among our contributions we include a study of the H.264 codec to achieve optimal tuning so as to transmit video in wireless ad hoc networks with high resilience to packet losses.

Our second contribution consists of a performance evaluation of H.264 streams in typical MANET environments with currently available technology, detecting the problems to be solved at the different protocol layers.

The third contribution is designing an end-to-end path model for MANETs. That model can be used to accelerate the tuning of video codecs for optimal performance on such environments by reducing the evaluation period drastically.

The fourth contribution is a proposal for enhancing a well-known routing protocol for MANETs, DSR, so as to reduce as much as possible the impact of mobility on real-time multimedia streams, especially video streams.

As a fifth contribution we set forth an analysis of the upcoming IEEE 802.11e technology to assess its adequateness to offer MAC level QoS support in multihop MANET environments, as well as the improvements it offers in terms of routing efficiency.

The last contribution of this thesis consists of proposing a novel soft QoS model for MANET environments based on distributed admission control.

After describing our distinct contributions in detail we proceed by making a joint evaluation of all the previous proposals, obtaining a clear picture of the overall improvements achieved.

Organization of the thesis

This thesis is organized in the following manner: in the next three chapters we make a background introduction to the different technologies used for this thesis, referring to some related works in each of the referred research fields. So, in chapter 1 we make an introduction to wireless networks. Our focus is given to the IEEE 802.11 standard since it provides the technology we use to create wireless links between MANET stations. In chapter 2 we analyze the state-of-the-art in mobile ad hoc networks. We focus on two different issues: routing protocols and QoS support. The last introductory chapter, chapter 3, is dedicated to the most recent video standard known as H.264.

In chapter 4 we begin by tuning the H.264 codec for optimum performance in MANET environments, and we then analyze the performance experienced in a simulated MANET. The results found on that chapter will make clear the different goals that have to be attained to develop a fully functional QoS framework.

In chapter 5 we propose a novel end-to-end path model for MANET environments. This model, though a complementary work to the primary purpose of this thesis, can be extremely useful by allowing the developers of video codecs to analyze their performance in MANET environments much faster than using simulation or actual testbeds.

Chapter 6 is dedicated to routing issues. We propose enhancements to both the route discovery techniques and the packet forwarding algorithms to improve the performance of real-time video streams in the presence of significant levels of mobility.

In chapter 7 we delve into the upcoming IEEE 802.11e technology, analyzing its effectiveness in multi-hop ad hoc networks. We also include a study on the benefits this technology brings in terms of increased routing responsiveness.

DACME, our proposal for distributed admission control in MANET environments, is the topic of chapter 8.

In chapter 9 we make a joint evaluation of the different enhancements proposed throughout the thesis, showing how we were able to gradually progress from legacy MANET technology to a reliable video transmission system capable of offering QoS support, increased support for mobility and admission control.

Finally, in chapter 10 we present a summary of the main results of this thesis, along with some concluding remarks. We also include a list of the publications related to the thesis, and we comment on possible future works that can derive from the work here presented.

Chapter 1

Wireless networks

The development of communication networks was significant step for mankind, undoubtly agilizing everyday's tasks and improving the quality of life of many. Both telecommunication and computer networks began with a strong emphasis on wires, both for the communication infrastructure and for the last hop where the actual connection towards the users' terminals takes place. In recent years this trend has shifted towards wireless networks, especially at the user side. This shift comes from the demand of improved mobility support and greater flexibility, so as to face the challenges of our fast-changing society.

In this chapter we will analyze the state-of-the-art of the wireless technologies that are being widely adopted by both consumers and industry.

The main focus of this chapter is given to the IEEE 802.11 technology since it is the technology of choice for mobile ad hoc networks (MANETs). So, section 1.2 will be dedicated to the IEEE 802.11 standard, as well as to some of its annexes that are relevant to this thesis.

We end this chapter offering an overview of another technology that is also receiving much interest from both consumers and industry - Bluetooth - evidencing its limitations in supporting mobile ad hoc networks. This will be the topic of section 1.3, which also presents a study on the mutual interaction between Bluetooth and IEEE 802.11b technologies.

1.1 Introduction

In the last few years wireless networks have become ubiquitous. This is due to reasons such as the current life style, where there is the need to stay constantly connected to local area networks or to the Internet, and also due to other needs such as supporting mobility and offering greater flexibility. Depending on their purpose, we can separate wireless networks into three groups: Personal Area Networks (e.g.: Bluetooth), Local Area Networks (e.g.: IEEE 802.11 or HiperLAN2) and Wide Area Networks (e.g.: GSM, TDMA, CDMA, GPRS, EDGE, W-CDMA, 3G).

On figure 1.1 we can notice the differences between the different wireless tech-

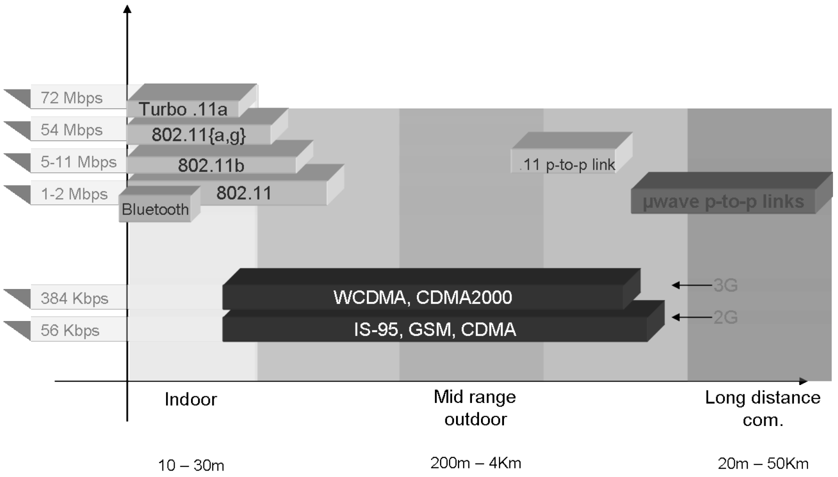


Figure 1.1: Bandwidth variation with range for different wireless technologies

nologies available in terms of communication range and bandwidth. That figure evidences the inverse relationship between bandwidth and range; in the wired networks field, a similar situation happened some years ago.

The appearance of wireless local area networks offers several advantages, besides those referred before for wireless networks in general. Among these is the compatibility with the already available wired networks, ease of installation, cost reduction, ease of management, scalability, passing through physical barriers, etc. From a commercial point of view we can also appreciate the advantages that this sort of networks offer. The installation of the so-called *hot-spots* allows cafes, pubs and restaurants to attract more clients, it allows airports, hotels and trains to receive an extra income by offering Internet access, and it can also improve the productivity of companies by allowing workers to access the internal network while moving through zones where wired connections are not possible.

The main purpose of wireless networks is to support computational and communication services while moving. Depending on the type of network used, though, the techniques employed vary. For instance, cellular networks are characterized by a single wireless hop before reaching the wired portion of the network. Also, the space is divided into cells, where each user is assigned to a cell's base station.

As we will see in the remaining of this chapter, IEEE 802.11-based wireless LANs differ from cellular networks in several ways. Concerning Bluetooth-based networks (see section 1.3) and mobile ad hoc networks (topic of the next chapter), the differences towards cellular networks are even more evident, as we will show.

1.2 IEEE 802.11

The IEEE 802.11 standard [WG99] is a technology whose purpose is to provide wireless access to local area networks (WLANs). Stations using this technology access the wireless medium using either the Point Coordination Function (PCF) or the Distributed Coordination Function (DCF).

The Point Coordination Function is a centralized access mode optionally used in a Basic Service Set (BSS) when a point coordinator (PC) is available. The PC is typically an *access point* (AP), and so the stations are said to operate in infrastructure mode. When relying on the PCF, contention-free periods (CFP) and contention periods (CP) alternate over time. The regular generation of beacons allows stations to associate and synchronize with the PC. Typically a CFP is started after a beacon management frame, followed by a CP; together they form a superframe. During the CFP there is no contention, and so stations are simply polled by the PC. The CF-End control frame is transmitted by the PC to indicate the end of the CF period, and the beginning of the CP. During the CP stations access the medium using the DCF.

The Distributed Coordination Function (DCF) uses a listen-before-talk scheme named *carrier sense multiple access* (CSMA) with *collision avoidance* (CA). It is used by stations in a BSS during the CP and also by stations in an IBSS (Independent Basic Service Set) operating in ad hoc mode. The CSMA technology distributes the medium access task among all stations, making every station responsible for assuring the delivery of *MAC service data units* (MSDUs) and reacting to collisions. The collision avoidance (CA) scheme is used to reduce the probability of collisions between different stations. To achieve this it applies a backoff procedure before initiating a transmission if the medium wasn't previously idle. Stations select a random number of slots to wait before transmission on an interval between 0 and the current *contention window* (CW) value. The value for CW is set initially to the minimum value defined for the radio technology being used (CWmin), being increased when consecutive collisions occur up to a maximum value (CWmax).

The CSMA/CA mechanism shows good adaptation to different numbers of transmitting stations, and probabilistically shares the channel equally among all transmitting stations. However it offers no mechanisms to perform traffic differentiation, making it very difficult to offer QoS support. The IEEE 802.11e working group was created to focus on this issue, and a new international standard was completed at the end of 2005.

Products using the IEEE 802.11 standard are experiencing a growing interest by companies all over the world. This is due to a good balance between cost, range, bandwidth and flexibility. The bandwidths defined by the standard currently range from 1 to 54 Mbps, but other standards being developed in the 802.11 family shall offer greater bandwidth, though maintaining the same frequency bands. The IEEE 802.11 standard offers the two operation modes referred before: *Point Coordination Function* (PCF) and *Distributed Coordination Function* (DCF). PCF can only be used in the infrastructure mode where access points are responsible for the coordination among nodes. DCF, on the other hand, is a fully distributed mechanism which allows stations to compete for the medium without requiring an

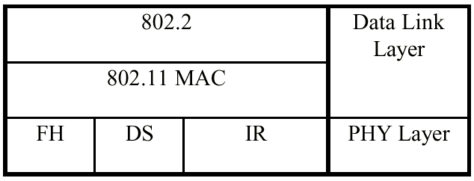


Figure 1.2: Different layers integrating the IEEE 802.11 architecture

access point for support.

As shown in figure 1.2, IEEE 802.11's architecture allows, at the physical level, to use *Frequency Hopping* or *Direct Sequence* modulation techniques, apart from the modulation techniques defined on the IrDA standard [IrD01]. Above IEEE 802.11's MAC level an 802.2 layer is available, being responsible for the logical control of the channel (LLC).

In the next section we offer more details relatively to IEEE 802.11's physical layers.

1.2.1 Physical level

The IEEE 802.11 standard specifies three physical layers that were standardized in 1997. Two of them were designed for operation at the ISM (Industry, Scientific and Medical) frequency band (2.4 GHz); these are the Frequency-hopping (FH) and Direct-sequence (DS) spread-spectrum techniques. A physical layer using infrared light (IR) was also defined.

At ISM bands, the initial IEEE 802.11 standard defined operation with data rates of 1 or 2 Mbps using either *Direct Sequence Spread Spectrum* (DSSS) or *Frequency Hopping Spread Spectrum* (FHSS) modulation techniques. We will now offer more details for the DSSS technology since the two remaining physical layer technologies are currently not being used.

The DSSS technology works by modulating data with a second pattern (chipping sequence). In IEEE 802.11 DSSS that sequence is known as Barker's code, which is simply a sequence of 11 bits (10110111000) that has some mathematical properties that makes it ideal to modulate radio waves. The basic data flow goes through a block that does the XOR operation with the Barker code to generate a series of objects known as chips. Each bit is coded by the 11 bits of the Barker code, and each 11 chips group codes a data bit.

To transmit at 1 Mbps it makes use of BPSK (*Binary Phase Shift Keying*) modulation with a phase change per bit. To achieve transmission at 2 Mbps it uses QPSK (*Quadrature Phase Shift Keying*) modulation. QPSK uses four rotations (0, 90, 180 and 270 degrees) to code two information bits on the same space where BPSK codes only one. We therefore have a trade-off between power and range. Notice that it is not allowed for mobile radios to transmit wirelessly with more that 1 Watt EIRP (*Equivalent Isotropic Radiated Power*) in the USA and with more than 100 mW in Europe. Therefore, as nodes separate, the radio adapts itself by using a slower and less complex mechanism to send data.

IEEE 802.11a The IEEE 802.11a technology is a physical layer annex to IEEE 802.11 for operating on the 5 GHz radio frequency. It supports several different data rates between 6 and 54 Mbit/s. Since the 5 GHz band is used it suffers from less RF interferences that those physical layers that operate on the 2.4 GHz band (e.g. IEEE 802.11b and 802.11g) since the penetration capability of radio waves at the 5 GHz frequency is lower. So, by offering high data rates and low interference, the IEEE 802.11a technology allows achieving good results supporting multimedia applications in environments with several users. The only drawback is that more access points are required to cover a similar area than for IEEE 802.11b or 802.11g.

Concerning modulation, it uses Orthogonal Frequency Division Multiplexing (OFDM) with 52 sub-carriers.

IEEE 802.11b The IEEE 802.11b specification enhances the IEEE 802.11 physical layer to achieve higher data rates on the 2.4 GHz band. It uses another modulation technique known as *Complementary Code Keying* (CCK) to achieve 5.5 and 11 Mbps. Instead of using the Barker code, CCK uses a series of codes known as Complementary Sequences. The CCK technique uses 64 unique codewords that can be used to code the signal, being that a particular codeword can encode 4 or 8 bits (instead of a single bit as represented through a Barker symbol).

The final solution consists of combining the DSSS (*Direct Sequence Spread Spectrum*) techniques based on CCK with DQPSK modulation, which is the key for achieving data rates of 5.5 and 11 Mbit/s.

IEEE 802.11g The IEEE 802.11g is the most recent specification available for IEEE 802.11's physical layer. It results of merging IEEE 802.11a with IEEE 802.11b, which allows the IEEE 802.11g technology to achieve data rates similar to IEEE 802.11a (54 Mbit/s). The main advantage of IEEE 802.11g is that it maintains compatibility with more than 11 million Wi-Fi products (IEEE 802.11b) already sold.

The IEEE 802.11g standard defines several modulation types:

- ERP-DSSS: refers to physical layers using Direct Sequence Spread Spectrum (DSSS) modulation, and exists to ensure compatibility with the original IEEE 802.11 standard.

- ERP-CCK: refers to physical layers using Complementary Code Keying (CCK) modulation as defined for IEEE 802.11b, and exists also to ensure compatibility with that standard.

- ERP-PBCC: refers to physical layers using extended rate Packet Binary Convolutional Coding (PBCC) modulation. PBCC was added as an option in the IEEE 802.11b. In 802.11g, the ERP-PBCC option also supports data rates of 22 and 33 Mbps.

- ERP-OFDM: refers to physical layers using Orthogonal Frequency Division Multiplexing (OFDM) modulation. OFDM was defined in the IEEE 802.11a supplement. In IEEE 802.11g, ERP-OFDM supports data rates of 6, 9, 12, 18, 24, 36, 48, and 54 Mbps.

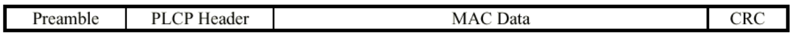


Figure 1.3: PLCP framing

- DSSS-OFDM: refers to physical layers using hybrid DSSS-OFDM modulation. DSSS-OFDM was added in the IEEE 802.11g standard and is an optional mode that does not use the Extended Rate PHY (ERP) protection mechanism. Instead, DSSS-OFDM combines the DSSS preamble and header with the OFDM payload, supporting rates similar to ERP-OFDM.

IEEE 802.11n In January 2004 IEEE announced that it had formed a new 802.11 Task Group (TGn) to develop a new amendment to the IEEE 802.11 standard for local-area wireless networks. The real data throughput is estimated to reach a theoretical 540 Mbit/s (which may require an even higher raw data rate at the physical layer), and should be up to 10 times faster than IEEE 802.11a or 802.11g, and near 40 times faster than IEEE 802.11b. It is projected that IEEE 802.11n will also offer a better operating distance than current networks.

IEEE 802.11n builds upon previous 802.11 standards by adding MIMO (multiple-input multiple-output) and orthogonal frequency-division multiplexing (OFDM). MIMO uses multiple transmitter and receiver antennas to allow for increased data throughput through spatial multiplexing and increased range.

1.2.2 IEEE 802.11 frame format

Though designed for total compatibility with existing IEEE 802.3 based networks, IEEE 802.11 frames differ from the former due to the requirements of the wireless medium itself.

The physical level for IEEE 802.11 is divided into two sub-layers: the Physical Layer Convergence Procedure (PLCP) and the Physical Medium Dependent (PMD) sub-layer. While the PMD sub-layer is the one responsible for actually transmitting the bits on the channel, the PLCP sub-layer works as the interface between the MAC layer and the radio transmission itself.

The PLCP sub-layer also adds its own headers to transmitted frames. Figure 1.3 shows the generic PLCP framing structure.

The preamble depends on the physical level and includes a synchronization sequence of 80 bits, apart from a frame delimiter of 16 bits whose sequence is: 0000 1100 1011 1101. Concerning the PLCP field, it is always transmitted at 1 Mbps for IEEE 802.11 and contains information that allows decoding the frame at the physical level, consisting of:

- PLCP_PDU length word, which contains the size of the packet in bytes.

- PLCP signaling field, which contains information relative to the transmission's bandwidth.

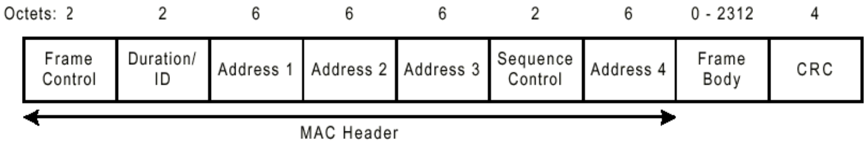


Figure 1.4: MAC layer frame format for IEEE 802.11

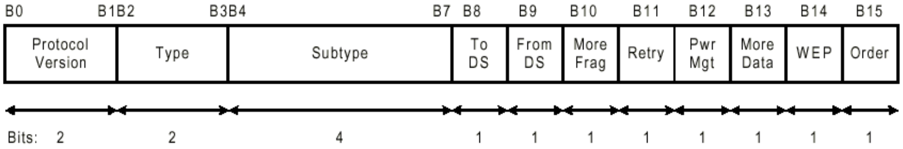


Figure 1.5: Information contained on the frame control field

- Header Error Check Field, that is a 16 bits CRC field used to detect errors on the frame's header.

We now proceed to analyze MAC layer frames in detail. An IEEE 802.11 MAC frame has the format shown on figure 1.4.

IEEE 802.11's MAC layer adds both a header and a trailer - the final 32 bits CRC field designed to detect errors on data.

The frame control field contains a large amount of data, as shown in figure 1.5, relative to the protocol version, the frame's type and subtype, as well as other flags with different purposes.

The *Duration/ID* field's content depends on the operation mode, being either the destination station identifier on power save mode or a duration used on the calculation of NAV (*Network Allocation Vector*).

The four addresses included are used to increase the flexibility, so that address 1 always refers to the destination and address 2 to the source; addresses 3 and 4 identify source and destination on those situations where one or more access points are used for communication among two wireless nodes. Table 1.1 resumes the contents of each of the fields depending on *To DS* and *From DS* bits. Notice that SA and DA refer to the actual source and destination addresses, while BSSID is the address of the access point involved in the frame relaying process. When two access points must communicate wirelessly, their addresses are refferred as RA and TA (for receiver and transmitter respectively), being SA and DA the actual source and destination of the data.

As for the sequence control field, it is used for both defragmentation and to discard duplicate frames. It consists of two fields - fragment number and sequence number - that define the frame and the fragment within the frame, respectively.

Table 1.1: Meaning of the contents of the four address fields depending on the value of the *To DS* and *From DS* bits.

Bit To DS	Bit From DS	Direction 1	Direction 2	Direction 3	Direction 4
0	0	DA	SA	BSSID	N/A
0	1	DA	BSSID	SA	N/A
1	0	BSSID	SA	DA	N/A
1	1	RA	TA	DA	SA

1.2.3 Distributed Coordination Function (DCF): CSMA/CA

The CSMA/CA access mechanism (*Carrier Sense Multiple Access/Collision Avoidance*) used by most Wireless LANs is the core of the protocol, specifying when a station must listen and when it should transmit. The basic principles of CSMA/CA are therefore: to listen before transmitting and sending messages asynchronously and without connection. There are neither bandwidth nor latency guarantees.

The CSMA/CA mechanism derives from the CSMA/CD (*Collision Detection*) mechanism used on Ethernets. The only difference is that it avoids collisions (see figure 1.6). The CSMA/CA protocol starts by listening to the channel (to this we call *carrier sensing*) and, if it is free, the first packet on the transmission queue is sent. If it is occupied (due to the transmission of other nodes or to interference) the node waits until the current transmission finishes, and it enters a contention period (waits for a random amount of time). When the contention timer expires the node sends the packet if the channel is still free. The node choosing the lowest contention time wins and transmits the packet. The remaining nodes merely wait for the following contention period to take place, which occurs when the packet transmission ends. Since contention depends on a random number and is done per packet, all the nodes have the same chances of accessing the channel (on average). Moreover, collisions can not be detected once transmission begins; since the radio requires some time to switch from transmission to reception mode, the contention mechanism makes use of time slots. So, transmission can only take place at the beginning of a slot, whose duration is 50 ms for 802.11 FH (Frequency Hopping) and 20 ms for 802.11 DS (Direct Sequence). This causes the contention time to increase, but reduces collisions significatively (it is impossible to eliminate them completely).

1.2.4 Point Coordination Function (PCF)

Apart from DCF, there is another operation mode known as PCF that can be used to support services with temporal restrictions, such as video and voice. This is possible since PCF operation is related to a smaller interval between frames (PIFS), so that access to the medium is obtained before any regular node using DCF. By making use of this higher priority, an access point can poll the different stations associated with it, thereby controlling access to the medium.

Though an access point has priority when accessing the medium, it must allow,

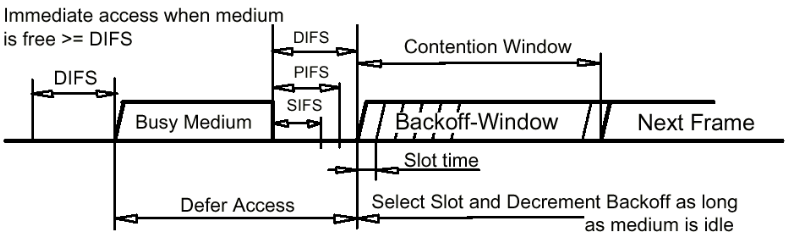


Figure 1.6: The CSMA/CA mechanism

between polling processes, nodes to send frames using distributed channel access techniques. Otherwise, nodes using such techniques would never have chances of transmitting, suffering from starvation.

1.2.5 MAC-level retransmissions

As referred before, the main problem of the CSMA/CA protocol is that the transmitter can not detect collisions on the wireless medium. Moreover, there is a greater error rate on the air than on a cable, so the chances for a packet to become corrupt are higher. TCP reacts poorly towards MAC level packet losses; so, most MAC layer protocols implement acknowledgment and retransmission functionality to avoid this problem. The principle is quite simple: each time a node receives a packet it immediately sends a short acknowledgment message (ACK) back to the source to indicate that the packet has been received without errors. If, after sending a packet, the source does not receive and ACK, it assumes that the packet has been lost and retransmits after a new contention period. Most MAC layer protocols use a *Stop & Wait* mechanism through which they only transmit the next packet on queue after the arrival of the current packet is confirmed (it retries up to a certain number of times). This technique simplifies the protocol and avoids that packets arrive unordered.

ACK messages are part of the MAC protocol, so it respects their delivery to avoid collisions - contention starts after the ACK packet is transmitted. These acknowledgments are totally different from those used in TCP since they operate at a different protocol layer.

Broadcast and *multicast* packets do not benefit from ACK frames since there is no specific destination, so they will frequently be lost.

1.2.6 RTS/CTS

The main effect to take into consideration when transmitting a radio wave is signal attenuation. It is due to attenuation that the hidden terminal problem is likely to occur. This problem takes place on those situations where not all the nodes can listen to the rest since the signal attenuation is too strong. Taking into account that transmissions are based on a carrier sensing mechanism, nodes ignore those that are far away and so can transmit at the same time. Generally this is good since it allows frequency reuse. However, for a node located between two other nodes, simultaneous transmissions have a similar power and so they collide. From the point of view of that intermediate node all the information transmitted is lost.

The main problem with the carrier sensing mechanism is that the transmitter tries to estimate if the channel is available on the receiver based solely on local information, which results in errors. A simple and elegant solution (proposed by Phil Karn on his MACA protocol [Phi90] for AX.25) is using RTS/CTS (*Request To Send/Clear To Send*). RTS/CTS is a handshake mechanism: before transmitting a packet the transmitter sends a RTS and waits for a CTS from the destination node. The reception of a CTS indicates that the receptor has received an RTS, which means that the channel is available on that area. At the same time, each node on the receiver's transmission range listens to the CTS (despite possibly not listening to the RTS), deducing that a transmission is going to take place. The nodes that listen to the CTS are nodes that potentially could provoke collisions on the receiver, supposing that the channel is symmetric. Since these nodes can not listen to the transmission, both RTS and CTS packets contain the expected duration of such transmission. This characteristic of RTS/CTS is the one that allows avoiding collisions: all the nodes avoid accessing the channel after listening to the CTS even if the carrier sensing mechanism indicates that the medium is free.

The RTS/CTS mechanism has another advantage: it reduces a collision's overhead on the medium (collisions are shorter). If two nodes try to transmit on a same slot of their contention windows, their RTS will collide and no CTS is received; that way all that is lost is a simple RTS. Without this mechanism an entire packet would be lost. Since the RTS/CTS mechanism adds a significant overhead it is not used with small packets or in lightly loaded networks.

1.2.7 IEEE 802.11e: MAC enhancements for QoS

The IEEE 802.11e working group is extending the IEEE 802.11 MAC in order to provide QoS support. This new standard introduces the *hybrid coordination function* (HCF) which defines two new medium access mechanisms to replace PCF and DCF. These are the *HCF controlled channel access* (HCCA) and the *enhanced distributed channel access* (EDCA).

With the HCF there may still exist a contention period and a contention-free period in a superframe, but now the HCCA is used in both periods, while the EDCA is used only during the CP. This new characteristic of HCF obviates the need for a CFP since it no longer depends on it to provide QoS guarantees.

With IEEE 802.11e, the *point coordinator* is replaced by a *hybrid coordinator* (HC) which also resides in an AP. A BSS including a HC is referred to as a QBSS. In this paper we focus on ad-hoc networks and, therefore, we are only interested in IEEE 802.11e stations implementing EDCA. For more information on HCs, the HCF and the HCCA please refer to [IEE05].

Concerning IEEE 802.11e enabled stations forming an ad-hoc network, these must implement the EDCA. The IEEE 802.11e QoS support is achieved through the introduction of different *access categories* (ACs), and their associated backoff entities.

Table 1.2: User Priority to IEEE 802.11e Access Category Mapping (according to IEEE 802.1D)

User Priority	Designation	Access Category	Common designation
1	BK (Background)	AC_BK	Background
2	BK (Background)	AC_BK	Background
0	BE (Best-effort)	AC_BE	Best effort
3	EE (Video/Excellent-effort)	AC_BE	Best effort
4	CL (Video/Controlled Load)	AC_VI	Video
5	VI (Video)	AC_VI	Video
6	VO (Voice)	AC_VO	Voice
7	NC (Network Control)	AC_VO	Voice

In table 1.2 we can see the mapping between different user priorities and the different access categories available in IEEE 802.11e stations.

Contrarily to the legacy IEEE 802.11 stations, where all MSDUs have the same priority and are assigned to a single backoff entity, IEEE 802.11e stations have four backoff entities (one for each AC) so that packets are sorted according to their priority. Each backoff entity has an independent packet queue assigned to it, as well as a different parameter set. In IEEE 802.11 legacy stations this parameter set was fixed, and so the inter-frame space was set to DIFS and CWmin and CWmax where set to 15 and 1023 respectively (for IEEE 802.11a and 802.11g). With IEEE 802.11e the inter-frame space is arbitrary and depends on the access category itself (AIFS[AC]). We also have AC-dependent minimum and maximum values of the contention window (CWmin[AC] and CWmax[AC]). Moreover, IEEE 802.11e introduces an important new feature referred to as transmission opportunity (TXOP). A TXOP is defined by a start time and a duration; during this time interval a station can deliver multiple MPDUs consecutively without contention with other stations. This mechanism, also known as *contention-free bursting* (CFB), increases global throughput through a higher channel occupation. An EDCA-TXOP (in contrast to an HCCA-TXOP) is limited by the value of TXOPLimit, which is a parameter defined for the entire QBSS and that also depends on the AC (TXOPLimit[AC]).

Table 1.3 presents the default MAC parameter values for the different ACs [IEE05]. Notice that smaller values for the AIFSN, CWmin and CWmax parameters result in a higher priority when accessing the channel; relative to the TXOPLimit, higher values result in larger shares of capacity and, therefore, higher priority.

The relationship between AIFS[AC] and AIFSN[AC], is the following:

$AIFS[AC] = SIFS + AIFSN[AC] \times aSlotTime$, $AIFSN[AC] \geq 2$, where $SIFS$ is the shortest inter-frame space possible and *aSlotTime* is the duration of a slot. AIFSN[AC] should never be less than 2 in order not to interfere with AP operation.

Table 1.3: IEEE 802.11e MAC parameter values for a IEEE 802.11a/g radio

Access category	AIFSN	CW_{min}	CW_{max}	TXOPLimit (ms)
AC_BK	7	15	1023	0
AC_BE	3	15	1023	0
AC_VI	2	7	15	3.008
AC_VO	2	3	7	1.504

Mapping QoS requirements to IEEE 802.11e parameters

QoS parameters are typically set at the application level depending on the requirements of a particular application. The Internet Protocol (IP) supports traffic differentiation mechanisms in the sense that it allows tagging the packets according to QoS requirements, so that successive network elements can treat them adequately. This is achieved using the 8 bits of the "Type of service" field in an IPv4 datagram header or the "Traffic class" field in an IPv6 datagram header. It is the 3 TOS bits, part of both "Type of service" (IPv4) or "Traffic class" (IPv6) fields, that are used to indicate the desired user priority at the IP level. These shall then be mapped to IEEE 802.11e ACs according to table 1.2.

The IEEE 802.11e draft [IEE05] states that stations that depend on IEEE 802.11e for communication are able to offer packets a differentiated treatment by negotiating with the IEEE 802.11e MAC Service Access Point. The IEEE 802.11e MAC Service Access Point (MAC_SAP) allows negotiating QoS specifications in two ways: either directly by setting a traffic category (TC), or indirectly by making a traffic specification (TSPEC) instead. It is the value of the user priority (UP) parameter which indicates to the MAC_SAP the desired choice using values in the range 0 through 15. Priority parameter values 0 through 7 are interpreted as actual user priority values according to table 1.2, and so outgoing MSDUs are therefore marked according to the correspondent access category. Priority parameter values 8 through 15 specify traffic stream identifiers (TSIDs), and allow selecting a TSPEC instead.

The value of the chosen user priority is mapped to transmitted packets by setting the QoS Control field, part of the IEEE 802.11e MAC header, accordingly. The QoS Control field is a 16-bit field that identifies the traffic category or traffic stream (TS) to which the frame belongs and various other QoS-related information about the frame that varies for the particular sender and by frame type and subtype. In particular, it is the TID field (part of the QoS Control field) the one that identifies the TC or TS of traffic for which a TXOP is being requested. The most significant bit of the TID field, when set to 0, indicates that the request is for data associated with prioritized QoS or, when set to 1, indicates that the request is for data associated with parameterized QoS. The remaining bits define the UP value or the TSID accordingly.

When receiving a packet, the IEEE 802.11e MAC analyzes the QoS Control field and also offers a differentiated treatment to packets with different QoS requirements when passing them to upper layers.

1.2.8 Network architecture

In this section we will describe the three possible network configurations available within the IEEE 802.11 framework. The three configurations are known as IBSS, BSS and ESS. We now proceed to detail the basic architecture for each of them.

- Independent Basic Service Set (IBSS)

An IBSS, also known as an ad hoc network, is an short duration IEEE 802.11 network established from a mesh of mobile stations without any sort of infrastructure for support. On IBSS networks each user has to be within the destination's range for communication to be feasible, unless these and other nodes are acting as routers.

- Basic Service Set (BSS)

BSS-based networks, also known as infrastructure networks, are formed around an access point that typically has a wired connection with a network of greater dimensions. With this configuration each mobile node communicates directly with the access point, being the latter responsible for the communication of all nodes that registered with it. On this mode of operation, and independently of the distance between mobile stations, all the communication must pass through the access point.

- Extended Service Set (ESS)

ESS networks are characterized by the existence of multiple access points whose coverage area partially overlaps. The distribution services for the access points integrating an ESS include cooperation techniques for the interchange of frames among them. This way we achieve communication among mobile terminals associated with different access points on an ESS in an entirely transparent manner, as if it was all a single large subnet. Access points also allow dynamic association of stations, so that terminals can roam between access points belonging to a single ESS.

Figure 1.7 shows an example where we can distinguish the BSS and ESS concepts, evidencing the coverage of each of the access points and the required overlapping in terms of coverage to make roaming possible.

1.3 Bluetooth

Bluetooth [Blu02] is a standard created by the industry designed to allow simple interconnection between laptops, PDAs, mobile phones and other devices at short distances (10 meters maximum). Bluetooth uses fast frequency hops at 1600 hops per second on the 2.4 GHz band at a rate of 1 Mbit/s on version 1.1 and 10 Mbit/s

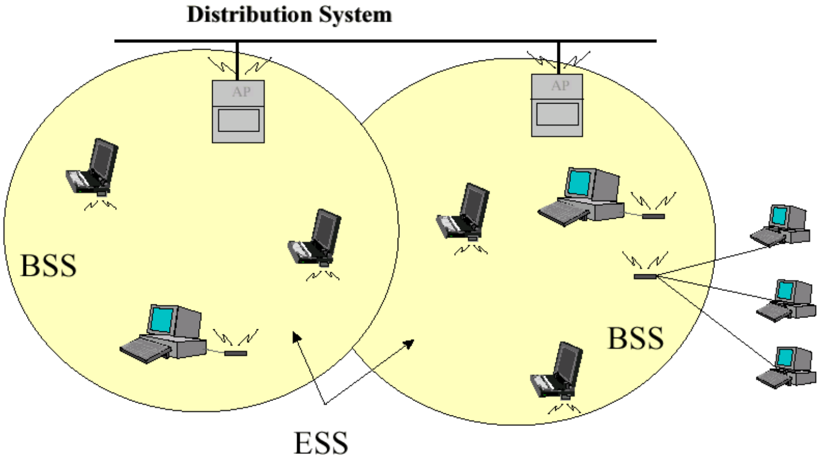


Figure 1.7: Example of a distribution ESS aggregating two BSSs

on version 1.2. The transmission power is limited to 1 mW, and access to the medium is done using a *Frequency Hopping Spread Spectrum* (FHSS) technique.

In the sections that follow we offer more information about this technology at different levels.

1.3.1 Specification

The Bluetooth specification defines all the necessary requirements to assure the interoperability among devices of this family. The specification is divided in two main parts: radio and protocol definitions (I) and requirements for interoperability (II).

Figure 1.8 presents the Bluetooth protocol stack defined in the specification.

The Radio layer handles all the issues related to the transmission and reception of modulated signals. The Baseband protocol defines temporization, frames and flow control on the channel. The Link Manager is responsible for managing the connection states, fair sharing among slaves, power management, as well as other management tasks. The logical link control layer multiplexes higher level protocols, being also responsible for the fragmentation and de-fragmentation of large packets and for device discovery tasks. Audio data are sent directly to the baseband, though audio control is done over the logical link control layer also. Above the LLC layer, both the RFCOMM layer and network level protocols offer different abstractions to communication. The RFCOMM layer offers the possibility of emulating a serial cable by using part of the *ETSI GSM 07.10* standard [GSM97]. Other parts of the Bluetooth specification handle the interoperability towards other protocols or protocol stacks. The use of TCP/IP over Bluetooth requires solving the problems associated with bridging, address resolution, MTU definition, multicast and broadcast.

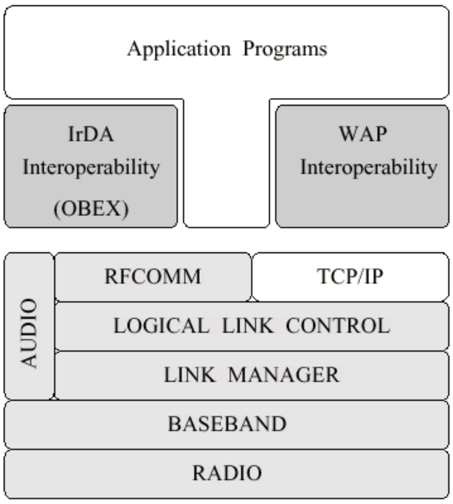


Figure 1.8: Bluetooth protocol stack

The second part of the specification defines those aspects related to interoperability. Due to the large variety of possible Bluetooth devices, different sets of requirements are necessary. For instance, the minimum requirements for headphones shall be different from those of a laptop. The purpose of the interoperability section is assuring that any device showing the Bluetooth logo is sure to offer a minimum set of benefits to the final user.

1.3.2 Architecture

Bluetooth has been developed and designed with the purpose of achieving a low-cost and robust communication system. Its implementation is based on a high performance radio transceiver, though cheap. Bluetooth aims at mobile users that need to establish a connection, or small network, using a computer, a mobile phone or other devices. The required and nominal range for the Bluetooth radio is of 10 meters (with an output power of 0 dBm). For other sort of applications (e.g. a home environment) Bluetooth devices can be enhanced with an external signal amplifier that allows increasing its range (up to 100 meters with an output power of 20 dBm). It is also possible to add hardware to support, for example, four extra voice channels or more. These extensions are totally compatible with the specification, and so their adequateness depends on the target application itself.

When two Bluetooth devices are within range they can start an ad hoc connection called *piconet*. Each *piconet* is formed by up to eight different units where there must always be a device operating as Master; the rest of the *piconet* members act as slaves. The unit establishing the *piconet* is the one responsible for acting as a Master, though this may change so that there is only a single device acting as Master instead of several.

When two or more piconets coexist in a same area we say that a *scatternet* is formed. On a *scatternet* all the units share the same frequency range, but each *piconet* uses different hop sequences and so transmits on different 1 MHz channels. Since all the piconets share a 80 MHz band, there won't be channel interference as long as different hop sequences are used.

Bluetooth devices operate on the international band of 2.4 GHz with a theoretical bandwidth of 1 Mbit/s and with a low power consumption, so that it can be used on battery operated devices. With the *scatternet* technology it is possible to achieve an aggregated bandwidth value of 10 Mbit/s or 20 voice channels instead. The structure also enables a range extension at the radio level simply by adding Bluetooth units working as bridges on strategic positions.

A single unit can support a data rate up to 721 kbit/s or a maximum of 3 voice channels. A mix of data and voice is also possible to support multimedia applications. Concerning voice coding, this process uses a quite robust scheme whose bandwidth is of 64 kbit/s. We should point out that Bluetooth offers a mechanism known as *graceful degradation* when operating on congested radio environments.

1.3.3 Establishment of network connections

When a connection is established for the first time, or when you must add components to a piconet, devices must be identified. These can connect and disconnect from the piconet at any time. The two available options for this process lead to connection times of 0.64 and 1.28 seconds on average. A device does not need to be always connected for a transaction to take place, on average, on less than a second. That way, when the unit is not being used, it can hibernate for most of the time (STANDBY), being a low power oscillator the only element active. This technique offers great power savings. Before connections are established all the units are on a *standby* state. On this mode a disconnected unit shall only listen to messages every 1.28 or 2.56 seconds, depending on the chosen option. Each time a unit becomes active it will listen to one of the 32 hop frequencies defined for that unit. The connection process is started by one of the units, the master. A connection is established with a PAGE message if the destination address is already known, or by an INQUIRY followed by a PAGE message if the address is unknown. On the initial PAGE state the unit performing the PAGE function (master) will send a sequence of 16 identical messages on 16 different hop frequencies defined for the destination unit (slave). This sequence covers half of the sequence of frequencies used by the slave when awakening. It is repeated 128 or 256 times (1.28 or 2.56 seconds) depending on the requirements of the slave unit. If no reply is received after this period, the master transmits an identical sequence of messages on the 16 remaining frequencies. The maximum delay achieved is therefore of 2.56 or 5.12 seconds. This technology offers a clear trade-off between power consumption and latency.

The INQUIRY message is typically used to find public printers, fax machines or similar equipment with an unknown address. This message is quite similar to the PAGE one, but it may require an additional search period to recollect all the answers. If no data requires being transmitted units can be configure to wait

(HOLD), situation where only the internal timer is active. When the units come out of this mode the data transmission may start immediately. The HOLD state is normally used to connect several piconets, being also used by units that require sporadic sending of data and where power savings are an important factor.

1.3.4 Service Discovery Protocol (SDP)

SDP defines how a client application using Bluetooth must act to find the services made available by Bluetooth servers, as well as their characteristics.

The protocol defines how a client can search for a service based on specific attributes, without the client knowing anything about the available services. The SDP provides means for the discovery of new services that become available when the client enters an area where a Bluetooth server is operating.

To activate a new service a server application must register with SDP; such information is added to the service record, which consists entirely of a list of service attributes. Based on that information the client can then proceed to establish a connection with the server, thereby making use of the service advertised.

The SDP also provides functionality for detecting when a service is no longer available.

1.3.5 Basic Bluetooth Profiles

To avoid different interpretations of the Bluetooth standard relatively to the interface between application and Bluetooth, the *Bluetooth Special Interest Group* defined certain user models and protocol profiles. A profile is merely a selection of messages and procedures obtained from Bluetooth specifications, offering a clear description of the interface used for the specified services. A profile can be described as a certain subset of the protocols defined in the Bluetooth stack. Moreover it defines, for each protocol, the options that are mandatory for the selected profile, as well as the parameter ranges that can be used with those protocols.

Four generic profiles have been defined, and the main user models are based on them. These four models are the *Generic Access Profile* (GAP), the *Service Discovery Application Profile* (SDAP), the *Serial Port Profile* (SPP) and the *Generic Object Exchange Profile* (GOEP).

- Generic Access Profile (GAP)

This profile defines how Bluetooth devices must find and establish a connection among them. GAP also assures that Bluetooth devices, independently of the manufacturer and the application they work with, can exchange information to find what sort of applications are supported by both. Conformity to this profile is essential to guarantee interoperability and coexistence.

- Service Discovery Application Profile (SDAP)

Is is expected that the number of services offered through Bluetooth channels increases at an undetermined and possibly uncontrolled manner. That way the required procedures for users to select one of the available services must be defined.

Though most of the services to be created are unknown at the moment, a standard procedure can be created to locate and identify them. The Bluetooth protocol stack offers the *Service Discovery Protocol* (SDP) which allows locating those services available on the vicinity.

- Serial Port Profile (SPP)

The Serial Port profile defines how to configure virtual serial ports on two devices and how to connect them through Bluetooth. This profile allows Bluetooth devices to emulate a serial cable with RS232 control signaling, assuring a bandwidth up to 128 kbit/s. This profile depends on the GAP profile.

- Generic Object Exchange Profile (GOEP)

This profile defines the protocols and procedures that will be used by applications, indicating also some models of use. These models can be, for example, *Synchronization*, *File Transfer* or *Object Push*. This profile depends on the SPP profile.

- Other profiles

Besides the profiles referred before there are others that, despite not being so important, are also relevant. Among these are the *Cordless Telephony Profile* that offers communication between phone terminals so that one terminal can use services of the other one, the *Intercom Profile* that allows phone devices to work as "walkie-talkies", the *Headset Profile* that allows the remote transmission of sound, the *Dial-up Networking Profile* that allows a device to use another to connect to a network (e.g. Internet), the *Fax Profile* that allows a device to offer fax functionality, the *LAN Access Profile* that allows accessing a LAN using PPP, the *Synchronization Profile* which defines the interoperability requirements between protocols, the *Exchange Profile* which allows performing the interchange of objects and doing management tasks, and also the *File Transfer Profile* that supports the transference of files.

1.3.6 Final considerations

Bluetooth was designed as an enabling technology for Wireless Personal Area Networks (WPANs). Therefore, the main focus was on how to design this technology so as to achieve inter-operation with a number of devices as large as possible. Obviously, this technology allows connecting two or more devices as if a cable was connecting them, which enables networking. Despite such fact, this technology was not designed to support mobile ad hoc networks with fast changing topologies. Two of the main impediments are the relatively high device discovery and device connection times. Another impediment is related to the organization of *scatternets* when there are a large number of nodes involved. As yet another drawback, we can also refer to the relatively low range achieved by most of the devices using this technology. These issues cause Bluetooth to be, currently, not the best technologic option to support mobile ad hoc networks.

In a preliminary work [CRP03] we studied the interaction of the Bluetooth and IEEE 802.11b technologies. Our purpose was to assess the most adequate manner

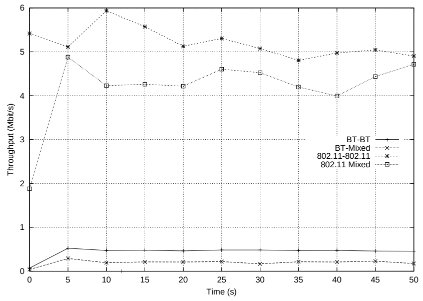


Figure 1.9: Mutual impact of the Bluetooth and IEEE 802.11 technologies in terms of throughput

for integrating Bluetooth terminals in IEEE 802.11-based MANETs. In that work we obtained interesting data concerning the mutual impact of the Bluetooth and IEEE 802.11 technologies in terms of measured TCP throughput. These results are shown on figure 1.9.

That figure shows that, when two IEEE 802.11-enabled devices are operating close-by without interference from Bluetooth devices the throughput achieved is around 5 Mbit/s. In a similar fashion, when two Bluetooth-enabled devices are operating close-by without interference from IEEE 802.11 devices the throughput achieved is around 0.5 Mbit/s. However, when both pairs of devices are close-by and transmitting at the same time there is a considerable mutual impact between them, being that the throughput between the two IEEE 802.11-enabled devices drops to about 4.2 Mbit/s and the throughput between the two Bluetooth-enabled devices drops to about 0.25 Mbit/s. These substantial performance drops experienced, especially in the Bluetooth case, lead to the conclusion that the segregation of both technologies is perhaps the most appropriate decision to take when performance is at premium.

1.4 Conclusions

In this chapter we have offered an overview of the most widely deployed wireless technologies currently available for free personal use: IEEE 802.11 and Bluetooth. We presented some architectural details of these technologies, including information about their physical layers and their basic architecture. By analyzing their characteristics and performance we reach the conclusion that currently the IEEE 802.11 technology is the most adequate choice for wireless mobile ad hoc networks, which is the area we focus on this thesis. Such networks will be the topic of the next chapter.

Chapter 2

Wireless Mobile Ad hoc Networks

In this chapter we will review the state-of-the-art in wireless mobile ad hoc networks. We start by explaining the ad hoc network concept, referring the main characteristics of these networks and their fields of application.

We then focus on some of the most important routing protocols developed for wireless ad hoc networks, evidencing their characteristics in terms of responsiveness to mobility and of route maintenance. Our analysis also includes a background study of multipath routing protocols since this will be the topic of chapter 6.

We conclude this chapter by analyzing previous proposals for Quality of Service (QoS) support in ad hoc networks, a topic we will focus in chapters 7 and 8.

2.1 History, definition, characteristics and applications

The history of wireless networks dates from the late 70s and interest has been growing ever since. Towards the end of the last decade, interest reached a peak mainly due to the fast growth of the Internet. Recent developments are centered around infrastructure-less wireless networks, more commonly known as ad hoc networks. The term ad hoc, despite sometimes having negative overtones and being equated with improvised or not organized, is used in this context to express a higher level of flexibility. All nodes within an ad hoc network provide a peer-level multi-hopping routing service to allow out-of-range nodes to be connected. Unlike a wired network, nodes in an ad hoc network can move, thus giving rise to frequent topology changes.

Such a network may operate in a stand-alone fashion or be connected to the larger Internet. An ad hoc architecture has many benefits, such as self-reconfiguration and adaptability to highly variable characteristics such as power and transmission conditions, traffic distribution variations, and load balancing. However, such benefits come with many challenges. New algorithms, protocols, and middleware have to be designed and developed to create a truly flexible and decentralized network. Protocols should be adaptable, that is, they should learn and anticipate the behavior of the network using parameters such as level of congestion, error rate and topology change rate. Resources and services have to be located and used automatically, without the need for manual configuration. Access and authentication issues should also be considered to ensure security and user privacy. Finally, Quality of Service (QoS) technologies and should be introduced to provide guarantees to specific time-constrained traffic sources (e.g., voice, video, etc.).

In terms of applications, ad hoc networks offer the required flexibility to adapt to situations were no sort of infrastructure is possible. Examples of such situations are army units moving inside hostile territories or organized teams such as firemen performing rescue tasks. Ad hoc networks are an optimal solution for these situations since communication among peers is achieved in a simple and straightforward manner.

There are other fields of application, such as context-aware environments, where ad hoc networks are also a possible solution. Context-aware applications necessarily require some kind of mobile wireless communication technology. This mobile wireless technology will interconnect computing devices together with various sensing technologies such as motion sensors or electronic tags, setting up a new kind of intelligent environment in which context-aware applications can search for and use services in a transparent way without user intervention.

In general, mobile ad hoc networks can be used on all those situations characterized by lack of fixed infrastructure, peer-to-peer communication and mobility support.

2.2 Classification of routing protocols

A routing protocol is required when a packet must go through several hops to reach its destination. It is responsible for finding a route for the packet and making sure it is forwarded through the appropiate path. Routing protocols based on algorithms such as *distance vector* (e.g. RIP [G. 98]) or *link-state* (e.g. OSPF [J. 98]) were available before solutions were sought in the field of wireless ad hoc networks. These routing protocols generate periodic control messages, a procedure that is not adequate for a large network with long routes since it would result in a large number of control messages. Moreover, the period between messages would have to be reduced in the presence of mobility. This effect is critical for mobile nodes where CPU use, as well as radio transmissions and receptions, would cause batteries to be quickly depleted. Also, all the conventional routing protocols assume bidirectional routes with a similar quality, something that is not always true on some kinds of networks (e.g. wireless ad hoc networks). Routing protocols can be classified according to three different criteria:

- Centralized or distributed: when a routing protocol is centralized, all the decisions take place at a central node. However, with a distributed routing protocol, all the nodes share the routing decisions.

- Adaptive or static: an adaptive routing protocol can change its behavior according to the network state, which can be the congestion on a certain connection or other possible factors, contrarily to a static one.

- Reactive, proactive or hybrid: a reactive routing protocol must act to find routes when necessary, while a proactive routing protocol finds routes before these are required. Reactive routing protocols are also known as *on-demand* routing protocols. Since these are executed on-demand, the control packets' overhead is considerably reduced. Proactive methods maintain routing tables, being these periodically updated. Concerning hybrid methods, these use a combination of both reactive and proactive techniques to achieve a more balanced solution.

2.2.1 Basic routing techniques

Independently of how a routing protocol is classified according to those criteria, the routing techniques used can be divided into three families: *distance vector*, *link state* and *source routing*. We now detail the basic principles of each of these techniques.

Distance Vector This technique maintains a table for the communication taking place and employs diffusion (not flooding) for information exchange between neighbors. All the nodes must calculate the shortest path towards the destination using the routing information of their neighbors.

Link State The protocols based on this technique maintain a routing table with the full topology. The topology is built by finding the shortest path in terms of link cost, cost that is periodically exchanged among all the nodes through a flooding technique. Each node updates its routing table by using information gathered about link costs. This technique is prone to cause loops on networks with a fast changing topology.

Source Routing Technique where all the data packets have the routing information on their headers. The route decision is made on the source node. This technique avoids loops entirely, though the protocol overhead is quite significant. This technique can be inefficient for fast moving topologies due to route invalidation along the path of a packet.

2.3 Routing in ad hoc networks

An ideal routing protocol for ad hoc networks must have certain properties that make it different from the rest. To begin with, it must be distributed to increase reliability: when all the nodes are mobile, it makes no sense to have a centralized routing protocol. Each node must have enough capabilities to take routing-related decisions with the aid of the rest of the nodes.

Also, a routing protocol should assume that the links detected are unidirectional connections. On a wireless channel a unidirectional connection may be formed due to physical factors, so that bidirectional communication may result impossible. Therefore, a routing protocol should be designed to take into account this possibility.

It is also important that an ad hoc routing protocol takes into account issues such as power consumption and security. Obviously, mobile nodes depend on batteries. This means that a protocol that minimizes the total power consumption of network nodes would be ideal. Concerning security, you must take into account that the wireless medium is very vulnerable. At the physical level, DoS attacks can be avoided by using frequency hopping or code-based *Spread Spectrum* techniques. At the routing level, though, both the authentication of neighbors and the encryption of data are required.

2.3.1 Routing protocol families for ad hoc networks

In section 2.1 we introduced the most important characteristics of an ad hoc network. Concerning the routing protocols used on these networks they should be, according to the classification of section 2.2, both distributed and adaptive.

Relatively to the third category (reactive/proactive/hybrid), there is no consensus over which is the most adequate strategy. Below we present the different proposals that are currently available for each of these protocol families, and we also include other non-cataloged proposals.

Proactive routing protocols

The concept of proactive routing means that all the nodes (routers) periodically interchange routing information (or upon detecting topology changes) with the aim of maintaining a consistent, updated and complete view of the network. Each node uses the exchanged information to calculate the costs towards all possible destinations. That way, if a destination is found, there will always be a route available; this avoids delays associated with finding routes on-demand.

Proactive techniques typically use algorithms such as *distance vector* or *link-state*. Both techniques require routers to periodically broadcast information and, based on that information, to calculate the shortest path towards the rest of the nodes.

The main advantage of proactive routing schemes is that there is no initial delay when a route is required. On the other hand, these are usually related to a greater overhead and a larger convergence time than for reactive routing techniques, especially when mobility is high. To increase the performance in ad hoc networks both *link-state* and *distance vector* algorithms were modified. Examples of routing protocols using *distance vector* techniques are the *Destination-Sequenced Distance Vector* (DSDV) [CP94] and the *Wireless Routing Protocol* (WRP) [MGLA96]. Examples of *link-state* based protocols are the *Open Shortest Path First* (OSPF) [J. 98], the *Optimized Link State Routing* (OLSR) [TPA$^+$01], the *Topology Broadcast Reverse Path Forwarding* (TBRPF) [BR99], the *Source Tree Adaptive Routing* (STAR) [GLAS99], the *Global State Routing* (GSR) [CQS98], the *Fisheye*

State Routing (FSR) [PGC00] and the *Landmark Routing Protocol* (LANMAR) [PGH00].

Reactive routing protocols

Reactive routing does not depend, in general, of periodic exchange of routing information or route calculation. Therefore, when a route is required, the node must start a route discovery process. This means that it must disseminate the route request throughout the network and wait for an answer before it can proceed to send packets to the destination. The route is maintained until the destination is unreachable or until the route is no longer necessary. By following this strategy reactive routing protocols keep to a minimum the resource consumption by avoiding the maintenance of unused routes. On the other hand, the route discovery process causes a significant startup delay and causes a considerable waste of resources. If the network is wide enough, the overhead will be similar or superior to that achieved with proactive routing protocols.

The most common routing algorithms found among reactive routing protocols are *distance vector* and *source routing*. Example of reactive routing protocols are the *Ad-hoc On-demand Distance Vector* (AODV) [PR99], the *Dynamic Source Routing* (DSR) [DDY04], the *Associativity Based Routing* (ABR) [C. 97], the *Signal Stability based Adaptive routing* (SSA) [DRWT96], the *Temporally Ordered Routing Algorithm* (TORA) [VS00] and the *Relative Distance Micro-discovery Adhoc Routing* (RDMAR) [AT99].

Other strategies

There are other strategies proposed for the design of routing protocols. There are, for instance, hybrid solutions such as the *Zone Routing Protocol* (ZRP) [ZM99] which uses both reactive and proactive concepts: each node maintains a zone (on a radius of 2 hops) where it employs proactive routing; to access nodes outsize that zone it uses reactive routing. The overhead is limited since the maintenance of proactive routes is only performed with neighbor nodes, and the reactive search for routes is limited to the communication within the selected subset of nodes.

It is common that the routing protocol has a flat architecture, though there are some protocols based on *clustering* and hierarchical architectures, such as the *Clusterhead Gateway Switch Routing* (CGSR) [Chi97], the *Distributed Mobility-Adaptive Clustering* (DMAC) [Bas99] and the *Cluster-based Energy Saving Algorithm* (CERA) [JDP03]. The advantage of these solutions is mainly the discovery of more robust routes with fewer control messages, though periodic messages may be required for the maintenance of clusters. A clear disadvantage is the centralization of routes through cluster leaders, which provokes congestion and also single points of failure which can cause large recovery periods.

The procedure of route discovery for reactive routing protocols such as AODV and DSR is based on a variant of the flooding technique, being typically quite resource consuming. The LAR protocol [YN98] tries to avoid this problem by using GPS information so that only those nodes on a certain geographic area between source and destination must retransmit route requests.

Another way of reducing the resource consuming process of finding routes is to have several additional paths available as backup, thereby reducing the need for a new route discovery when a link breaks. The DSR protocol implements this behavior.

Reducing the control overhead achieves, in general, improved scalability and reduced power consumption. There are several techniques that intend to improve the power consumption. PAR [SWR98] is a solution that takes into account the battery lifetime, selecting those routes that minimize the energy consumption of the system. Another solution is to reduce the transmission power, which also reduces interference and improves spacial reuse [M. 01]. Normally, this kind of protocols work in cooperation with the MAC layer protocol so that there is a per-packet energy assessment, as with the PARO [GCNB01] protocol. This technique also suffers from some problems, such as generating unidirectional routes and decreasing the transmission power, which increases the bit error rate and reduces the bit-rate under IEEE 802.11-based network environments.

2.3.2 The Optimized Link-State Routing Protocol (OLSR)

The Optimized Link State Routing protocol [TP03] is a proactive routing protocol specifically designed for Mobile Ad Hoc Networks (MANETs). It is based on the definition and use of dedicated nodes, called multipoint relays (MPRs). MPRs are selected nodes which are responsible for forwarding broadcast packets during the flooding process. This technique allows to reduce the packet overhead compared to a pure flooding mechanism where every node retransmits the packet when it receives the first copy of it. Contrarily to the classic link-state algorithm, partial link-state information is distributed throughout the network. This information is then used by the OLSR protocol for route calculation. The protocol is particularly suitable for large and dense networks as the technique of MPRs works well in this context.

Basic principles

The OLSR protocol inherits its stability from link-state algorithms. Due to its proactive nature, it offers the advantage that available routes can be used immediately.

Pure link-state algorithms declare and propagate the list of neighbors for each node throughout the network. OLSR tries to improve this solution by using different techniques. To start with, it reduces the size of control packets since it does not declare all of its neighbors, but only a subset of these referred as Multipoint Relay Selectors. A node's Multipoint Relay is in charge of retransmitting its broadcast messages. The use of MPRs serves the purpose of minimizing the amount of retransmissions upon a flooding or broadcast event.

Besides periodic control messages, the protocol does not generate additional control traffic in response to failures or association with new nodes. The protocol maintains routes towards all networks destinations, being useful in those situations where a great number of MANET nodes is communicating, especially when source/destination pairs are changing frequently. This protocol is more adequate

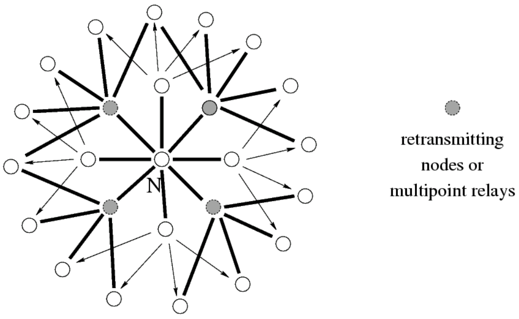


Figure 2.1: Illustration of the multipoint relay concept for node N

for large and dense networks, where the optimizations achieved by introducing Multipoint Relays offer important benefits.

The protocol is designed to operate in a distributed fashion, so it does not depend on a central entity. Moreover, it does not require reliable transmission of its control messages: each node sends periodic control messages, being tolerant to sporadic losses of control packets. Packet reordering, a frequent phenomena in ad hoc networks, will not cause OLSR to misbehave since each message carries a different sequence number.

The OLSR protocol uses per-node packet forwarding, which means that each node uses its most recent information to route a packet. That way, when a node is moving, its information is successfully transmitted as long as neighbor nodes can keep track of the mobile node. The ability to follow a node can be adjusted by setting the interval between consecutive control messages.

Multipoint Relays

The Multipoint Relay concept consists in trying to minimize the flooding caused by broadcast traffic by eliminating duplicated transmission on a same region. Each network node selects a subset of those nodes in its vicinity to retransmit its packets. Nodes belonging to this subset are a node's Multipoint Relays (MPRs). The neighbors not part of the MPR subset of a certain node N will still receive packets from it, but will not re-transmit them again. That way, each node maintains a table with the nodes which have selected it as their MPR.

Each node selects its own set of MPRs among their neighbors with a criteria that consists of assuring that all those nodes two hops away from it can be reached with a minimal number of MPRs. Figure 2.1 illustrates this concept.

OLSR trusts on the MPR node selection to calculate routes towards all the destinations having these as intermediate stations. This solution requires each node to periodically broadcast the list of neighbor nodes chosen as its MPRs. When receiving this information, each neighbor node updates the routes towards all known stations.

MPR nodes are selected among those neighbor nodes where bidirectional communication is feasible, which avoids attempting to transmit packets through unidirectional links.

Neighbor detection

Each node must detect those neighbor nodes towards which bidirectional communication exists. To achieve this purpose a node periodically broadcasts HELLO messages containing information about its neighbors and the state of the channel towards them. These messages are received by all neighbor nodes but not retransmitted.

These HELLO messages allow a node to discover those other nodes that are two hops away. It is based on this information that a node selects its MPR node set. The MPR set is indicated on HELLO messages through the MPR identifier; such information is used by other nodes to construct their MPR Selector table.

For adequate operation each node will then maintain a table with a list of all the nodes it can see either directly or indirectly. Links to one hop neighbors are tagged as either unidirectional, bidirectional or MPR. Each table entry has a both a sequence number and a timeout value associated, so that old entries can be removed.

Multipoint Relay selection

Each network node independently chooses its MPR set. This set is calculated among the direct neighbors so as to cover all the nodes two hops away. To maintain a list of the two-hop neighbors requires analyzing HELLO messages and filtering all the unidirectional links. The higher the degree of optimality of the MPR selection algorithm, the more benefits will it bring to all nodes.

The MPR set is only altered when a change is detected in terms of one-hop or two-hop neighbors (bidirectional connections only).

MPR information broadcasting

Each node must broadcast topology control messages (TC) in order for all nodes to maintain their database updated. These messages are broadcasted throughout the network using a technique similar to the one used for traditional link-state routing protocols, with the only difference that it employs MPRs to improve scalability.

A TC message is sent periodically to each network node to declare its MPR selector set. This means that the message must contain a list with those direct neighbors that have selected it as their MPR. This list always has a sequence number associated.

The list of addresses on each TC message can be partial, but it must be complete before each refresh period ends. These messages will allow each node to maintain its own table with the network topology. If a node has not been selected as any other node's MPR it does not send TC messages, thereby saving power and bandwidth.

The interval between the transmission of two TC messages depends on whether there have been changes on a node's MPR selector set. If so, the next TC message can be transmitted before the time scheduled, though respecting the minimum inter-message time.

Calculation of the routing table

Each node maintains a routing table with information on how to access other network terminals. When nodes receive a TC message they store sets of two addresses indicating the last hop before reaching a certain destination node, as well as the destination node itself. By combining the information in these address pairs the node is able to find what is the next hop towards a certain destination node. Minimum distance criteria should be followed to restrict the search options.

Routing table entries are composed of destination, next hop and estimated distance to destination. On this table we only register those entries for which the route towards destination is known. This means that the routing table must be constantly updated according to the topology changes detected.

In a real implementation the OLSR daemon must update the kernel's forwarding table according to the routing table it maintains, so that packets are sent through valid routes.

2.3.3 Ad hoc On-Demand Distance Vector Routing Protocol (AODV)

The Ad-hoc On-demand Distance Vector (AODV) is a reactive routing protocol that, as the name indicates, uses distance-vector algorithms to create and maintain routes.

An important advantage of AODV is that it generates no extra traffic for communication along existing routes. Also, distance vector routing is simple and doesn't require much memory or calculation. However, AODV requires more time to establish a connection and, besides, the initial process required to establish a route provokes more routing overhead than proactive approaches.

Route discovery

AODV only finds routes to a destination on demand, which means that no routing packet flows through the network until a connection is needed. When a route is required the source node starts a route discovery process by broadcasting a route request packet (RREQ) that is re-broadcasted by intermediate nodes until the destination is reached. During this process each intermediate node makes sure that it re-broadcasts a RREQ packet only once for the same route discovery action; this is possible because each request for a route has a sequence number associated. Intermediate nodes also update an internal table where they keep temporary route information about how to reach the source node. Once RREQ packets reach the destination a route reply (RREP) message is sent back to the source, and it can then determine which of the available routes offers the least number of hops.

Management of the routing table

To avoid keeping uncertain routes, the routes formed during each route discovery process are assigned a timer. Routes that are not used are invalidated when this timer is triggered.

Routes towards neighbors nodes are also maintained. A neighbor node is considered active if it originates or forwards at least one packet within the most recent *active_ timeout* period. This information is used to notify neighbors when a link of a path being used breaks, thereby stopping traffic if the path to the destination becomes unavailable.

The contents of AODV's routing table are the following: destination, next hop, number of hops, sequence number for the destination, active neighbors and expiration time. Notice that it is very important for all the routes in the routing table to be tagged with a destination sequence number. This technique will allow assuring that no routing loops can be formed, even in the presence of out-of-order packet delivery or high degrees of node mobility.

Maintenance of routes

AODV's route maintenance is required due to node mobility. To detect that a link being used breaks, the source has two options. One of them consists in sending HELLO messages frequently. These messages allow detecting when a link no longer is available as described below. A much better option is using information from the link layer to detect link failures. These link failures occur every time that an attempt to send a packet through that link fails.

When a link failure is detected the node detecting it will send a RERR message back to the source(s) using that link. It must state the problem encountered and use an updated sequence number. This process ends when all active sources receive the message. Source nodes still requiring the connection can start a new route request process to find new routes to active destinations.

Neighbor management

Detecting the availability of new neighbors and, more important, detecting that a neighbor has become unavailable is an important issue in mobile ad hoc networks. AODV allows nodes to detect their neighbors in two different ways. When a neighbor receives a broadcast packet from one of his neighbors it updates its internal tables to include that neighbor. Sometimes a node does not send any packets downstream, which could case downstream nodes to consider it unavailable. In those cases such nodes broadcast an unsolicited RREP packet within a *hello_ interval* time so that downstream nodes become aware of their liveliness. This packet has a TTL of 1, so it is not re-broadcasted. When a node fails to listen to up to *allowed_hello_ loss* consecutive packets from a node participating on an active path it considers the link with the upstream node to be lost and, therefore, generates a route error message to be sent to the source of that stream.

In a real implementation this technique can suffer in those situations where the channel is very congested, causing frequent losses of broadcast packets. Nodes could then erroneously infer that a link is down, when in fact it is not.

2.3.4 Dynamic Source Routing Protocol (DSR)

The Dynamic Source Routing (DSR) protocol [JM96, DDY04] is a high performance reactive routing protocol for MANETs. Its route discovery process is on-demand, which means that routes are only built when needed; route maintenance also depends on the existence of traffic. Therefore, when there is no data traffic on the network the routing traffic is effectively reduced to zero. In some aspects it is quite similar to the AODV protocol described before; the main difference is that DSR uses source routing, which means that the source determines the entire route towards the destination. So, packets are sent to their destinations with the entire route in their IP header. This method, despite provoking a small increase in terms of overhead, has other advantages such as avoiding routing loops in a simple and efficient manner.

One of the main differences between DSR and other routing protocols for MANET is its intensive use of caching. Each node participating in the MANET maintains a route cache where it saves all the routes it has learned. So, when a packet must be sent to a particular destination, nodes first check if a route is available on their cache. When there is no route for a packet a route discovery process is started. During that time DSR can be configured to hold on a buffer those packets waiting for a route or it can choose to discard them, relying on higher-layer protocol software to recover from that loss. DSR typically allows a data packet to be queued while waiting for a route for up to 30 seconds.

Route discovery

When initiating a route discovery the source broadcasts a route request (RREQ) packet, which is then successively broadcasted by other nodes until the destination is reached. Each node forwards only the first packet it receives for a certain route request ID originated at that same source. This aims at reducing the broadcast storm generated as much as possible. When re-broadcasting RREQ packets, nodes add themselves to the DSR header as elements of the route. This allows other nodes (including the destination) to also learn about the path. Therefore, when the destination receives a RREQ packet, it constructs the entire route for the route reply packet merely by calculating the opposite path through route reversion.

The destination of a route request can send a reply only for the first RREQ packet arriving or for all of them. Replying only to the first one allows reducing the routing overhead, but when that route is invalidated a new route request cycle has to be initiated. If it replies to all of the route requests the source is able to cache the different routes found and, when the first route is lost, it can try using the remaining routes successively.

If no route reply arrives to the source before the established timeout a new route discovery procedure is launched.

Route maintenance

DSR's route maintenance procedure consists of acting when link breaks occur. DSR uses the information from lower layers in order to detect broken links. This method allows it to react very quickly to link failures, but only for unicast packets since only these are acknowledged. The node which detects the broken link sends a route error message (RERR) back to the source indicating the link that broke. The source, as well as the rest of the nodes through which the RERR packet passes by, removes from its cache all routes including that broken link.

Another option in the case of detecting a link break event is to save the route error packet locally in a buffer, perform a route discovery for the original sender, and then send the route error packet along with the new route as soon as it receives the respective route reply.

Optimizations

From all the protocols designed for MANETs to this day, DSR is the one for which more optimizations have been proposed. We will start by analyzing optimization related to the use of cache.

Every time a host learns about a new route it adds an entry to its route cache. This occurs, for example, when a host is forwarding a packet towards another node since the entire route traversed is put on each packet's header. Also, a node may put its interface in promiscuous mode and add to its route cache any information it can overhear.

Another optimization still related to cache has to do with early route replies. If a host has a route cache entry for the target of the request, it may append this cached route to the accumulated route record in the packet, and may return this route in a route reply packet to the initiator without propagating (re-broadcasting) the route request. This avoids problems of many simultaneous replies in a simple manner; it also attempts to eliminate replies indicating routes longer than the shortest reply by causing each mobile host to delay the reply from its cache slightly before transmitting.

A last optimization involving full use of the route cache consists in allowing the originator of a route request to limit the maximum number of hops through which a packet may be propagated. This aims at exploiting the use of cached routes by favoring early route replies and avoiding many redundant request packets to propagate.

DSR also allows applying other kinds of optimizations such as piggybacking, route shortening and advanced error handling.

Piggybacking consists of accelerating the initial connection period when using connection-oriented protocols (e.g. TCP) by piggybacking connection-start messages (e.g. SYN) on route request packets, therefore doing route discovery and connection setup with a single step.

Concerning route shortening, the purpose is to reflect the availability of shorter routes as soon as possible. This becomes possible if network hosts operate with their network interfaces in promiscuous mode and find that they can overhear communication with nodes that were considered to be two hops away. In that situation the node making the discovery sends an unsolicited route reply packet to the original sender of the packet, informing it of the shortened route. The source host should then add this new route to its route cache.

We finally refer to some advanced error handling functionality. As for the discovery of routes, nodes can also improve on error detection by listening to packets in promiscuous mode. Since route error packets clearly indicate both ends of the link that broke, any host receiving a route error packet can update its own routing information so as to reflect that loss. This is done by analyzing all the routes on cache and truncating those entries using the broken link at the appropriate hop. All hosts on the route before this hop are still reachable on this route, but subsequent hosts are not.

The last optimization to improve the handling of errors is to support the caching of negative information on a host's route cache. This consists on the source maintaining information about a broken link on memory for a certain time. If during that time a route request is initiated and some nodes reply from cache using routes that still use that broken link, their information is discarded.

2.3.5 Multipath routing protocols

In the past there have been several approaches in the literature related to the discovery and use of multiple routes in MANETs.

In the work of Wang et al. [LYM$^+$01] a probing technique is used in order to assess the quality of available routes so that the traffic is forwarded based on the delay of each route. Their objective was to achieve load distribution as well as improved throughput, end-to-end delay and queue utilization.

In [MD01] the AODV protocol has been extended in order to provide multi-path capabilities, though no new route discovery mechanism was proposed. Both node disjoint and link disjoint approaches are presented. In their work there is no traffic splitting. Also, neither of these two works referred before propose enhancements to the route discovery technique itself.

Nasipuri et al. [ARS01] proposed a strategy for quick route recovery through packet re-direction on intermediate nodes in order to reduce the frequency of query floods. Their solution aims at reducing the number of *lost route* messages, as well as performing fewer route discoveries. However, the source is unaware of any extra routes, which means that their solution does not aid in the task of splitting traffic through disjoint routes.

Wu [Wu02] proposes a more selective route discovery procedure to DSR to increase the degree of disjointness of routes found without introducing much extra overhead. However, it allows the source to find a maximum of only two paths (node disjoint paths) per destination and required two consecutive route discovery processes to take place.

In the work of Lee and Gerla [LG01] the traffic is evenly split among the two first routes found in order to achieve load distribution; they analyze the options of starting a new route discovery process when one of the routes is lost or only when both are lost. The authors find that DSR's standard route discovery mechanism not only returns a few routes, but also that these routes are mainly overlapped (not disjoint). To solve this problem they enhance the route discovery mechanism of DSR to find more node disjoint paths; we will compare Lee and Gerla's proposal with our own method in chapter 6.

We now proceed to study different QoS models for mobile ad hoc networks.

2.4 Quality of Service models for MANETs

The Internet was initially created to handle only best-effort traffic. This means that there is no resource reservation, so all users compete for bandwidth. For this reason the Internet Protocol (IP) is connectionless, requiring no set-up "signaling" for admission control. Later, enhancements in terms of available bandwidth and terminal's capabilities brought up the need for supporting new services in the Internet. These new services, though, performed poorly due to the best-effort policy. There was, therefore, a need to enhance the Internet in order to perform resource reservation in a similar fashion to telephony networks. The RSVP protocol [RLS$^+$97] was created to fulfill this need as part of the Internet's Integrated Services (IntServ) architecture [RDS94]. RSVP follows a receiver based model since it is the responsibility of each receiver to choose its own level of reserved resources, initiating the reservation and keeping it active. The actual QoS control, though, occurs at the sender's end. The sender will try to establish and maintain resource reservations over a distribution tree. If a particular reservation is unsuccessful, the correspondent source(s) is notified.

The Integrated Services architecture proved to be complex and required too many resources, suffering from scalability problems. So, the Differentiated Services (DiffServ) architecture [S. 98] emerged as a more efficient alternative. In the latter, Service Level Agreements (SLA) are achieved between different domains. One of the main virtues of the Differentiated Services architecture is that it drops the traditional concept of signaling, since it no longer requires the reservation of resources in all the network elements involved. The strategy consists in performing admission control on domain boundaries, and then treating them in a differentiated manner inside the domain according to packet tagging on the domain borders, which is a much faster and lightweight process.

MANET environments differ greatly from the wired environments the DiffServ and IntServ models were created for. The difference stems not only from the new problems encountered in MANETs (mobility, collisions, variable channel conditions, etc.), but also because MANETs do not follow the Client / Service Provider paradigm inherent to both IntServ and DiffServ models. In MANETs the network is typically formed by users that cooperate and, except in situations where there is some centralized management entity (e.g. army), it relies on users good will and limited resource sharing. So, new proposals were presented in order to achieve reliable QoS support in MANETs. Examples of such proposals are INSIGNIA, SWAN and FQMM. We will now expose the main characteristics of these proposals

2.4.1 INSIGNIA

Lee et al. [SAXA00] proposed INSIGNIA, an in-band signaling system that supports fast reservation, restoration and adaptation algorithms. With INSIGNIA all flows require admission control, resource reservation and maintenance at all intermediate stations between source and destination to provide end-to-end quality of service support. In figure 2.2 we offer an overview of INSIGNIA's architectural components, showing that the INSIGNIA framework is independent of the routing and MAC protocols used.

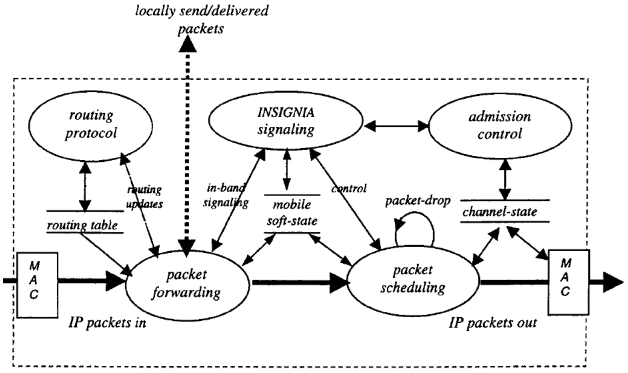


Figure 2.2: The Insignia QoS framework

The INSIGNIA signaling system is designed to be lightweight in terms of the amount of bandwidth consumed for network control and to be capable of reacting to fast network dynamics such as rapid host mobility, wireless link degradation, intermittent session connectivity and end-to-end quality of service conditions. We will now proceed by detailing several elements that conform INSIGNIA's QoS framework.

In-Band Commands

INSIGNIA's commands are put inside the IP option field; these include service mode, payload type, bandwidth indicator and bandwidth request fields.

When a node wants to perform a flow reservation it activates a reservation mode bit (RES) in the IP option service mode field of a data packet and sends the packet to the destination. Intermediate nodes will then decide if to admit or deny the reservation request. Acceptance means that node's resources are committed and so further packets belonging to the same flow will receive the requested QoS-enhanced service. On the contrary the reservation is denied and packets from that flow are treated as best-effort packets.

When the destination of a reservation request receives a RES packet it sends a QoS report to the source node notifying it that the reservation succeeded. If during the reservation, or at a later time, the situation changes and the flow can no longer receive the requested QoS, the destination node can issue scaling/drop commands to the source node. Detection of downgraded flows is achieved through monitoring the IP option fields, especially the reservation mode bit which is switched from RES to BE.

Fast Reservation

The support for adaptive flows requires setting and processing the IP option field adequately. QoS reservation packets have the service mode set to RES, payload set to either BQ or EQ (Base QoS or Enhanced QoS), bandwidth indicator set to either MIN or MAX, and the minimum and maximum bandwidth fields set to valid values.

When a reservation is being established each node along the path checks if it can offer the maximum QoS requested. In the event that all nodes can offer this maximum QoS requested the destination will become aware of it by noticing that the bandwidth indicator is set to MAX. On the contrary this field will be set to MIN, which means that all the packets sent by the source pertaining to the enhanced QoS traffic (payload type is EQ) will be degraded to best effort traffic at the first bottleneck node.

Soft-State Management

A soft-state approach is used to manage resources in the presence of mobility. This means that, as the route being used changes, new reservations along that new path are done automatically by a restoration mechanism. Once a new flow is accepted by the admission control mechanisms a soft-state timer is started and associated with the new or re-routed flow. Every new packet of the flow that is received at any intermediate node causes the timer to be restarted. If the flow is re-routed and no longer passes through that intermediate node, the timer expires and the reservation is de-allocated. This mechanism has the advantage of operating in a fully distributed manner.

Fast Restoration

Mobility forces on-going flows to be re-routed. The purpose of restauration is to re-establish the connection as quickly as possible. However, the speed of restauration depends on routing tasks, and so on the routing protocol being used. Ideally the flow would use a new route as soon as the path was found to be lost. However this is not always the case, and so the connection re-establishment may take longer.

The first intermediate node that cannot meet the QoS requirements of a flow downgrades it to best effort, which means that no further nodes after that point will attempt to reserve resources for that flow. If later the bottleneck node becomes uncongested it can allow the reservation to take place throughout the entire path, thereby achieving what is called degraded restoration.

QoS Reporting

QoS reporting is helpful to inform source nodes of the status of their flows. This requires the destination nodes to constantly monitor on-going flows and check if their status information has changed or if the delivered QoS has changed. The destination will send QoS reports to the source periodically, or immediately if there was some sort of adaptation.

In the case that the source finds that only BQ packets can be supported, it switches the service mode of EQ packets from RES to BE. This allows freeing partial reservations along the path and, therefore, allowing other competing flows to use these resources.

Adaptation

INSIGNIA provides several adaptation levels that can be selected. Typically, an adaptive flow operates by performing resource reservation for both its base and enhanced components. Scaling flows down depends on the adaptation policy selected. Flows can be scaled down to their base QoS delivering enhanced QoS packets in a best-effort mode hence releasing any partial reservation that may exist. On the other hand, the destination can issue a drop command to the source to drop enhanced QoS packets (i.e., the source stops transmitting enhanced QoS packets). Further levels of scaling can force the base and enhanced QoS packets to be transported in best effort mode. Either way, the time scale over which the adaptation actions occur depends on the application itself. These scaling actions could be instantaneous or based on a low-pass filter operation.

Drawbacks

In [LPB04] Georgiadis et al. show that link interferences (due to the hidden terminal problem) in multihop wireless networks make the problem of selecting a path satisfying bandwidth requirements an NP-complete problem, even under simplified rules for bandwidth reservation. In the literature this is commonly referred to as the "coupled capacity" problem. This means that a local assessment of available bandwidth may not offer accurate-enough data to reach an accept or deny decision in a MANET environment where hidden-node effects are prone to occur. Such fact makes deploying INSIGNIA on a real MANET environment a non-trivial task.

2.4.2 FQMM

FQMM [HWAK00] is a flexible QoS model for MANETs designed for small to medium sized MANETs (less than 50 nodes) with a flat topology. In that model a hybrid per-flow and per-class provisioning scheme is used, so that traffic of the highest priority is given per-flow QoS provisioning while other category classes are given per-class QoS provisioning. FQMM's hybrid scheme combines the per-flow granularity of IntServ and the per-class granularity of DiffServ.

It defines three kinds of nodes as in DiffServ: ingress, interior and egress nodes. An ingress node is a mobile node that sends QoS data. Interior nodes are the nodes that forward data for other nodes. An egress node is a destination node. Figure 2.3 illustrates these concepts.

In this model a MANET represents one DiffServ domain where traffic is always generated by applications running on an ingress node and terminate in an egress node. This means that the roles of nodes will depend on each data flow.

Provisioning

Provisioning refers to the determination and allocation of resources needed at various points in the network.

The FQMM model proposes using a hybrid provisioning scheme that tries to combine both IntServ and DiffServ solutions available for the Internet. The QoS

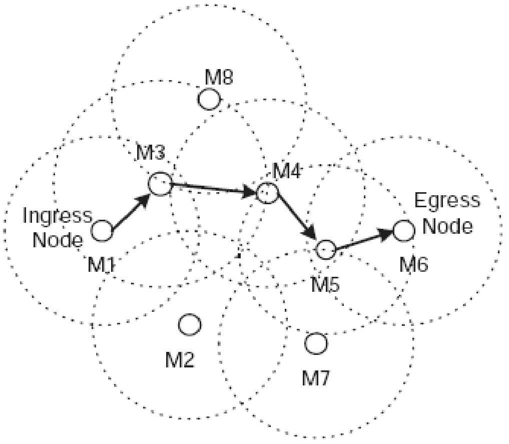


Figure 2.3: Illustration of the ingress and egress node concepts for FQMM

provisioning scheme proposed offers per-flow granularity to some kinds of traffic and per-class granularity to other kinds of traffic. This way, streams of the highest priority are handled on a per-flow basis so as to achieve optimum performance. Traffic belonging to the remaining QoS classes is handled on a per-class basis so as to allow traffic differentiation without saturating MANET elements with QoS-related tasks.

Conditioning

Traffic conditioning is done at ingress nodes to affect the traffic source. Traffic policing tasks are done according to the traffic profile selected.

Components of the traffic conditioner include traffic profile, meter, marker and dropper. The traffic profile element is the most important since other components will change their configuration according to the traffic profile.

Since the bandwidth between two nodes is time varying, the traffic profile is defined as a percentage of the effective bandwidth available. The purpose is to achieve a relative and adaptive differentiation traffic profile. The goal is to keep consistent differentiation between sessions, which could be per flow or per aggregate of flows to adapt to the dynamics of the network.

Metering and marking tasks are done with the aid of a token bucket policer. Relatively to the dropper element, it is responsible for dropping packets when the QoS requirements can not be met.

QoS routing and resource management

Typical routing protocols find routes without taking into account any QoS-related information. An example of such a protocol is the DSR routing protocol studied before. In FQMM authors propose extending the functionality of routing proto-

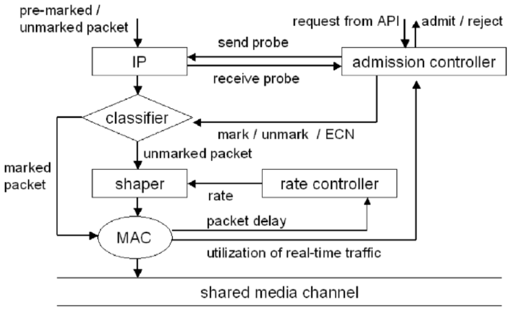


Figure 2.4: Architecture of the SWAN model

cols to make them more cooperative with the FQMM framework by assessing the quality of paths discovered, though that work is considered to be out-of-scope.

Concerning resource management, authors consider that two important issues to focus on are link bandwidth sharing and buffer management. In their analysis they propose applying bandwidth shaping at the source node and queue management techniques on routers along the path.

Drawbacks

The experiments using FQMM focus mostly on differentiating TCP traffic by applying different queue management techniques.

In a later work [HKWA01] authors find that the priority buffer and scheduling schemes proposed fail when UDP traffic gets higher priority than TCP. Since UDP is the transport protocol used to support real-time multimedia applications, this becomes an important drawback of the FQMM proposal.

Sobrinho and Krishnakumar [SK99] point out that the main performance requirement for these applications is bounded end-to-end delay, implying that bounded packet delay at the MAC layer is a *sine qua non* condition for success.

2.4.3 SWAN

Ahn et al. [GAAL02] designed SWAN, a stateless network model aiming at providing service differentiation in MANETs. One of the main advantages of SWAN is that it does not require the support of a QoS-capable MAC layer to provide service differentiation. Instead, it relies on rate control mechanisms that shape best-effort traffic at each node. SWAN's framework uses sender-initiated admission control and explicit congestion notification for real-time traffic in order to adapt to mobility and congestion conditions.

The SWAN model includes a number of mechanisms used to support rate regulation of best effort traffic, as illustrated in figure 2.4. The main elements of SWAN are a traffic classifier, a traffic shaper, a rate controller and an admission controller. The classifier is used to differentiate real-time from best effort traffic. The shaper will act upon packets classified as best effort traffic with a simple leaky bucket to regulate the traffic rate. The rate controller will act upon the shaper setting the shaping rate. Concerning the admission controller, this element is responsible for allowing new connections to enter the MANET and also for estimating locally available bandwidth.

Local Rate Control of Best Effort Traffic

Every node in a SWAN-enabled MANET must regulate best effort traffic. The rate controller determines the departure rate of the shaper using an AIMD (additive increase / multiplicative decrease) rate control algorithm based on feedback from the MAC layer. This is simply measured at each node by subtracting the time that a packet is passed to the MAC layer (from the upper layer) from the time an ACK packet is received from the next-hop station.

The rate controller constantly monitors the actual transmission rate. When the difference between the shaping rate and the actual rate is greater than g percent of the actual rate, then the rate controller adjusts the shaping rate to be g percent above the actual rate. This gap (i.e., g percent) allows the best-effort traffic to increase its actual rate gradually.

Source-Based Admission of Real-Time Traffic

The admission controller measures the rate of real-time traffic in terms of bits per second at each node. The admission control element must obtain a running average of channel measurements in order to filter small-scale variations.

When a new flow must be admitted into the network the source station sends a probing request packet to assess end-to-end bandwidth availability. Each node along the path will update the bandwidth value if its locally available bandwidth is lower than the one stated on the packet. The destination node sends a probing response packet back to the source node with the bottleneck field copied from the probing request message it received.

If the new flow is accepted all of its packets are tagged as QoS packets. The network elements along that new path are unaware of the new flow, and they merely keep shaping best effort traffic to offer good performance to QoS traffic.

Dynamic Regulation of Real-Time Traffic

The functioning of SWAN can be disturbed by mobility and false admission conditions. Mobility may cause traffic to be re-routed through new paths where there are no resources available to accommodate the new traffic. False admission may occur when several sources probe the path, all finding that there are enough resources to admit new flows when there really aren't. Such problem can be alleviated by using congestion regulation algorithms.

SWAN uses ECN (Explicit Congestion Notification) to mark packets when a node finds that traffic is experiencing congestion. This is done by setting the ECN bits located on the IP header.

The use of ECN can be done in two different ways. The first one consists in marking all the packets flowing through a congested node. When receiving

ECN-marked packets, the destination node notifies this occurrence to the source. However, the source node does not immediately initiate reestablishment upon receipt of a regulate message. Rather, it waits for a random amount of time before initiating the reestablishment procedure. This avoids that all sources probe the network at the same time.

Another solution would be for congested/overloaded nodes to randomly select a c*ongestion set* of real-time sessions and only mark packets associated with the set. This can be done using a hash function without keeping any per-flow state at the intermediate nodes. A congested node marks the congested set for a period of time T seconds and then calculates a new congested set. As in the case of the previous algorithm, nodes stop marking packets congested when the measured rate of the real-time traffic drops below the admission control rate. A disadvantage on this scheme is that it requires some intelligence at intermediate nodes to manage the congested sets,as well as for determining if a flow is new or old in order to correctly respond to false admission. However, it enables a better utilization of resources than the source-based regulation technique.

Drawbacks

SWAN's admission control mechanism requires all stations to keep track of the MAC's transmission delay of all packets in order to estimate available bandwidth; however, the association of a global estimate for transmission delay with a certain bandwidth in the link towards a specific target station is not straightforward, especially outside simulation scope. Also, the overhead introduced by the proposed shaping and measurement techniques proposed can be significative for mobile terminals with few resources.

2.5 Conclusions

In this chapter we offered an overview of the current state-of-the-art in mobile ad hoc networks. We focused on two different subjects: routing protocols and quality of service models.

Concerning routing protocols, there is still no agreement among the scientific community members on which is the most adequate routing strategy for mobile ad hoc, and so a lot of research is still being done.

In this chapter we explored the main differences between the three most important routing protocols for MANETS (OLSR, AODV and DSR) showing how each routing protocol handles problems such as route discovery and route maintenance. We also offered an overview of the different multipath routing protocols available, evidencing the proposal by Lee and Gerla. We will compare their proposal to our own on chapter 6.

The issue of Quality of Service support on MANETs is still on an early stage. Most of the proposals currently available only focus on improvements at a specific layer, not offering an overall QoS architecture.

The few QoS models currently available for MANETs are INSIGNIA, FQMM and SWAN. In this chapter we have shown their main architectural principles, as well as some of the drawbacks we have detected for each of these proposals. In chapter 8 we will propose a new QoS architecture for MANETs intending to offer a solution that is more lightweight and more easily deployable than those referred here.

Chapter 3

Digital video coding

Digital video is being adopted by an increasing array of applications, ranging from video telephony and videoconferencing to DVD and digital TV. The adoption of digital video in many applications has been fueled by the development of efficient video coding standards, resulting in many video coding standards that simultaneously cover a wide range of application areas. These standards were defined taking into account the interoperability between systems designed by different manufacturers for any given application, hence facilitating the growth of the video market.

This chapter is dedicated to digital video coding, more specifically to the recent H.264 standard [H2603a]. We start by offering an overview of the most important video codec standards developed to this day, and we then proceed to analyze the H.264 standard in more detail. Our analysis of H.264, though embracing the standard as a whole, focuses especially on those issues related to H.264 transmission in IP networks and error-resilience issues.

3.1 Evolution of video coding standards

The ITU-T [ITU] is now one of two most important standard organizations that develop video coding standards, the other being ISO/IEC JTC1 [JTC]. The ITU-T video coding standards are called recommendations, and they are denoted with H.26x (e.g., H.261, H.262, H.263 and H.264). The ISO/IEC video coding standards are denoted with MPEG-x (e.g., MPEG-1, MPEG-2 and MPEG-4). Most ITU-T recommendations have been designed for real-time video communication applications, such as video conferencing and video telephony. On the other hand, MPEG standards have been designed to address the requirements of video storage (CD/DVD), video broadcast (broadcast TV), and video streaming (e.g., video over the Internet, video over DSL, video over wireless) applications.

H.261 was the first video coding standard available, and it was designed for videoconferencing applications. On the other hand, the MPEG-1 video coding standard was accomplished for storage in compact disks (CDs).

The H.262/MPEG-2 standard, which was developed jointly by the two committees, is an extension to the MPEG-1 standard designed to support digital TV and

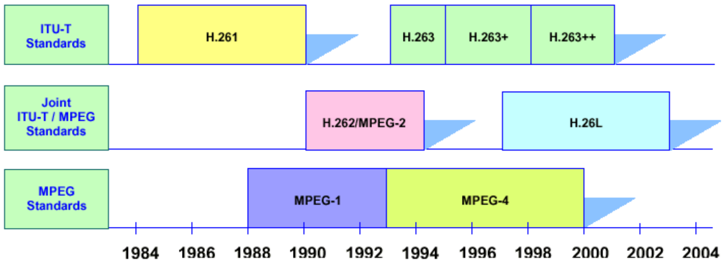


Figure 3.1: Progress of ITU-T recommendations and MPEG standards.

HDTV. Concerning MPEG-4 part 2 [mpe01], it covers a very wide range of applications including video objects that can be either rectangular pictures or shaped regions. This includes also natural and synthetic video / audio combinations with interactivity built in.

The ITU-T, on its part, developed the H.263 standard [H2695] to improve the compression performance of H.261, and the base coding model of H.263 was adopted as the core of some parts in MPEG-4 part 2. After the H.263 standard was ready the same study group enhanced H.263's video compression technology, which led to the appearance of two new video standards: H.263+ and H.263++.

Figure 3.1 summarizes the evolution of ITU-T recommendations and ISO/IEC MPEG standards.

Recently, the ITU-T VCEG and the ISO/IEC JTC1 have agreed to join their efforts once again to develop the H.264 / MPEG-4 part 10 standard, forming the JVT (Joint Video Team). H.264 was initiated by the ITU-T committee under the name H.26L, but it was adopted by both committees because they were working with very similar techniques for developing extensions of current standards in order to significantly increase coding performance. So, joining efforts to exchange ideas and develop a common framework was the main reason for working together.

We will now proceed to analyze with more detail the new H.264 standard, evidencing the technical improvements that it offers.

3.2 The H.264 standard

The main objective behind the H.264 project [H2603a] is to develop a high-performance video coding standard by adopting a back to basics approach where simple and straightforward design using well-known building blocks is used. The ITU-T Video Coding Experts Group (VCEG) has initiated the work on the H.26L standard in 1997. Towards the end of 2001, and witnessing the superiority of video quality offered by H.26L-based software over that achieved by the existing most optimized MPEG-4 based proposals, ISO/IEC MPEG joined ITU-T VCEG forming the Joint Video Team (JVT) that took over the H.26L project of the ITU-T. JVT's purpose is to create a single video coding standard that will simultaneously result in a new part (Part 10) of the MPEG-4 family of standards and a new

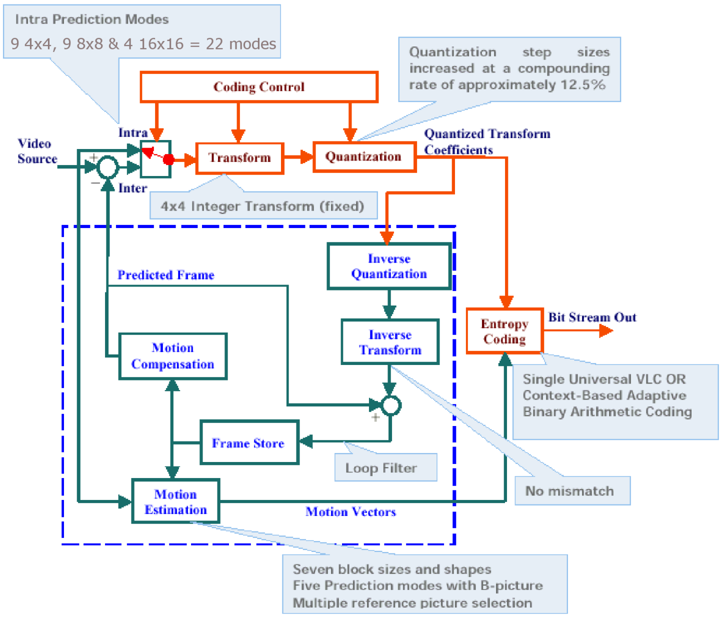

Intra Prediction Modes
9 4x4, 9 8x8 & 4 16x16 = 22 modes
Quantization step sizes increased at a compounding rate of approximately 12.5%
Coding Control
Video Source
Intra
Inter
+
−
Transform
Quantization
Quantized Transform Coefficients
4x4 Integer Transform (fixed)
Inverse Quantization
Predicted Frame
Entropy Coding
Bit Stream Out
Inverse Transform
Motion Compensation
Single Universal VLC OR Context-Based Adaptive Binary Arithmetic Coding
+
Frame Store
Loop Filter
No mismatch
Motion Estimation
Motion Vectors
Seven block sizes and shapes
Five Prediction modes with B-picture
Multiple reference picture selection

Figure 3.2: Block diagram of the H.264 encoder

ITU-T (H.264) Recommendation. The H.264 development work is an on-going activity, being the first version of the standard finalized technically before 2002 and officially in 2003. The emerging H.264 standard has a number of features that distinguish it from existing standards while, at the same time, sharing common features with them.

Some of the key features of H.264 are: up to 50% in bit rate savings, high quality video (including low bit rates), adaptation to delay constraints (real-time communication applications as well as video storage and server-based video streaming applications), error resilience tools to deal with packet loss in packet networks and bit errors in error-prone wireless networks. Network friendliness is also aimed by this standard through a conceptual separation between a Video Coding Layer (VCL), which provides the core high-compression representation of the video picture content, and a Network Adaptation Layer (NAL), which packages that representation for delivery over a particular type of network.

The above features can be translated into a number of advantages for different video applications.

The underlying approach of H.264 is similar to that adopted in previous standards such as H.263 and MPEG-4, and consists of the following four main stages:

1. Dividing each video frame into blocks of pixels, so that the processing of the video frame can be conducted at the block level by using the well known DCT transform.

2. Exploiting the spatial redundancies that exist within the video frame by coding some of the original blocks through transform, quantization and entropy coding (or variable-length coding).

3. Exploiting the temporal dependencies that exist between blocks in successive frames, so that only changes between successive frames need to be encoded. This is accomplished by using motion estimation and compensation. For any given block, a search is performed in the previously coded one or in more frames to determine the motion vectors that are used by the encoder and the decoder to predict the subject block.

4. Exploiting any remaining spatial redundancies that exist within the video frame by coding the residual blocks, i.e., the difference between the original blocks and the corresponding predicted blocks, again through transform, quantization and entropy coding.

From the coding point of view, the main differences between H.264 and the other standards are summarized in figure 3.2 through an encoder block diagram. From the motion estimation/compensation side, H.264 employs blocks of different sizes and shapes, higher resolution sub-pixel motion estimation, and multiple reference frame selection. In the transform side, H.264 uses an integer-based transform that approximates the DCT transform used in previous standards, but does not have the mismatch problem in the inverse transform. In H.264, entropy coding can be performed using either a single Universal Variable Length Codes (UVLC) table or using Context-based Adaptive Binary Arithmetic Coding (CABAC).

3.2.1 Profiles and levels

The H.264 framework defines different profiles and levels. These consist of subsets of the bit-stream syntax, as well as the minimum requirements that decoders conforming to that profile must have. The three most important profiles developed are: Baseline, Main, and Extended. The Baseline is the most simple, being adequate for real-time services such as videoconferencing and videophone. The Main Profile is designed for digital storage media and television broadcasting. Finally, the Extended Profile was designed to support multimedia services over the Internet.

Apart from these three, there are also extension profiles for H.264 aiming at applications related to content contribution and distribution, as well as for studio editing and post-processing. These extensions consist of four high profiles denoted as High, High 10, High 4:2:2, and High 4:4:4, which offer increasing degrees of accuracy representing pixels.

All H.264 profiles share some common parts that are mandatory for basic H.264 functionality. Among these are the use of *Intra-coded and Predictive-coded slices*, as well as *CAVLC* (Context-based Adaptive Variable Length Coding) for entropy coding.

The different profiles then extend these mandatory requirements according to the purpose being sought. For instance, the Baseline Profile requires support for

Flexible macroblock ordering, *arbitrary slice ordering* and *redundant slices.* Flexible macroblock ordering consists of using a map to assign macroblocks to a slice group, which allows macroblocks to be transmitted and retrieved in a non-scan order. Arbitrary slice ordering allows doing something similar, but acting on slices instead of macroblocks. Concerning the use of redundant slices, it allows re-sending a same slice obtained by using the same or a different coding rate.

The Main Profile includes *B slices*, *weighted prediction* and *CABAC* (Context-based Adaptive Binary Arithmetic Coding) for entropy coding. B slices (Bi-directionally predictive-coded slices) are more flexible than P slices since the former are coded using inter prediction from both past and future pictures using two different motion vectors and reference indices to predict the sample values of each block. Concerning weighted prediction, it is a scaling operation by applying a weighting factor to the samples of motion-compensated prediction data in P or B slices.

The Extended Profile is an extension of the Baseline Profile including *SP, SI and B slices*, as well as *data partitioning.* SP and SI slices are specially coded to offer efficient switching between video streams, being similar to P and I frames respectively. Relatively to data partitioning, it consists in placing coded data in separate data partitions, where each partition can be placed on a different layer unit.

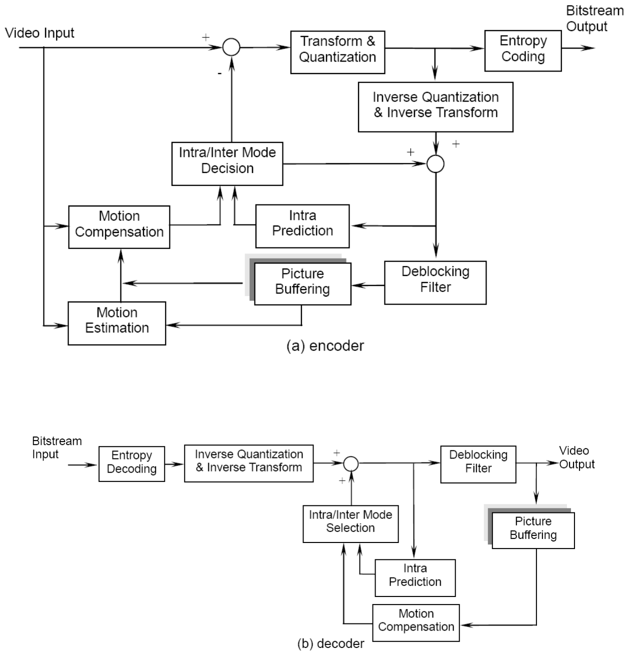
Finally, all the High Profiles are extensions to the Main Profile offering also adaptive transform block sizes, as well as quantization scaling matrices where different scaling can be applied according to the specific frequency associated with the quantized transform coefficients. The purpose is to optimize the subjective quality.

3.2.2 Layered structure

An H.264 codec generates output bitstreams with two distinct layers: Network Abstraction Layer (NAL) and Video Coding Layer (VCL).

The NAL operates at a higher abstraction level than the VCL and is used to prepare data to be sent over a communication channel or to be stored on a media. The NAL defines two stream formats: byte-stream and packets. The byte-stream format is defined for any storage media and also for non-packet oriented networks, such as video broadcasting. On the other hand, the packet-based format is used mostly on applications that run over IP networks, typically using a RTP/UDP/IP combination.

A NAL unit is classified into VCL and non-VCL. A non-VCL unit contains extra information to form self-contained packets; the purpose is to support the loss of packets by inserting those parameters that are required for the decoding process. Concerning the VCL unit, it contains the video sequence itself, which is divided into pictures; pictures are in turn divided into slices, and these into macroblocks.



(a) encoder

(b) decoder

Figure 3.3: Block diagram of H.264's encoder (a) and decoder (b)

3.2.3 Video Coding Algorithm

H.264 does not define an encoder/decoder pair, but instead defines the video stream syntax and the methods for decoding the bitstream. We can, however, extract the main functional elements of both encoder and decoder. These are depicted in figure 3.3. Most of the basic functional elements presented are not new, being already present in previous standards such as MPEG-1, MPEG-2, MPEG-4, H.261 and H.263, being the de-blocking filter the only exception. However, in terms of details, H.264 offers great enhancements for all functional blocks.

Starting our analysis by the encoder, it may select either intra or inter coding for each picture block. Intra coding is especially important since it does not require other pictures for a correct decoding of information. It uses various spacial prediction modes, as well as spacial redundancy, in a single picture's scope.

Inter coding, used for both P and B frames, is a more efficient technique that consists in using previously coded pictures for the prediction of each block. To achieve that purpose it calculates motion vectors to reduce the temporal redundancy between nearby pictures. Prediction values are obtained by applying a deblocking filter to the pictures encoded previously. This filter is able to reduce block-related image distortion on block boundaries.

After the prediction process is done the residuals are then quantized and compressed using a transform that allows removing any spatial correlation that may exist. The final step consists in entropy coding the motion vectors or intra predic-

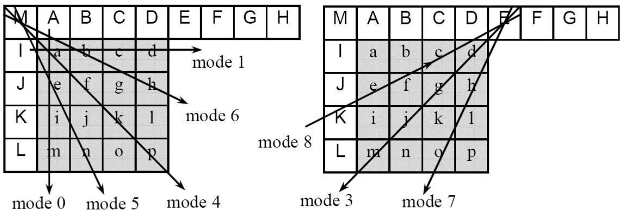


Figure 3.4: Intra 4 x 4 prediction mode directions (vertical : 0, horizontal : 1, DC : 2, diagonal down left : 3, diagonal down right : 4, vertical right : 5, horizontal down : 6, vertical left : 7, horizontal up : 8)

tion modes, along with the quantized transform coefficients, using either context-adaptive variable length codes (CAVLC) or context adaptive binary arithmetic codes (CABAC).

Intra prediction

There are two situations where intra prediction is used. The first one is when Intra frames are being coded - in that situation all the macroblocks are intra-coded. The second situation occurs when the motion compensation prediction offers very poor results - in that situation only the macroblocks that suffer from poor prediction accuracy are intra-coded.

One of the main drawbacks of using intra-coded pictures is low compression performance, which can become a bottleneck when operating in constrained bitrate environments. So, the H.264 standard gives special attention to this issue in order to increase performance.

The technique used consists of predicting intra-coded macroblocks based on previously decoded macroblocks. It is the residual signal from the difference between the current block and the prediction that is finally encoded.

The prediction block can have several sizes: 4x4, 8x8 or 16x16 (for luma samples). So, 4x4 and 8x8 luma blocks can use 9 different prediction modes, while 16x16 luma blocks can use 4 different prediction modes. To exemplify the prediction process, figure 3.4 shows the 9 prediction mode directions for a 4x4 luma block. For mode 0 (vertical) and mode 1 (horizontal), the predicted samples are formed by extrapolation from upper samples [A, B, C, D] and from left samples [I, J, K, L], respectively. For mode 2 (DC), all of the predicted samples are calculated using both upper and left samples [A, B, C, D, I, J, K, L]. For mode 3 (diagonal down left), mode 4 (diagonal down right), mode 5 (vertical right), mode 6 (horizontal down), mode 7 (vertical left), and mode 8 (horizontal up), the predicted samples are formed from a weighted average of the prediction samples A-M. For example, samples a and d are respectively predicted by round($I/4 + M/2 + A/4$) and round($B/4 + C/2 + D/4$) in mode 4, also by round($I/2 + J/2$) and round($J/4 + K/2 + L/4$) in mode 8. The encoder may select the prediction mode for each block that minimizes the residual between the block to be encoded and its prediction.

Inter prediction

Inter prediction of video frames is used to reduce the temporal correlation between nearby frames through motion estimation and compensation.

As referred before, H.264 allows splitting a 16x16 macroblock into smaller block sizes with a minimum of 4x4 blocks. For the 16x16 macroblock mode there are 4 cases: 16x16, 16x8, 8x16 or 8x8, and for the 8x8 mode there are 4 cases : 8x8, 8x4, 4x8 or 4x4 . The purpose is that the extra bits consumed to code the motion vectors of these smaller blocks become compensated by an increase in terms of estimation accuracy. So, the best choice will depend on the characteristics of each video input, being that regions with more detail will benefit from smaller partition sizes.

Another feature included in the H.264 framework is the support for sub-pixel motion compensation. This consists in improving the accuracy of motion vectors up to a quarter of a pixel. Obviously, the improvement in terms of estimation accuracy has a price both in terms of complexity (it requires interpolating sample values) and number of bits to code the motion vector. Sub-pixel accuracy is expected to improve the coding efficency at high bitrates and high video resolutions.

Yet another improvement introduced by the H.264 framework is the possibility of using Multiple Reference frames. Previously, the motion compensation process used one reference frame for P frames and two reference frames for B frames. H.264 introduces the possibility of using several reference frames, which can be pictures before or after the current picture. Such reference pictures are classified as either short-term or long-term reference frames, and must be stored in the picture buffer. The improvement offered by H.264 consists of these long-term reference frames, unavailable until this standard appeared. Long-term reference pictures extend the motion search range by using multiple decoded pictures, instead of using just one decoded short-term picture.

The use of multiple reference frames requires improving memory management so that pictures can be deleted from the buffer as soon as they are not used.

Transform and quantization

Besides temporal redundancy, video codecs also have to deal with spacial redundancies in an efficient manner. These spatial redundancies are also relevant in the case of prediction residuals. The H.264 standard makes use of a block-based transform to remove spacial redundancies. So, once both inter and intra prediction processes complete successfully, the resulting prediction residual is split into either 4x4 or 8x8 blocks which are then transformed and quantized. Compared to previous video coding standards where only an 8x8 DCT was used, the introduction of 4x4 transforms allows reducing the typical ringing artifacts encountered with these transforms, and also removes the need for multiplications while calculating the transform value.

H.264 introduces yet another coding improvement that consists in combining the DC values of nearby 4x4 transforms since it was found that the correlation among these was typically high. The structure of the transform will, thereby, be hierarchical.

Some applications can benefit from a reduced quantization step size. The purpose it to achieve very high PSNR levels. So, H.264 increases the quantization step size by two octaves, making the QP range vary from 0 to 51. To avoid the need for several multiplications, the transform and the quantization processes are combined using a modified integer forward transform, quantization and scaling.

Entropy coding

The previous standards developed by the ISO/MPEG and ITU-T/VCEG groups used variable length codes (VLC) to perform entropy coding. This was done by using fixed tables with sets of codewords built using generic probability distributions. H.264 behaves differently by adapting to context characteristics using Exp-Golomb codes [S. 66]. Residual data is obtained using either a zig-zag or an alternate scan, and coding is done using such context-adaptive variable-length codes (CAVLC). Basically, the main enhancement introduced by CAVLC has to do with the coding of zero and ±1 coefficients separately. Since the probability of these values after transformation and quantization is rather high, it results in a good coding efficiency.

Main and High profiles use a different algorithm, CABAC, since it offers more coding efficiency. CABAC achieves improved compression through an adaptive arithmetic coding algorithm where the probability model for each symbol is constantly updated.

The encoding process used by CABAC starts with binarization, which is a mere transformation from non-binary values such as transform coefficients or motion vectors to a binary sequence. This is an essential step since all the processing that follows requires symbol binarization. The following step consists of context modeling, which consists of a probability model for a symbol's elements whose selection depends on previously encoded syntax elements. The last and most important step consists of binary arithmetic coding. This element must perform two tasks for each element processed. The first one consists in encoding according to the probability model used; the second one consists of an update of the model itself.

Deblocking filter

Avoiding blocking artifacts is a very important issue for all block-based transforms that use coefficient quantization. The need to avoid them becomes relevant since these can be perceptually noticed by users, and so reduces the quality of the video visioning experience.

The purpose of the deblocking filter is to reduce these blocking artifacts that are present on block boundaries, and also to prevent the propagation of accumulated coded noise. Previous coding standards did not include this element because of its complexity; however, H.264 developers considered it to be relevant for higher coding performance. When the accuracy of motion vectors is as low as half a pixel it can be avoided.

The filtering process is applied to the four edges of a 4x4 macroblock, and it operates on both luma and chroma components.

The deblocking filter operates in an adaptive manner, and it is applied at different levels: slice, block-edge and sample. At slice level the deblocking filter allows the global filtering level to adjust to the characteristics of the video sequence itself. At the block-edge level the encoder also applies an adaptive filtering level depending on factors such as differences in terms of motion, coded residuals or choices relative to inter/intra coding for the participating blocks. When macroblocks are characterized by being very uniform, the deblocking filter must operate especially well to remove tilting artifacts. Finally, at the sample level there are also adjustments since filtering can be turned on or off depending on a sample's values and quantizer-dependent thresholds.

3.2.4 Error resilience

Error resilience support on video codecs is a very important issue. Currently there is a increasing demand for streaming video applications, as well as for real-time video communication. Since the transmission media used is normally a lossy environment, error detection and correction algorithms are an important requirement for video coding standards.

In this section we will discuss the different mechanisms made available in the H.264 framework to improve error resilience. We start by discussing mechanisms that were already available on previous video coding standards such as placement of intra-coded macroblocks/slices/frames, picture segmentation, detection of reference pictures and data partitioning. We then proceed by exposing the new error resilience mechanisms introduced by H.264, which include flexible macroblock ordering and redundant slices.

Intra placement

The use of intra-coding is an efficient method to reduce the propagation of errors. The H.264 framework offers the possibility of using intra-coding at macroblock, slice or picture levels.

One of the issues to take into account when good error resilience is to be achieved is that the error reseting process should be as effective as possible for complete re-synchronization. When working with an H.264 codec we should take care of two issues: that intra-coded portions of a picture do not depend on inter-coded ones, and that intra-coded slices remove references to previously coded pictures. The first of this issues is solved by enabling the ConstrainedIntraPrediction Flag on the sequence level. The second one is solved by using IDR (Instantaneous Decoder Refresh) slices instead of Intra slices. IDR pictures, which are formed completely by IDR slices, empty memory buffers from any short-term reference frames, and so avoid that subsequent pictures make reference to other pictures that are older than themselves. This is to make sure that error drifting is completely eliminated.

Finally, we should refer that H.264 also integrates a test model for lossy environments. It uses a loss-aware rate/distortion optimized coder that, based on probabilistic loss analysis, makes the best decisions about whether to apply inter or intra coding to a specific macroblock.

Picture Segmentation

Picture segmentation is achieved in the form of slices. A slice is a group of macroblocks; it can be as small as a single macroblock or as big as the entire picture. By using a higher number of slices per picture more overhead is generated but, on the other hand, error resilience improves. This improvement is due to the fact that macroblocks from one slice can not be used to predict macroblocks from another slice, which effectively reduces error propagation, and also due to the fact that each slice carries its own parameter set. The slice size is typically set so that it can adapt to the network's MTU, thereby assuring that the information on each packet can be decoded independently.

Macroblocks are assigned to slices in scan order, and every macroblock must be assigned to a single slice. The only exceptions to this rule are Redundant Slices and Flexible Macroblock Ordering (both described below).

Reference Picture Selection

The use of multiple reference pictures can also be considered an error resilience mechanism of H.264. However, its application will depend on the characteristics of the video transmission system. If feedback mechanisms are available then the destination can notify the encoder when parts of reference frames are lost, making it use older pictures for prediction instead of directly using intra-coding of data. When a feedback channel is not available we can make use of video redundancy coding to provide alternative information in the presence of losses.

Data Partitioning

The data partitioning process allows differentiating data according to its degree of importance. The purpose is allowing more relevant data to receive a higher QoS from the underlying networks, thereby minimizing the impact of losses.

The H.264 framework differentiates data into three partition types: A, B and C. The A type is the most important one, and contains header information which includes the types of macroblocks used, the quantization parameters and the motion vectors applied. Without this information the data of the two remaining partition types can not be decoded. Also, in case information of partition types B and C is missing, the available header information does still allow to improve the error concealment efficiency. This is due to the availability of the MB types and the motion vectors, which are able to offer a relatively high reproduction quality since only texture information is missing.

The B type is a partition containing intra information, more specifically Intra CBPs (coded block patterns) and Intra coefficients. Despite it requires type A data to be available for its information to be useful, this layer is important since the intra information it provides allows to reset error propagation.

The least important partition type - type C - contains inter information, more specifically Inter CBPs and Inter coefficients, which typically occupies the largest share of a slice. This partition, as occurred for type B, requires information from partition type A to be available for its information to be useful, but not information from partition type B. The information from this partition (C) is the least important one since it does not allow coder and decoder to re-synchronize (with feedback) or the error propagation to be reset (no feedback).

Flexible Macroblock Ordering (FMO)

Flexible macroblock ordering is an error resilience technique designed to avoid the impact of losing information from a same sector of a picture, making error-concealment in that zone a very difficult task.

The FMO technique consists in assigning macroblocks to slices in non-scan order, which requires using a macroblock allocation map. The purpose is to avoid that neighbor macroblocks belong to a same slice. Though this technique offers very good results in terms of error resilience, it typically causes the in-picture prediction to become unavailable, which results in a lower coding efficiency. For highly optimized environments, the macroblock re-arranging process also has an impact in terms of delay.

Redundant slices (RSs)

The use of redundant slices is particularly useful in highly error-prone mobile environments. Redundant slices consist of repeating macroblocks with different coding parameters on the same bit stream. For example, a macroblock could be quantized for high quality for its primary representation, and quantized for low quality on a redundant representation. This way, when the primary data arrives, it is used for the decoding process; however, in case the reception of primary data fails, the redundant representation, if available, is used to fill-in the information gap. Therefore, higher error resilience is achieved with a small cost in terms of additional bandwidth required.

3.2.5 H.264 over IP

The use of H.264 over IP networks has three main purposes: conversational applications, download of videos and non-real-time video streaming.

Conversational applications such as videotelephony and videoconferencing are characterized by imposing low latency values. These require using real-time video encoders and decoders, which restricts the type of tools used for coding (e.g. bipredicted slices) and also for error-resilience.

The download of complete video streams typically uses reliable protocols for data transfer, and the video coding process is typically not done in real time. Therefore, the complexity of the encoder and the time taken to generate the encoded stream are not an issue, which allows achieving higher coding efficiency.

The third application - IP-based streaming - is a technology than is not so demanding as conversational applications, but that has more requirements than a mere video download. Video data is typically played-back a few seconds after transmission begins, and the video stream is either pre-recorded or transmitted on demand. Anyway, the information is typically sent with different encoding parameters to the different users, so that the video characteristics can adapt to the user's bandwidth and display type. Also, it can use unicast, multicast or even broadcast.

In terms of the network protocols used, IP networks operate over a great number of different physical and link-layer technologies. One of the issues that must be taken into account, however, is the maximum transmission unit (MTU) on an end-to-end path. The size of coded slices should be close, but never superior, to the MTU size for optimum performance. By surpassing this value data would be fragmented, which would cause packets to be dropped entirely if one of the fragments is lost.

Except for downloaded videos, H.264 is typically transported in IP networks using a combination of IP/UDP/RTP. This combination is optimal is terms of service, but introduces an overhead for each packet equal to 20+8+12 = 40 bytes. The choice of UDP instead of TCP is mandatory when supporting applications with real-time requirements because only UDP offers a connectionless service; the UDP header also contains an error detection field. Concerning the RTP protocol, it also offers several advantages to H.264 streams such as packet loss detection, time stamping, marker bits and other administrative information. RTP packetization issues in the scope of the H.264 framework are the topic of the next section.

3.2.6 RTP packetization

H.264's network adaptation layer (NAL) is prepared to offer full support for RTP packetization. In fact, H.264's NAL also generates the RTP header.

A NALU (NAL unit) consists of a byte string of variable length that contains syntax elements of a certain class. Examples of such classes are slices, data partitions and parameter sets.

When designing the packetization system several issues must be taken into account. The first one is that fragmentation into RTP packets must generate a low overhead so that MTU sizes as small as 100 bytes are possible. Also, distinction between essential and non-essential RTP packets should be possible without decoding the bit stream, so that undecodable data may be discarded. Finally, both NALU fragmentation into multiple RTP packets, as well as aggregation of several NALUs in a single packet, should be supported.

NALU fragmentation may become necessary when the encoder can not produce NALUs smaller than the network's MTU. In that situation the IP protocol will be responsible for packet fragmentation and reassembly, which invalidates some of the error recovery techniques made available on the H.264 framework.

NALU aggregation becomes relevant when there are NALUs that contain only a very small number of bytes. An example of such NALUs are those which contain solely parameter sets. In those cases it is interesting to join NALUs into a single RTP packet to avoid the overhead introduced by the different lower-layer protocols used.

In the H.264 framework both the fragmentation and the aggregation techniques have been designed so that media aware network elements can perform fragmentation, aggregation, and the respective inverse functions without parsing the media stream beyond the NALU header byte.

The use of RTP to support H.264 video transmission in IP networks offers several advantages. By using RTP's sequencing information the receiver is able to find out when packets have been lost, reordered, and it can also detect duplicated packets. The latter could possibly be sent by the source on purpose to achieve error-resilience through data redundancy. Concerning reordered packets, these are typically not a problem for the decoder in the scope of H.264, except when using the main profile which requires all packets to be ordered. Finally, the timing information sent in RTP packets simplifies the rendering process by allowing the decoder to know the exact time for picture presentation.

3.3 Conclusions

In this chapter we presented the evolution of video coding standards, which has culminated in the recent H.264/MPEG-4 Part 10 specification. We analyzed the main features of H.264, evidencing its error resilience tools as well as its good support for IP-based networks.

In terms of error-resilience we analyzed the purpose of the different tools that H.264 offers, relating them to their fields of applicability.

Concerning support for IP-based networks, we referred to the most common application for H.264 that can be found in these networks. We then explained the purpose of the NAL, evidencing its good integration with lower protocol layers. Specifically, we referred to the packetization process when relying on a combination of IP/UDP/RTP for video transmission.

The outstanding improvements introduced by the H.264 standard have made it the video technology of choice for the work developed in this thesis. The next chapter is dedicated to assessing the performance of the H.264 technology both in terms of data compression and error resilience.

Chapter 4

Improving H.264's performance in ad hoc networks

In the previous chapter we analyzed the main characteristics of the new H.264/MPEG-4 Part 10 technology. The current chapter is dedicated to tuning the H.264 codec for optimal operation in wireless ad hoc networks. Our strategy consists of starting with a general purpose performance analysis to assess the impact of the different coding tools available on H.264's reference software. We then proceed by analyzing the effectiveness of the different error-resilience tools available in the presence of random packet losses. Our work also includes simulations of packet loss bursts in order to test tools like flexible macroblock ordering for short bursts, and multi-frame prediction for longer bursts.

We end the study of this chapter by evaluating the performance of H.264's error resilience tools on a MANET environment. This evaluation will be done using the results found in preceding sections, so as to use those options which get better performance results. Finally, we present our conclusions.

4.1 Tuning of the H.264 codec

In this section we evaluate the impact of the most relevant parameters related to the H.264 framework. The chosen parameters consist of *intra frame period, quantization parameter, search range, number of reference frames, macroblock line intra updates, B frames, SP picture interval, motion vector resolution, Hadamard transform, rate/distortion optimization, constrained intra prediction* and, finally, *CABAC vs. UVLC.*

This evaluation was done using H.264's reference software [H2603b]. The only restrictions in terms of available parameters were set by the selected version of the reference software, which is JM2.0. These restrictions, however, were essentially related with options to increase the robustness under packet loss (ex.: FMO

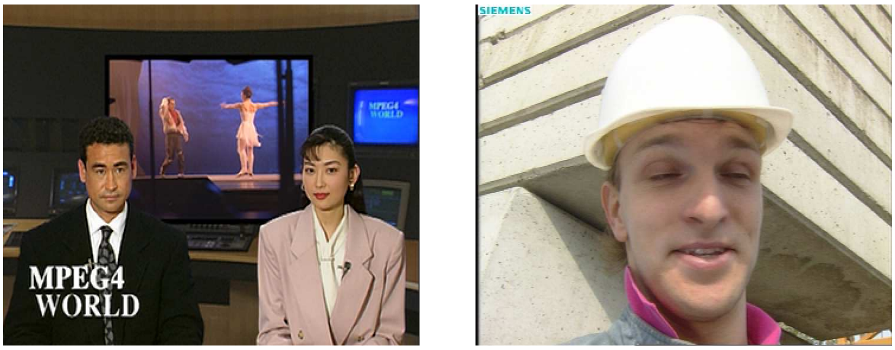


Figure 4.1: News (left) and Foreman (right) video sequences

reordering), a topic that will be discussed on later chapters only.

Concerning the test sequences used, these were the well known News and Foreman sequences (see figure 4.1) in the QCIF format. The need for different test sequences is related to different levels of movement between both, which results, at times, in different conclusions. The H.264 codec was configured to bypass frames, so that these sequences, originally captured at 30 Hz, will be coded at 10 Hz (except when stated otherwise).

Results are shown in terms of the bitrate generated for the sequence, the total encoding time, and the PSNR (Peak Signal-to-Noise Ratio). Concerning the PSNR metric, it is the most widely used objective quality metric for video signals, being usually expressed in terms of the logarithmic decibel scale.

To compute the PSNR you need a source image I(i,j) containing m by n pixels, and a reconstructed image K(i,j) where K is reconstructed by decoding the encoded version of I(i,j). Error metrics are computed on the luminance signal only, and so the pixel values I(i,j) and K(i,j) range between black (0) and white (255).

The first step to obtain the PSNR value consists of computing the mean squared error (MSE) of the reconstructed image as follows:

$$MSE = \frac{1}{m \cdot n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \| I(i,j) - K(i,j) \|^2 \tag{4.1}$$

The actual PSNR value can then be derived using the following expression:

$$PSNR = 20 \cdot log_{10} \frac{255}{\sqrt{MSE}} \text{ (dB)} \tag{4.2}$$

PSNR values in a range between 20 and 40 dB are typically considered useful for evaluation, being other values outside this range considered either too bad or unnecessarily good.

4.1.1 Intra Frame Period

Intra frames are coded independently from other frames, using simply what is called intra coding. This coding is similar to the one used for static pictures, which consists of methods which simply take care of spatial redundancy issues. By not

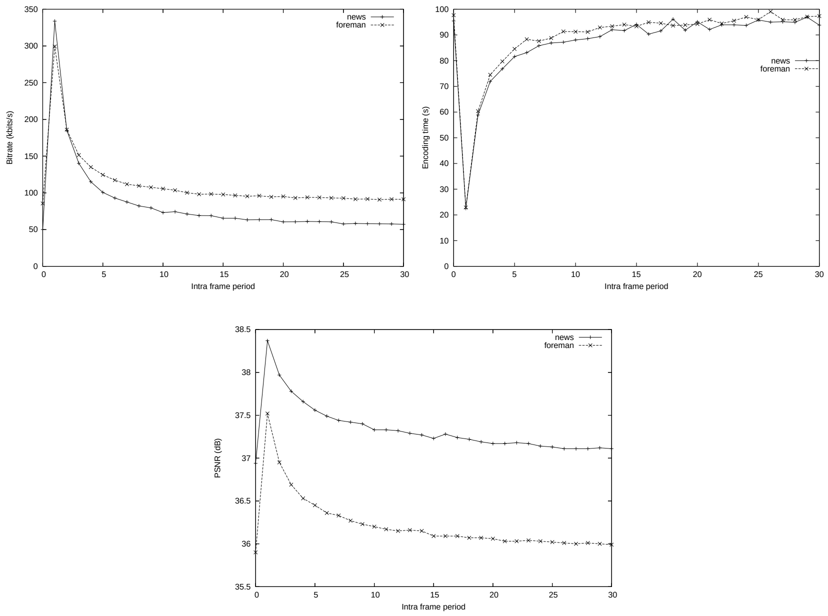


Figure 4.2: Bit rate (top-left), encoding time (top-right) and PSNR variation (bottom) when varying the GOP size.

taking advantage of temporal redundancy, the compression rates are typically low compared to other frame types (e.g. B and P). Intra-coded frames are particularly important in terms of error propagation, since the introduction of an I frame will stop the propagation of errors. Hence, the prediction process can start again from an error-free situation. So, our first analysis consists of determining the behavior of the H.264 codec when varying the GOP size (distance between two consecutive I frames in number of frames).

In our experiments our GOP size (Intra frame period) ranged from 1 to 30 frames; we also tested the value zero, which consists in not using I frames except for the first one. The results from our experiments are shown in figure 4.2.

As it can be appreciated in that figure, using mostly intra coded frames results in small encoding times but very high bit rates, as expected. The peak variations in PSNR are not very significative, and are due mostly to the differences in the quantization parameters between I and P frames.

An analysis of the graphics included in figure 4.2 evidences that using a period of 15 (typical MPEG 2/4 GOP size) or more has almost no effect in terms of bit-rate and PSNR relative to a solution where no intra frames are used. This aspect is particularly interesting in terms of error-resilience, where intra-coded frames offer better results.

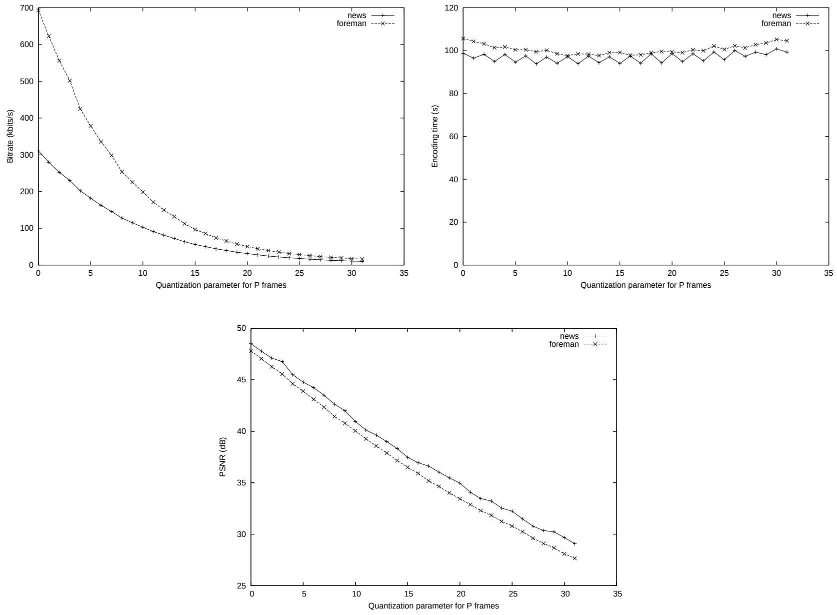


Figure 4.3: Bit rate (top-left), encoding time (top-right) and PSNR variation (bottom) when varying the quantization parameter for P frames

4.1.2 Quantization parameter for P frames

A real-time video transmission relies essentially on I and P frames, being the latter the most frequent. This means that the quantization of those frames mostly defines the quality of the output signal. Therefore, we will now focus on the performance of the H.264 codec in terms of quantization of P frames.

Our evaluation of P frames was done using only a single I frame at the beginning. We tested the full quantization range, that goes from 0 to 31. Figure 4.3 shows that the encoding time is quite stable, and that PSNR decreases linearly with increasing quantization as expected. It is the bit rate curve which gives more relevant data since non-linearities appear. The quantization value of 15 seems to be the frontier between a zone where the actual data is dominant and a zone where other data such as RTP and stream headers dominate the bit-rate curve.

This bit-rate non-linearity relative to P frames gives a hint on how to choose the best option for streaming. In general, quantization values between 15 and 20 seem appropriate in terms of both bitrate - it is quite low already - and PSNR - the quality is still very good (close to 35 dB).

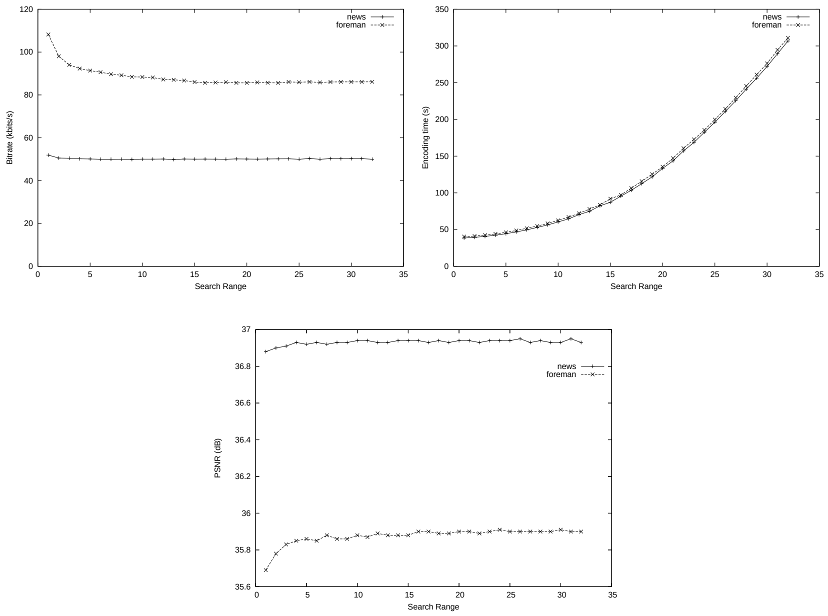


Figure 4.4: Bit rate (top-left), encoding time (top-right) and PSNR variation (bottom) when varying the Search Range

4.1.3 Search Range

The search range sets the searching area in what refers to motion estimation methods. A greater search area typically achieves greater estimation accuracy; the cost of that improved accuracy is an increased estimation time. Relatively to this parameter, the tests performed included all possible values for it, which range from 1 to 32.

As it can be seen in figure 4.4, the encoding time grows almost exponentially with increasing search range; concerning PSNR and bitrate variations, these are minimal because the quantization values are not changed. We can see, though, that both are slightly affected, being that the bitrate becomes lower and the PSNR becomes higher as the search range increases. This phenomena is related to the motion degree of a video sequence. In our example the *foreman* sequence presents higher motion than the *news* sequence. As motion increases, the search range must also be increased to achieve the same estimation accuracy. If the estimation accuracy improves then the required bitrate will be decreased; this is mainly because the rate/distortion decision block will decide to intra code blocks less often. Also, the quality of the signal will be slightly increased since the residual values from prediction will be smaller.

These factors can be clearly seen in the PSNR plot, where the difference between the minimum and maximum values is about 0.1 dB for the News sequence and about 0.2 dB for the Foreman sequence (higher level of movement). In terms of bitrate the situation is equally clear, being the difference between the minimum and maximum values of about 3 and 22 kbit/s for the News and Foreman sequences respectively.

We find that the News sequence reaches values close to optimum for a search range of 4, while the Foreman requires a search range of 7. This, as referred before, shows the impact of the motion degree of a video sequence on codec requirements.

4.1.4 Number of Reference Frames

The use of multiple reference frames in the motion estimation process aims at obtaining better results in terms of both compression and robustness.

The H.264 standard offers the option of employing multiple reference frames in inter picture coding. Up to five different reference frames can be selected, resulting in better subjective video quality and more efficient coding of the video sequence. Moreover, using multiple reference frames might help making the H.264 bit stream error resilient. As shown in figure 4.5, using multiple reference frames significantly increases the complexity since the encoding time will increase proportionally to the number of reference frames used. We have therefore a trade off between bit-rate/PSNR and encoding time, which deserves scrutiny since the obtained bit-rate reduction is not very significative and the maximum PSNR increase does not surpass 0.1 dB only (Foreman sequence). The reasons for the improvements observed are the same as for the search range: better prediction accuracy.

To support multiple reference frames, both encoder and decoder require much more memory resources. In general, we could say that it is a good choice for asymmetric video distribution (as in DVDs), but its use among devices with limited resources, such as PDAs or mobile phones, may become prohibitive. So, the choice of whether to use them or not will depend mostly on their error-resilience capabilities, which is a topic focused on later sections.

4.1.5 Macroblock Line Intra Updates

This parameter, when active, instructs an H.264 encoder to intra-code a GOB (Group of blocks) every N frames. Its usefulness is related to stopping error propagation as in I frames, but now this behavior is distributed throughout the sequence. So, a similar effect is achieved in terms of error resilience, but the bit rate is kept at more constant values. In our experiments we have tested values for the MbLineIntraUpdate parameter ranging from 1 to 30; a value of zero means that this option is not used, and so we also depict it for reference.

As it can be seen in figure 4.6, the bit rate constantly decreases as we increase the MbLineIntraUpdate value from 1 to 30; the PSNR values behave the opposite way, showing an increase instead. For a MbLineIntraUpdate value greater than 10 the bit-rate is almost unchanged, though differences of a few tenths of dBs can still be noticed for the Foreman sequence. The results relative to PSNR also evidence that the quantization values used in the intra update process are different from those globally defined. Should the PSNR be kept constant, the peak experienced in terms of bit-rate would be even higher.

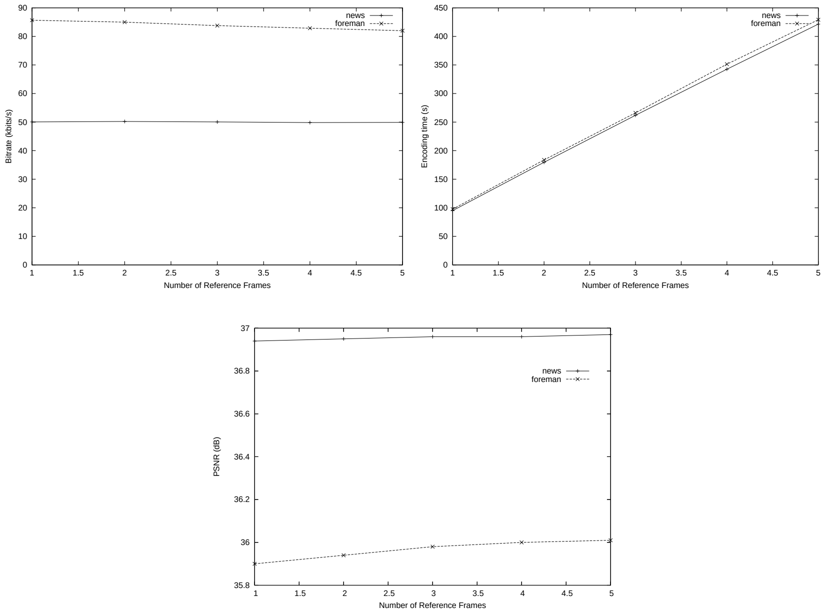


Figure 4.5: Bit rate (top-left), encoding time (top-right) and PSNR variation (bottom) when varying the number of reference frames

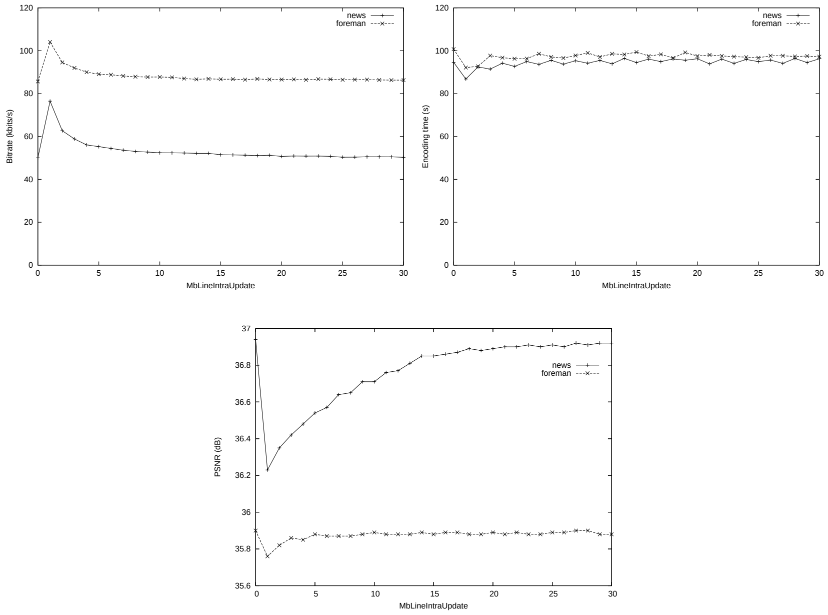


Figure 4.6: Bit rate (top-left), encoding time (top-right) and PSNR variation (bottom) when varying the amount of lines using Intra Macroblock Updating

4.1.6 B pictures

The use of bidirectionally coded pictures is not appropriate for real-time video transmissions due to the temporal restrictions inherent to real-time communication. However, IP-based video streaming does not have such strict restrictions, which means that B frames can be used within certain bounds.

In this section we will compare two options for using B frames based on the configuration used until now. So, the first experiment consists in analyzing the impact on the codec and on the output data of switching the frame rate from 10 Hz to 30 Hz. In this strategy two B frames will be introduced between previously available I and P frames, so that the GOP will be changed from IPPPPP... to IBBPBBPBBPBBP....

In our second experiment we will maintain the frame rate at 10 Hz, but we will use either one or two B frames between I/P frame types so as to assess the PSNR improvements for a same output bitrate.

Quantization parameter for B pictures @ 30 Hz

The use of B pictures for higher frame rates makes sense for high quality video encoding systems. However, encoding B frames significantly increases the encoding time, which means that a higher performance processor system should be employed when encoding time is critical. As shown in figure 4.7, the use of B pictures increases quite significatively the encoding time, which means that a highly efficient codec will have to be designed for the support of B pictures in devices like PDAs or mobile phones. In fact, the encoding time has grown from about 100 seconds using the previous configurations, to values above 450 seconds; notice that, despite we had a three times frame rate increase, the encoding time has grown by more than four and a half times.

Bit rate and PSNR graphs show that there is a "knee effect" when quantization values reach 15, leading us to consider that for values above 15 the improvement on bit-rate does not compensate for the PSNR loss.

B pictures @ 10 Hz

The use of B pictures allows the encoder to achieve much greater compression values than regular P pictures do. However, each B picture needs on average twice more processing than P pictures do. Also, due to the bidirectional prediction performed on B pictures, the transmission of real time video can be affected since each B frame will need to reference future P or I pictures, especially on the encoder side. This will restrict its use severely for real time applications, contrarily to what happens on stored video where large GOPs with abounding B pictures are commonly used.

In our experiments we tested the impact of changing the frame sequence from IPPPPPP... to either IBPBPBPBP.... or IBBPBBPBBP....

As shown in figure 4.8, mixing B frames will produce substantially better results as expected, even in the simplest case (left).

In general, the use of B frames for real-time video streaming will be restricted due to the temporal restrictions that have to be met on either the encoder, the

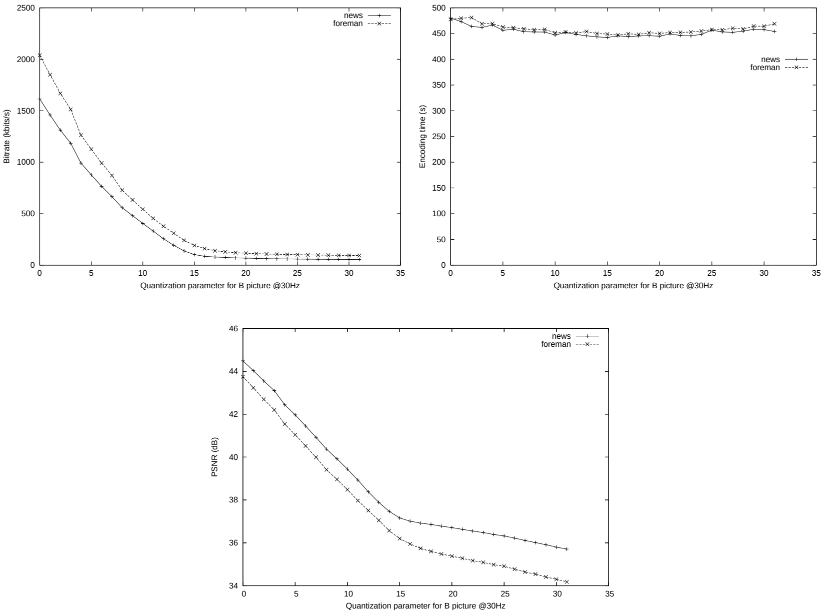


Figure 4.7: Bit rate (top-left), encoding time (top-right) and PSNR variation (bottom) when varying the quantization parameter for B pictures

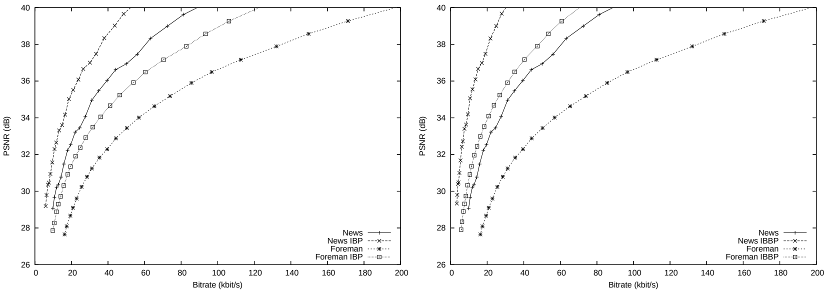


Figure 4.8: Bitrate vs. PSNR results when introducing 1 (left) and 2 (right) B frames on the GOP.

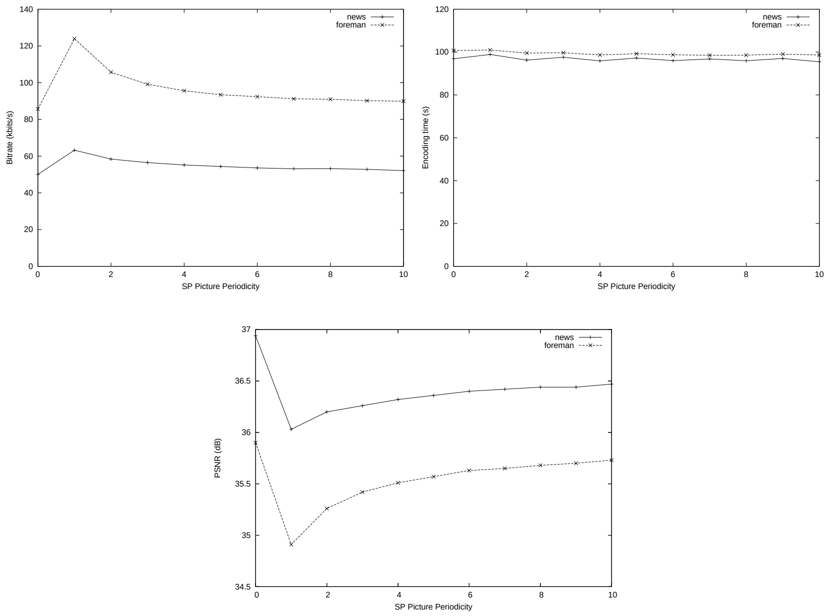


Figure 4.9: Bit rate (top-left), encoding time (top-right) and PSNR variation (bottom) when varying the SP picture periodicity

decoder or both. Since the subject of this thesis is centered on real-time video communication, we will drop the use of B frames in our experiments from now on.

4.1.7 SP Picture Interval

S-pictures are a special frame type designed to provide functionalities such as bit-stream switching, splicing, random access, VCR functionalities such as fast-forward and also error resilience/recovery.

There are two types of S-pictures, namely SP-pictures and SI-pictures. SP-pictures make use of motion compensated predictive coding to exploit temporal redundancy in the sequence similarly to P-pictures; SP-picture coding allows identical reconstruction of a frame even when different reference frames are being used, which is very useful for switching between similarly coded sequences. SI pictures, on the other hand, are used to switch between completely different video sequences by using 4x4 Intra Prediction modes instead of inter-frame prediction.

The version of the H.264 codec used supports SP pictures only, and so we will focus on the error-resilience features of SP pictures alone. Our tests consisted of varying the periodicity of this frame type in a range from 1 to 10; a value of zero means that SP pictures are not used, and it is our reference.

As it can be seen in figure 4.9, the effect of introducing SP pictures in terms of

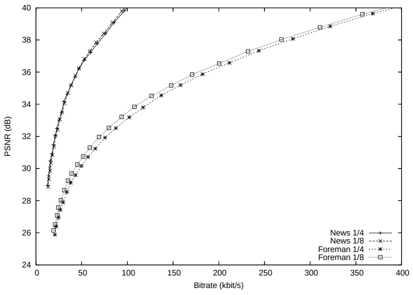


Figure 4.10: Bitrate vs. PSNR results comparing 1/4 to 1/8 Motion vector resolution

bit-rate and PSNR is essentially the same as the one provoked by the introduction of I frames. Therefore, choosing this kind of frames seems more appropriate in situations where other properties of SP frames, such as random-access, are also desired. In a later section we will revisit SP pictures to assess their properties in terms of error-resilience.

4.1.8 Motion vector resolution

The motion vector resolution sets the granularity allowed in prediction tasks. Since the H.264 framework introduces sub-pixel resolution, our encoder may be tuned to use either 1/4 th or 1/8 th of a pixel accuracy. Our tests consisted in assessing the goodness of either option for different data rates; these different data rates have been generated by varying the quantization parameter used.

As it can be seen in figure 4.10, higher motion vector resolution leads to slightly better results, being the improvement more significant on video sequences with medium/high degrees of movement. However, these improvements are only slight and are not worth the significant increase of complexity of motion estimation.

4.1.9 Hadamard transform

The H.264 encoder allows the use of either the Hadamard transform or the DCT transform, which was the one used by default in previous standards (e.g., MPEG-2 and H.263). The Hadamard transform offers greater compression levels, though it introduces a slight processing overhead.

In our experiments we compared both transforms at different data rates by varying the quantization values.

As it can be seen in figure 4.11, the Hadamard transform achieves better results than the DCT as expected, which validates and justifies its integration into the H.264 standard.

This improvement, though, does not surpass 0.5 dB for the same bit rate, which means that perhaps there is little room for improvement using this family of transforms.

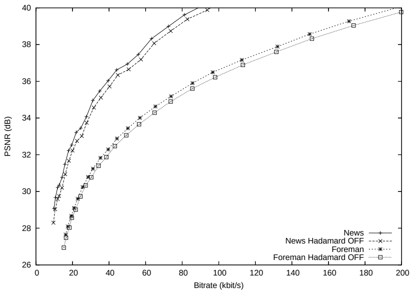


Figure 4.11: Bitrate vs. PSNR results comparing Hadamard and DCT transforms

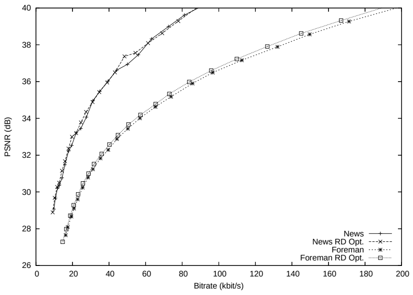


Figure 4.12: Bitrate vs. PSNR results when applying Rate Distortion Optimization algorithms

4.1.10 Rate Distortion Optimization

The rate-distortion optimization mechanism made available by the H.264 codec consists of an enhancement to the prediction process, so that the codec is able to compare the compression achieved by performing prediction tasks for a certain macroblock, in contrast to the choice of intra coding.

Our experiments concerning this parameter consisted in comparing the improvements in terms of signal distortion at different bitrates.

As shown in figure 4.12, this method achieves better results for sequences with high levels of movement, even though the gain is small. However, in terms of robustness, intra-updated macroblocks are always a better solution. This means that by activating Rate Distortion Optimization there are gains in terms of both bit-rate and error-resilience. This aspect will be addressed again in later sections.

4.1.11 Constrained Intra Prediction

The aim of constraining the intra prediction process is to increase the error resilience by avoiding the use of inter macroblock residual data and decoded samples from neighbor macroblocks for the prediction of intra macroblocks. This way the

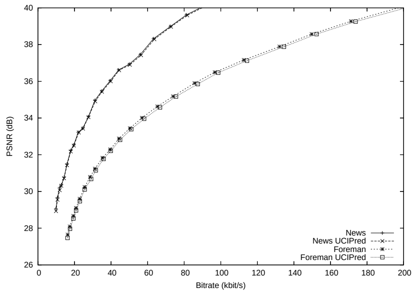


Figure 4.13: Bitrate vs. PSNR results when enabling the Constrained Intra Prediction parameter

loss of data will not affect negatively intra predicted macroblocks, which results in an effective method to block the propagation of errors. As shown in figure 4.13, the rate-distortion results are only slightly affected by activating this parameter.

In terms of error-resilience, though, this parameter may show good results, as exposed in later sections.

4.1.12 CABAC vs. UVLC

Universal VLC Entropy coding, based on the use of Variable Length Codes (VLCs), is the most widely used method for the compression of quantized transform coefficients, motion vectors, and other encoder information. VLCs are based on assigning shorter codewords to symbols with higher probabilities of occurrence, and longer codewords to symbols with less frequent occurrences. The symbols and the associated codewords are organized in look-up tables, referred to as VLC tables, which are stored at both the encoder and decoder. In some video coding standards such as H.263, a number of VLC tables are used, depending on the type of data under consideration (e.g., transform coefficients, motion vectors). H.264 offers a single universal VLC table that is to be used in entropy coding for all the symbols in the encoder, regardless of the type of data those symbols represent. Although the use of a single UVLC table is simple, is has a major disadvantage, which is that the single table is usually derived using a static probability distribution model, ignoring the correlations between the encoder symbols.

Context-Based Adaptive Binary Arithmetic Coding (CABAC) makes use of a probability model at both the encoder and decoder for all the syntax elements (transform coefficients, motion vectors). To increase the coding efficiency of arithmetic coding, the underlying probability model is adapted to the changing statistics within a video frame, through a process called context modeling.

Our experiments consisted in comparing the performance in terms of signal distortion between the CABAC and UVLC algorithms at different data rates.

Figure 4.14 shows that using CABAC the performance increases up to 1 dB for the same bit-rate, or reduce the bit-rate as much as 14,4% for the same distortion (Foreman). Overall, we consider that this distortion gain justifies the increase in terms of processing overhead.

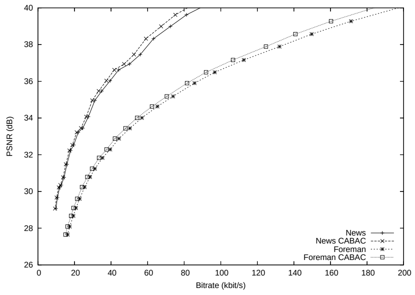


Figure 4.14: Bitrate vs. PSNR results when comparing CABAC and UVLC algorithms

4.2 Improving H.264's resilience against random packet losses

H.264 makes available error resilience mechanisms both on the encoder and on the decoder side. In the encoder we find several parameters that can be tuned. So, a trade-off between compression rate and error resilience can be made targeting the different types of problems found in heterogeneous environments.

Random intra macroblock updates and the insertion of intra-coded pictures (I frames) are the most commonly used methods to stop the temporal propagation of errors when no feedback channel is available. While intra frames reset the prediction process, thereby avoiding error propagation, their use has a generally high bandwidth cost causing also severe bit rate variations. The use of random intra macroblock refreshes is more effective than I frames because they help to achieve CBR-like streams. Moreover, random intra-updating of macroblocks provides more constant quality in terms of signal distortion by statistically reseting the error for each of the macroblocks. The Macroblock Line Intra Update is another robustness option where a group of blocks will be intra coded every N frames. It is just another form of macroblock updating.

The use of slices is also a very important method to improve robustness by stopping spatial error-propagation. The macroblocks belonging to a slice can be decoded independently from other slices since no inter-slice dependencies are allowed. In our work we have used slices intensively since this mechanism is closely related to the RTP packetization process performed by the encoder.

Another method which deserves consideration is Flexible Macroblock Ordering (FMO), whereby the sender can transmit macroblocks in non-scan order. This method, although similar to slice interleaving, provides much greater flexibility and can be tuned to be more effective in terms of error resiliency. It aims essentially at dealing with packet loss bursts.

SP slices make use of motion-compensated predictive coding to exploit temporal redundancy in the sequences, like P slices do. Unlike P slices, however, SP slice coding allows identical reconstruction of a slice even when different reference pictures are being used. They aim essentially at bit stream switching, splicing, random access, VCR functionalities and error resilience issues.

Rate Distortion Optimization [TDT02] is yet another tool integrated in the H.264 framework related to error-resilience. If the prediction process does not offer good results, this element allows to intra-code a pixel block instead. Concerning encoder tuning, it can be set to OFF for no optimization and ON if such optimization is desired. However, such values will only be optimal in the absence of errors in the network. For that reason, a third mode is available where the encoder takes into account the expected packet loss rate of the network, as well as the decoder's methods to cope with errors in order to decide weather to intra or inter code a block. See [TS02] for more details on that subject.

The constrained intra prediction option is related with the H.264 intra prediction mode. When it is active avoids using inter macroblock pixels to predict intra macroblocks.

Multi-frame compensation prediction is another tool targeting to increase both compression performance and error resilience, since the loss of an entire reference frame will have less critical effects on later predicted frames [MJ97, MJ01].

Concerning the decoder, it also plays a fundamental role in error resilience since it is responsible for error concealment tasks. With that purpose it keeps a status map for macroblocks which indicates, for each frame being decoded, weather a certain macroblock has been correctly received, lost or already concealed. The methods used vary between intra and inter frames. For intra frames the task mainly consists of performing a weighted pixel averaging on each lost block in order to turn it into a concealed one. For inter frames the task performed consists mainly of guessing the adequate motion vector for lost macroblocks, although intra-style methods can also be used. For a more complete description of such methods please refer to [YMV$^+$02].

The decoder also has other tasks like handling multiple reference frames or entire frame losses.

As exposed in [TS02], the reference decoder for H.264 does not incorporate bit error resilience features since it increases significantly the complexity of the decoder, with only slight improvements as a result. Therefore, bit error detection and handling has to be processed externally.

The evaluation done in this chapter aims at verifying the effectiveness of the robustness tools developed under the H.264 framework. To achieve this we used version JM3.9a of the H.264 reference software.

In the previous chapter we picked the News and Foreman sequences for our evaluation. Now we additionally introduce the Bus sequence (see figure 4.15), which has a higher degree of movement than previous test sequences, so that the impact of losses can become more noticeable. The three video sequences are in the QCIF format and are 10 seconds long. Our study employs a 10 Hz frame rate.

4.2.1 Rate control

In order to perform a detailed evaluation of the H.264 codec we are going to work with different configuration parameters, different test sequences and different packet loss rates. In figure 4.16 we show the selected performance metrics we are going to employ.

Figure 4.15: The *bus* video sequence

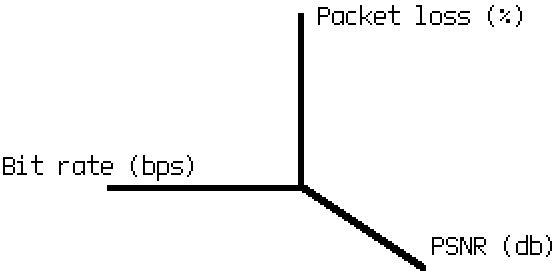


Figure 4.16: Parameters of interest in error-resilience evaluation.

Making our analysis as straightforward as possible requires fixing one of the three configuration parameters so that it becomes possible to depict the remaining two on a bidimensional plane. Since we are interested in observing the evolution of the PSNR decay with increasing packet loss rates, we opt to use a fix bitrate value for each sequence.

The aforementioned approach requires using a rate control mechanism. Since the H.264 codec used does not have such a mechanism, a program in Perl was created with that aim.

The Perl rate controller is an external mechanism that tests different quantization values until it achieves a value that best matches the bit-rate selected by the user. The next sections will present not only the error-resilience results, but also the results of the rate controller here described.

4.2.2 The *Video Robustness* parameter

Our evaluation of the H.264 codec was done using both PSNR measurements and a Robustness parameter (R) that we defined. This parameter, contrarily to PSNR, does not aim at providing a measure of the quality of the sequence but, instead, offers a mean through which the ability to sustain the image quality in the presence

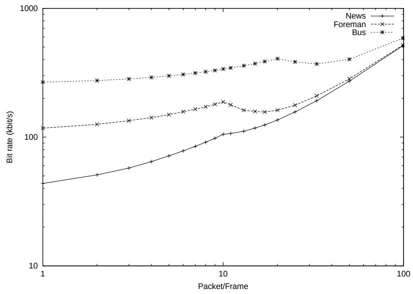


Figure 4.17: IEEE 802.11 bit rate vs. Packets/Frame for the three test sequences.

of error is quantified.

This parameter has been defined as:

$$R = \frac{1}{N} \cdot \sum_{i=0}^{N} \frac{MSE^{i}_{error-free}}{MSE^{i}} \text{ , 0< R } \leq 1 \tag{4.3}$$

In the absence of any kind of error the Robustness will remain at the maximum value of 1. As the error-rate increases, the R values decrease quadratically down to a minimum of 0.

4.2.3 Bit rate vs. Packets/Frame

When streaming video through a network there is always a need for data packetization. This process can be tuned in order to obtain optimal performance in terms of throughput and error-resilience.

From the error-resilience point of view, fine-grain packetization will be optimal since the lost of a packet is translated in the loss of just a small amount of data. From the network point of view, though, fine-grain packetization means higher bit-rate, and so it is prone to increase congestion.

Figure 4.17 shows the trade off between packetization and bit-rate. The bit-rate is calculated considering the actual IEEE 802.11 MAC-layer headers, including the overheads introduced by RTP, UDP and IP. As it can be seen, the curves are not linear presenting a peak at some point. This is due to the relationship between the number of macroblocks in a frame and the packetization process itself.

In this work we have configured the H.264 codec to use 7 packets per frame, except when stated otherwise.

4.2.4 Evaluation under loss

In this section we will perform different experiments to assess the error-resilience features of H.264. Our tests include the analysis of different intra-frame periods (GOP sizes), the impact of constraining the intra-prediction process, of using intra-updating of entire macroblock lines, of using a different number of reference frames, of introducing SP frames and also of performing random intra-updates of macroblocks.

Concerning the test sequences, we have chosen different target bitrates according to mid-scale quantization values as reference. This resulted in a target bitrate of 43 kbit/s for the *news* video sequence, 125 kbit/s for the *foreman* video sequence, and 275 kbit/s for the *bus* video sequence. Due to the coarse nature of the rate control method employed, the bitrates achieved will not always adjust perfectly to these target bitrates defined; this was, however, the best results possible using the available H.264 software.

Relatively to losses, these were simulated using a stochastic packet loss process following a uniform distribution. Since the packet loss probabilities are equal for all packets, large packet loss bursts are not prone to occur. We will assess the effects of loss bursts later, more specifically on section 4.3.

In all experiments the packet loss probabilities have been tested in the range from 0.1% to 20%, and we present our results in terms of both PSNR and the robustness indicator introduced before.

4.2.4.1 Results for I period variation

In this section we evaluate the impact in terms of error resilience of using different GOP sizes. GOPs start with an I frame followed by P frames. So, the GOP size is also the interval between consecutive I frames. As referred in the previous chapter, intra-coded frames reset the propagation of errors. This can be a very useful tool in environments with high packet loss rates.

Our experiments begin by testing the accuracy of the rate control mechanism when varying the interval between consecutive I frames; we test intervals between 1 and 30, including also the value 0 which corresponds to a situation where the only I frame used is the first one. We also present the corresponding PSNR results in order to appreciate the initial video distortion value for each I frame interval.

Figure 4.18 shows the output from the external rate controller, as well as the PSNR results for different intra frame periods. The rate controller's output shows more difficulty to maintain a constant bit-rate at higher bit-rate values due to the coarse granularity available. The intensive use of intra-coded frames has a negative impact on the PSNR value in no-loss scenarios, especially for the *news* sequence that has low levels of movement.

Using the values presented before as reference, we now proceed to study the impact of packet losses on these three sequences.

As shown in figures 4.19 and 4.20, the robustness and distortion results achieved by using low intra-frame intervals are in general quite superior. We find that, as expected, video quality will benefit from the intensive use of intra frames when the packet loss rate is high.

In general there will be an optimal I period for each sequence and for each packet loss rate. A method that would adapt itself to the sequence and network congestion would be the best one, but would require a feedback channel. However, in the absence of such method, we can say that using an I period ranging from 5 to 15 produces good overall results.

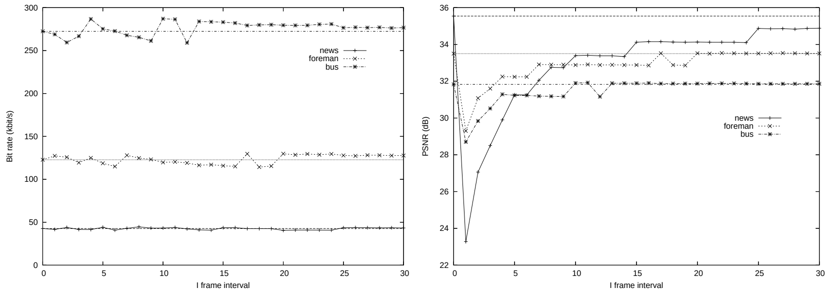


Figure 4.18: Bit rate and PSNR results for loop control with variable period for I frames

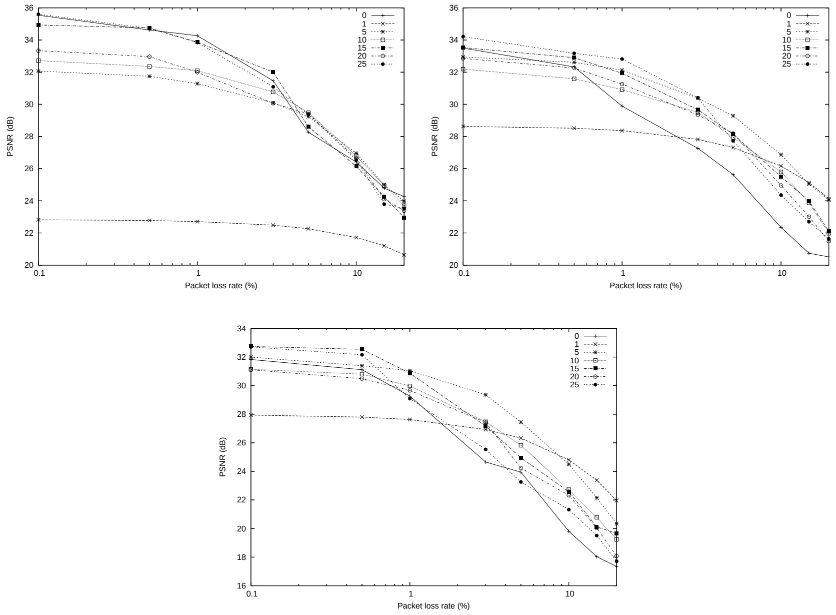


Figure 4.19: PSNR results with different I periods for News (top-left), Foreman (top-right) and Bus (bottom) sequences.

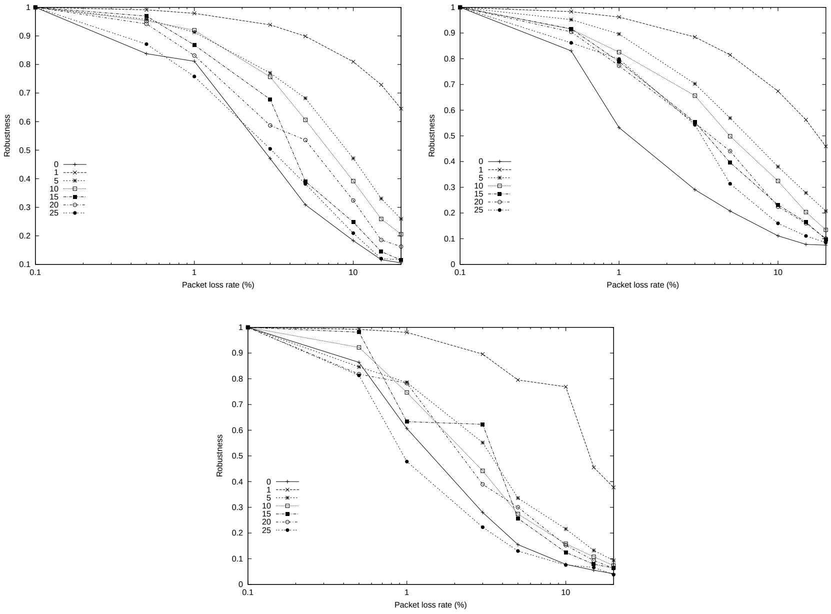


Figure 4.20: Robustness results with different I periods for News (top-left), Foreman (top-right) and Bus (bottom) sequences.

Table 4.1: PSNR and rate control results for Constrained Intra Prediction

	News		Foreman		Bus	
	Original	With CIP	Original	With CIP	Original	With CIP
PSNR (dB)	35,54	35,52	33,50	33,46	31,83	31,82
Bit-rate (kbit/s)	42,64	43,26	122,80	122,85	272,46	272,93

In terms of robustness, there is almost a direct relationship between the intra frames period and the robustness itself. The results obtained using the proposed metric are, as it can be seen, not the same as for the PSNR one.

4.2.4.2 Rate control results for Constrained Intra Prediction.

As stated in the previous chapter, there is a slight loss in picture quality for a same bitrate by constraining the intra prediction process to not using inter-predicted pixels. However, concerning error-resilience, it can be a factor that deserves serious consideration since the results achieved on packet-loss scenarios are theoretically better.

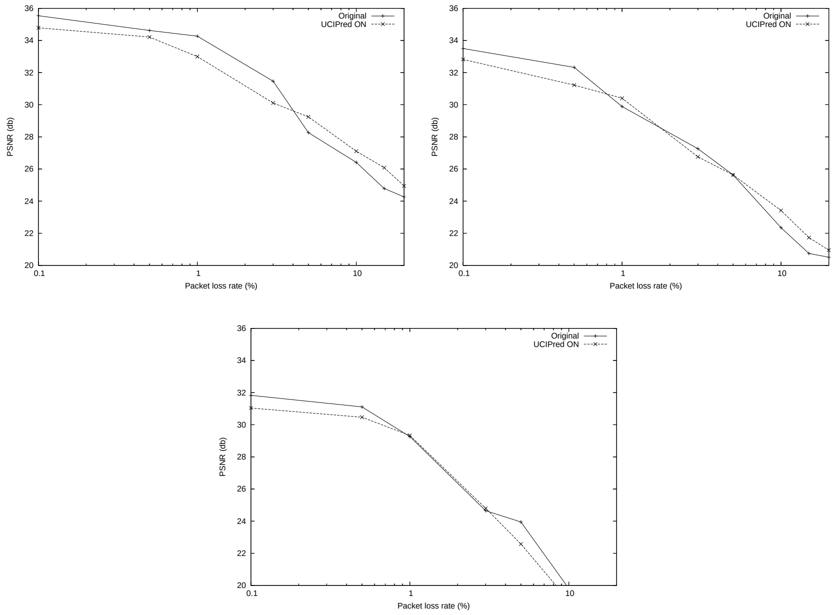


Figure 4.21: PSNR results with packet losses for News (top-left), Foreman (top-right) and Bus (bottom) sequences.

Our experiments were setup as previously so as to compare the benefits of using this technique. Table 4.1 shows the output of the rate controller in terms of both bitrate and distortion. As was supposed to happen, the use of Constrained Intra Prediction generates slightly worse distortion values and a minimum increase in terms of bitrate.

We now proceed to analyze the effectiveness of this error-resilience method by observing the decay in terms of both PSNR and robustness as the packet loss rate increases.

As it can be seen by inspecting figures 4.21 and 4.22, there is in general an increase in terms of both robustness and PSNR when the packet loss rate surpasses about 3-5%. The only exception is the Bus sequence, where the results are not clear as for the other two video sequences.

The results obtained justify the constraining of the intra-prediction process, being useful in scenarios and networks that suffer from intensive packet losses.

4.2.4.3 Macroblock Lines intra update

This parameter offers another method to intra code parts of a frame; more specifically, it allows to intra-code an entire macroblock line randomly chosen every N frames. Even though no benefits are obtained by using this option in a no-loss environment, the results change when the environment is error prone.

Our experiments consisted of varying the *MB Line Intra Updates* parameter

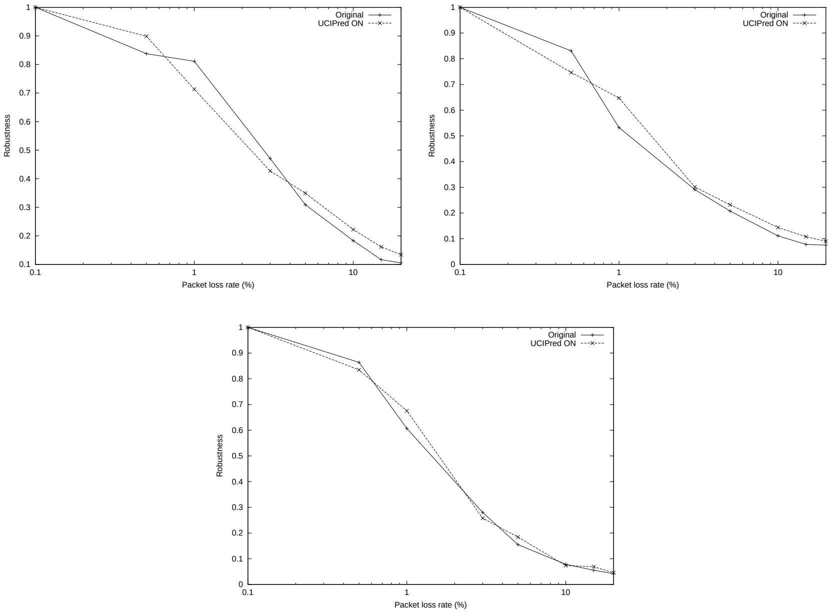


Figure 4.22: Robustness results for News (top-left), Foreman (top-right) and Bus (bottom) sequences.

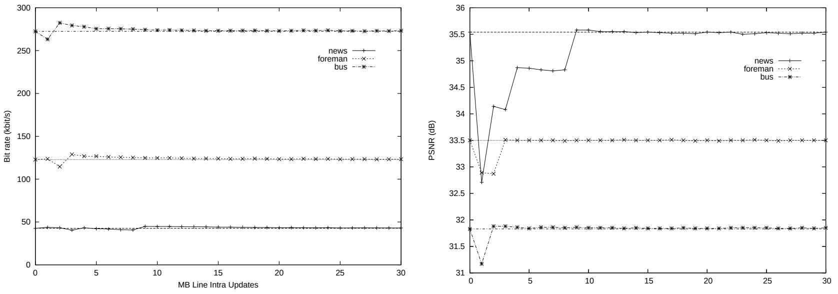


Figure 4.23: Bit rate and PSNR results for loop control with variable number of Macroblock Intra Updates

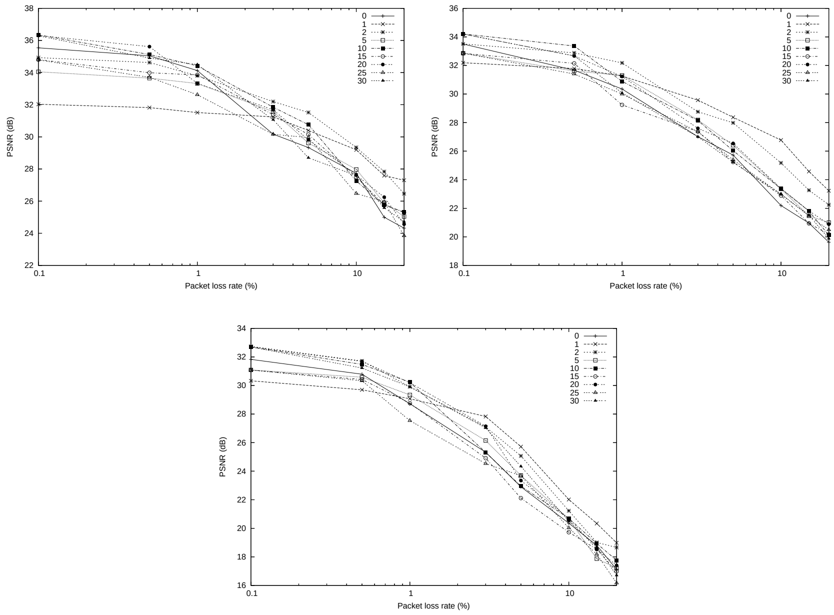


Figure 4.24: PSNR results with packet losses for News (top-left), Foreman (top-right) and Bus (bottom) sequences.

in a range from 1 to 30; the value of zero means that this technique is not used and works as reference.

The rate controller output is presented in figure 4.23, along with the corresponding PSNR results. As expected, very small values for this parameter cause the video distortion value to decay under no loss.

As shown in figures 4.24 and 4.25, the results obtained under loss are quite similar to those obtained using intra coded frames: higher updating frequency can lead to improved error resilience, especially on scenarios that provoke a high number of packet losses.

Concerning robustness we can generally say that, as happened with I frames, more frequent line intra updates result in higher robustness.

The performance observed with this method makes it a possible tool to apply when the packet loss rate is high and the GOP size is large.

4.2.4.4 Multiple reference frames

The use of multiple reference frames has two main purposes: to increase compression by improving the prediction process, and to increase error-resilience by offering a method that attempts to bypass those situations where a previous frame cannot be used for reference since it was lost. The price to pay is both increased complexity and buffer size at encoder and decoder.

Our experiments consisted of testing the error resilience achieved when using

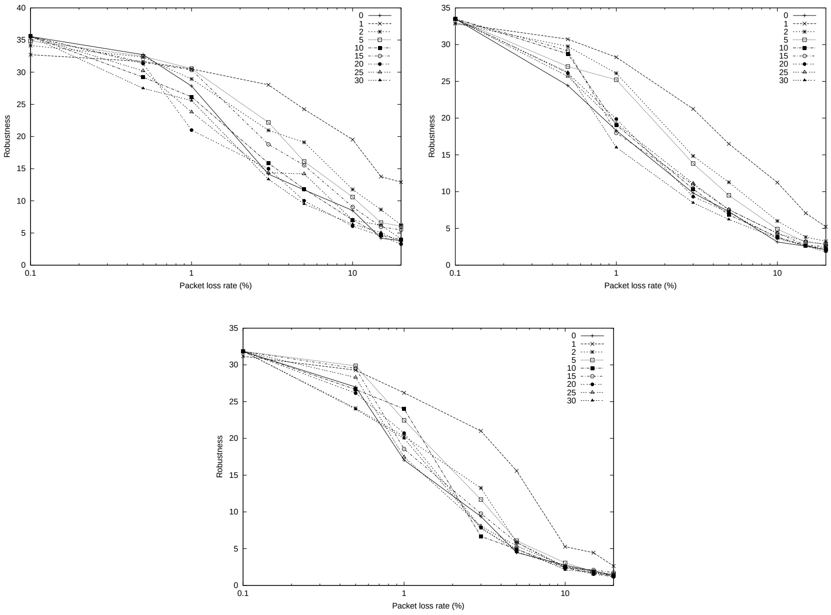


Figure 4.25: Robustness results for News (top-left), Foreman (top-right) and Bus (bottom) sequences.

a different number of reference frames. The H.264 framework allows using up to 5 reference frames, and so our tests where performed with a number of reference frames between 1 and 5.

Figure 4.26 shows strange effects on the distortion output of the rate controller. It shows that the bit-rate control process has almost perfect results, with the distortion showing oscillations. This phenomena is possibly related to the number of bits used to code the number of the reference frames by using Exp-Golomb bit strings.

As it can be seen in figures 4.27 and 4.28, there are differences in distortion by using this technique, but its error-resilience features are not so evident. In fact, it can be noticed that the use of multiple reference frames may provide better behavior on error-prone scenarios, but it is not clear which value is best.

In terms of robustness the distinction is unclear too. This can be due to the fact that, although there are frequent losses, these losses do not result in the loss of entire frames. This explains why this method does not show great success. Anyway, we will again assess the error resilience feature of multi-frame prediction when the losses are bursty in later sections.

4.2.4.5 Number of SP frames

Though the concept behind SP frames does not take error resilience issues as the primary target, this frame type also performs well in error-prone environments.

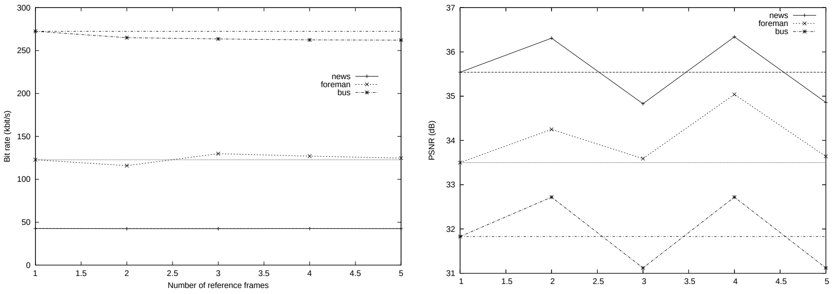


Figure 4.26: Bit rate and PSNR results for loop control with variable number of reference frames

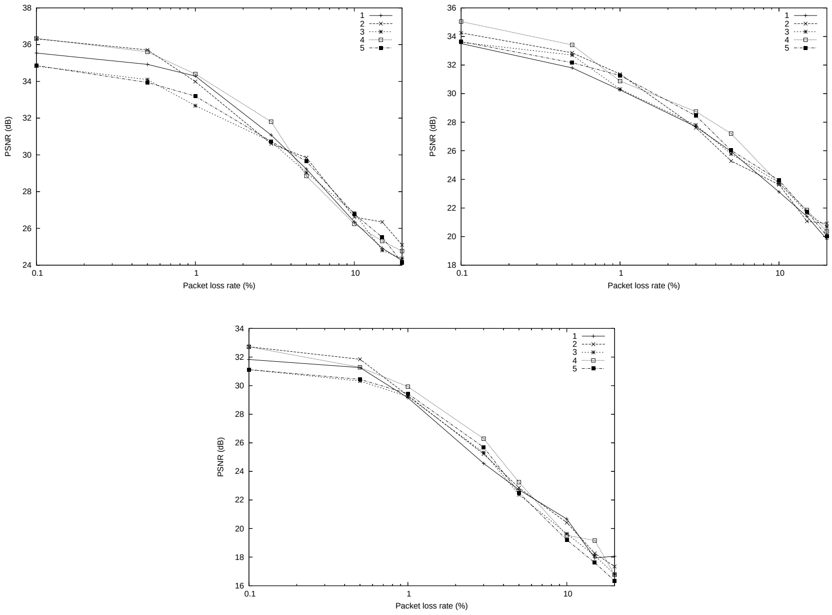


Figure 4.27: PSNR results with packet losses for News (top-left), Foreman (top-right) and Bus (bottom) sequences.

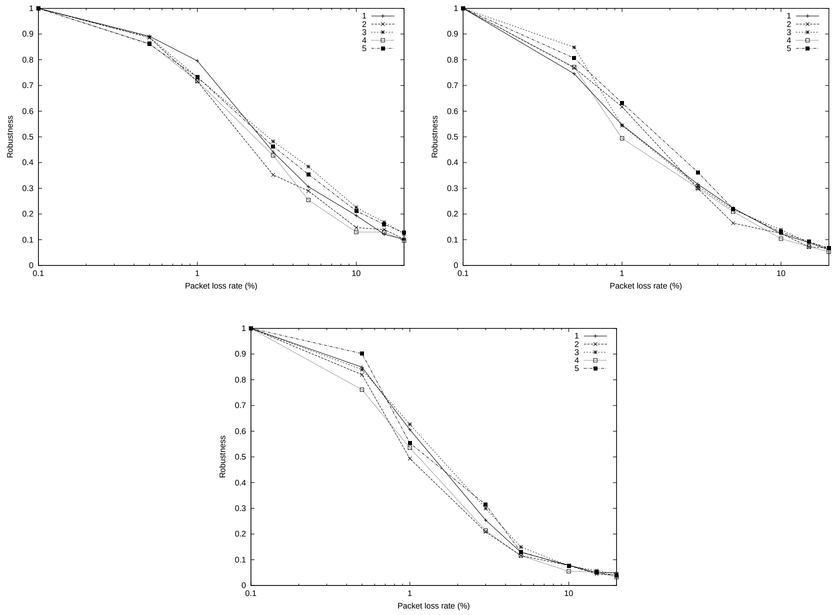


Figure 4.28: Robustness results for News (top-left), Foreman (top-right) and Bus (bottom) sequences.

So, despite their use may cause a quality drop for a same bit rate, they also have different functionality such as quickly advancing on the video stream, as well as other features.

Our tests consisted of varying the periodicity of SP frames in a range from 0 (no SP frames used) to 30 (one every 30 frames is SP coded). Figure 4.29 shows the rate controller's output in terms of bitrate and PSNR. We see that SP frames show a similar behavior to other types of intra coding methods presented previously.

The results presented in figures 4.30 and 4.31 show that this kind of frames can be efficiently used in error-prone environments. Though we cannot reach a final conclusion by observing the PSNR and robustness results, we consider that an SP frame interval around 20 seems to offer a good trade-off.

As referred before, and as we will show in the next section, H.264 offers other mechanisms that are more effective in terms of error-resilience that SP frames.

4.2.4.6 Random Intra macroblock refresh

This technique offers an excellent method to cope with the most demanding scenarios in terms of error-resilience since it does distributed intra-macroblock updating. So, it becomes an alternative to using intra coded frames and, moreover, it offers a simple method to approach a constant bit rate video stream.

Figure 4.33 shows that careful tuning of this parameter can achieve good results in scenarios with packet losses between 5% and 20%, which are acceptable values

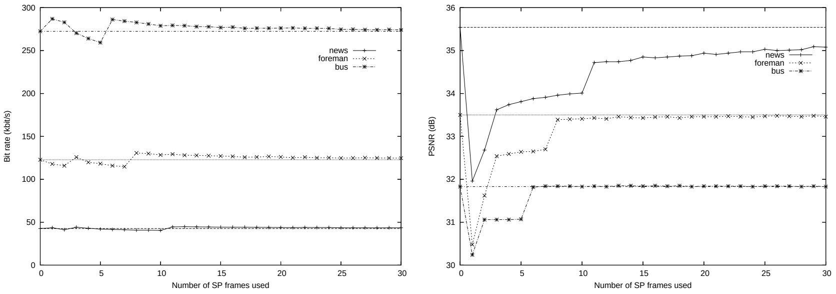


Figure 4.29: Bit rate and PSNR results for loop control with variable number of SP frames

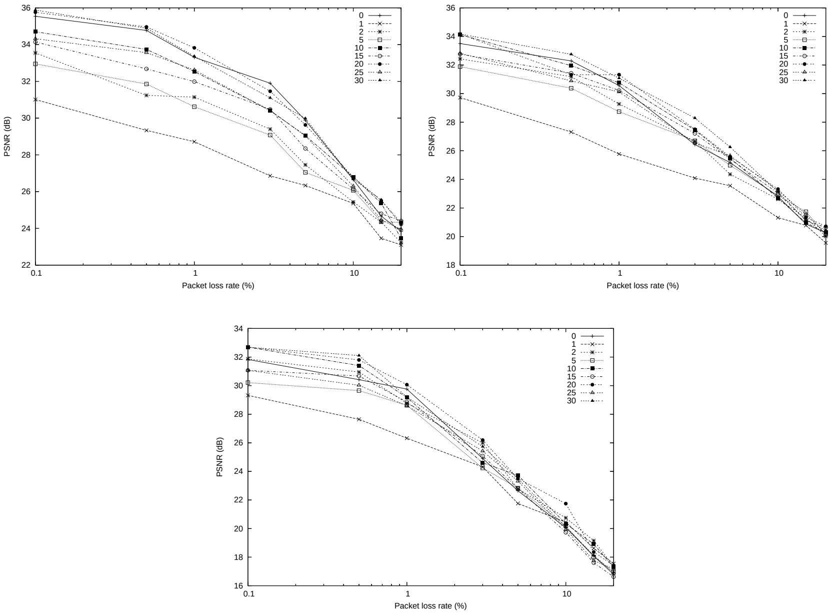


Figure 4.30: PSNR results with SP frames for News (top-left), Foreman (top-right) and Bus (bottom) sequences.

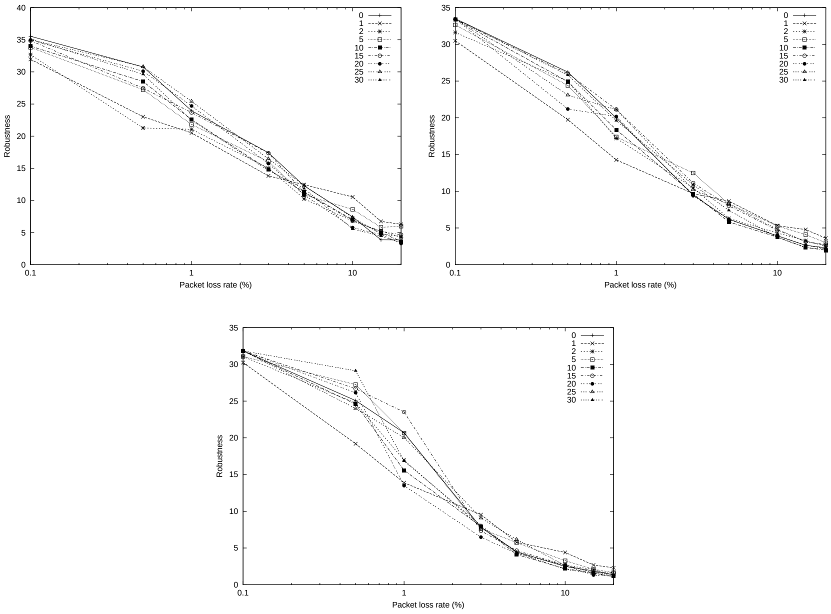


Figure 4.31: Robustness results with SP frames for News (top-left), Foreman (top-right) and Bus (bottom) sequences.

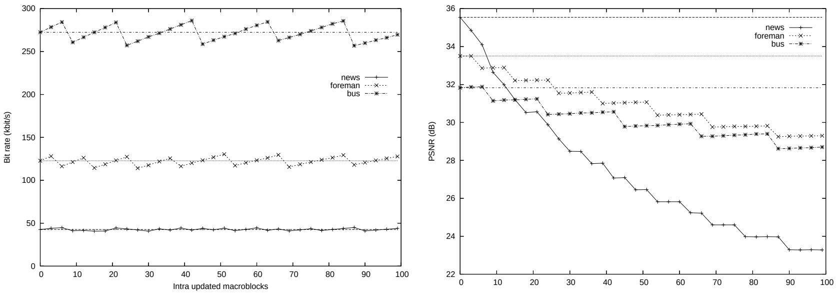


Figure 4.32: Bit rate and PSNR results for loop control with variable number of random intra macroblock refreshes

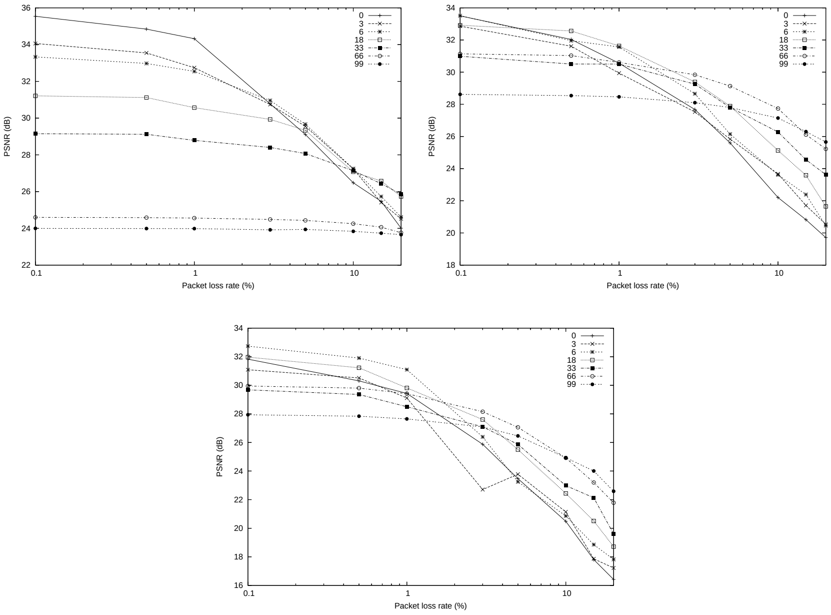


Figure 4.33: PSNR results with packet losses for News (top-left), Foreman (top-right) and Bus (bottom) sequences.

for, e.g., wireless ad-hoc networks. Intra-updating 1/3 of each frame offers a good balance between distortion and error-resilience, and can be considered as a good rule in such scenarios.

In terms of robustness, the obtained conclusions are similar to those achieved with intra frames: higher intra updating rates lead to higher robustness, since the capability of sustaining quality will be improved.

After concluding this first analysis of H.264's error-resilience features, we now proceed to analyze the performance of H.264 in environments characterized by bursty packet loss events.

4.3 Improving H.264's resilience against bursty packet losses

In section 4.2 we evaluated the performance of H.264 in lossy environments. These losses where randomly generated using a uniform distribution; hence, packet loss bursts, if any, where quite small and typically affected a single frame only.

In this section we proceed with the packet-loss analysis by provoking different loss burst sizes and measuring the effectiveness of the H.264 algorithms in coping with these errors. The H.264 framework offers two techniques to cope with packet loss bursts: Flexible Macroblock Ordering (FMO) and the use of multiple reference frames.

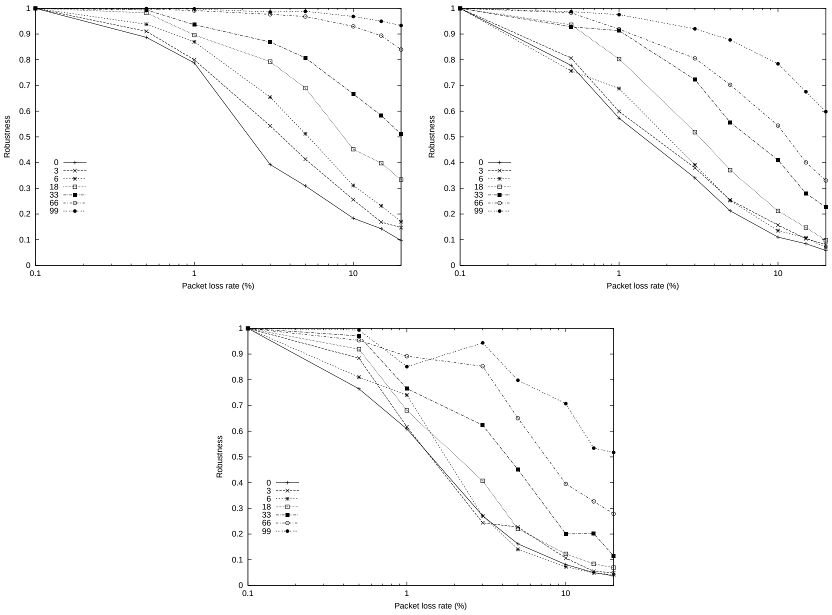


Figure 4.34: Robustness results for News (top-left), Foreman (top-right) and Bus (bottom) sequences.

Concerning the FMO mechanism, it does macroblock rearrangement in order to distribute the burst error throughout a frame. Due to problems with the reference software used (JM3.9a), FMO tasks were performed by external software that we have developed. Our evaluation was done using two and three groups of macroblocks; such options also belong to the H.264 framework.

We conclude this section by analyzing the improvements achieved by the multiple reference frames mechanism.

4.3.1 Effect of macroblock reordering (FMO) on packet loss bursts

Macroblock reordering is a new feature of the H.264 codec that provides increased error-resilience when macroblocks are lost in a bursty manner. It aids the macroblock concealing process by spreading the error throughout the affected frame, thereby reducing the temporal propagation of errors.

In this evaluation we used the Bus sequence since it characterized by a high degree of movement. We considered that for that reason it properly stresses the codec for the evaluation being made. In our tests we used the first 30 frames and tuned the quantization to mid-scale for a distortion of 31.73 dB at 10 Hz.

The burst simulated occurs on the first P frame and was set to a duration of one quarter of a frame, half a frame and three quarters of a frame.

Concerning the macroblock reordering process, the reference software used

Burst start relative to frame	PSNR	PSNR (FMO)	Robustness	Robustness (FMO)
Beginning of frame	27.70	27.52	0.201	0.227
Middle of frame	26.22	28.70	0.175	0.346
End of frame	26.12	28.62	0.178	0.276
Average Value	26,68	28,28	0.185	0.283

Table 4.3: PSNR and robustness average results for bi-partitioning after the loss of half frame.

(JM3.9a) had that option broken, which led us to evaluate it by performing an external reordering of macroblocks. Since this early version of the H.264 codec crashed when a one macroblock per packet mapping was made, we had to use a minimum granularity of 2 macroblocks for testing. This means that we split each picture into 2-macroblock slices for our experiments.

Relatively to the reordering strategy, we divided the image in two groups (first the even, then the odd) and in three groups (first macroblocks 1, 4, 7, etc., then 2, 5, 8, etc., and finally 3, 6, 9, etc.). The H.264 reference software is expected to offer more complex reordering (such as a spiral pattern distribution or a used defined one) in future implementations.

4.3.1.1 Loss burst of half a frame

We now proceed to analyze the benefits of the FMO technique when losses provoke half of a frame to be lost. We begin by testing the benefits of splitting the picture's macroblocks into two groups using a checkerboard approach (even/odd scanning), and we then proceed to compare it to a solution using three groups instead.

Even / odd scanning

Our experiments using FMO based on two different macroblock groups offer a good insight into the usefulness of this technique. As shown in table 4.3, FMO reordering achieves an increase in terms of average distortion values of 1,6 dB compared to a solution with no FMO.

Robustness always shows an improvement by using FMO, though distortion values may present slightly worse results when the error occurs at the beginning of the frame.

We consider that the difference of effectiveness for the FMO technique is related to the levels of movement in the image, which are lower on the top and higher on the middle and bottom.

In relation to the initial distortion value for this sequence (maximum), we can see that on average the use of FMO reordering closes the gap towards that value by 31,68%, showing its effectiveness in relation to the best results that could be achieved.

Analysis of figure 4.35 shows that, on the long term, the spreading of errors by FMO reordering can lead to higher convergence times. Since the solution with FMO always produces improved results before frame 16, it gives us a hint relative

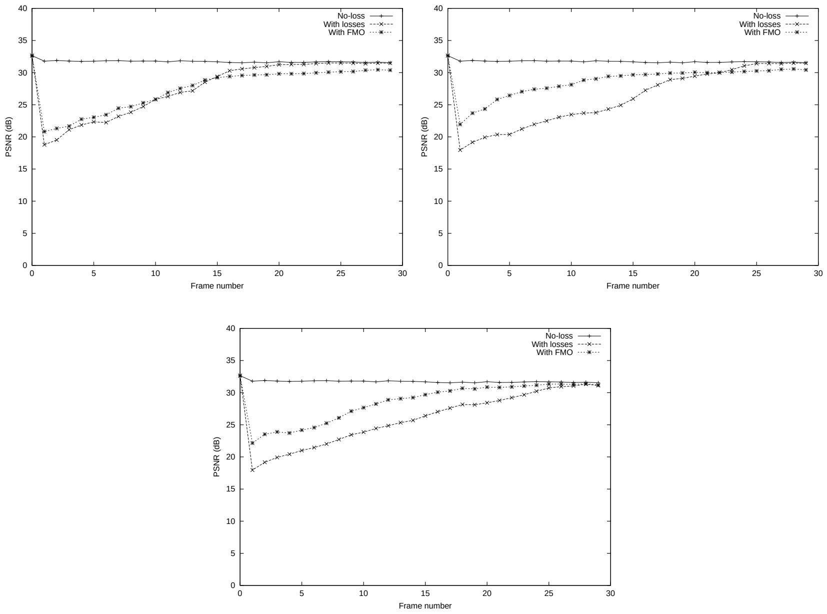


Figure 4.35: PSNR recovery with 2 groups after half-frame loss on the beginning (top-left), middle (top-right) and end (bottom) of frame 1.

to minimum intra-updating frequency for enhanced performance, which should be at least every 15 frames according to the results found for this video sequence.

As expected, the highest improvements in terms of PSNR normally occur on the frame which suffers the loss, validating the adequateness of the FMO technique completely.

Figures 4.36, 4.37 and 4.38 show what occurs when the loss burst occurs at the beginning, middle and end of a frame, respectively. The error presented in those pictures was obtained through subtraction of the correspondent lossless frames; afterwards, the color levels were inverted so that white pixels mean that no error exists. The effect of using FMO is clearly evidenced on these figures, being the error spread throughout the frame as expected. This behavior helps error-concealment techniques at the decoder side, showing better rate-distortion performance.

After 8 frames we can see that the error propagation has been reduced by using the FMO technique, being almost residual.

Having completed this analysis for dual scanning, we now proceed with a similar analysis using triple scanning instead.

Triple scanning

Our experiments using triple scanning required splitting the pictures into three distinct groups, and the splitting was made so that consecutive macroblocks belong to distinct groups. This is to assign macroblocks from different picture zones to all macroblock groups.

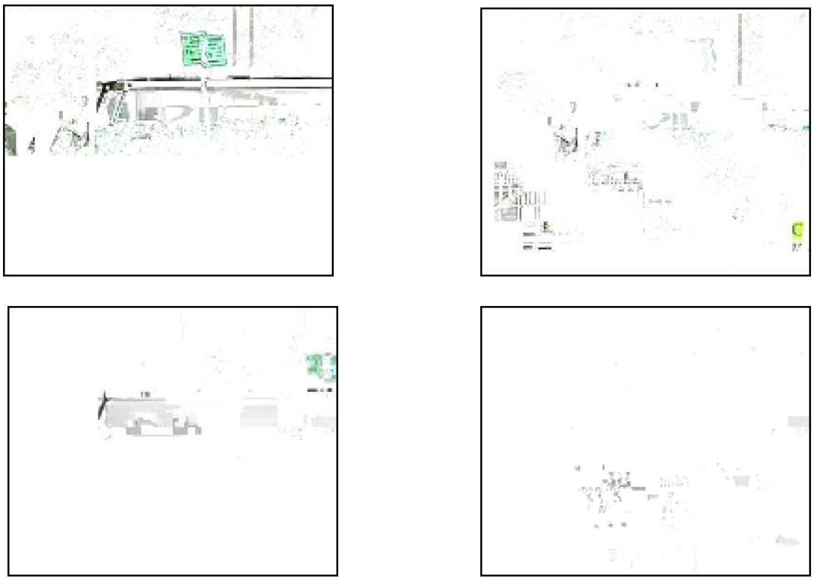


Figure 4.36: Error with (right) and without (left) macroblock reordering for a burst occurring at the beginning of the frame: on the affected frame (top), after 8 frames (bottom).

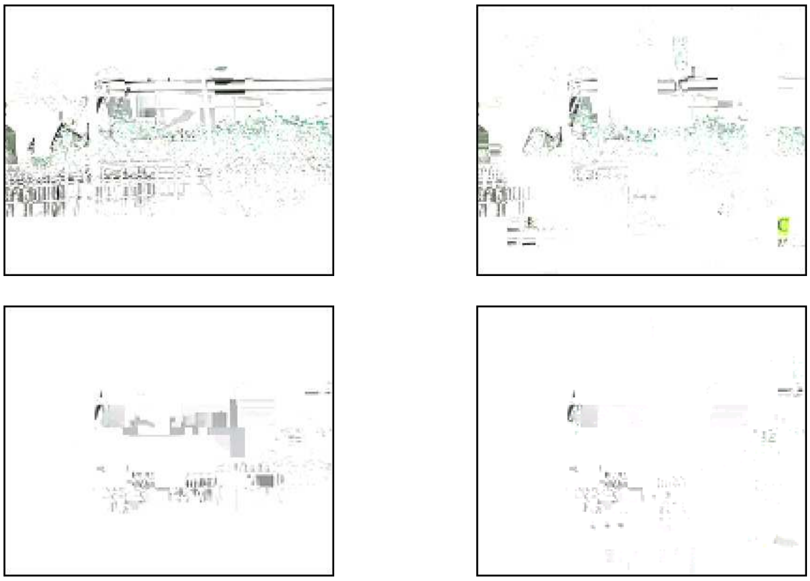


Figure 4.37: Error with (right) and without (left) macroblock reordering for a burst occurring at the middle of the frame: on the affected frame (top), after 8 frames (bottom).

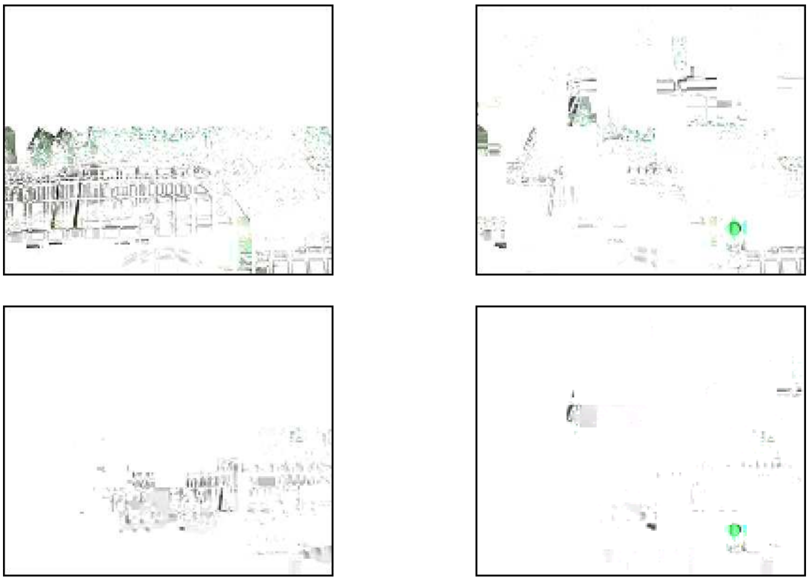


Figure 4.38: Error with (right) and without (left) macroblock reordering for a burst occurring at the end of the frame: on the affected frame (top), after 8 frames (bottom).

We now present the results achieved using our triple scanning technique. We found that, as expected, the performance is slightly inferior to the 2 groups solution analyzed before. Table 4.4 presents the results achieved.

As it can be seen from that table, even though this method leads to inferior results, there is still an average improvement of 1,27 dB relatively to the original solution. Therefore, the gap towards the maximum value is reduced by 25,15% (6% less than the 2 groups solution).

After obtaining these results, we now proceed to validate them in situations where the amount of information lost is reduced to a quarter of a frame, and increased to three-quarters of a frame.

Burst position	PSNR	PSNR (FMO-3)	Robustness	Robustness (FMO-3)
Beginning of frame	27.70	27.06	0.201	0.210
Middle of frame	26.22	27.77	0.175	0.254
End of frame	26.12	29.03	0.178	0.302
Average Value	26.68	27.95	0.185	0.255

Table 4.4: PSNR and robustness results for triple-partitioning with half frame loss.

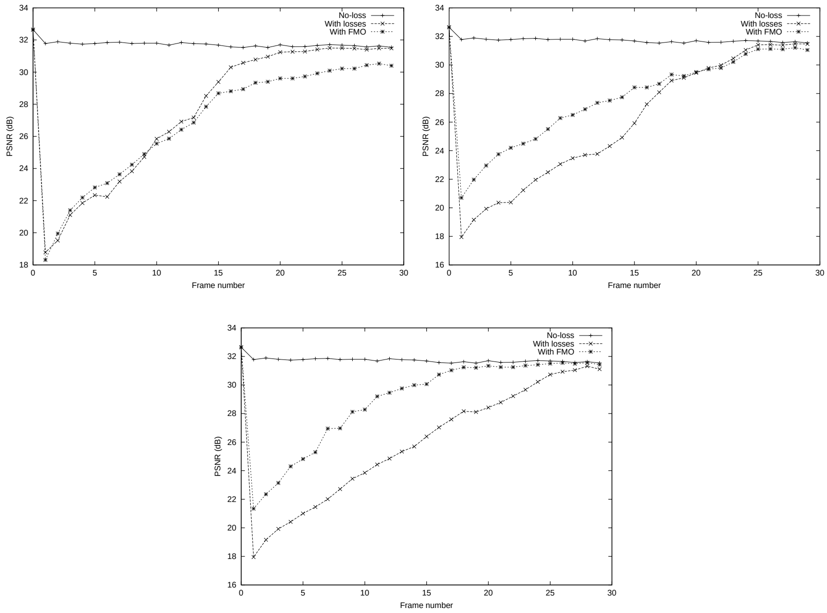


Figure 4.39: PSNR recovery with tri-partitioning after half-frame burst occurring at the beginning (top-left), middle (top-right) and end (bottom) of a frame.

4.3.1.2 Loss burst of 1/4 frame

This evaluation is used to compare the results obtained in the half-frame loss evaluation in order to check the generality of the conclusions achieved. Therefore, the sequence of analysis and the methodology used are the same. Our analysis begins with a dual eve/odd scanning and proceeds with triple scanning.

Even / odd scanning

By forcing quarter-frame losses, the drop in terms of video distortion is not as severe as was previously. The use of FMO reordering, however, still brings benefits, as it can be seen in table 4.5.

The average increase in terms of distortion is of 0,57 dB. This means that the gap towards the maximum value is reduced by 20,7% on average, which in terms of performance is not as good as with half-frame losses.

If we observe figure 4.40 we can see that when the loss occurs at the beginning of the frame (upper left picture), using FMO will actually produce worse results (FMO curve is below the one with no FMO reordering). We should take into consideration, though, that the upper quarter of the picture is much more static than the rest. We can also see that when the error burst occurs at the end of the frame there is only a slight gain for the same reason too. For losses occurring in the rest of the picture FMO produces better results. After several frames,

Burst position	PSNR	PSNR (FMO)	Robustness	Robustness (FMO)
Beginning of frame	30.40	29.51	0.498	0.334
1/4 of frame	27.91	28.87	0.230	0.375
1/2 of frame	28.25	30.44	0.268	0.532
3/4 of frame	29.15	29.23	0.348	0.347
Average Value	28.93	29.51	0.336	0.397

Table 4.5: PSNR and robustness results after quarter-frame loss with macroblock bi-partitioning

Burst position	PSNR	PSNR (FMO-3)	Robustness	Robustness (FMO-3)
Beginning of frame	30.40	28.74	0.498	0.284
1/4 of frame	27.91	28.45	0.230	0.324
1/2 of frame	28.25	30.77	0.268	0.628
3/4 of frame	29.15	29.70	0.348	0.368
Average Value	28.93	29.42	0.336	0.401

Table 4.6: PSNR and robustness results after a quarter-frame loss with macroblock tri-partitioning

though, FMO is no longer the best option because of difficulties to stop the error-propagation entirely.

We now proceed with a similar analysis using triple scanning instead.

Triple scanning

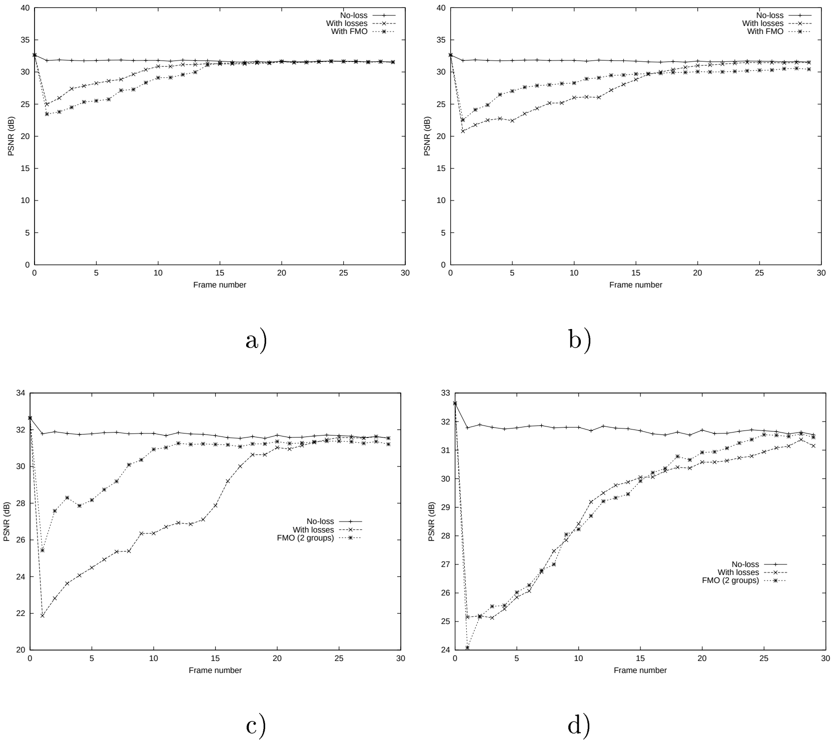
The results obtained when performing a triple scan show that this alternative technique continues to result in a worse performance compared to a dual scan macroblock reordering. Table 4.6 shows the actual values. Now the average increase is of just 0,49 dB, so that the gap towards the optimum value is reduced by 17,5% only.

The results presented in figure 4.41 allow us to reach similar conclusions to those with frame bi-partitioning. Again, the loss of the first quarter of the frame is the only situation where FMO produces worse results. Error-propagation phenomena occurs in this situation too, and again frame 15 seems to be a transition point where applying FMO leads to worse results, as it can be seen in the picture at the upper right.

We now proceed with our final analysis concerning the FMO technique, which consists of increasing the loss area to three-quarters of a frame.

4.3.1.3 Loss burst of 3/4 of frame

Previously we presented the results of quarter frames losses to perform a comparison towards the reference evaluation - half frame loss. We now do so with 3/4



a)

b)

c)

d)

Figure 4.40: PSNR recovery with bi-partitioning after quarter-frame loss at a) beginning of frame b) 1/4 of frame c) half frame d) 3/4 of frame.

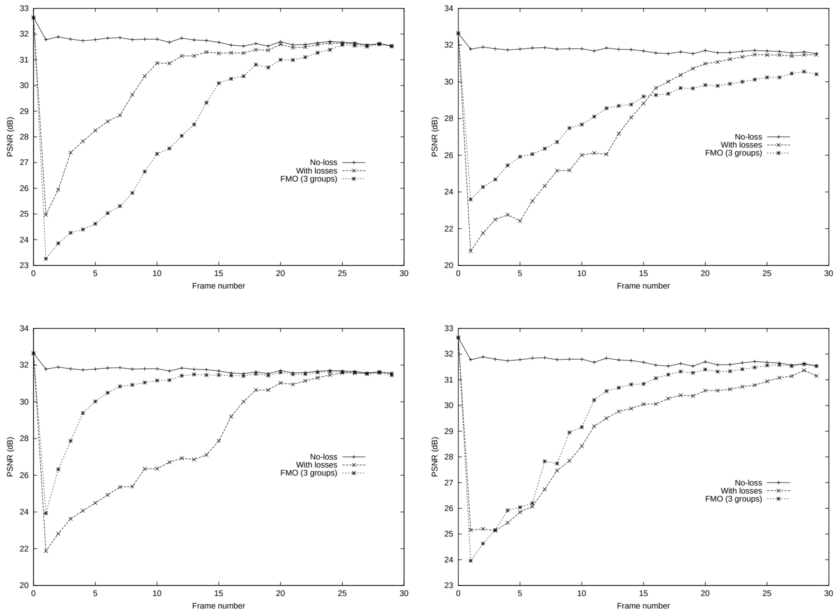


Figure 4.41: PSNR recovery with tri-partitioning after quarter-frame loss at different frame positions.

frame losses to see what happens for bursts larger than those used in the reference situation. As previously, we first analyze the performance of FMO with a dual scan, and we then proceed to triple scanning.

Even / odd scanning

As before, our experiments consisted in comparing the performance improvements by using FMO in terms of both PSNR and Robustness. Table 4.7 shows that, on average, FMO presents worse distortion values by a difference of 0,26 dB. Robustness, though, still shows a slight increase.

Taking a look at figure 4.42 we can see that what is actually happening is that FMO still provides superior results, though only slightly at times. However, after

Burst position	PSNR	PSNR (FMO)	Robustness	Robustness (FMO)
Beginning of frame	26.61	26.51	0.183	0.183
End of frame	25.58	25.17	0.158	0.172
Average Value	26.10	25.84	0.171	0.178

Table 4.7: PSNR and robustness results after loss of 3/4 of frame with macroblock bi-partitioning.

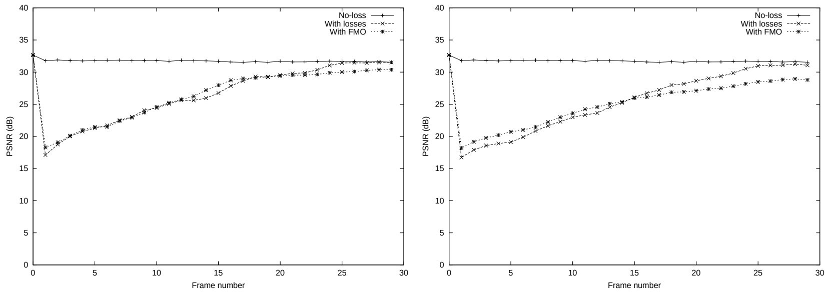


Figure 4.42: PSNR recovery after burst occurring at the beginning and end of frame 1.

Burst position	PSNR	PSNR (FMO-3)	Robustness	Robustness (FMO-3)
Beginning of frame	26.61	26.74	0.183	0.189
End of frame	25.58	27.88	0.158	0.230
Average Value	26.10	27.31	0.171	0.210

Table 4.8: PSNR and robustness results after loss of 3/4 of frame with macroblock tri-partitioning

several frames, FMO reordering can show a much worse behavior. This explains the worse average results found before.

We now proceed with the triple scanning analysis.

Triple scanning

Contrarily to dual scanning, applying FMO with triple scanning does produce improved average results, as exposed in table 4.8. Now the increase is on average of 1,21 dB, and so the gap towards the optimum value is reduced by 21,49%.

This result is quite unexpected, since the dual-scan option (bi-partitioning) is in theory more effective at spreading errors.

As shown in figure 4.43, the improvement is mainly due to the performance achieved when the error occurs in the lower part of the frame, which in this case is the one with a higher level of movement.

We also find that, again, some sort of error resetting mechanism - such as intra-coded frames or other - should be introduced in conjunction with FMO techniques in order to achieve optimum overall results.

4.3.2 Using multiple reference frames to decrease error propagation

In this section our experiments consist of analyzing the technique of using more than one reference frame to reduce the impact of very large packet loss bursts.

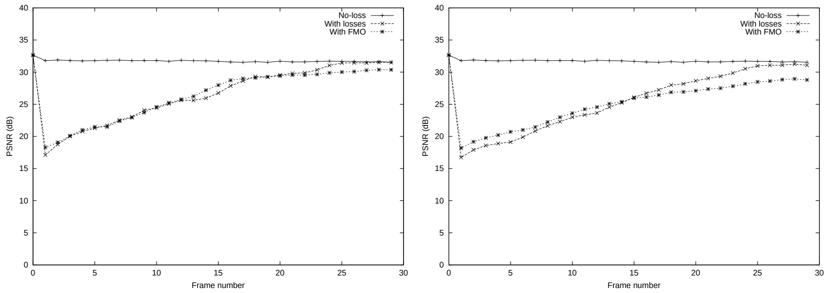


Figure 4.43: PSNR recovery with tri-partitioning after loss burst during the beginning (left) and end (right) of the frame.

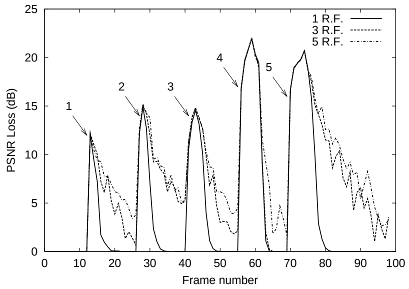


Figure 4.44: Analysis of error propagation by simulating an increasing number of entirely lost frames

Since large bursts can cause one or several consecutive frames to be lost, we want to verify if using more reference frames causes the error-resilience of an H.264 video stream to improve.

To evaluate the error-resilience properties of the multiple reference frames technique, we took the Foreman QCIF sequence and provoked calculated losses ranging from 1 to 5 consecutive frames, so that the error propagation effect was presented as clearly as possible. The analysis was done using 1, 3 and 5 reference frames for comparison.

Figure 4.44 presents the results achieved. Number/arrow pairs refer to how many consecutive frames were lost at each point.

As it can be seen, the behavior experienced is quite clear: using more reference frames leads to worse results every time.

This result was unexpected according to previous works [MJ97, MJ01] which state that the use of multiple frames of reference is effective in terms of error-resilience; these works, however, focus on resilience to bit errors instead of entire frame losses.

From our results we can conclude that using a single reference frame is the most effective choice to stop temporal error propagation. Demands in terms of memory on both encoder and decoder are also reduced by this setup.

Since the version of the codec used (JM3.9a) is still under development, the results found previously should be validated with a completed version of the H.264 codec.

4.4 Evaluation of H.264 in simulated MANET environments

In the previous chapters our analysis began by tuning the different H.264 parameters for an adequate transmission over MANETs. Concerning the main parameters of interest after this first analysis, the Hadamard transform, CABAC and Rate Distortion Optimization were used since they offered the best results. The use of adaptive block transforms for inter and intra macroblocks was set to the fully flexible mode taking into account the results available in [TM02].

We then proceeded to study those error resilience properties of H.264 that offered optimum performance under both random and bursty packet losses. We found that the two best options were enabling random intra macroblock updates - set to 1/3 of frame size - and applying two-group FMO reordering.

This section is dedicated to evaluating the effectiveness of the parameter setup performed previously on a real MANET environment. The test sequence we have chosen for our evaluation is the well-known QCIF Foreman sequence. We consider that real-time video over ad-hoc networks will essentially be used to provide communication between peers; since the Foreman sequence shows in essence the upper body of an individual, it fits perfectly in our model. Concerning its size (176x144), it is considered adequate for display in current PDAs and other mobile devices. The chosen frame rate for the sequence is of 10 frames per second, and the sequence's average bit rate is of 178.64 kbps.

4.4.1 MANET simulation setup

In order to perform the desired evaluations we used the Network Simulator (ns-2) [KK00] version 2.1b9a. Ns-2 is the most widely used discrete event simulator. Changes and enhancements to the simulator can be done in a straightforward manner since it is open source.

The radio propagation model used in our simulations is based on a two-ray ground reflection radio propagation model. The physical and MAC layers are implemented using IEEE 802.11. Evaluation of MANETs requires using IEEE 802.11's Distributed Coordination Function (DCF), and so the Media Access Control Protocol (MAC) is CSMA/CA - Carrier Sense Multiple Access with Collision Avoidance. All protocols simulated maintain a send buffer of 64 data packets, containing the data packets waiting for a route. Packets sent by the routing layer are queued at the interface queue until the MAC layer can transmit them. This queue has a maximum size of 50 data packets, and it gives priority to routing packets being served. The transmission range for each of the mobile nodes is set to 250m, and the channel capacity to 11Mbps (full rate in IEEE 802.11b).

Concerning the routing protocols used, we test AODV, OLSR, TORA and DSR. Besides the AODV implementation integrated in the ns-2 simulator, we also test the AODV implementation of Uppsala University, denoting it with AODVUU. We also test a version of AODV based on "Hello" packets for neighbor discovery, denoting it AODVUU-H.

To evaluate the desired video flows, we make a conversion of the RTP output from the H.264 encoder to ns-2's native format. That way we are able to stress the network with a trace of real-life video traffic, instead of relying on CBR flows to mimic them.

Though the sequence is only 10 seconds long, ns-2 automatically re-reads its input so that the sequence automatically restarts. Our evaluation is done over 100 seconds, and so we average the distortion values found for each 10 second interval. Moreover, all results presented are average results from 20 random simulation processes.

A 10 second wait period was introduced at the beginning of each simulation in order to allow the different routing protocols to converge, and also to start the background traffic; the purpose is to evaluate the conditions experienced by the video flow on an almost steady-state situation.

After the ns-2's simulation process ends, we converted its output results in order to ascertain which packets from the video flow under evaluation had been lost. This is achieved through a series of Perl scripts. Afterwards, the output from the scripts is used in order to filter the original H.264 RTP file according to the packet losses that occur. That filtered video file is then passed through the H.264 decoder, which provides the final result.

The steps described envisage achieving as much consistency and real-life proximity as possible to our evaluation.

4.4.2 Performance Results

We now proceed to present the actual simulation results found. This section is divided into 6 parts. We begin by obtaining some preliminary results to assess the re-routing time of the different MANET routing protocols under evaluation. We then proceed with a performance analysis of the different routing protocols at different levels of mobility. Our tests continue with a study on the impact of variable congestion levels on the performance of the H.264 video streams, followed by a detailed analysis of the effects of re-routing and congestion on the video streams.

To conclude our evaluation we again review the goodness of different codec choices made, finishing with a study on the most adequate data rate values for congested networks.

4.4.2.1 Preliminary analysis

Protocols used for routing in MANETs are usually divided into two main categories: reactive and proactive. Moreover, another division can be made according to the way in which they detect link failures. While the method of sending "Hello" messages is more universal, the IEEE 802.11 technology enables the use of a more

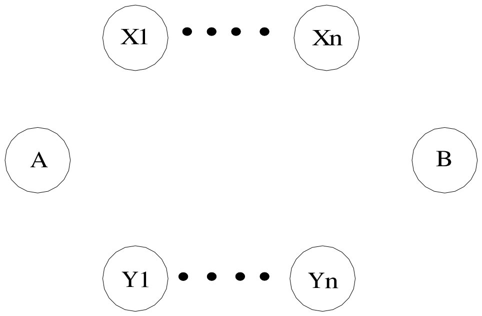


Figure 4.45: Simple scenario for re-routing evaluation

effective and efficient method to detect link breaks by using the information it provides. Using that information nodes can react to broken links more quickly, avoiding sending packets to nowhere.

Broken links are the main cause of long packet loss bursts in MANETs. In fact, long packet loss bursts can be a major source of problems for video flows in MANETs. This problem is more evident when "Hello" packets are used to detect broken links. Typical "Hello" intervals [PR99, TPA$^+$01] range from 1 to 2 seconds, and so re-routing times can be as high as 6 seconds or more - connection is considered lost usually after 3 missing "Hellos". Since such failures are too long to be handled even by the most versatile video codec, we recommend the use of Link Level aware protocols such as AODV (link aware version), DSR or TORA in order to perform re-routing tasks as soon as possible.

In this preliminary evaluation a simple scenario was devised, as presented in figure 4.45. The purpose is to evaluate the re-routing times of different routing protocols by setting a CBR flow from node A to node B through path {A,X1,...,Xn,B} and provoke a re-routing process using path {A,Y1,...,Yn,B}. To achieve that, the last intermediate node from the upper path (Xn) moves quickly away making that route unusable; just before that Y1 moves into the range of A. Choosing the departure of the last node on the upper path (Xn) and the arrival of the first node on the lower path (Y1) aims at achieving a worst re-routing scenario.

The results for the evaluation under this scenario are presented in figure 4.46. "Hello" based protocols such as OLSR and AODVUU-H perform significantly worse than protocols that include broken link detection, as expected. Moreover, the re-routing time for "Hello" based protocols depends essentially on the "Hello" period and on the number of missed "Hellos" for a link to be considered lost. OLSR's RFC [TP03] states that the default value for the "Hello" period is 2 seconds, twice that for AODV (1 second); both consider the link to be lost after 3 failed "Hellos". This explains the gap between both.

The best performing protocols are AODV and DSR. Concerning AODV, we choose the AODVUU implementation for further testing since it follows the AODV specification more rigorously than the one that comes bundled with ns-2.

To make the evaluations that follow more clear, we drop ns-2's AODV im-

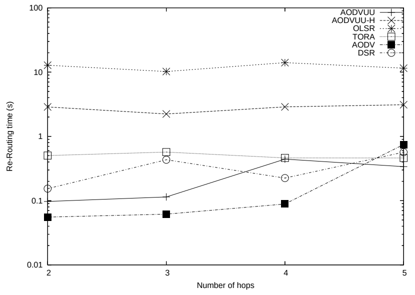


Figure 4.46: Re-routing times for different protocols in the simple test scenario

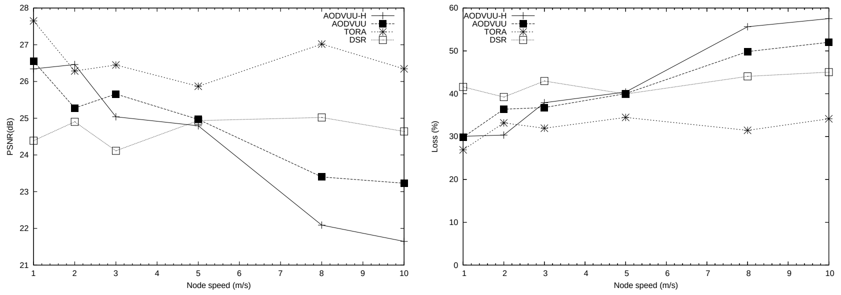


Figure 4.47: Evaluation of different routing protocols for varying mobility in terms of packet losses and perceived PSNR for the video test sequence

plementation, as well as OLSR since the interruptions produced are too high for real-time communication, and we maintain TORA, DSR and both AODVUU and AODVUU-H.

4.4.2.2 Mobility evaluation

After this initial evaluation, we devised a typical MANET scenario consisting of 30 nodes moving inside a 670x670 m area. Mobility was generated through the random waypoint model bundled in the ns-2 tool, and we test different node speeds with a wait time of 5 seconds; this applies to all nodes. For more information on MANET mobility models refer to [JCPM04].

In addition to the H.264 video flow, 5 background FTP flows are also set (1 every 6 nodes). Figure 4.47 shows the results achieved in terms of distortion and packet loss rate when using different routing protocols.

“Hello” based AODV (AODVUU-H) performs well in situations of very low mobility because route changes do not occur often. Also, there are less chances that background congestion causes one link to be considered lost (3 consecutive “Hellos” have to be lost). Its link aware counterpart (AODVUU) performs well on low and average mobility scenarios. TORA shows the best overall behavior

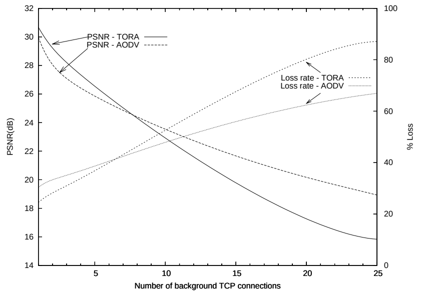


Figure 4.48: PSNR and packet loss rate performance for a variable number of background TCP connections

under this scenario, showing good distortion levels at all speeds and good ability to maintain the packet loss rate at high mobility levels. DSR is also able to maintain steady levels of distortion and packet loss rate, although not so efficiently as TORA.

This analysis does not pretend, though, to be an in-depth examination of these routing protocols, but rather a study on best choices to achieve good video robustness at different mobility levels in the presence of TCP-based congestion. The results achieved contrast with those found in the literature concerning the goodness of routing protocols; for instance, AODV and DSR are known to be superior to TORA and "Hello"-based AODV. However, the presence of resource-greedy TCP sources is prone to cause routing protocols to malfunction, as shown in section 7.9.

Please, refer to works such as [SCM99] for a more general study on the performance of different routing protocols.

4.4.2.3 Performance under congestion

Based on the results of the mobility evaluation made before, we choose both TORA and the AODV implementation from Uppsala University to proceed with our analysis. We now pretend to evaluate their performance when submitted to different levels of congestion at user mobility levels; the values used to configure the random waypoint mobility model were 2 m/s for speed, and 5 seconds for the wait time.

These results were achieved using the same 30 node square scenario described in the previous subsection.

Our experiments consist of varying the number of TCP connections from 0 to 25 and assessing both the video distortion (PSNR) and the loss rate. We then make a similar experiment by varying the number of background video connections instead. We vary the number of video connections in a range from 1 to 15. These video connections are similar to our reference H.264 stream.

Figure 4.48 allows us to compare the performance of TORA and AODV with a variable number of TCP connections in the background (TCP traffic is currently the most common - FTP, Web, Peer-to-peer, Database Access, etc. - but perhaps

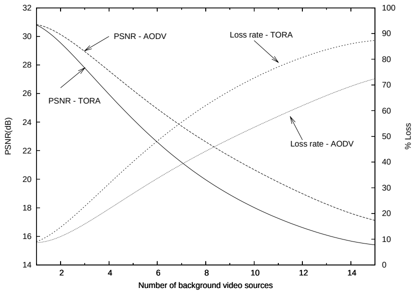


Figure 4.49: PSNR and packet loss rate performance for a variable number of background video connections

not in a near future). We can see from that figure that acceptable distortion levels cannot be reached with more than 10 background connections using either TORA or AODV. TORA is, therefore, the best choice for this range and, even though AODV performs significantly better under critical levels of congestion, the results in terms of distortion are almost at noise levels.

In figure 4.49 we show the results of a similar analysis made using a variable number of video connections identical to the one under evaluation as background traffic. In this scenario AODV always performs better than TORA and, overall, we consider AODV to be an adequate choice to support video flows as reliably and uninterruptedly as possible, especially in environments where there are no QoS or congestion-control mechanisms available as the ones tested here.

4.4.2.4 On the effects of re-routing and background traffic

To complete our analysis we altered the scenario so that it maintained the same number of nodes and the same area as before, but was made rectangular instead (1500×300m); the purpose is to increase the average number of hops. Envisaging a differentiated analysis of mobility and congestion, we started with a situation having no background traffic nor mobility. We then analyzed separately the effect of assigning a high degree of mobility to all nodes (10 m/s and no background traffic) and of congesting the network by setting all the nodes to transmit a moderated amount of CBR traffic (all nodes are static). In all situations, the average (or exact) number of hops was three; the routing protocol used was AODVUU, though with DSR we obtain similar results.

Figure 4.50 shows the effects of mobility and congestion on the distortion as perceived by the user. It can be seen that mobility affects distortion in a bursty fashion, typically causing the loss of multiple frames and consequent freezing of the image. On the other hand, traffic congestion causes packets to be lost in a more random manner, and so the distortion variations are smoother though more frequent.

The delay analysis also evidences the nature of both kinds of losses, as presented in figure 4.51.

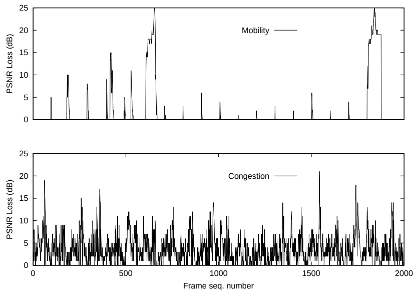


Figure 4.50: Effect of congestion and mobility on user perceived PSNR

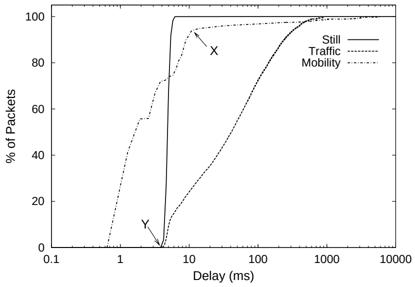


Figure 4.51: Delay effects of congestion and mobility

In the reference situation, more than 99,9% percent of the packets arrive before 7 ms; with high mobility, 92% of the packets arrive in less than 10 ms. Point X is the frontier between two distinct regions: the one on the right where a very small number of packets have very high delays (as much as 6 seconds or more), and the one on the left where packet forwarding is uninterrupted. In the "mobility" scenario, although the average number of hops is 3, the actual value varies throughout the simulation. This explains why some of the packets arrive earlier than those in the reference scenario, and others arrive later (for points before X). The phenomena whereby some packets arrive with very high delays (after X) is expected since AODV causes packets to wait in a queue when re-routing tasks are being performed. So, the percentage of packets that experience high delays depends on the path change rate experienced between source and destination.

Congestion causes a very different behavior, so that all packets that arrive at the destination do so in less than 1 second. The delay between consecutive packets, though, can vary greatly. The start point (Y) for both reference and congestion scenarios is the same because the destination is 3 hops away on both.

The jitter analysis of figure 4.52 also aids at visualizing the fundamental differences between both test scenarios. Even though the peaks of jitter for the mobility scenario occur rather infrequently, they are an order of magnitude superior than those caused by congestion. We conclude that jitter peaks in the order of seconds

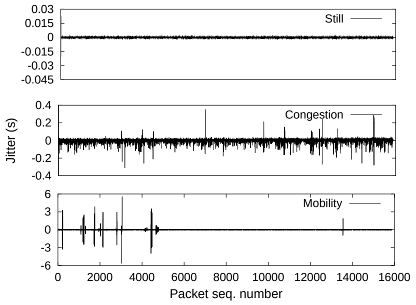


Figure 4.52: Jitter due to congestion and mobility

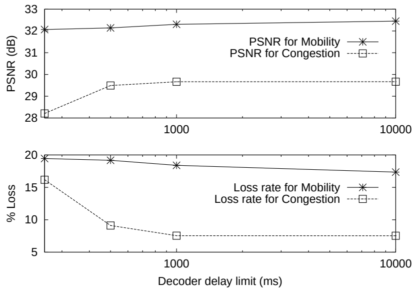


Figure 4.53: PSNR and Packet Loss Rate variation for different delay thresholds

usually translate into a change of route when using reactive protocols.

Real-time video tightens the limits on end-to-end delay and jitter. Depending on the decoding strategy and buffer size, different degrees of flexibility can be achieved. However, certain limits are imposed in order that jerky video visualization is avoided. The results presented in figure 4.53 show the variation in terms of distortion and packet loss rate by applying different delay thresholds to the video stream. As it can be seen, the effects of congestion are apparently more negative since worse distortion values are achieved at smaller loss rates. It should be noticed, however, that long bursty packet losses results in image freezing. This effect, even though much more annoying to a human receptor, is not reflected well in terms of PSNR since the minimum value achieved when the image freezes is around 13 dB, and not zero. Random packet losses, on the other hand, provoke more uniform distortion levels, being therefore more suitable for acceptance from a human receptor point of view.

As it could be inferred from the previous results, tightening the limits on packet delay causes more negative effects on distortion in high-congestion scenarios than in high-mobility ones. These effects can, however, be countered by QoS policies at either the MAC or higher levels. Transmission gaps due to mobility, though, are much more difficult to counter and are more critical.

Solutions to the congestion problem could be introduced at the MAC level itself

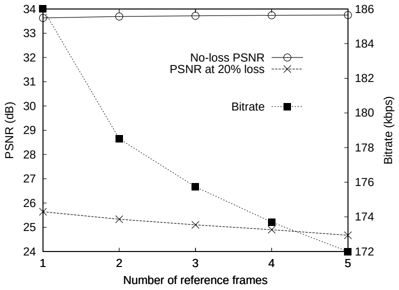


Figure 4.54: Performance under high congestion of a variable number of reference frames

by assigning QoS traffic a higher probability of accessing the wireless medium, in a manner similar to what IEEE 802.11e[IEE05] does, or by assuring that collisions are minimal by assigning it a dedicated inter-frame space. The latter solution, though, would require new changes to the specification. Neither of this options, however, is able to provide a delivery guarantee even to a single surrounding node.

Relatively to the mobility problem, it requires enhancing routing protocols in order to reduce to a minimum the communication gaps provoked by route disruption.

4.4.2.5 Evaluation of video codec choices

In this section we assess if some of the conclusions reached in sections 4.2 and 4.3 relatively to codec tuning also apply to MANET environments.

Our evaluation focuses on two topics: the number of reference frames used and the best method to do intra-macroblock updating.

We start by evaluating the impact of using a different number of reference frames. The experiments are made using the heavy congestion scenario presented in the previous subsection.

In figure 4.54 we present the distortion values achieved in this scenario. According to the results depicted, using multiple reference frames increases compression slightly, being this the expected result. In terms of error-resilience, though, we find that there is a monotonous distortion decrease; in fact, figure 4.54 shows that a 1 dB drop results from using 5 reference frames instead of just 1. These results are in accordance to those found in section 4.3.2, and so we restate that the optimum H.264 configuration for MANETs in terms of number of reference frames consists in using a single reference frame, which not only allows simplifying the codec's tasks, but also offers improved error resilience.

Concerning macroblock intra-updating, H.264 provides several choices to the user. We have evaluated the main choices available in the reference software used, which are: use of I frames, intra update a pre-defined number of macroblocks randomly and intra update a whole line chosen randomly for each frame. The results of our evaluation are shown in figure 4.55.

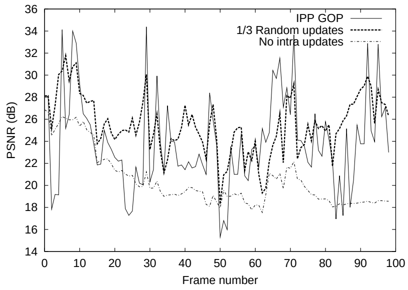


Figure 4.55: Evaluation of different techniques for intra macroblock-updating

Table 4.9: Average PSNR results evaluating strategies for intra MB updating at 23% loss

Updating method	Avg. PSNR (20% loss)	Avg. PSNR (12% loss)	Avg. PSNR (4% loss)
1/3 random MB updates	25,58	28,22	30,63
IPP sequence	24,01	27,18	30,19
IPPPPP sequence	23,35	27,04	29,32
Random line intra update	22,79	26,12	28,93
No intra MB updates	20,62	22,93	25,84

In this process all the test files are encoded at the same bit-rate by varying the global quantization values. Such process is required because the used H.264 software codec does not currently provide a loop-back mechanism for bit-rate control. This allows a fair comparison between different parameter choices, paving the way for more meaningful conclusions.

The scenario used was fixed so that the packet loss ratio is of 23% for all the options being tested.

Figure 4.55 shows the results for the two best performing solutions - Random macroblock updates and IPP GOPs - along with the worst solution - not using any sort of intra macroblock updating strategy.

With this setup, random intra-macroblock updating is by far the best option, offering more stable video distortion values. Table 4.9 presents the average distortion values for all solutions tested sorted by performance. We can see that, for a same target bitrate, choosing an adequate macroblock updating method can bring benefits in terms of PSNR of up to 5 dB.

The process of random intra-macroblock updating could, however, be tuned to adapt to network congestion interactively. This process would require a feedback channel, which would also increase network congestion. Therefore, we didn't consider it the best option for MANETs with a fast-changing topology, though this matter may deserve further analysis in other fields of application.

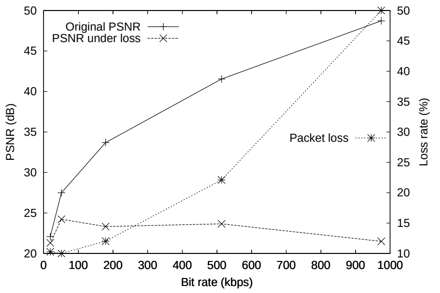


Figure 4.56: Behavior experienced by varying the sequence's original distortion

4.4.2.6 Source distortion tuning

To achieve the results presented in previous sections we used a bit-rate value for the sequence which was obtained by setting the quantization parameter to a mid-scale value. However, in IP-based networks, and contrarily to telephony networks, increasing the data rate does not translate into a better performance always. The reason for this has to do with the TCP rate-control paradigm. TCP sources adapt their data rate to avoid states of congestion, and TCP-friendly applications follow the same paradigm, despite possibly using different transport protocols (e.g. UDP).

Concerning our study, we will now perform a simple test that consists of, starting from a scenario with a pre-defined congestion, assessing which is the data rates that offers the best results from the receiver point of view. For our experiments we used different quantization values, which produce distinct source distortion and bit-rate values. The scenario used was the same one used in the previous section - 1500×300 in size and under high congestion.

Figure 4.56 presents the results of our evaluation. The congestion effect referred before can be noticed in that figure, whereby improving the original distortion does not translate into better PSNR at reception. This effect happens because the growth in PSNR is surpassed by the packet loss increase. The distortion experienced by the user is, therefore, almost constant, with a maximum around 50 kbps.

By looking at table 4.10 we observe that high bit-rate values provoke another drawback: not only do they increase congestion, resulting in higher PSNR decays, but the standard deviation is also much higher. The user will, consequently, perceive less stability on the quality of the video stream.

4.5 Conclusions

In this chapter we focused on the H.264 video coding standard. Our analysis started with a study on the behavior of the various tuning parameters included in the reference software, presenting the results in terms of distortion, bit-rate and

Table 4.10: PSNR decay and deviation for different bit-rates

Bitrate (kbps)	Avg. PSNR decay (dB)	Standard deviation for PSNR
19.08	0.77	0,81
51.15	3.27	1,64
178.64	10.37	2,55
513.35	17.88	4,22
974.18	27,23	4,08

processing overhead. Using those results we tuned the codec in order to achieve optimum performance in terms of compression.

We proceeded to analyze the effectiveness of the error-resilience tools integrated into the H.264 framework. Concerning the error resilience tests, our analysis was centered on two types of error: random and bursty packet losses. The random-loss results obtained allow tuning the encoder according to the expected packet loss rates inside the network, and show that a careful choice can increase significantly the overall PSNR of a video sequence. We also presented the effects of packet loss bursts on the quality of video and propose methods to efficiently handle these situations. Assuming a typical situation where there is a 10% packet loss in the network, tuning the Random Intra Macroblock Update to 1/3 of the total number of macroblocks improves error-resilience on random and burst error situations at the cost of only a marginal increment in terms of bit-rate.

Concerning the use of multiple reference frames, our study points out that this technique increases the temporal error propagation in the presence of large loss bursts, and so it should be avoided in MANET environments.

We found that the FMO technique performs significantly well in the presence of small bursts, and the increase in terms of processing time is not relevant.

After tuning the codec using the aforementioned findings, we then proceeded to evaluate the performance of H.264 streams in actual MANET environments. Since one of the main problems of MANETs is the large interruption time caused by routing protocols every time a route breaks, we made a preliminary analysis focusing on typical re-routing times associated with common MANET routing protocols. This analysis made evident the effectiveness of link-level aware routing protocols in re-routing tasks.

We proceeded with a mobility evaluation under average congestion, where TORA has shown to offer the best distortion results to the video stream. Variable congestion tests followed using TORA and AODVUU. When using FTP sources as background traffic, TORA has only provided slightly better results with less than 10 connections. If a variable number of video connections is employed we find that AODVUU performs better, supporting up to four extra video connections and maintaining a common level of video distortion.

Scrutiny of our results evidenced that, even though routing protocols detect broken links in milliseconds, they are not able to perform re-routing tasks as quickly as would be desired. This phenomena occurs because congestion causes collisions (losses) and higher delays, which prevent routing protocols from operating adequately; this is prone to cause long transmission breaks. In fact, increasing background traffic intensifies this problem, causing routing tasks to become more and more impossible. That way, routing protocols which are known to be more efficient (e.g. AODV and DSR) may offer a worse performance that other routing protocols (e.g. TORA) when the level of congestion is high.

An analysis of delay and jitter followed, showing the effects of congestion and mobility on video streams separately. We found that the ON/OFF behavior with high mobility causes the loss of communication for several seconds, being therefore prone to cause annoyance by the receptor.

Concerning the video codec, we have also showed that the tuning performed was effectively resilient in terms of macroblock updating. The use of more than one reference frame, though effective in reducing bit-rate, increases the temporal propagation of errors and so should be discarded in MANET environments.

We ended our study analyzing the effect of varying the sequence's bit-rate. Results show that, under high congestion, no distortion improvement can be achieved by increasing the bit-rate. In fact, the optimum value found for the video sequence under study is around 50 kbps, a very low one.

Work plan

The findings of this chapter show that, independently of the performance of a video codec in terms of compression and error-resilience, there are several issues requiring further scrutiny and improvements in order to achieve a reliable and robust video transmission system for MANET environments. These problems are essentially: large communication gaps due to re-routing processes, lack of QoS support at the MAC level and lack of an admission control system for QoS streams.

The core of this thesis will propose solutions to the different problems that were put in evidence in this chapter. So, the main contributions to be detailed along this thesis are the following:

- A formal study of the transmission gaps provoked by mobility when operating with both proactive and reactive routing protocols, including a model for end-to-end paths in MANETs to characterize them, and also to accelerate the evaluation of different video coding techniques. This is the topic of chapter 5.

- Proposal of an enhanced route discovery algorithm, along with a traffic splitting technique, to reduce the impact of mobility on real-time multimedia streams, especially video streams. This is the topic of chapter 6.

- Evaluation of IEEE 802.11e as an enabling technology for QoS support in MANET environments, including a study on the interaction between reactive routing protocols and the IEEE 802.11e technology. This is the topic of chapter 7.

- Proposal of a distributed admission control system for MANETs that operates optimally in conjunction with the IEEE 802.11e technology. This is the topic of chapter 8.

In chapter 9 we will assess the overall benefits of the different proposals of this thesis by jointly evaluating them all.

Chapter 5

A novel end-to-end path model for MANETs

On the previous chapter we found that routing related packet losses can provoke quite large packet loss bursts, even when operating in ideal conditions. These loss bursts can extend for up to several seconds, depending on the routing protocol used. Since this degree of burstynessburstiness is unusual and affects QoS streams greatly (e.g. real-time video), we consider adequate to pursue two purposes: to design metrics that are able to properly measure loss bursts, and to MANET's end-to-end paths as experienced by higher layer applications, in our case video applications.

The topic of this chapter is, therefore, a formal study of packet loss bursts occurring on MANETs. We start by proposing a methodology to find an adequate end-to-end path model for MANETs based on hidden Markov chains. We use both a reactive and a proactive routing protocol as example of how to tune a model for adequate operation, evidencing the different requirements of both families of routing protocols. We also include a set of heuristics on how to find good initial estimates for the different parameters of the proposed models.

We proceed by proposing different metrics which allow assessing the magnitude of loss bursts; these metrics are used to validate our models, showing their accuracy. We also demonstrate, with an example, how these models can be used to reduce greatly the time required to tune a video codec for optimum performance in MANET environments.

5.1 Model description and proposed methodology

In a MANET, the associated nodes execute a routing protocol daemon that can be in different routing states. Independently of the routing state, packet losses can occur for a variety of other reasons (collisions, channel noise, queue dropping, etc.). Therefore, an outside observer cannot relate a packet loss with a certain routing state. We deal with a situation where the observation is a probabilistic function of the state, that is, only the output of the system and not the state transitions are visible to an observer. We will therefore try to solve the classification problem using a *hidden Markov model* (HMM) [Rab89].

HMMs are well known for their effectiveness in modeling bursty behavior, relatively easy configuration, quick execution times and general applicability. So, we consider that they fit our purpose of modeling end-to-end paths in MANETS. Such a model shall accelerate the process of evaluation of multimedia streaming applications, while offering results similar to simulation or real-life testbeds.

5.1.1 General methodology

Relatively to the methodology proposed in this work, we start by focusing on a single data stream (e.g. audio, video, etc.) for analysis, as well as the criteria for considering a packet good or unusable by the application at hand. We can take into account factors such as which packets arrive to destination within a maximum delay, the delay jitter limits, the dependency among packets, etc. We then have to map each packet sequence number with values 1 - when the packet is considered good - or 0 - if the packet does not arrive to destination or does not meet any of the chosen criteria. This output mapping is stored in a trace file *(ST)* that will be parsed to obtain the distributions of consecutive packets arriving (CPA) and consecutive packets lost (CPL), stored respectively in trace files *C1* and *C0*. We then use the latter two to tune the proposed HMMs.

In a HMM the number of states is not defined by the number of possible output events. To choose an adequate HMM configuration we propose starting from a very simple 2-state model as presented in the next section. We consider that one of the states models a currently broken path, where the probability for a packet to reach destination is zero. The other state models path availability, and the probability for a packet to reach destination is given by function *h(s)*, where s is the packet size. This function models packets lost mostly due to collisions, but also due to channel noise, packet fragmentation, buffer overflow, etc.

Starting from the 2-state model we can compare the model's output with the distributions used for its tuning, and assess if the desired degree of accuracy is achieved. If the results are not accurate we have to add more states to the model, and repeat the process until the results are satisfactory.

The characteristics of the routing protocol used can be useful to provide an insight on how to enhance the model (see section 5.1.3 for an example). In our experiments we did not have to use more than three states, showing that the model complexity can be kept low and still provide the desired behavior.

In the two following sections we show how to model the transmission of data streams on MANETs using a proactive routing protocol (OLSR) and a reactive one (DSR) by using 2-states and 3-states HMMs, respectively. In order to speed up the determination of the optimum values for the model parameters we also present, for each case and for each parameter, a set of heuristics that offer good initial estimates.

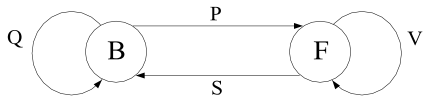


Figure 5.1: Two-state Markov chain for the multi-hop wireless path model

5.1.2 Two-states packet loss burst model

In this section we present our basic HMM that is, despite its simplicity, able to model large lost bursts. The idea is to focus on two distinct situations: when a path towards the destination is lost and no packet can arrive successfully, and when a path to the destination exists but some of the packets are dropped due to congestion, transmission errors, buffer overflow, etc. It consists of a two-states HMM based on the Markov chain shown in Figure 5.1 (also known as the Gilbert model).

State B models the situation where a path towards the destination has been lost; the probability for a packet to reach the destination is zero. In state F packets arrive to the destination with a probability defined by function $h(s)$, where s is the packet size. Mapping state B with 0 and state F with 1 we obtain the following transition probability matrix:

$$A_2 = \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix} = \begin{bmatrix} Q & P \\ S & V \end{bmatrix} \tag{5.1}$$

In this work we estimate the different parameters of the HMM using experiments based on the ns-2 simulator [KK00] as input. We have tested several different scenarios with different mobility and traffic patterns, and we have chosen one that was particularly representative in terms of large packet loss bursts. This choice aimed at stressing the model using a very demanding example.

Our setup consists of a 200 m × 200 m indoor scenario with 80 nodes. The wireless interfaces are based on the IEEE 802.11b standard with radio range limited to 50 m. The medium access used is the distributed coordination function (DCF). Node mobility is generated using the random waypoint model with the node speed uniformly distributed in a range from 0 to 2.4 m/s. The source of the reference flow sends packets with random sizes ranging between 10 and 2300 bytes at a rate of 50 packets per second. The background traffic consists of 4 UDP sources generating 512 bytes packets at a rate of 4 pkt/s. We evaluate both a reactive (DSR) and a proactive (OLSR) routing protocol. Applying a filter to the simulation's output we obtain a trace file (ST) where incrementing packet sequence numbers are tagged with either a 1 or a 0 (for packets received and packets lost/unusable respectively). Our criteria for this example is that all packets that arrive to destination in less than 300 ms are considered good packets (tagged 1); the remaining were tagged as lost (0). From this trace we obtain two other with the lengths of the sequences of *consecutive packets lost* (trace $C0$) and of the sequences of *consecutive packets received* (trace *C1*). These traces will be used as training sequences for the model calculation.

Using trace ST we first analyzed the correlation between packet size and the event of losing, or not, a packet. The correlation coefficient found is $r^2 = 6.03 \times 10^{-6}$, which indicates that the event of losing a packet is basically independent of the packet size. Hence, the probability function associated to state F will be fixed at a constant value $h(s) = H$. We found that this simplification is applicable to simulation results most of the times.

Using trace *C0* we calculate the ratio between the total number of packets lost and the sum of the lengths of CPLs sequences bigger than one (sum of packets lost in a bursty fashion). With trace *C1* we do the same with the packets received. We find that the ratios for the packets received are high (above 99%), as expected. The interesting result is that the ratios for packets dropped are also high (above 97%), indicating that packet loss bursts are the dominant cause of losses, contrarily to a random-loss situation. Since the main reason for packets lost in a burst is a route failure, these events shall take place mostly in state B.

From these results, and since parameter H accounts mainly for non-consecutive packet losses, we consider that $H = 1 - \varepsilon \approx 1$. This allows us to find an initial estimate for vector $v_e = (S_e, P_e, H_e)$, which contains the minimum set of parameters required to determine the entire system. We estimate S_e and P_e taking into account that the runs at each state of a Markov chain are memoryless, having by definition a geometric distribution. Using this information we find that run lengths for B and F states have an average size of $\frac{1}{P}$ and $\frac{1}{S}$, respectively. Therefore

$$S_e = \frac{1}{\mu_c}, \text{ and } P_e = \frac{1}{\mu_b},$$

where μ_c is the average length of the sequences of consecutive packets arriving (CPA), and μ_b is the average length of the sequences of consecutive packets lost (CPL) imposing CPL> 1, that is, after removing all isolated packet losses.

The value H_e is estimated using the transition probability matrix A_2. We can find the steady-state probability π for all states by evaluating $\pi = \pi A_2$. After finding π we can define the exact probability for a packet to arrive to destination, $p_{arrival}$, using the following expression:

$$p_{arrival} = H_e \cdot \pi_1 = H_e \cdot \frac{P_e}{P_e + S_e} \tag{5.2}$$

Since we have already estimated values for P and S, and since $p_{arrival}$ can be found using the simulation results, we can obtain from Equation 5.2 the value for H_e. This concludes our preliminary analysis whose purpose was to determine an initial value for vector v_e.

Starting from the vector of estimated parameter values $v_e = (S_e,\ P_e,\ H_e)$, we proceeded to find a more precise solution through an iterative process, which can be any one of the many available in the literature [Rab89]. We consider that our estimates v_e are close to the definitive ones, and so the method we use is a hybrid iterative/brute force technique. Starting from the estimated parameter values we select a search interval for each parameter testing several points in this interval and choosing the one that minimizes error function *f*. In the next iteration we reduce the search interval around the point that minimizes f in the previous iteration. We proceed with this algorithm until the output from function f is smaller than

Table 5.1: Estimated parameters values (v_e) vs. the values obtained through the iterative process (v_i) for both routing protocols.

DSR	v_e	v_i
P	10.786×10^{-3}	11×10^{-3}
S	1.357×10^{-3}	1.3×10^{-3}
H	0.99560	0.99998

OLSR	v_e	v_i
P	5.44×10^{-3}	5×10^{-3}
S	2.868×10^{-3}	1.85×10^{-3}
H	0.969	0.999

a pre-defined error value (ξ). This value defines the desired degree of accuracy of the model.

The minimization function we used was:

$$f = \mid \frac{\mu_{CPA-M} - \mu_{CPA-S}}{\mu_{CPA-S}} \mid + \mid \frac{\mu_{CPL-M} - \mu_{CPL-S}}{\mu_{CPL-S}} \mid \tag{5.3}$$

where μ_{CPA-S} and μ_{CPA-M} refer to the mean values of the consecutive packet arrival distribution for the simulator and the model output, respectively. In a similar way, μ_{CPL-S} and μ_{CPL-M} refer to the mean values of the consecutive packet loss distributions. We have chosen this function for minimization since it also allows to set bounds on the probability of packet arrivals. If we impose that $f < \xi$, and since $p_{arrival}$ can also be defined as:

$$p_{arrival} = \frac{\mu_{CPA}}{\mu_{CPA} + \mu_{CPL}} \tag{5.4}$$

we find that the relative error for $p_{arrival}$ (e) is bounded by $\frac{1-\xi}{1+\xi} < e < \frac{1+\xi}{1-\xi}$. We consider f to be a good choice because similar values for $p_{arrival}$ obtained from the simulator and the model will allow us to perform consistent comparisons when evaluating multimedia application's software. In fact, if we achieve similar distributions for CPL and CPA, but do not achieve very similar values for $p_{arrival}$, it would not be possible to validate the model against the simulator correctly. It would mean that different goodput values are achieved with the simulator and with the model, making any kind of comparison unfair.

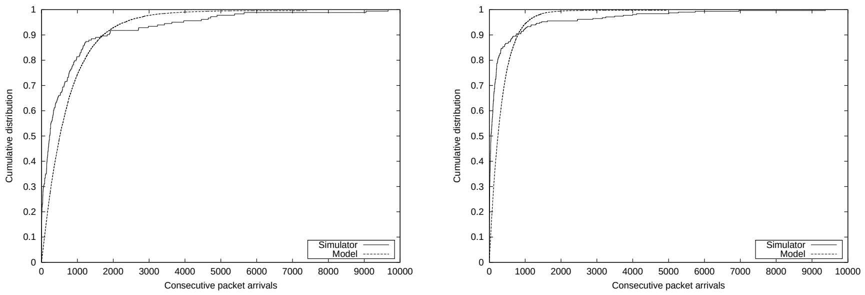
Table 5.1 presents both the v_e values and the values obtained through the iterative process (v_i). As it can be seen, the initial parameter estimates are very accurate.

In table 5.2 we present a comparison, in terms of consecutive packets arriving (CPA) and consecutive packets lost (CPL), of the accuracy of the initial and final estimated values relatively to the reference values obtained with the simulator. The simulator's results were obtained from traces *C1* and *C0*, and vectors v_e and v_i represent the initial and final estimates for the two-states model. We find that the heuristic proposed offers more accurate initial estimates for the DSR routing

Table 5.2: Statistical average matching for the estimated and iterated model values

DSR	Simulator	Model			
		v_e	error	v_i	error
μ_{CPA}	737,04	554,65	24,75%	746,58	1,29%
μ_{CPL}	86,91	71,53	17,70%	88,99	2,39%

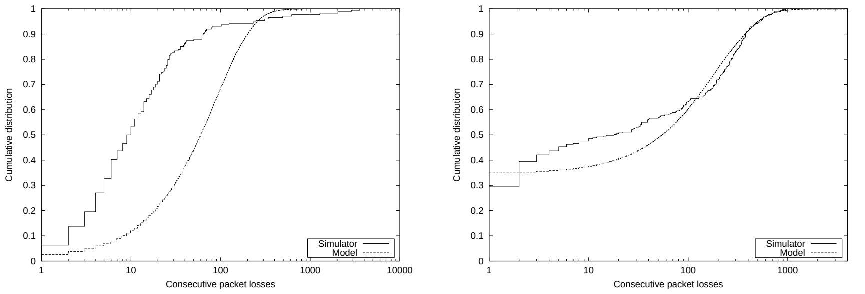
OLSR	Simulator	Model			
		v_e	error	v_i	error
μ_{CPA}	348,69	29,58	91,54%	346,96	0,50%
μ_{CPL}	129,99	17,49	86,55%	129,92	0,05%



a) DSR

b) OLSR

Figure 5.2: Cumulative distribution function of consecutive packet arrivals (CPA) for DSR (a) and OLSR (b).



a) DSR

b) OLSR

Figure 5.3: Cumulative distribution function of consecutive lost packets (CPL) for DSR (a) and OLSR (b).

protocol. When the iterative search concludes, though, we find that the error for the OLSR routing protocol is lower.

Figures 5.2 and 5.3 show a comparison, for both routing protocols, of the consecutive packet arrival patterns and the consecutive packet loss patterns, respectively. These results were obtained using vector v_i, allowing us to compare the probability density function and the cumulative distribution function for the simulation and model outputs. Figure 5.2 shows that the statistical distribution provided by the model has a close resemblance with the simulator's output.

Concerning the distribution of consecutive packet losses, figure 5.3 shows that the two-state model fails in accurately modeling the desired consecutive packets loss pattern for DSR. Concerning OLSR, we consider that the HMM is able to approximate the consecutive packet loss distribution satisfactorily.

The different precision of the results for the two routing protocols is due to their different routing nature. DSR belongs to the reactive family of routing protocols. These protocols are able to reestablish a path very quickly until there are no more available routes on the source node's cache. Afterwards they have to proceed with the possibly high time-consuming process of route discovery until communication is resumed. Proactive protocols, such as OLSR and TBRPF, rely on frequent "Hello" and topology update messages to manage the routing tables. Hence, these are not prone to present the asymmetry encountered with DSR, being more closely modeled with the two-states HMM presented before. Modeling more accurately DSR's distribution for consecutive packet losses can be done at the cost of introducing more complexity into the model. In the next section we show how a more accurate solution can be achieved through a three-states Markov model.

5.1.3 Three-states packet loss burst model

In this section we present an enhancement to the model described in the previous section which offers a higher degree of similitude to DSR's packet loss bursts distribution. Analyzing DSR's behavior we find that path breaks can either be short - if breakage is handled by a quick re-routing process using the node's cache - or long - if a route discovery process is required. Taking into account this different behavior, we replace state B from the two-states model with states L and R, where state L models short path breakages and state R models route discovery processes. The resulting three-states HMM is shown in Figure 5.4.

As in the two-states model, packets arrive to destination on state F alone, and with a probability of H. In states L and R all packets are lost. Mapping state L as 0, state F as 1 and state R as 2, we obtain the following transition probability matrix:

$$A_3 = \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} Q_1 & P_1 & 0 \\ S_1 & V & S_2 \\ 0 & P_2 & Q_2 \end{bmatrix} \tag{5.5}$$

We work with the traces described in the previous section: trace *ST* with the mapping between packet sequence number and packet received/lost events, trace *C0* with the lengths of the sequences of packets lost, and trace *C1* with the lengths

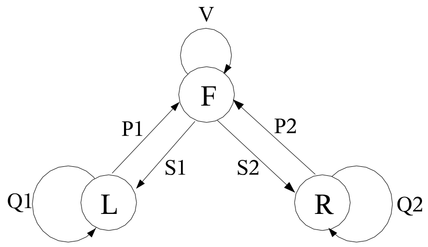


Figure 5.4: Three-states Markov chain for the multi-hop wireless path model

of the sequences of packets received. These traces will again be used as training sequences for model calibration.

As in the previous section, we maintain $H = 1-\varepsilon \approx 1$. We classify consecutive packet loss (CPL) events into two groups choosing a threshold t. The value for t can be chosen by determining the inflection point of the cumulative distribution for consecutive packets lost, or using any other criteria. Notice that the determination of the final parameter values through iteration is independent of the threshold t, but better guesses for t allow finding the final values with fewer iterations. In this example the value we have chosen for t is 200 (see figure 5.5), which is slightly above the inflection point and corresponds to an interruption of 4 seconds at a source rate of 50 pkt/s.

We now wish to determine the vector of estimated values $v_e = (S_{1_e}, S_{2_e}, P_{1_e}, P_{2_e}, H_e)$. We consider that μ_c is the average length of the sequences of consecutive packets arriving (CPA), μ_b is the average length of the consecutive packets lost (CPL) when their length is grater than 1 and less or equal than $t-1$, and μ_B is the average length of the CPL when their length is equal to or greater than t. We can then calculate the values for S_{1_e}, S_{2_e}, P_{1_e} and P_{2_e}using the following equations:

$$S_{1_e} = \frac{1}{\mu_c} \cdot P(\text{CPL} < t \,|\, \text{CPL} > 1), \tag{5.6}$$

$$S_{2_e} = \frac{1}{\mu_c} \cdot P(\text{CPL} \geq t \,|\, \text{CPL} > 1) \tag{5.7}$$

$$P_{1_e} = \frac{1}{\mu_b}, \text{ and } P_{2_e} = \frac{1}{\mu_B} \tag{5.8}$$

The values obtained from these expressions allow to evaluate the transition probability matrix A_3, and after that to determine the steady-state probability for all states, π, obtaining:

$$p(F) = \pi_1 = \left(1 + \frac{S_{1_e}}{P_{1_e}} + \frac{S_{2_e}}{P_{2_e}}\right)^{-1} \tag{5.9}$$

The expression $p_{arrival} = H_e \cdot \pi_1$ gives us the exact probability for a packet to arrive to destination, and it is used to calculate the value for H$_e$, thus completely

Table 5.3: Estimated parameters values (v_e) vs. the values obtained through the iterative process (v_i).

DSR	v_e	v_i
S_1	1.2735×10^{-3}	1.173×10^{-3}
S_2	0.08324×10^{-3}	0.07669×10^{-3}
P_1	59.21×10^{-3}	59.2×10^{-3}
P_2	0.79821×10^{-3}	0.7982×10^{-3}
H	0.99916	0.9999

Table 5.4: Statistical average matching for the estimated and iterated model values

DSR	Simulator	Model			
		v_e	error	v_i	error
μ_{CPL}	86,91	28,59	67,1%	85,82	1,25%
μ_{CPA}	737,04	268,15	63,6%	737,12	0,01%

defining vector v_e. We then find the final parameter values using the same iterative methods exposed in the previous section. Table 5.3 presents both the vector of estimated values (v_e) and the vector of values obtained through the iterative process (v_i). As it can be seen, the initial estimate is very accurate, reducing the number of iterations to find the final solution to a minimum.

Table 5.4 presents a comparison of the mean errors when comparing traces *C1* and *C0* obtained from the simulator with the same traces obtained using the three-states model using either the values from vector v_e or from vector v_i. The comparison is made in terms of consecutive packets arriving (CPA) and consecutive packets lost (CPL). Notice how slight differences in terms of the values of the different parameters result in great differences in terms of error values.

Figure 5.5 shows that, with this enhanced model, the probability density function and cumulative distribution function obtained resemble the desired distribution with a much higher degree of accuracy than the original model. It becomes evident that the introduction of two loss-states instead of one improves the behavior of the model's CPL cumulative distribution curve, reproducing very large bursts with more accuracy.

Concerning the consecutive packet arrivals distribution, both density and cumulative distribution functions are very similar to the ones obtained with the two-states model, as expected. Though the model could be further extended in order to achieve small values of consecutive packet arrivals, thus offering a better approach to the cumulative distribution curve of the simulator, we consider that it is an irrelevant issue to our purpose.

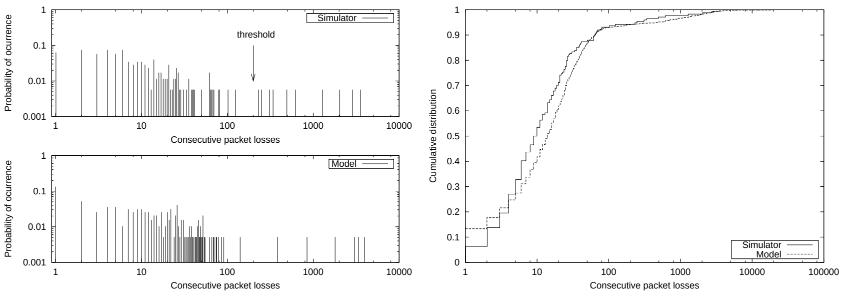


Figure 5.5: Probability density function (left) and cumulative distribution function (right) for packet loss bursts

5.2 Validation

We now validate the models proposed in sections 5.1.2 and 5.1.3, verifying their correctness and adequateness for the purpose of evaluating multimedia streaming applications.

With that aim we start by defining a set of metrics that allow measuring packet loss bursts. Afterwards we will apply these metrics to both simulator and model outputs in order to verify the effectiveness of our models.

5.2.1 Measuring packet loss bursts

Before detailing the different metrics proposed to characterize packet loss bursts we provide a definition of the boundaries of a packet loss burst. This definition is specifically designed for video and audio data flows. We consider that data flows belonging to different applications will not be affected by packet loss bursts in the same way. It is also important to point out that loss burst measurements are always done focusing on a single traffic flow, and not for all the traffic in the network, even if there are other similar flows.

We define the *burst start threshold* as the minimum number of consecutive packets that have to be lost to consider the presence of a burst. We also define the *burst end threshold*, as the minimum number of consecutive packets arriving to the destination after a loss burst to consider that communication has been adequately resumed. The burst start and burst end thresholds will depend on the type of information sent and the packetization granularity. For example, if we consider that at least one entire video frame has to be lost for a burst to be meaningful, and that one entire frame has to arrive for communication to be resumed, the burst start and end thresholds will have the same value, which will be equal to the number of packets per frame defined in the video codec.

We will now proceed to define some indicators that characterize packet loss burst occurrences. The most simple indicator is *packet burst percentage* (PBP), defined as:

$$PBP(\%) = \frac{\sum_{i=1}^{K} B_i}{N} \tag{5.10}$$

where B_i is the size of loss burst i in number of packets, K is the total number of loss bursts and N the total number of packets sent. The PBP gives a measure of the relative burst incidence. To measure the relative weight of bursts over the total number of packets lost L, we define the *Relative Burstiness* (RB) metric as:

$$RB = \frac{\sum_{i=1}^{K} B_i}{L}, \, 0 \leq RB \leq 1 \tag{5.11}$$

In a situation where most packets are lost in a random manner the RB parameter approaches 0, while, when packet loss bursts dominate, RB will be greater than 0.5.

This parameter allows us to detect where a MANET needs more improvements: if on the routing protocol side ($RB > 0.5$) or on the support for QoS to mitigate the effects of congestion ($RB < 0.5$).

Both these indicators are burst size independent. They equally penalize very small bursts occurring in a distributed fashion and very large bursts if the total number of packets lost is the same. From the user point of view, however, long communication breaks may be far more unacceptable. To take into account such discrepancies we introduce a new metric, the *Burstiness Factor* (BF):

$$BF = \frac{\sqrt{\sum_{i=1}^{K} B_i^2}}{N}, 0 \leq BF \leq 1 \tag{5.12}$$

The BF is a metric of the impact of the re-routing time of different routing protocols on a given flow; smaller values indicate that interruptions caused by routing protocols are either fewer or smaller.

Though BF is a good indicator to measure improvements on routing protocols, it does not take into account the relative position of the bursts, which might have a different impact on multimedia streams from a codec point of view. We therefore introduce a new metric called *media smoothness factor* (MSF), which we define as:

$$MSF = \frac{\sqrt{\sum_{i=1}^{T} F_i^2}}{N}, \, 0 \leq MSF \leq 1 \tag{5.13}$$

where T is the total number of *inter-bursts* or *burst delimited* periods, identified as F_i, and N is the total number of packets. Figure 5.6 shows an example on how to determine F_i and B_i zones. In this example the burst start and burst end thresholds are set to 3 packets, thus resulting in $K = 2$ and $T = 3$. Applying this threshold we have two well defined loss bursts and three burst delimited zones.

The MSF measures the *fluidity* experienced by a multimedia data stream; obviously, $MSF \gg BF$ must hold for communication to be sustainable. To better understand the different properties of BF and MSF we propose a case study scenario, depicted in figure 5.7, where we have a train of K bursts of length G, separated by exactly X packets. The burst sequence is centered so that Y packets

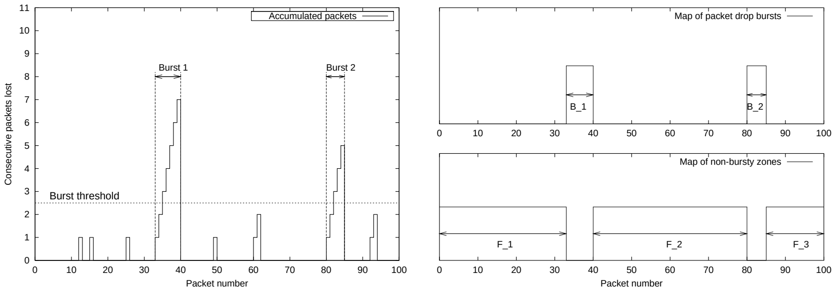


Figure 5.6: Example of plotting the values of F_i and B_i

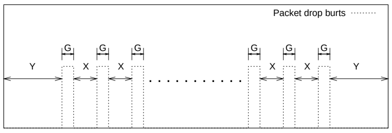


Figure 5.7: The BF vs MSF case study scenario

separate the first and last bursts from the beginning and end of the observation period, where $Y = (N - K \cdot G - (K-1) \cdot X)/2$.

In this scenario $BF = \sqrt{K} \times G/N$, which is independent from the inter-burst value (X), while:

$$MSF = \frac{\sqrt{(K-1) \cdot X^2 + 2 \cdot Y^2}}{N}, \tag{5.14}$$

depends not only on the size and number of bursts, but also on the distance between them. Considering that the upper limit for X (achieved when $Y = 0$) is given by:

$$X_{max} = \frac{N - K \cdot G}{K - 1} \tag{5.15}$$

we can normalize Equation 5.14 using $z = X/X_{max}$, thus obtaining:

$$MSF = \frac{N - K \cdot G}{N} \cdot \sqrt{\frac{z^2}{K-1} + \frac{1}{2} \cdot (1-z)^2} \tag{5.16}$$

Figure 5.8 shows the behavior of Equation 5.16 as a function of K, taking $\frac{G}{N} = 0.02$.

Equation 5.14 reaches its minimum when $x_m = y_m = \frac{N-K \cdot G}{K+1}$. This indicates that the minimum value of MSF is reached when interruptions on communication are evenly separated, that is, when distance between loss bursts is equal to the distance to the extremes.

The normalized expression for z_{min} is:

$$z_{min} = \frac{x_m}{X_{max}} = \frac{K-1}{K+1}, K \geq 2 \tag{5.17}$$

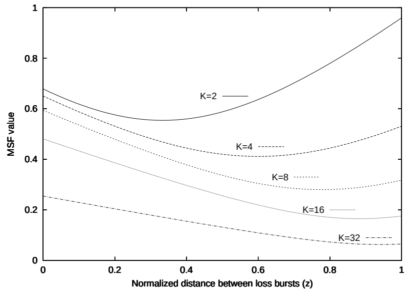


Figure 5.8: MSF variation with distance between bursts

which depends solely on the number of loss bursts present on the sequence.

We can directly check the correctness of this expression from the results of figure 5.8, and also check that it approaches 1 for large values of K. Since typically we will have a large number of gaps ($K \gg 1$), the MSF will be monotonically decreasing. This result allows us to conclude that MSF offers a measure of burst concentration for similar values of BF, increasing as the concentration of bursts increases.

Summarizing, we have defined four metrics for analyzing packet loss bursts: the PBP (burst percentage), the RB (relative burstiness), the BF (burstiness factor), and the MSF (media smoothness factor). These metrics give us different information about loss burst patterns, and they will help us in the model validation process that follows.

5.2.2 Validation process

We now apply the previously defined metrics to compare the two-states and three-states HMM results with the simulator's output when using the DSR protocol.

We set the burst start threshold equal to the burst end threshold for the sake of simplicity in the presentation of results.

Results show that the bursts' percentage over the total number of packets sent (PBP), as well as the RB parameter, vary with increasing thresholds for burst start/end values (see figure 5.9). The variation in terms of PBP, though, is minimal.

The RB parameter represents clearly the relation between packet losses that pertain to bursts and those that don't. As it can be seen, the three-states model approaches the reference curve from the simulator with much greater accuracy than the two-states model.

In figure 5.10 we compare the output from the HMMs and the simulator in terms of the Burstiness Factor (BF) and the Media Smoothness Factor (MSF). In terms of the BF metric, the impact of varying the burst start/end threshold is not significant, as occurred for the PBP metric.

We also observe that, in terms of both BF and MSF metrics, the results for the three-states HMM are much closer to the reference values. We consider that the degree of accuracy achieved is acceptable for applications such as video codec

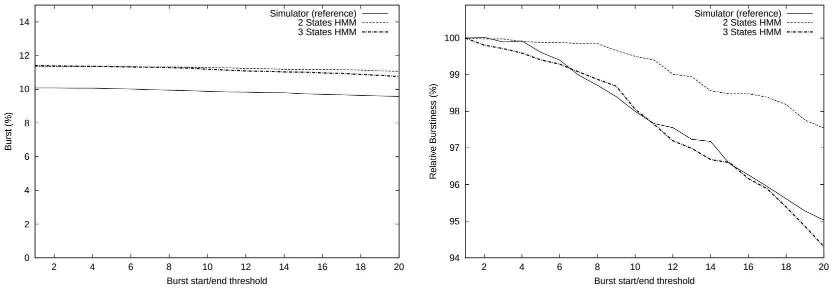


Figure 5.9: Comparison of the two-states and the three-states HMMs in terms of PBP (left) and RB (right) accuracy.

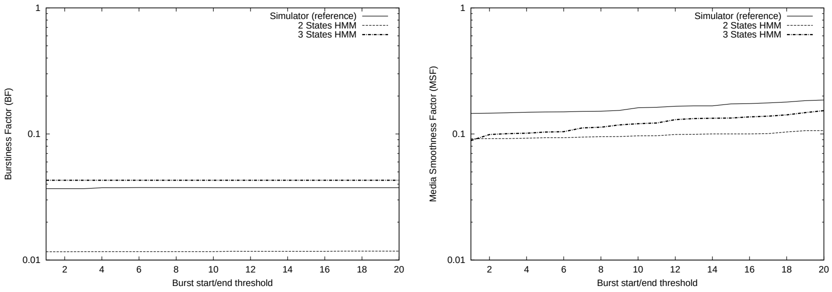


Figure 5.10: BF (left) and MSF (right) comparison

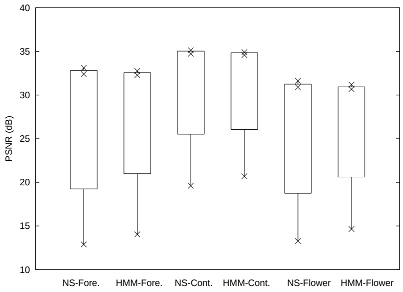


Figure 5.11: Box plots for the video distortion achieved with the simulator and HMMs for the *foreman*, *container* and *flower* QCIF video sequences.

enhancing and tuning. In terms of the MSF, which takes into account consecutive packet arrivals instead, notice that the three-states HMM approaches the reference MSF value with increasing thresholds. This slight difference is expected since the accuracy of the consecutive packet arrivals distribution was not the main focus of our model. It could be improved by increasing the number of states in the HMM, similarly to was done for DSR's consecutive packet losses distribution.

To further validate our model, we now proceed comparing the results in terms of video quality achieved with the model against the ones obtained with the simulator. To perform this comparison we replace the randomly generated data stream used initially with the video streams of well-known QCIF video sequences, namely: *foreman*, *container* and *flower*, each replicated to be 300 seconds long. In our analysis we used the H.264 video codec framework [H2603a] in order to obtain both the input video trace files and the output video quality results for the simulator and the model.

The metric we use is video distortion, also known as Peak Signal-to-Noise Ratio (PSNR), which is the most commonly used objective video quality metric.

In figure 5.11 we present a box-plot comparison between the simulation values and the model values using the three video sequences under test. In each box plot we represent the minimum, the median, and the maximum values (the three crosses), as well as the box (rectangle) which contains the values between the 0.250 and the 0.750 quantiles of the data. The figure clearly shows that the results achieved with the model closely resemble the ones achieved via simulation. These results, along with the previous ones, allow us to conclude that the model proposed offers a behavior quite similar to the one we wish to obtain.

5.3 A model's application example

In this section we illustrate the applicability of our models by using them as a tool to speed up the evaluation and tuning of a video codec (in this example, an H.264 video codec). We measure the impact of the different steps required for simulation and data extraction when using either the ns-2 simulator or HMMs; we do it for

Table 5.5: Duration of the different simulation steps using a) the ns-2 simulator or b) HMMs.

a)

	DSR	OLSR
Mobility generation time (s)	840	840
Single simulation time (s)	1320	9720
Extraction of packet loss details (s)	60	60
Total time for 100 simulations	61h40m	295h

b)

	DSR	OLSR
Mobility generation (s) - once	840	840
Single simulation time (s) - once	1320	9720
Extraction of packet loss details (s) - once	60	60
Determination of model parameters (s) - once	3600	3600
Single simulation time using model (s)	0,40	0,39
Total time for 100 simulations	1h38m	3h58m

both the DSR and the OLSR routing protocols.

We simulate the streaming of a typical movie 1 hour and 30 minutes long. The results presented in table 5.5 allow comparing the time consumed at each step using the ns-2 simulator alone, or using HMMs in conjunction with ns-2. The values presented are achieved on a dual 2,6 GHz Pentium-IV server with 2 Gbytes of RAM running GNU/Linux version 2.4.22.

Notice that, when using ns-2 only, steps I, II and III must be repeated every time. When using HMMs steps I to IV are only performed once, and step V is the only one repeated for each additional simulation run.

Results show that, when relying on HMMs, most of the time is consumed in simulation and in the determination of model parameters. Once that is done, though, the execution of the model is very quick. Relatively to the entry named "Determination of model parameters", we wish to point out that this time takes into account not only the time to determine the initial estimates for the different parameters (v_e), but also to find the final iterated values (v_i). In the bottom of both tables we present the estimated time to run 100 simulations, a value required to extract statistically significant results when tuning, e.g., a video codec.

Relatively to the improvements achieved, we find out that our algorithm allows executing experiments 38 times faster when using DSR, and up to 74 times faster when using OLSR. This difference is due to the time taken by each simulation run when using OLSR: 9720 s.

In terms of trace file output we find that, comparing trace file sizes, the model's output is 300 to 12000 times smaller than the simulator's output, though the output from the last can be reduced. Concerning real-life experiments, the trace file size can be reduced to the size of the HMM's trace file.

5.4 Conclusions

In this chapter we proposed an alternative to evaluate multimedia streaming applications in MANETs avoiding repetitive, time-consuming simulations or tests in real environments. Our solution, based on hidden Markov models, allows assessing the effects of packet loss and arrival patterns when streaming a compressed audio/video sequence in MANETs using different routing protocols. Results show that a two-states model is effective in modeling packet loss bursts when using a proactive protocol such as OLSR, though failing to accurately model MANET behavior when using a reactive protocol such as DSR. This occurs because DSR's mechanisms present a higher level of asymmetry, thus requiring a three-states HMM. We finally validated our models showing that the proposed HMMs provide similar results in terms of the loss burst metrics we defined, and also in terms of video distortion.

Finally we showed an application of our models to the design and evaluation of a video codec, obtaining very significant gains in terms of simulation time and disk space usage.

Overall, we consider that the strategy presented in this paper has proved to be an adequate alternative to the developers of multimedia streaming applications for MANETs, showing excellent results in terms of both the accuracy and the speedup achieved.

Chapter 6

Reducing the impact of mobility on video streams

As discussed in chapter 4, path instability provoked by the mobility of stations is one of the main problems that real-time multimedia streams encounter in MANET environments.

This chapter is dedicated to proposing techniques to reduce the impact of node mobility on MANET traffic in general, and particularly on real-time video streams. Such techniques consist in altering one of the routing protocols available for MANETs (DSR) so that it can use multiple routes, reducing the chances that communication between end stations is completely interrupted.

This chapter is organized as follows: in section 6.1 we propose some changes to DSR's route discovery mechanism so that stations are able to find more routes. Section 6.2 is dedicated to the tuning of one of the route discovery mechanisms proposed in section 6.1, assessing the benefits it brings to multimedia streams by increasing the number of backup routes for each node. In section 6.3 we propose enhancements to the packet handling mechanism in order to do per-packet traffic splitting through different routes. A joint evaluation of the different techniques proposed throughout the chapter is made in section 6.4, and section 6.5 presents the conclusions that we derive from the analysis made in this chapter.

6.1 Extending DSR to find more routes

DSR is a protocol that, by default, finds only a small number of routes; it does so by provoking a controlled broadcast storm that has a relatively small routing overhead, and that operates rather quickly. By extending the route discovery mechanism in DSR we are able to increase the average number of routes found per node. This extra information increases the route choices at nodes when a route is lost, so fewer route discovery processes are required.

As an enhancement, nodes can use the extra routes available for other useful purposes, like packet splitting or packet replication over disjoint routes, route congestion analysis, QoS routing, etc. Obviously, the success of these techniques depends on how good the available routes are, and how low are we able to keep the additional routing overhead generated. When referring to the goodness of a set of routes we are talking about their degree of disjointness. This is because, when mobility causes a link breakage, all the paths using that link are removed from cache. So, if routes that do not rely on that link are available (link disjoint routes), communication can be resumed with low interruption times.

In this section we propose enhancements to DSR's route discovery mechanism based on the proposal of Lee and Gerla [LG01] referred in section 2.3.5. For the sake of clarity, we refer to their solution as LG from now on.

In the LG route discovery algorithm the route discovery process is altered so that during the "RREQ" propagation phase packets with the same route request ID can be forwarded if they arrive *"through a different incoming link than the link from which the first RREQ is received, and whose hop count is not larger than the first RREQ"*. It is up to the source to analyze the disjointness of the paths found. In this work we tried to follow their route discovery proposal strictly in order to evaluate the performance of our own proposals.

Based on the LG proposal, we will propose an alternative route discovery strategy which we denoted as "Super Restrictive mode" (SR). Starting from the basic DSR approach, the SR algorithm enhances it in order to allow more routes to be found. We compare SR with the LG proposal in order to assess its effectiveness. In addition to SR, we also propose the RLG (Relaxed LG) solution which, contrarily to SR, surpasses the LG proposal in terms of the routing overhead generated.

6.1.1 The Super Restrictive route discovery algorithm (SR)

The solution we propose in this section, the SR algorithm, aims at offering an enhancement to DSR's route discovery technique so that the average number of routes found per node is increased.

The implementation of the SR mode required enhancing DSR's code by adding a new data structure to the already existing route discovery table structure on each node. So, every node is able to keep track of all the last hops that forwarded the current route request.

The LG solution presented previously is simpler, keeping track of just the first node through which a RREQ packet arrived. This approach has the drawback of not limiting the maximum number of route request packets propagated per node, possibly allowing the routing overhead to grow without bound. So, the main difference between the SR and the LG algorithms is that the former discontinues the propagation of a route request arriving through a repeated last hop. The reason behind this choice is that, when the last hop is the same than that of a previous request, there will be at least one link in common between the routes. Therefore, the usefulness of these extra routes will be reduced. With the SR mode we expect to decrease the routing overhead significantly when compared to the LG solution.

For the moment we will limit the maximum route size to the one of the first route request, the same heuristic as for the LG solution. However, the propagation process of the "Super Restrictive" mode will be further enhanced and tuned in section 6.2.

6.1.2 Relaxed LG (RLG)

The RLG solution we propose here is similar to the LG solution, except that the restriction concerning the last hop through which the first message arrived is no longer used. This means that nodes will propagate RREQs packets with the same route request ID even if they arrive through the same input link. The route size restriction, though, is maintained.

To better explain the difference between DSR, LG and RLG route discovery techniques we will now expose what would happen in a scenario such as the one presented in figure 6.1.

During a route discovery process using the standard DSR implementation a *RREQ* packet would arrive to node 3 through both sub-paths *a* and *b*. Only the first one arriving would be propagated to the receiver through both sub-paths *c* and *d*. As it can be seen, these routes are not link disjoint, though link disjoint routes do exist.

The propagation technique used in LG allows node 3 to propagate both RREQ packets arriving through sub-paths *a* and *b*. However, only the first one to be forwarded will arrive to D through both c and d sub-paths; the second one is retained at nodes 4 and 5 because the previous node on the path (node 3) is the same.

By weakening of restrictions of the LG solution, RLG allows up to 4 possible routes to be found: {S-1-3-4-D}, {S-2-3-4-D}, {S-1-3-5-D} and {S-2-3-5-D}. As it can be observed, this solution is less restrictive that the LG solution, which means that routing overhead shall be increased accordingly.

6.1.3 Analytical comparison of LG, RLG and SR route discovery strategies

To further evidence the differences between the LG, RLG and SR route discovery methods we will now show, with an example, what is the upper bound on the routing overhead generated by each of them. Our example (see figure 6.2) is based on the use of groups, which represent stations that are nearby. Every node on group *i* is able to listen to packets coming from every node on group *i-1*, and is

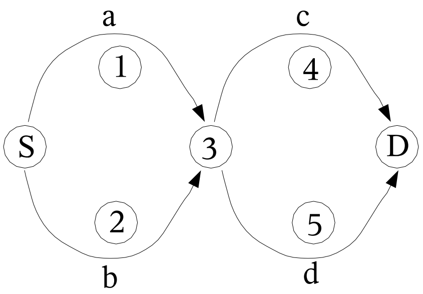


Figure 6.1: Multiple paths applying the RLG solution.

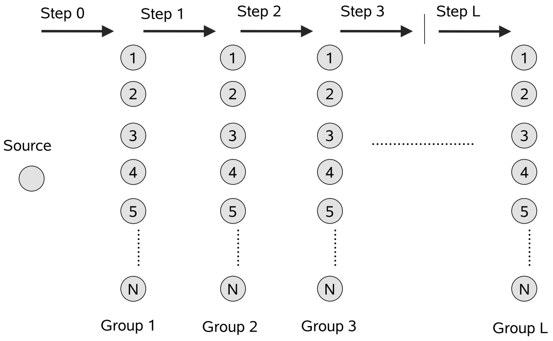


Figure 6.2: Example scenario based on the use of groups

Table 6.1: Upper bound on the number of packets generated at each step for the different propagation strategies

Prop. strategy	Step 0	Step 1	Step 2	Step 3	Step L
Normal	1	N	N	N	N
RLG	1	N	N^2	N^3	N^L
LG	1	N	N^2	$N^3 - N^2 + N$	$N \times \sum_{i=1}^{L} (N-1)^{i-1}$
SR	1	N	N^2	N^2	N^2

able to transmit to every node on group *i+1*. Nodes belonging to groups that are not nearby (e.g., 1 and 3) can not establish communication; every group is composed by N nodes.

In table 6.1 we present the maximum routing overhead generated (number of packets) at each step of propagation. We include data relative to the DSR routing protocol (denoted as “Normal”) for comparison.

On step 0 the source sends a RREQ packet to be propagated throughout the network. Nodes of group 1 receive this packet and propagate it, causing N packets to be generated; this is in common for all propagation strategies. Nodes of group 2 then propagate the route request again, causing the number of packets generated to grow to the square value of those that are normally generated by DSR. So, we find that after L steps the normal overhead of DSR is maintained at N packets, and that the SR mode is able to bound this overhead to N^2 packets. Both LG and RLG strategies are prone to generate an excessive overhead, with a growth in the order of $\theta(N^L)$.

In section 6.2 we will further reduce the overhead of the SR propagation technique, so that the routing overhead generated grows with $\theta(N)$ instead of $\theta(N^2)$.

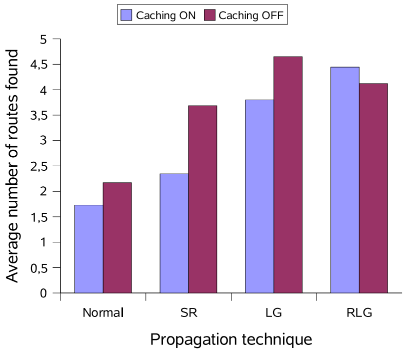


Figure 6.3: Average number of routes found

6.1.4 Simulation setup and results

For the evaluation being made on this section we will again use the ns-2 simulator [KK00] since it models collisions occurring at the physical level, an issue which we believe to be very important. Otherwise, the route request process would not suffer performance degradation caused by losses, being able to find all the best routes to a certain destination on every attempt, a result which is not accurate.

In order to perform an initial evaluation of the different propagation strategies under study we generated a scenario with dimensions 1500×300; 40 nodes are randomly positioned in that scenario. We impose a static setup which aims at measuring the total number of routing packets generated by each technique in a controlled environment. In later sections we will present further results varying mobility and traffic conditions.

In the interval comprised between 0 and 20 seconds half of the nodes start a route discovery process towards a destination node belonging to the other half. This is achieved by creating a client application that generates only a single small UDP packet, which as a consequence starts a route discovery process on that node.

Concerning the use of cache in DSR, we performed tests with both cache on and off. Turning the cache on means that intermediate nodes can reply to RREQ packets instead of the destination if they have a valid route in their cache; turning it off means that no node except the destination itself can reply, and so RREQ packets are propagated all the way. By default, DSR turns cache on to reduce the routing overhead generated.

In figure 6.3 we present the average number of routes found per node with the different propagation techniques under test. As it can be seen, our purpose was met correctly since the SR algorithm allows a node to obtain a number of routes which is higher than that achieved with the normal route discovery method, though lower than that achieved with the LG and RLG techniques.

Contrarily to what was expected after the analysis made in the previous section, the RLG solution can not find a superior number of routes when the cache is turned off. To understand the cause of this phenomena we analyze the routing overhead

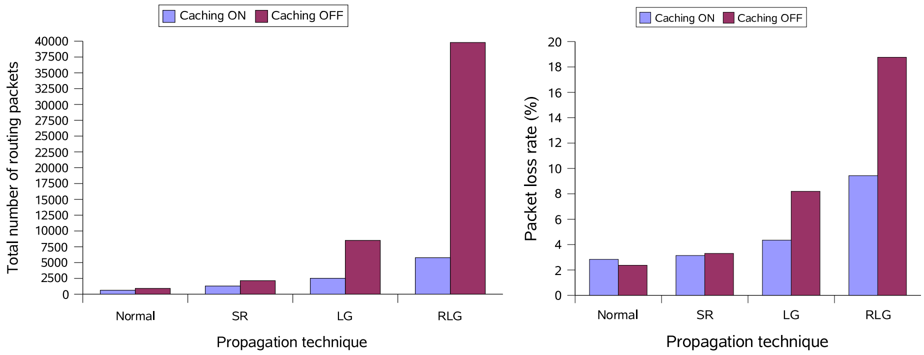


Figure 6.4: Routing overhead (left) and packet loss rate (right) for different propagation techniques

Table 6.2: Packet loss reason (Cache ON)

Packet loss reason	Normal	SR	LG	RLG
Address resolution	13,3	36,8	92,2	297,6
MAC Collision	2,0	3,0	10,7	19,9
Queue full	0,0	0,0	5,9	225,7
Other	2,8	0,9	0,8	1,2
Total	18,1	40,7	109,6	544,4

and the routing packets lost during the RREQ propagation period (see figure 6.4). We find that the RLG solution generates too much routing overhead, especially when route caching is turned off. Consequently, the packet loss rate for routing traffic is also quite high.

Concerning the LG solution, we see that it does not perform very well compared to the normal and SR route discovery methods. In fact, it generates more that twice the overhead with cache turned on, and more than five times the overhead when replies from cache are not allowed.

On tables 6.2 and 6.3 we detail the reasons why packet losses occur. By looking at the number of packets lost on the interface queue we find that the RLG solution clearly generates too many control packets, and so its use is not recommended. On the other hand, both normal and SR methods practically do not suffer from losses on the queue, being the total amount of losses quite lower than for the LG and RLG solutions.

Queue dropping is an event that is related to network overloading, and that in our example becomes especially relevant when the cache is turned off. On the other hand, the number of packets lost at the physical level (MAC collisions) gives a measure of the magnitude of the broadcast storm generated, being also a measure of adequateness for the different route discovery techniques used.

Table 6.3: Packet loss reason (Cache OFF)

Packet loss reason	Normal	SR	LG	RLG
Address resolution	17,7	64,0	214,4	510,5
MAC Collision	2,6	6,2	30,7	157,5
Queue full	0,0	0,3	452,1	6751,3
Other	1,6	0,6	1,0	46,9
Total	21,9	71,1	698,2	7466,2

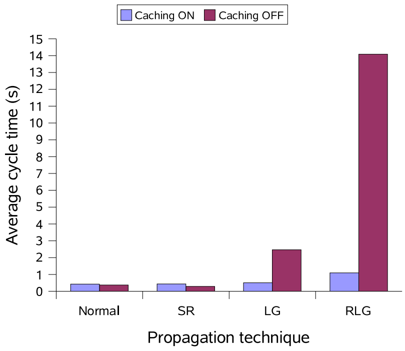


Figure 6.5: Average time for route discovery cycle

To clearly notice the duration of the broadcast storm generated during a route request cycle, we present at figure 6.5 the average time it takes to complete.

As it can be seen, the normal and SR solutions have acceptable values (much lower than one second), while the LG and RLG solutions present much higher values. In fact, except for the LG technique with caching on, the rest of the values are too high, being therefore prohibitive.

Based on the results found on this section, we drop the RLG proposal by considering it inadequate for MANET environments.

6.2 Enhancements to the SR algorithm

In the previous section we presented an enhanced route discovery algorithm for DSR, denoted as *Super Restrictive* (SR).

In this section we propose new enhancements to the SR technique by restricting even more the propagation process, and also by increasing its flexibility. Both proposals are independent and can be used together.

6.2.1 Increasing propagation restrictions

As described previously, the SR algorithm maintains a list with all the last nodes through which a route request arrived. However, as long as the last hops differ, there is no limit to that list's size. This could result in the propagation of an excessive number of route request packets when the MANET is too dense. So, we now propose an extension to this method by restricting the list to a certain size. By doing so a route request is not propagated by a node if, upon arrival, the list of that node for that particular route discovery event is already full; this means that only a pre-defined number of route requests is forwarded.

From now on we shall refer to the maximum size for this list as PNC (Previous Node Count). If the list size is very large, we obtain the same behavior as for the original SR algorithm; if it is reduced to one, the behavior becomes similar to the default DSR propagation algorithm.

If we now go back to the analysis presented in table 6.1 we conclude that, after L steps, the number of packets propagated can be maintained at $PNC \cdot N$ if the restriction proposed is applied. So, this simple change is enough to avoid a $\theta(N^2)$ increase, as we desired.

Besides finding extra routes, it would also be desirable to improve the goodness of the additional paths found. A frequently used measure of the goodness of a path towards another path is the degree of disjointness. When referring to path disjointness we can distinguish between link disjointness and node disjointness. Node disjoint paths are those which do not have nodes in common except for the source and destination nodes; link disjoint paths are those where all links differ, though nodes in common may exist. So, node disjointness implies link disjointness, but the opposite does not apply.

The link disjointness condition is less restrictive than the node disjointness condition, and so we will adopt the later. In practical terms this means that, in order to propagate a route request packet, a station must first assess if the path traversed by that packet is node disjoint relatively to any other paths traversed by the previous route request packets received (for that same route discovery event).

Though we find that the routing overhead is not greatly affected when introducing this last constraint (we experience a reduction of about 10%), in terms of path quality we expect to have relevant improvements. If traffic splitting algorithms are applied, these improvements will be reflected in the performance experienced by real-time streams in the presence of mobility. This is topic of later sections.

6.2.2 Increasing the flexibility on propagation

The SR algorithm, as well as its enhancement presented in the previous section, both aim at restricting the number of route requests propagated during a route discovery process compared to the LG algorithm. Also, both SR and LG algorithms restrict the route request forwarding process to route sizes not superior to the first one arriving.

In this section we propose an increase of flexibility by allowing routes with an additional number of hops up to a certain value (the flexibility parameter) to be also forwarded. This technique allows alternate routes to be found in scenarios

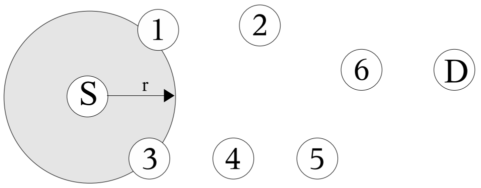


Figure 6.6: Scenario that benefits from increased flexibility on route discovery

similar to the one presented on figure 6.6, where source S wants to find routes to D; the transmission range for all stations is r.

As it can be seen, if flexibility is increased to 1 or more the source is able to find two routes to the destination, which are mostly disjoint (except for the last node). These are: {S,1,2,6,D} and {S,3,4,5,6,D}. Otherwise, it would just find the first of these routes.

In general, the increase of flexibility can have two different effects in terms of route propagation. On one hand it allows more routes to be found, and so the routing overhead should increase. However, if the number of requests that each node is going to propagate is small (as when PNC is as low as 2), then it can result in a reduction of the route request cycle time, especially for dense scenarios. Such analysis is made in section 6.2.3.

6.2.3 Tuning of the Super Restrictive mode

In this section we will vary both the PNC and the Flexibility values to evaluate the trade-off achieved between number of routes found, routing overhead generated and cycle time.

This evaluation, as well as the ones presented in the next sections, are done using the ns-2 simulator [KK00]. In our experiments we use the same 1500×300 static scenario with 40 nodes used in section 6.1.4. We tested several <***Flexibility, PNC***> pairs in order to study the impact on each of these parameters.

Figure 6.7 shows the average number of routes found varying the flexibility in a range from 0 to 3, for values of the PNC parameter ranging from 1 to 4. As it can be seen, all solutions allow more routes to be found compared with the original DSR implementation (PNC=1). When the cache is turned off the average number of routes is further increased. Figure 6.7 also shows that increasing PNC does usually translate into more routes found. On the contrary, increasing the flexibility parameter may, at times, reduce it instead.

In terms of routing overhead, we can see from figure 6.8 that the extra routes found have a cost, as expected. We find that, when increasing the flexibility value from 0 to 1, there is a considerable increase on the routing overhead, which remains more or less stable afterwards.

Concerning packet losses, the packet loss rate is below 4% in all simulations that we executed, which is an acceptable value taking into account that there is no way to avoid collisions for packets broadcasted using the IEEE 802.11 technology.

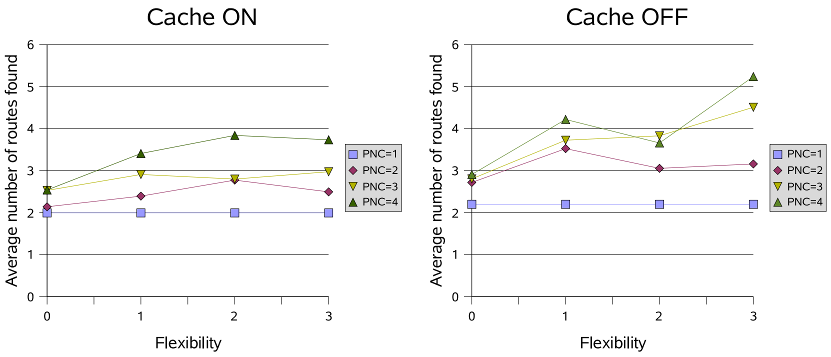


Figure 6.7: Average number of routes found

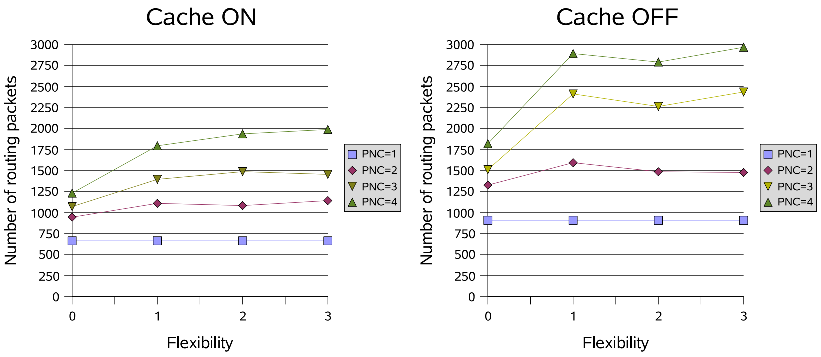


Figure 6.8: Routing overhead

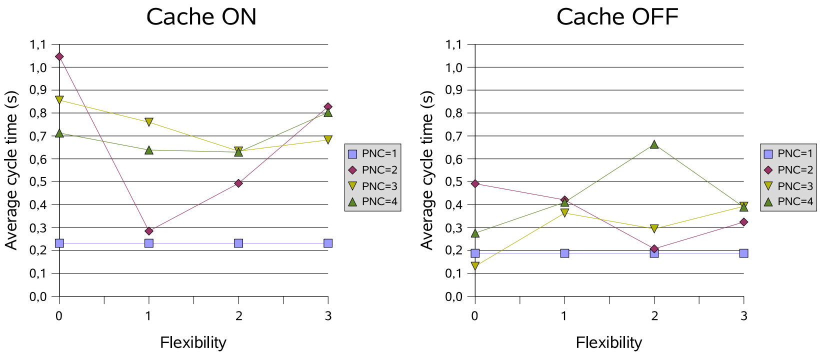


Figure 6.9: Average time for the route request propagation cycle

Table 6.4: Different test modes using SR

Mode	Flexibility	PNC
1	0	2
2	2	2
3	0	4

To conclude this evaluation of the SR mode we also analyzed the average cycle time, this is the time that goes from the moment a route request is generated to the time when the last message associated with that request is received. The results are presented on figure 6.9. In general, turning the cache off results in a reduced cycle time, which is rather unexpected. When the maximum propagation count is 2 we find clearly a minimum when Flexibility equals 1 for cache on and 2 for cache off. This phenomena has been referred to in section 6.2.2.

Based on the results of this section, we have chosen three different *Flexibility / PNC* pairs to proceed with our study. These are presented on table 6.4. We have assigned a test mode number to each pair of parameters for the sake of simplicity on the tests that follow.

In mode 1 the propagation using the SR technique is restricted to the maximum, so that only one extra route per node is allowed relative to the default DSR behavior. Modes 2 and 3 maintain one of the parameters of mode 1, but in mode 2 we increase the value of the flexibility parameter by two, and in mode 3 we increase by two the PNC parameter instead.

We consider that having the cache turned off is the most reasonable choice when attempting to maximize the number of disjoint paths found; therefore, the cache is implicitly turned off for all three test modes in the experiments that follow.

6.2.4 Validation of the proposed SR modes

In the previous section we analyzed the impact of using different combinations for the PNC and Flexibility parameters in a static scenario, and we finished by choosing a set of modes for further study. We now proceed with a performance evaluation comparing the impact of the different SR test modes defined on table 6.4 in terms of their scalability properties.

Our validation of the different SR modes consisted of comparing the results obtained under different conditions against the standard DSR implementation. Tests are made with the ns-2 simulator, and we use different scenario sizes and variable node densities.

Each node is equipped with an IEEE 802.11b radio interface transmitting at 11Mbps, and the radio range is 250 meters. We loaded the network with 10 UDP sources, each sending 512 byte packets at a rate of ten packets per second. The traffic from each source starts at a random time between zero and ten seconds. Afterwards the simulations run up to 900 seconds.

Concerning node movement, it was generated using the random waypoint mo-

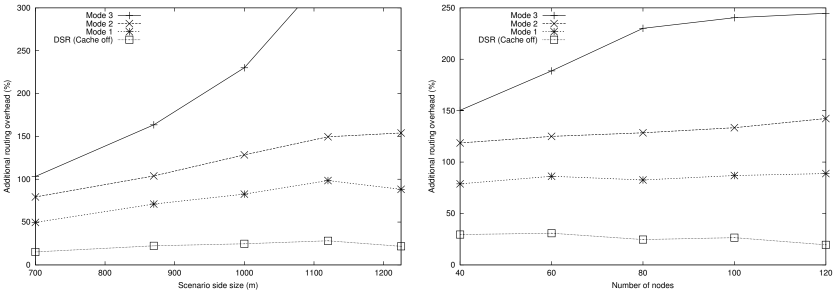


Figure 6.10: Fixed node density and variable size (left) and fixed size and variable node density (right)

bility model bundled in ns-2; we set the node speed to be generated in a range from 0 to 10 m/s; an extra script was created in order to accept only scenarios without network partitioning (no unreachable destinations), in order to obtain a connected graph. To perform a fair comparison, each scenario generated is tested with all protocols under evaluation. Each depicted value is an average of 5 simulation runs.

We now present the results achieved for the different SR modes in terms of additional routing overhead compared to the standard DSR implementation. We also include values for DSR with cache turned off as reference.

On the left side of figure 6.10 we present the relative routing overhead generated by the different modes with respect to DSR. We use a squared scenario with side sizes varying from 700 to 1225 meters; the node density is fixed at 80 nodes per squared kilometer. In terms of packet arrivals, the difference of all modes compared with DSR does not exceed 3%, except for the 1225×1225 m scenario where the difference is slightly higher.

We can observe from figure 6.10 that mode 1 does not generate much additional overhead, being that this value never surpasses 100%. Mode 2 provokes a higher routing overhead than mode 1 as expected, though the increase is not excessive. Mode 3, on the other hand, faces scalability problems for large scenarios.

To complete our scalability analysis, we make a similar evaluation varying the node density in a squared scenario with size fixed at 1000×1000 meters. The right side of figure 6.10 presents the results achieved in terms of relative routing overhead with respect to the standard DSR implementation. In terms of packet arrivals, the difference compared with DSR, for all modes, does not exceed 3%. Concerning modes 1 and 2, they perform well for all node densities, being that again mode 1 offers lower overhead than the remaining SR modes.

6.2.5 Evaluation of the SR mode in typical MANET scenarios

Previously we evaluated how well did the different SR modes scale. In that evaluation we generated only a little amount of UDP traffic, so as not to cause routing protocols to collapse due to congestion. However, we consider that it is also im-

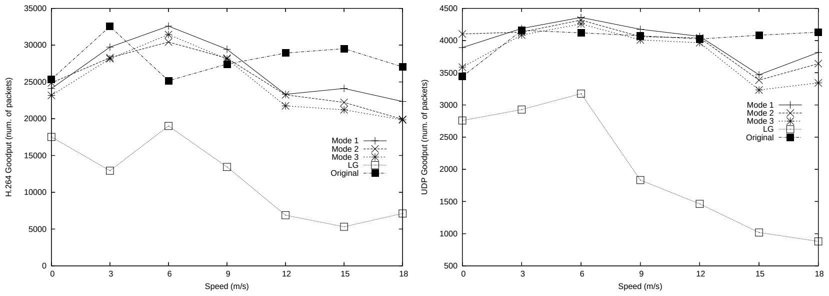


Figure 6.11: H.264 Goodput and UDP Goodput at different mobility levels

portant to assess how routing protocols behave when we have different degrees of mobility and different kinds of traffic sources in the MANET. We will study if, under varying conditions, the traffic delivery rate does not decrease, and the routing overhead is kept within acceptable bounds.

In our framework we used the same *Flexibility / PNC* pairs presented in table 6.4.

The scenario used in our simulations has 40 nodes in a 1500×300 area, and the simulations run for 310 seconds. The scenario generation process uses the random waypoint mobility generator that comes bundled with the ns-2 simulator; the output of the generator has been restricted with the help of a script that assures that no node partitioning phenomena occurs during the simulation. This is done in order to assure that all packet drops are solely due to congestion or routing problems.

We used 3 FTP/TCP traffic sources, 3 CBR/UDP sources and 2 H.264/RTP/UDP video sources. The different sources of traffic start at a random time between 0 and 10 seconds, and generate traffic until the end of the simulation. One second before the beginning of each H.264 video flow, one UDP packet is sent to the destination to initiate the session; this is to assure that a route is available for video streaming. The average bit rate for each H.264 video flow is 186 kbit/s. H.264 packets are identified in the simulator as video packets, so that they have a higher priority on the interface queue. Therefore, only routing packets have higher priority that video packets. Concerning CBR traffic, each source sends 512 byte packets at a rate of 20.5 kbit/s.

For each speed value we generated 10 different scenarios where all DSR modes have been evaluated. So, all the points depicted are the average values obtained from these 10 distinct scenarios.

We will now present the simulation results achieved by using the simulation framework described above. We compare each of the three SR modes proposed, using the original DSR implementation as reference; we also include data for the LG proposal, so that it becomes evident that it is a route discovery mechanism which does not offer high performance under heavy traffic loads.

Figure 6.11 shows the results achieved in terms of goodput at different speeds. As it can be seen, even though nodes are not using the extra routes available to

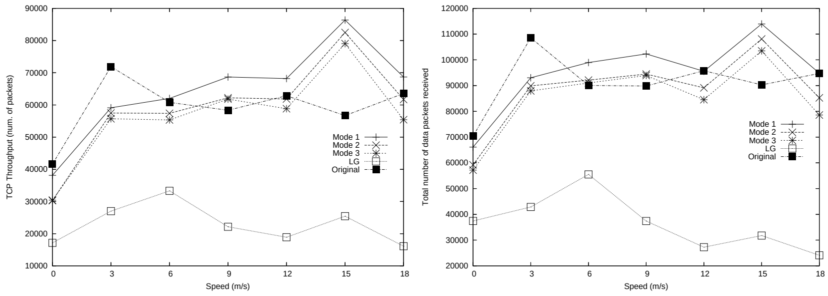


Figure 6.12: TCP Goodput and total number of data packets received at different mobility levels

perform traffic splitting or replication through them, the effect of having more routes in the cache is quite visible, being the results superior to the original DSR implementation for moderate mobility (H.264 streams) and for low mobility (UDP traffic). Relative to the LG solution, we find that its performance is quite poor, as we expected from previous results.

Figure 6.12 shows the results achieved in terms of TCP performance and total number of data packets received on the network. The TCP improvements at high node speeds show clearly the increased route stability caused by having more routes on cache. We therefore believe that such improvements are the main cause of the worse video and UDP results at high speeds. In terms of global performance, mode 1 surpasses DSR for speeds over 5 m/s. We can also see from that figure that, when the speed reaches 18 m/s, the advantage of having more routes available is lessened; this can be explained by the fact that having many routes becomes less useful as the topology change rate achieves very high levels.

In terms of routing overhead, figure 6.13 shows that the difference between the different SR modes and DSR is minimal; in fact, the overhead for SR mode 1 is even lower than DSR's at average speeds. Obviously, when the speed is zero, there is no benefit by obtaining further routes since no route will ever be lost. The lack of efficiency of the LG algorithm is clearly put in evidence on this figure and, as it can be seen, consumes most of the network bandwidth with routing packets. The use of the SR strategy is, therefore, a more stable and recommendable technique.

In summary, these simulations have shown that SR mode 1 performs better when compared to modes 2 and 3. Notice that mode 1 is the most restrictive propagation method and, therefore, can find extra routes without much extra routing overhead. We believe that this is the main cause for the superior results achieved when compared to the other two modes.

6.2.6 Effects of route stability on real-time video streams

The performance of real-time video streams on wired packet networks (such as the Internet) depends on queue management, scheduling policies, etc. Video streams' performance in MANETs also depends on other factors, such as the medium access

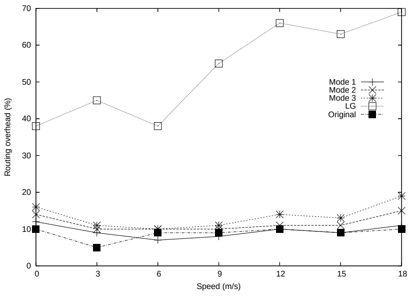


Figure 6.13: Relative Routing Overhead at different mobility levels

mechanism, the degree of node mobility, the routing protocol used, etc.

When video packets sent by one node compete for the medium with several TCP sources, the bandwidth is shared fairly among them due to TCP's adaptive nature. If the resulting bandwidth is not enough for the video stream, packets will experience high delays and queue drops. This problem can only be solved through congestion control techniques (e.g. SWAN [GAAL02]) or through QoS mechanisms at the MAC level (e.g. IEEE 802.11e [IEE05]), being fundamental in congested MANETs. Concerning mobility and the number of hops, these can only be handled by appropriate routing techniques.

In this section we are going to study how severely does node mobility affect a real-time video stream in terms of video communication disruptions (video gaps). We will prove that such video gaps are intimately related to route discovery events, and that the enhanced route discovery solution presented before, the SR mode, considerably mitigates the effects of mobility.

Experiments are made with the ns-2 simulator, and take place on a 1000×1000 *m* squared scenario with 80 nodes. Concerning traffic load, it consists of the trace of a single H.264 [H2603a] video stream obtained from the well known Foreman sequence; we encoded this video at 10 frames-per-second in the QCIF format. Each video frame is split into 7 RTP packets, which means that the packet generation rate is of 70 packets per second; the target bitrate is 186 kbit/s.

We chose to perform our study with only one video stream in order to clearly observe the impact of the different route discovery mechanisms under analysis, making it independent from other traffic flows and from congestion-related routing misbehavior.

Figure 6.14 shows that modes 1 to 3 always perform slightly better that the original DSR implementation in terms of packet arrivals. The best performing mode is SR mode 1, where the improvements over DSR reach 4,5%. Concerning routing overhead, SR mode 1 provokes only a moderate routing overhead increase compared to the standard DSR implementation, and generates fewer routing control packets than the remaining two test modes, again proving to be the best choice.

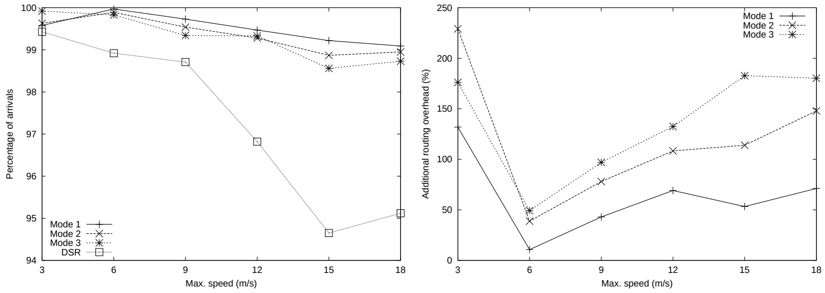


Figure 6.14: Percentage of arrivals for the H.264 video flow (left) and additional routing overhead generated (right)

6.2.6.1 Loss pattern analysis

Recent video compression standards, such as H.264, offer a wide range of tools to reduce the effects of degradation in the presence of losses. In fact, different types of intra macroblock updating strategies are offered, and an H.264 decoder is equipped with different error concealment tools which aim at estimating parts of frames which are not received. Observing the results of figure 6.14, and taking into account these facts concerning H.264, an observer could erroneously conclude that the difference in terms of PSNR (Peak Signal-to-Noise Ratio) between receiving 99% or 95% of the packets would be slight.

Packets dropped in bursts long enough to cause video gaps affect an H.264 video decoder in a drastically different manner. Occasional losses can easily be concealed by the video decoder; however, when no information relative to one or several consecutive frames arrives to the decoder, it will freeze the last decoded frame until communication is resumed. After resuming communication the decoder's effort is also increased since it must resynchronize and recover from losses as quickly as possible. We therefore argue that the PSNR is not a representative metric, and we adopt the Burstyness factor defined in section 5.2.1 instead. We repeat its definition below:

$$BF = \frac{\sqrt{\sum_{i=1}^{K} B_i^2}}{N}, \; 0 \leq BF \leq 1$$

What remains undefined, though, is the minimum number of consecutive lost packets to create a video gap. In this work we set that threshold to 7 packets (one entire frame), that is, $B_i \geq 7$.

Concerning BF limits, the lower limit, zero, indicates that there are no video frame gaps (though losses can occur) and 1 is the upper limit, meaning that the entire sequence was lost. Notice that the quadratic relation allows taking into account the fact that many distributed 1-frame gaps are almost imperceptible to a viewer, while a single 50 frames gap (5 seconds interruption at 10Hz) is quite disturbing from a user's point of view.

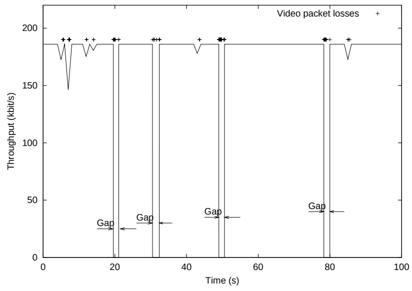


Figure 6.15: Video throughput variation

Table 6.5: Video annoyance statistics

Protocol	BF (10^{-3})	BF % towards DSR
DSR	6,17	-
Mode 1	0,709	11,5
Mode 2	0,841	13,63
Mode 3	0,876	14,20

We now analyze a typical packet drop pattern on a simulation using the standard DSR implementation and a single H.264 video flow. The results are presented on figure 6.15, and allow us to observe that some of the packet losses are bursty, a common occurrence in MANETs. This causes the video flow to be stopped since several entire frames are lost.

Table 6.5 presents a comparison between the different routing protocols in terms of the BF parameter. As it can be seen, the BF value associated with modes 1 to 3 is only a small fraction (less than 15%) of the BF achieved with the original DSR implementation. This result alone proves how the different SR modes improve the video experience in terms of video disruptions.

To further analyze and validate the improvements achieved in terms of the BF parameter, we study (see figure 6.16) the video gap histogram for all protocols at very high mobility levels (maximum node speed set to 18 m/s). We find that DSR performs much worse with respect to the three SR modes, being mode 1 the one that which achieves the best results overall.

Another important factor shown in figure 6.16 is that SR modes 1 to 3 present gap sizes of no more than 20 frames, contrarily to DSR. In fact, DSR is prone to lose as much as 217 consecutive frames (more than 20 seconds of interruption), while for mode 1 the maximum gap size experienced is of only 13 frames (1,3 seconds interruption).

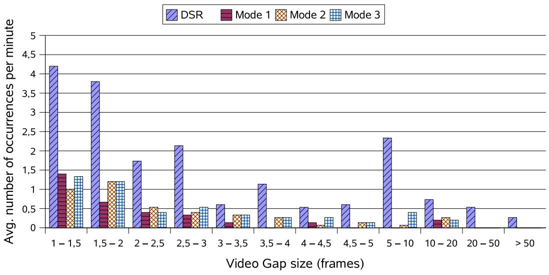


Figure 6.16: Video frame gaps histogram when setting the maximum node speed to 18 m/s

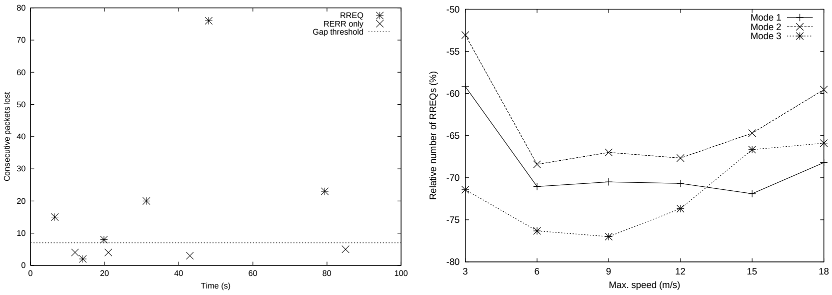


Figure 6.17: Relationship between RREQs, RERRs and video gaps (left) and relative number of RREQs generated (right)

6.2.6.2 Gap causes and solutions

In scenarios such as the one selected, where congestion is not a problem, packet losses are intimately associated with link breaks and subsequent route failures.

However, DSR uses link layer information to detect broken links, and so the interval between the detection of a broken link and the reception of the respective notification by the source is, in terms of video streaming, not excessively long. We therefore conclude that the cause of such long video gaps is related to route discovery procedures, and not just route failures. Figure 6.17 (left) proves that this statement is true. As it can be seen, only RREQ procedures are related to situations where the number of consecutive packets dropped is above the gap threshold, while route failure (RERR) events are usually not associated with video gaps at all. We also verified that, in this scenario, all video gaps are associated with a route request (RREQ) event since there is no competing traffic. Relative to the Gap threshold, it is equal to 7 packets as referred in the previous section.

Proven this direct relationship between video gaps and route request events, we studied the behavior of the different SR modes relative to DSR in terms of route request events generated so as to obtain a clear explanation of the results of table

6.5 and figure 6.16. We calculated the number of route requests generated by the video source at different speeds, achieving the results presented on the right side of figure 6.17. Results are relative to those found for the standard DSR protocol implementation.

These results show that modes 1 to 3 present less video gaps for all speeds; this is expected due to the higher number of routes found by these enhanced route discovery algorithms. The relationship between the number of RREQ events and video gaps also explains the improvements in terms of BF. The mode with best overall performance (mode 1) shows an average reduction of 68% on the number of RREQs generated in relation to DSR, and so we will use it as a basis for further improvements.

6.3 Multipath routing

In the previous section we have shown that video streams flowing in MANETs are prone to be degraded by mobility through packet loss bursts which cause gaps on communication. We presented some techniques to measure those gaps, such as the BF parameter and the video gaps histogram, and proved that by using any of the SR modes proposed we can reduce the number and size of these communication gaps. We also showed that video gaps are intimately related to route discovery processes.

In this section we will present further improvements by analyzing how the use of simultaneous paths on data transmission effectively reduces the down-time of multimedia flows, making the communication experience smoother and more pleasant to the user. Though the use of multipath routing can be extended to any type of data, in this proposal we concentrate on multimedia streams alone.

6.3.1 Traffic splitting strategies

Traffic splitting in the context of multipath routing refers to the technique of distributing the packets of a certain stream through different paths. An optimal strategy in terms of traffic splitting would be one where the shortest disjoint paths are used. In general, node disjoint paths are preferable since these achieve the best usage in terms of both bandwidth and node resources. There are some cases where no node disjoint paths are available and, therefore, link disjoint paths are recommendable. In fact, the link disjointness condition is enough to reduce the effect of mobility in ad-hoc networks. The algorithm we propose for maximum disjointness is algorithm 1. Notice that this algorithm works independently of the route discovery strategy used, though it clearly benefits from extensions such as the SR modes proposed for DSR. From now on, we shall refer to *Disjoint* solution as the one which makes use of this algorithm, in conjunction with SR mode 1 for route discovery and the preventive route discovery technic proposed in the next section.

The action of finding the disjointness of one route is always done relatively to the previously used route. The technique employed by our algorithm allows to easily adapt to extra routes found through the forwarding or interception of

Algorithm 1 Fully adaptive route disjointness algorithm

```
if (no path has been chosen previously) {
  choose the first shortest path;
} else {
  find the shortest node disjoint path;
  if (not found) { find the shortest link disjoint path; }
  if (not found) { find the shortest path with least common links; }
  if (not found) { choose first shortest path; }
}
```

Table 6.6: Comparison of traffic splitting strategies

Mode	Video arrivals(%)	Routing overhead	End-to-end delay (ms)	Dispersion
Disjoint	99,70	6759	39,54	0,71
R_2	97,60	11346	51,19	0,32

routing packets, as well as adapting to lost routes. To provide a good basis of comparison we define a metric that clearly evidences the exact gains in terms of path dispersion, that is, an average degree of path disjointness. So, the dispersion between two consecutive paths for a single stream would be:

$Dispersion = 1 - \frac{CL}{NL}$,

where CL is the number of common links relative to the previous path and NL is the number of links of the current path.

We aim at dispersion values near 1, which is the optimal solution; dispersion values near 0 reflect a bad traffic dispersion policy.

Though this algorithm aims at finding the best choices in each situation, it could be considered computationally expensive for small embedded systems. Therefore, we also propose a much simpler solution which consists of randomly choosing routes which are not larger than a certain size (s) relative to the first one. This alternative solution, referred as R_s, also aims at providing a basis of comparison to assess the goodness of the Disjoint solution.

Using the same simulation setup used in section 6.2.6, we evaluated the Disjoint algorithm comparing it with the R_s solution. Table 6.6 shows the average results obtained when setting $s = 2$ and varying the maximum allowed node speed from 3 to 18 m/s. Values presented are average values over 10 simulation runs.

As it can be seen, the maximally disjoint solution always achieves the best results. In terms of end-to-end delay, the fact that the Disjoint solution performs better than the R_2 one means that the paths used are shorter. In what refers to the routing overhead, again the Disjoint solution performs much better since it avoids using many different paths.

If we observe the results concerning the dispersion achieved with both methods, we verify that the Disjoint solution presents dispersion values that more than double the ones from solution R_2. We also found that the dispersion value almost

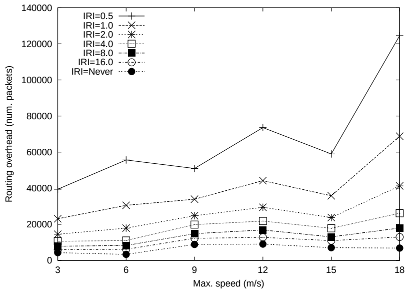


Figure 6.18: Effect of choosing different inter-request intervals on the routing overhead generated

does not vary with speed.

This analysis allows us to conclude that the results achieved fully justify the extra computational complexity required by the Disjoint solution, being the R_2 a possible solution for environments with few resources (though the performance will drop). From now on we will always use the Disjoint solution when performing traffic splitting.

6.3.2 Preventive route discovery

In order to complete the *Disjoint* solution, whose traffic splitting strategy was presented in the previous section, we now propose a mechanism to perform preventive route discovery processes. Its objective is to minimize the effect of video gaps on the video quality delivered to the user.

Normally, the DSR protocol initiates a route discovery process when it has no available routes to the destination. So, until a new valid route is discovered, our video flow transmission will lose a burst of packets, producing a video gap. We propose performing preventive route discovery processes in order to avoid that situation. So, we think that it is reasonable to look for new routes when there are no disjoint paths available.

Also, it is necessary to take into account the following situation: after completing a preventive route discovery cycle, no disjoint routes are found. In this case we have to start another route discovery process to avoid video flow stall if the current route is lost. So, by doing sporadic route discoveries we increase the probability of falling in video stall states, producing more video gaps. On the other hand, doing frequent route discoveries may result in an excessive routing overhead. Then, we need to find a trade-off.

With that purpose we made some experiments varying the preventive route discovery period. We tested route discovery intervals of 0.5, 1, 2, 4, 8 and 16 seconds, including the default situation (“never”) for comparison. Remind that, when all routes to the destination are lost, a new route discovery process is started; in that situation the probability of provoking a video gap is high.

In figure 6.18 we present the routing overhead differences among the different inter-request values. As expected, the routing overhead is higher than the default solution in all cases: the lower the inter-request interval, the higher the routing overhead becomes. We consider that for values under 4 seconds the routing overhead becomes prohibitive. In terms of packets arrivals, around 99% of the packets arrive for all solutions under test; that applies to all speeds.

Relative to the BF parameter, we achieve the best average results for an inter-request interval of 8 seconds, which also offers very good results in terms of packet arrivals and routing overhead. Therefore, this parameter is set to 8 seconds from now on.

6.3.3 H.264 enhancements for wireless scenarios

The techniques presented in the previous sections focus on enhancing the routing protocol to provide a better video service. In this section we present further enhancements to be applied in conjunction with the Disjoint routing strategy by focusing on an optimal adaptation of H.264 video flows to that solution. We start with a study on the impact of video-aware packet replication in the presence of packet losses, and we then proceed by analyzing the need for efficient data distribution through different paths.

6.3.3.1 Video-aware packet replication

Splitting traffic through two different routes towards a destination opens new possibilities for enhancing the video streaming process. A simple solution consists of replicating all the information through both paths, so that if we have a packet loss probability p on each path the receiver sees it as if losses were occurring with a probability of p^2. The only problem with this technique is that such improvement doubles the data rate generated.

If we analyze the organization of video data, we observe that not all the information is equally important. Intra coded frames, which are an essential source of reference for predicted frames, contain information which is more relevant than the rest. By replicating I frames alone we are able to improve the video distortion performance, while at the same time avoiding an increase of 100% in the video data generated.

For our study we use an H.264-encoded video of the Foreman sequence where 10% of the frames are intra-coded (I). The metric we use to assess video quality is Peak Signal-to-Noise Ratio (PSNR), also known as video distortion. The video distortion for the Foreman test sequence (under no loss) is of 33,51 dB.

By simulating a statistically random loss process on both paths we analyzed the video distortion decay by increasing the packet loss ratio from 1 to 20%. The results presented in figure 6.19 are average values over 10 simulation runs.

The tag “None” indicates that no packet replication is being performed, which means that packets are just evenly split among the two paths. The “All” tag indicates a 100% packet replication. The remaining tags refer to the replication of 10% of the packets. In the experiment referred to as “Random” the replication process selects 10% of the packets in a random manner, while in the experiment

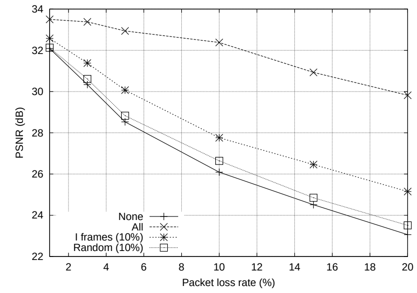


Figure 6.19: Distortion values for different packet replication strategies.

tagged “I frames”, the replication mechanism replicates only those packets related to I frames.

From that figure we find that, by replicating 10% of the packets randomly, the distortion increases by 0.5 dB relative to no-replication, so it almost does not justify the 10% increase in terms of bandwidth. We do achieve considerable gains by performing an intelligent packet replication, though still maintaining the same bandwidth increase. In fact, the introduction of video awareness in the packet replication process allows to achieve up to 2,14 additional dBs when packet losses reach 20%.

6.3.3.2 Optimal video data distribution in multipath scenarios

We now focus on some aspects of video distribution in multipath scenarios. As exposed previously, packet splitting attenuates the effects of mobility by assuring that, even when one path is lost, the other one remains available for data transmission. As soon as the loss is detected, the source stops sending packets through that invalid route. However, from the moment a broken link is detected to the moment the source receives the correspondent notification, a considerable amount of video packets could have been transmitted. Contrarily to random error scenarios, now the losses will affect well defined data - packets assigned to the invalid path. For example, if we split each frame in two packets and we route data through two disjoint paths, the loss of one path will always result in the loss of the upper or lower part of the frame, depending on the affected path. This situation will go on until the failure is detected.

To further examine this problem we use the same video sequence as before (Foreman), and simulate the loss of one of the routes. Our evaluation envisaged scenarios where the source splits packets through 2 and 3 disjoint paths, for comparison.

The results presented in figure 6.20 refer to the average video distortion perceived in the first 10 affected frames. Also, when using 2 disjoint paths, we average the distortion values resulting from losing one path or another; when using 3 paths we average the three distortion values obtained. Notice that, in terms of video distortion, we present experimental values for both intra-coded sequences (I) and predictive-coded sequences (P).

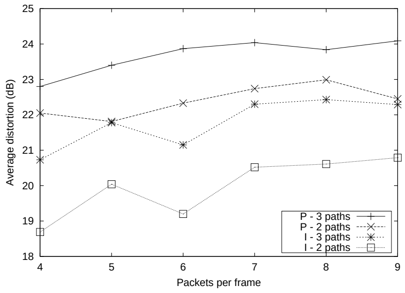


Figure 6.20: Average distortion value for varying packetization granularity in the first 10 affected frames

Figure 6.20 shows that the number of packets per frame is a parameter which affects the sequence's distortion, causing differences of up to 2 dB. Here we have two distinct factors acting simultaneously: (1) a higher granularity in packetization will cause information loss to be spread in a more uniform manner throughout frames. This effect helps in increasing the picture quality, but has the drawback that it does also increase the network's traffic. And (2), there is a relation between the number of packets per frame and the number of routes used. So, when the number of packets per frame is a multiple of the number of paths, the lost information will always refer to the same frame areas.

As a consequence of these interactions, we can see that using 7/8 packets per frame is the best solution in terms of video signal distortion for both predicted and intra-coded video sequences.

Another aspect shown in figure 6.20, and that should also be taken into consideration, is that I frames are more severely affected by losses than P frames. However, once the route is established, an intra-coded sequence will recover to maximum quality instantly, while sequences that rely on prediction alone will recover slower. Therefore, a good balance between intra coded (I) and predicted (P) frames on a sequence should be found in order to simultaneously optimize the use of bandwidth and the propagation of errors. We tuned the sequence for a GOP size of 10 frames, that is, frame groups formed by an I frame followed by 9 P frames.

6.4 Joint evaluation of proposals

In this section we are going to perform a global evaluation of the DSR, SR and Disjoint solutions, adjusting them with the options found to be the best in preceding sections. The simulation setup consists of a 1000×1000 meters scenario with 80 nodes. The mobility pattern is generated using the random waypoint mobility model bundled in ns-2, and an extra script was created in order to accept only those scenarios where node topology forms a connected graph.

A single video flow, with the same characteristics described in section 6.2.6, conforms the injected traffic. The Disjoint solution uses the multipath routing

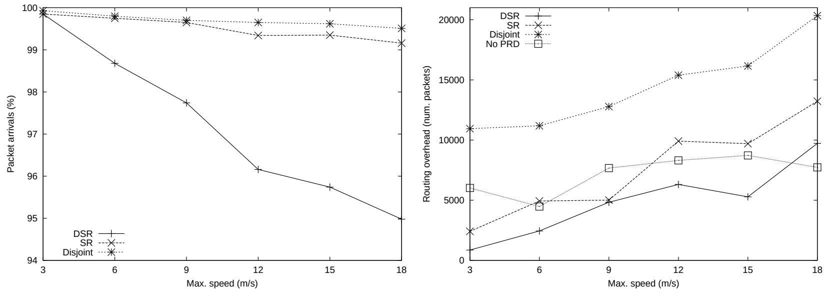


Figure 6.21: Comparison in terms of video packet arrivals (left) and routing overhead (right) of the different routing algorithms for different levels of mobility

algorithm 1, SR mode 1 for route discovery and also preventive route discovery, being the latter mechanism constrained to a minimum period of 8 seconds between consecutive route requests.

We can clearly see on the left side of figure 6.21 that, in terms of packet arrival rate, DSR performs worse than the remaining proposals for moderate/high speeds. The Disjoint solution proves to be the best for all speeds, though using SR alone already achieves very good performance.

In terms of routing overhead, we can see on the right side of figure 6.21 that SR mode 1 generates slightly more control packets than DSR. The Disjoint mode causes a much greater increase in terms of routing overhead since this protocol is performing preventive route discoveries. If preventive routing discovery mechanism is turned off (notice the *No PRD* tag) the routing overhead values are similar to those for SR.

Notice also that the rate of growth between the three solutions is quite different. Comparing the routing overhead at minimum and maximum speeds we can see that, while DSR control packets have increased by a factor of 9, in the SR mode there has been a 5 times increase, and just a twofold increase for the Disjoint mode. This shows the better adaptation of the later ones to very high mobility scenarios.

If we now look at what we wished to reduce - video streaming gaps - we can see that there has been a gradual improvement as shown in figure 6.22. As it can be seen, both the SR mode and the multipath Disjoint solution are able to significantly reduce the video gap occurrence, being the latter the most effective as expected.

Table 6.7 shows the improvements in terms of global gap percentage and the BF parameter. The reason why the difference in terms of BF between SR and Disjoint modes is not greater is due to the fact that both approaches avoid large video gaps, being large video gaps those that provoke significant differences in terms of the BF metric. For video gaps of less than 3 frames, though, the disjoint mode shows its effectiveness without doubt.

Relatively to the benefit of including or not the preventive route discovery mechanism, we achieve a reduction of 37% in terms of BF and of 50% in terms of

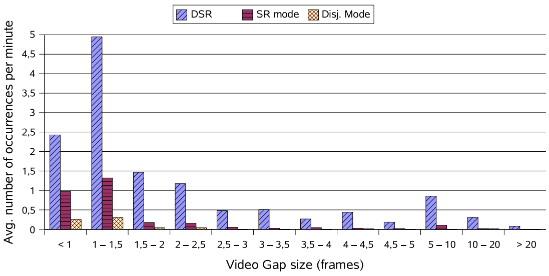


Figure 6.22: Video gaps distributions for the 3 protocols under test at high mobility (18 m/s)

Table 6.7: Comparison between the different routing protocol versions in terms of gap percentage and the BF parameter

Protocol version	DSR	SR	Disjoint
Gap (%)	2,41	0,303	0,0776
Avg. BF (10^{-3})	1,80	0,46	0,25

gap percentage by turning it on. The main reason for this difference are the few situations where preventive routing saves us of video gaps.

To conclude our analysis, we now apply the I-frame replication method developed specifically for video streams in multipath routing environments to determine global improvements. The scenario is the same one used in section 6.2.6, and we use both the *Disjoint* multipath routing solution and the standard DSR implementation for comparison.

We add background traffic, so 10% of the nodes are either sources or destinations of traffic. Each background source sends CBR/UDP traffic using 512-byte packets, at a rate of four packets per second.

Table 6.8 shows the results achieved with different protocol/replication combinations. In terms of packet arrivals for video data we see that the difference is

Table 6.8: Video arrivals and routing overhead for different protocol/replication combinations

Mode	Video arrivals (%)	Routing overhead
DSR	97,2	7011
DSR+Rep.	95,9	6438
Disj.	97,4	18380
Disj.+Rep.	97,6	18953

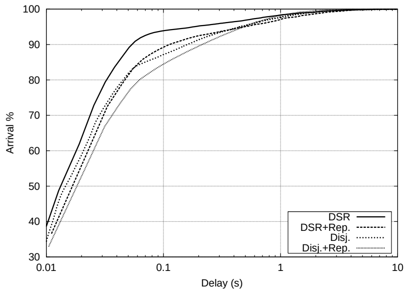


Figure 6.23: Distribution function for video packet delays

minimum, except for DSR plus replication where the arrival percentage is slightly reduced. Concerning the use of the *Disjoint* mode with or without I frames replication, it causes a 2% increase in the background traffic, along with a considerable increase of the routing overhead, as expected. Notice that this additional routing overhead can be reduced by about 50% if the preventive route discovery mechanism is turned off.

In terms of end-to-end delay we found that, except for very high packet arrival values, DSR alone performs better than the remaining modes, as shown in figure 6.23. This result is expected since both packet splitting and packet replication mechanisms are prone to cause additional delays to a video stream. It is also interesting to notice that, when using multipath traffic splitting, 96% of the packets arrive faster without replication, but the remaining 4% arrive faster with replication on.

In order to assess the effect of the different strategies under study on the video stream we evaluate the results, in terms of average distortion, for varying end-to-end delay thresholds. Such results are shown in figure 6.24. We should point out that the PSNR differences found are due to packet drops that are related to routing and delay thresholding, but are not related to congestion.

Contrarily to what would be expected by observing figure 6.23, DSR alone never performs better than its multipath counterpart. Concerning *Disjoint* mode plus replication, we also notice that it behaves better than DSR for end-to-end delay thresholds above 500 ms.

In this scenario it is not possible to observe the goodness of I frames replication, as obtained theoretically in section 6.3.3.1, since the main cause for packet losses are path breakages, which are not frequent, randomly distributed packet losses. It is clear, though, how the 10% increase in terms of video traffic affects the overall performance, which leads us to think that this option shall be optimal only in situations where packet loss is more severe and more randomly distributed, and also when the available radio technology offers higher bandwidths.

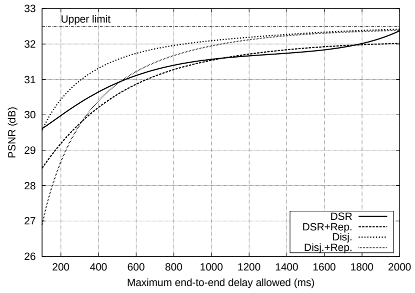


Figure 6.24: PSNR variation for varying end-to-end delay thresholds

6.5 Conclusions

In this chapter we presented several enhancements to the DSR routing protocol in order to provide a better support to real-time multimedia streaming, taking an H.264 video stream as reference. We started by presenting the SR algorithm, an extension to DSR's route discovery method based on a previous proposal by Lee and Gerla [LG01].

In order to assess the effectiveness of the SR algorithm we simulated a static scenario, and obtained values for several quality parameters. We compared it with the LG solution, noticing that the SR algorithm achieves excellent improvements in terms of routing overhead, lost packets and route discovery time.

We also proposed two enhancements to the SR algorithm which aim at both increasing and decreasing the route discovery overhead, varying the number of routes found accordingly. Using low values for the PNC parameter, in conjunction with path disjointness enforcement, successfully restricts the propagation overhead; on the other hand, the Flexibility parameter can be tuned to allow finding more routes. We make an evaluation of the behavior of these parameters through simulation and analysis of the results.

We proceeded with an evaluation in typical scenarios using different speeds and with typical traffic loads. By observing the improvements relative to the LG solution and the default DSR implementation we conclude that the LG solution, though allowing nodes to find more routes, generates too much overhead. We therefore consider it not appropriate for networks under average/high load. The SR solution, on the other hand, allows nodes to find extra routes without introducing too much routing overhead. Results showed that the routing overhead generated is very similar to that of DSR's.

Concerning data packets, the extra routes provided by the SR mode enhance route stability for average speeds (6 to 15 m/s), which was clearly evidenced by the improvements on TCP goodput, and was also reflected on the total number of data packets received.

We proceeded by evidencing the existence of video transmission gaps in wireless ad-hoc networks. We studied their causes, as well as the relation between video gaps and route stability. Concerning video streams, we also proposed using an alternative metric to PSNR - BF -, developed in chapter 5, in order to measure video gaps in a clear and straightforward manner.

We presented additional solutions to the video gap problem through packet splitting procedures and performing preventive route discoveries. The first of these mechanisms aims at reducing small but frequent video interruptions, while the second aims at eliminating very large video interruptions caused by the loss of all routes to a destination. We achieved reductions in terms of video gaps of up to 97% by applying both techniques simultaneously.

After the multipath framework was set, we analyzed the interactions between the video codec and the packet splitting mechanism, tuning the former so as to obtain the best performance.

We concluded our work with an overall evaluation of the improvements achieved, showing that there are clear improvements in terms of video distortion values for different delay thresholds, where the mobility-related gains can surpass 1.2 dB.

Chapter 7

MAC layer support for QoS in MANETS

One of the most important requirements for MANETs supporting real-time flows is having a QoS-enabled MAC layer. Without traffic differentiation on channel access it becomes very difficult to avoid that bandwidth-greedy protocols (e.g. TCP) occupy most radio resources. Also, the action of routing protocols is much more important in MANETs than in wired networks due to the presence of mobility; therefore, it is important to assure that routing-related tasks are not affected by the rest of the traffic.

In this chapter we study the viability of using the IEEE 802.11e [IEE05] technology to achieve MAC layer QoS in multi-hop wireless network environments. We begin our study by evaluating how well does the prioritized channel access mechanism proposed for IEEE 802.11e perform in static MANET scenarios. We then proceed by assessing the impact of mobility on QoS traffic.

We conclude this chapter with an analysis of the benefits that the IEEE 802.11e technology offers to reactive routing protocols, such as DSR and AODV, proving that it is enough to mitigate routing misbehavior under high loads of best effort traffic.

7.1 Evaluation of the IEEE 802.11e technology in MANETs

In order to assess the performance and effectiveness of IEEE 802.11e in multi-hop environments we perform several simulation tests focusing on throughput and end-to-end delay values for different degrees of network saturation, different number of hops from source to destination, and different number of competing sources.

We created both a static reference scenario and a mobile reference scenario. The static reference scenario, shown in figure 7.1, is a controlled environment used in order to make strict performance measurements. Therefore, all changes in static scenario situations will always be made relative to this reference scenario.

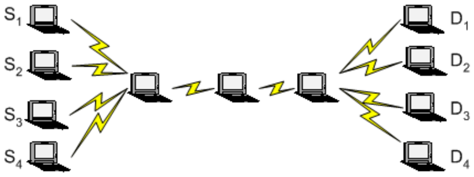


Figure 7.1: Static multi-hop scenario

Table 7.1: ns-2 PHY settings for IEEE 802.11a/g

Parameter	Value
SlotTime	9 μs
CCATime	3 μs
RxTxTurnaroundTime	2 μs
SIFSTime	16 μs
PreambleLength	96 bits ≅16 μs
PLCPHeaderLength	40 bits
PLCPDataRate	6 Mbit/s
DataRate	54 Mbit/s

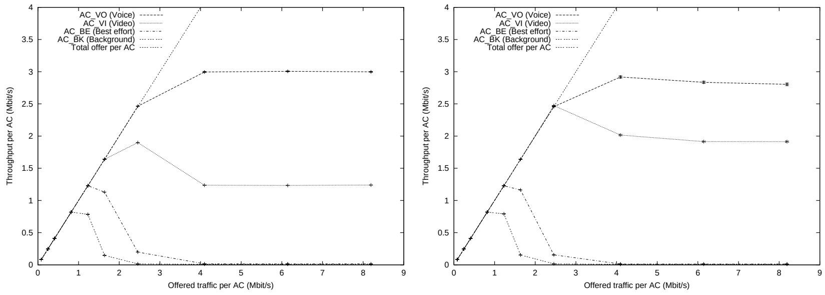
As shown in figure 7.1, we have four source/destination pairs (S_i, D_i) which are 4 hops away from each other (3 intermediate nodes). All traffic sources are set to generate the same data rate in all four ACs, and the traffic type chosen consists of CBR/UDP sources with packet size fixed at 512 bytes.

Relatively to the mobile reference scenario, it represents a typical mobile MANET environment. It consists of a rectangular area sized 1900×400 meters, where the average number of hops from source to destination is four - the same as for the static reference scenario presented above. The number of stations participating in the MANET is 50, and all of them are moving at a constant speed of 5 m/s according to the random waypoint mobility model. The routing protocol used is AODV, and routing traffic is assigned the highest priority (AC_VO).

To conduct our experiments we used the ns-2 simulator [KK00] with the IEEE 802.11e extentions from Wietholter and Hoene [SC03]. We setup the IEEE 802.11 radio according to the parameters exposed in table 7.1. These values are valid for both IEEE 802.11a and IEEE 802.11g since the radio model of the simulator does not differentiate between both.

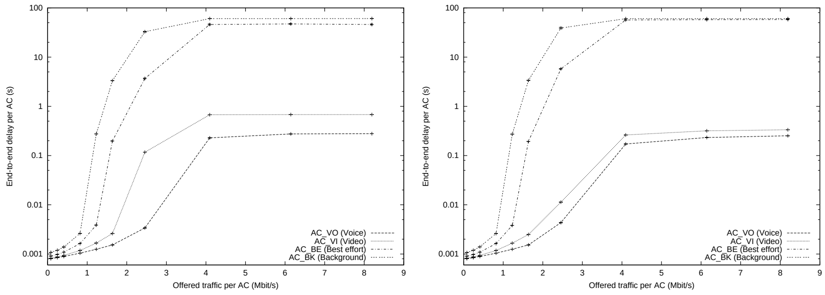
Concerning the IEEE 802.11e MAC, it was configured according to the values presented earlier in table 1.3.

The simulation results using the static scenario show a 99% confidence interval in all depicted points. Concerning the results of section 7.1.2, which use the mobile scenario instead, all points depicted are an average of 5 simulation runs using different mobility traces; this is because, contrarily to a static scenario,



a) b)

Figure 7.2: Throughput achieved with a) no CFB and b) CFB activated



a) b)

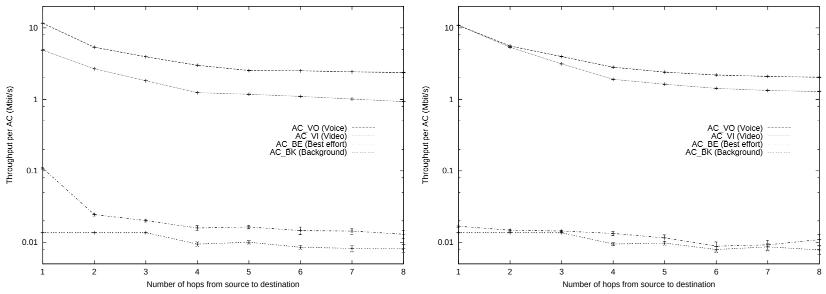
Figure 7.3: End-to-end delay achieved with a) no CFB and b) CFB activated

the completion time is much higher and strict confidence intervals would require a very large number of simulation runs since mobility has a strong impact on results. Simulation times are of 300 seconds in each scenario.

7.1.1 Determining the saturation limits

Using the static reference scenario, we start our analysis by observing the behavior in terms of throughput and end-to-end delay when choosing different traffic generation rates for the sources. Remember that the same packet generation rate is used for all ACs and for all stations acting as traffic sources.

Figure 7.2 shows the results achieved in terms of throughput with and without the *contention-free bursting* (CFB) mechanism. We can clearly observe how the points of departure from the offered traffic load line change when this mechanism is activated. This is especially relevant for the Video Access Category (AC_VI), where the saturation bandwidth reaches much higher values. We also find that the cost of this improvement in terms of Voice traffic is not high, being mainly due to a better utilization of the wireless channel capacity thanks to the large TXOPLimit assigned to both AC_VO and AC_VI.



a) b)

Figure 7.4: Throughput achieved with a) no CFB and b) with CFB activated

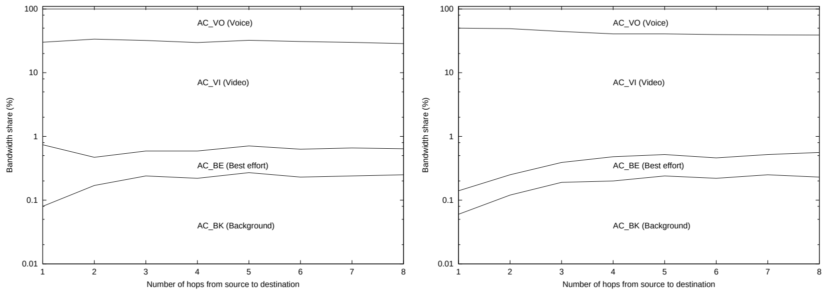
In terms of end-to-end delay, figure 7.3 shows that the probabilistic prioritization mechanism proposed by the IEEE 802.11e technology is quite effective in providing traffic differentiation. In fact, the end-to-end delay difference (for low values of offered traffic) between the highest priority AC (AC_VO) and the lowest one (AC_BK) is almost of one order of magnitude. We also observe that, under strong saturation (more than 4 Mbit/s per AC), best-effort and background traffic suffer from starvation, which can be verified analyzing both throughput and end-to-end delay graphs.

We now proceed with our analysis performing a different evaluation. The purpose is to observing the decay in terms of throughput as the number of hops is increased. Our purpose is also to assess if the share of bandwidth assigned to each AC in a single hop situation remains the same as the number of hops increases. If traffic with lower priority obtains significatively higher bandwidth shares with increasing number of hops, we could conclude that the effectiveness of IEEE 802.11e is reduced.

In this experiment we vary the number of hops from source to destination by varying the number of intermediate nodes. The total offered traffic per AC is set to 12 Mbit/s (3 Mbit/s per source), so that we operate under network saturation even when sources and destinations are within radio range. In such situation the delay values are very high, as expected, offering no significant data.

In figure 7.4 we present the saturation throughput when varying the number of hops with and without the CFB mechanism. As expected, the throughput for all traffic categories decreases as the number of hops increases. In terms of the total aggregate throughput, we observe that it drops from 16,60 Mbit/s (1 hop) to 3,32 Mbit/s (8 hops) with no CFB, and from 21,74 Mbit/s to 3,34 Mbit/s with CFB active. This result shows that, as the number of hops increases, the improvements in terms of channel utilization offered by the CFB mechanism decrease. However, it still influences the channel shares obtained by each Access Category.

In terms of bandwidth share, figure 7.5 shows the allocation of bandwidth to the different ACs with and without CFB. We observe that both Voice (AC_VO) and Video (AC_VI) traffic maintain a steady share of the available bandwidth when CFB is off, as desired. When the CFB mechanism is activated we observe



a) b)

Figure 7.5: Bandwidth share for varying number of hops with a) no CFB and b) CFB activated

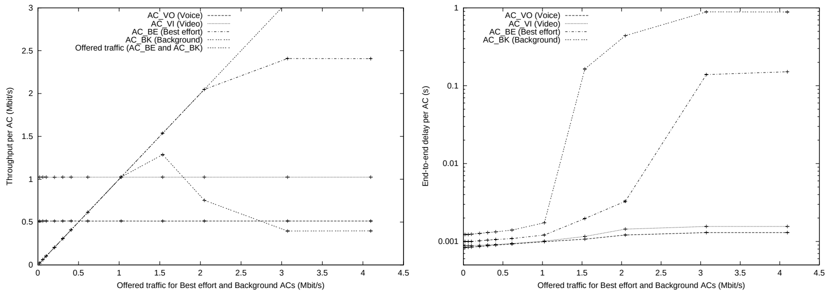
that the Voice traffic share increases as the Video traffic share decreases. This variation, up to ten percent for eight hops, is due to the loss of effectiveness of the CFB mechanism with increasing number of hops, as referred before.

The Best effort (AC_BE) traffic share slightly decreases with no CFB; on the contrary, it increases if CFB is turned on. Background (AC_BK) traffic increases its share as the number of hops increases in both cases, though it is always maintained very low.

To complete the analysis performed in this section we now examine the stability of Voice and Video traffic when only Best effort and Background traffics vary. Our purpose is to assess the impact of non-real-time data packets (e.g. peer-to-peer file sharing) on real-time traffic, such as video or voice streaming. With this purpose we set source S_1 to transmit nothing but Voice traffic at a rate of 0.5 Mbit/s; likewise, source S_2 is set to transmit solely video traffic, at a rate of 1 Mbit/s. Sources S_3 and S_4 transmit variable rates of Best effort and Background traffic respectively. With this setup we find no difference between using the CFB mechanism or not, since it is never activated (only becomes active when traffic is generated in a bursty manner).

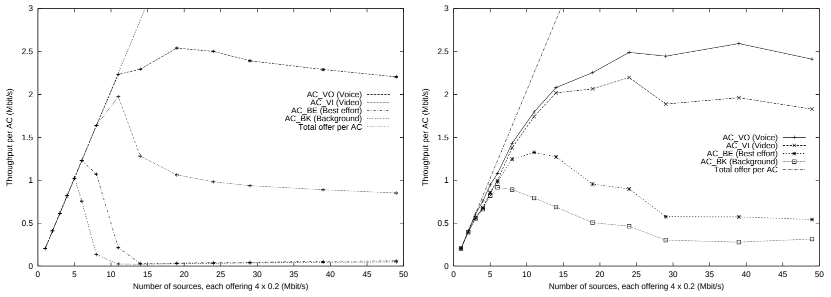
Figure 7.6 shows the results obtained with this new configuration. In terms of throughput we observe that neither Video nor Voice traffic are affected by increasing Best effort and Background traffic loads. When the channel becomes saturated, Best effort traffic obtains a bandwidth share about six times greater than Background traffic. In terms of end-to-end delay, we find that Voice traffic suffers from delay variations up to 70%, while for Video traffic the delay variations can reach 91%. Nonetheless, the actual delay values can be considered low enough to support real-time applications adequately.

Overall, results show that the prioritization mechanism of IEEE 802.11e retains most of its effectiveness independently of the number of hops traversed by traffic, or the load of Best Effort and Background traffic. As with legacy IEEE 802.11 networks, though, the impact of the number of hops on available bandwidth is considerable.



a) b)

Figure 7.6: Throughput variation (a) and end-to-end delay variation (b) with different loads of Best effort and Background traffic



a) b)

Figure 7.7: Throughput achieved in a) the static scenario and b) on the mobile scenarios

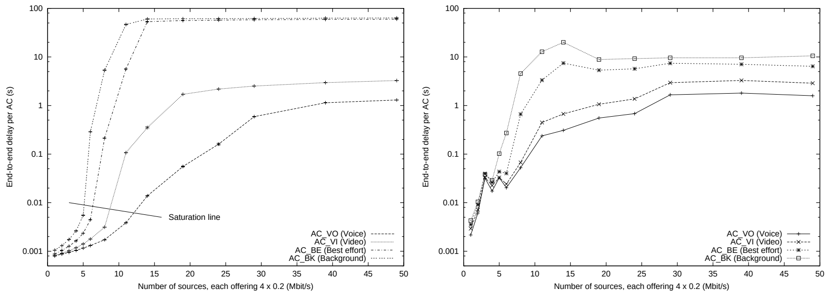
7.1.2 Impact of mobility on QoS performance

In this section we evaluate the performance of IEEE 802.11e in both static and mobile environments. The purpose is to compare the differences between both environments so that the impact of mobility on performance is put in evidence.

Concerning the simulation setup, we vary the number of traffic sources and each traffic source generates 0.2 Mbit/s (50 packets per second) per AC. The CFB mechanism is turned off.

Figure 7.7 shows the throughput behavior in both static and mobile scenarios. The results for the static scenario show that throughput values follow the offered traffic load line closely before saturation. After saturation is reached, the throughput increase rate is no longer maintained, and it starts dropping after a certain point due to the contention mechanism inherent to IEEE 802.11.

Relative to the scenario with mobility, we observe that throughput values no longer follow the offered traffic load so strictly, though the points of maximum productivity for the different ACs are reached for a higher number of source sta-



a) b)

Figure 7.8: End-to-end delay achieved in a) the static scenario and b) on the mobile scenarios

tions. This is due to the higher degree of path diversity achieved in the mobile scenario. So, while in the static scenario the maximum aggregated throughput is of 4,1 Mbit/s (6 sources), in the mobile scenario this value is increased to 6 Mbit/s (14 sources).

Concerning end-to-end delay, the results shown in figure 7.8 are quite representative in the sense that they allow observing two rates of growth: the one before saturation (below the line shown) and the one when saturation starts to cause collisions (above the line). We also see that Best effort and Background ACs almost can not transmit data with 8 traffic sources or more.

In the mobile scenario we observe that the minimum end-to-end delay values are higher compared to the static scenario. Moreover, the interval between the various ACs is not very high when there are only a few sources of traffic. This is due to mobility itself, which causes the routing protocol to react to route changes by buffering traffic on its queue. Similarly to what was found for throughput, now the end-to-end delay values do not reach saturation limits so quickly due to the expected traffic dispersion effect. In terms of traffic differentiation, we observe that in both scenarios the prioritization mechanism of IEEE 802.11e effectively offers better QoS to higher priority traffic, and so we consider that the effectiveness of this mechanism in multi-hop environments is preserved.

7.2 Interaction between IEEE 802.11e and reactive routing protocols

In the previous section we studied the performance experienced by QoS data sources in MANET environments which take profit from the IEEE 802.11e technology. We have showed that this technology is able to differentiate flows with different QoS requirements with great effectiveness, and so we consider that it creates the adequate conditions for deploying QoS services in MANETs.

From the results of the experiments made in chapter 4 we concluded that best-effort traffic can have a negative impact on routing protocols if there is no way to differentiate it from routing traffic. So, this section is dedicated to assess the benefits that IEEE 802.11e technology brings to MANETs in terms of routing responsiveness compared to the traditional IEEE 802.11 technology. We consider that these benefits become more relevant when the routing protocol used is reactive and relies on broadcast storms for route discovery; further benefits are obtained if the routing protocol uses link-layer feedback for the detection of broken links. Since currently the most well-known reactive routing protocols for MANETs are AODV and DSR, our experiments will focus on both of them.

To conduct our experiments we used the same mobile scenarios defined in section 7.1. We wish to evaluate the effects of injecting variable loads of both UDP and TCP best-effort traffic when using either legacy IEEE 802.11 or IEEE 802.11e. When referring to legacy IEEE 802.11 we are actually using an IEEE 802.11g physical layer; tests performed with IEEE 802.11e also make use of the IEEE 802.11g physical layer, being the MAC layer enhanced with IEEE 802.11e technology.

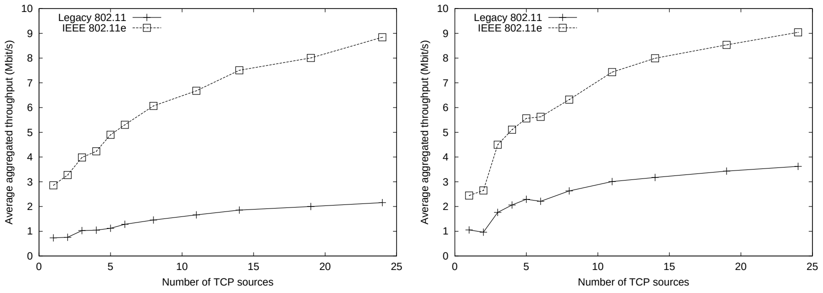
In all the simulations made we set an initialization period of 100 seconds. During that period all traffic sources send ICMP echo requests to destination at a very slow rate; the purpose is to find all the routes required, and also to allow routing protocols to converge. This technique aims at making measurements in a steady-state environment. After that period the actual traffic being measured starts, and our measurement period lasts for 300 seconds on each scenario being evaluated.

Relatively to the sources of traffic, TCP traffic sources are greedy for bandwidth throughout the entire measurement period. We simulate this behavior using FTP file transfers which go on forever. Since TCP uses an automatic bandwidth throttling mechanism, we vary the load injected into the MANET by vary the number of TCP connections. When simulating UDP traffic we take the opposite approach: we fix the number of sources (at four sources) and we vary the data generation rate. The purpose is to saturate the network gradually.

In all the experiments our purpose is to compare the performance achieved in terms of throughput and routing overhead when using either plain IEEE 802.11 or an IEEE 802.11e-enhanced MAC layer. For the comparison to be fair, both TCP and UDP packets are assigned to the best-effort access category (AC_BE) under IEEE 802.11e. Concerning routing packets, these are assigned to the highest MAC Access Category under IEEE 802.11e; such assignment is the most logical, and is also the one recommended in the IEEE 802.11e specification (see table 1.2). This way we are able to decouple routing traffic from best effort traffic, and so all the improvements experienced in terms of best effort traffic are solely due to an increased responsiveness of the routing mechanism itself.

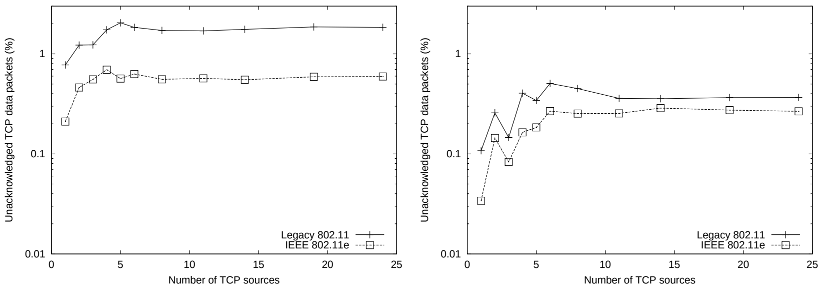
7.2.1 Improvements on TCP traffic

In this section we focus on the improvements in terms of TCP throughput by using IEEE 802.11e compared to legacy IEEE 802.11. We experiment with a variable number of FTP/TCP sources, and all tests are conducted with both DSR and AODV routing protocols.



a) b)

Figure 7.9: TCP throughput improvements with a) AODV and b) DSR by using the IEEE 802.11e technology



a) b)

Figure 7.10: Differences in terms of unacknowledged TCP data with a) AODV and b) DSR when using different IEEE 802.11 MAC layers

In figure 7.9 we show the TCP throughput performance results. When using the AODV routing protocol we encounter the most significant improvements. In fact, TCP throughput increases by around 300% for all points depicted. When using DSR the increment is also significant, being close to 150% for all points. In [GN99, TR01] authors show that TCP suffers from poor performance in mobile networks because it is not able to differentiate between congestion related packet losses and mobility related ones, treating all losses as congestion. To obtain an insight into the packet loss phenomena, and related it to the findings of those works, we evaluate (see figure 7.10) the number of unacknowledged TCP data packets using both MAC technologies. Notice that the lack of acknowledgments can be due both to the loss of TCP data packets on the direct path, or to the loss of TCP ACK packets on the reverse path.

The results show that there is a significative difference in the percentage of unacknowledged TCP data packets, especially when using the AODV routing protocol. In fact, when using AODV and the legacy 802.11 MAC, we find that there are up to 3 times more unacknowledged packets than with 802.11e. When using

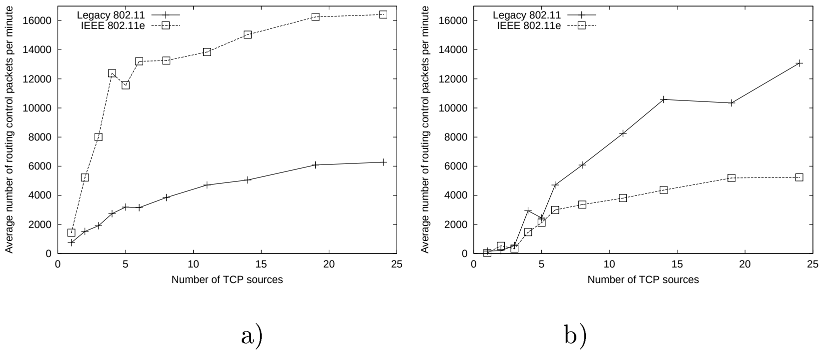


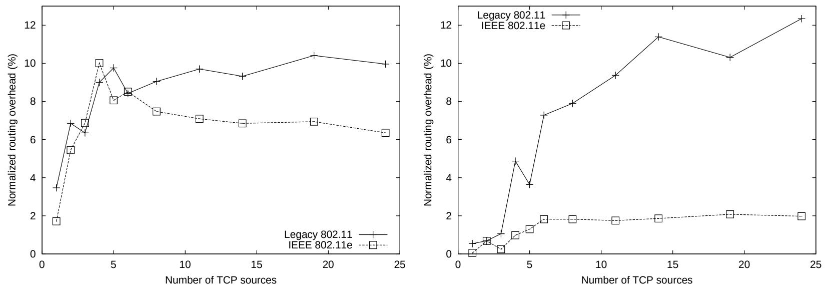
Figure 7.11: Number of routing control packets with a) AODV and b) DSR when using different IEEE 802.11 MAC layers

DSR the difference between both MAC technologies is lower, which is in accordance with the throughput results of figure 7.9. This difference is due mainly to a better performance of DSR when relying on legacy IEEE 802.11 for data transmission. DSR differs from AODV in its intensive use of caching and snooping of routes from packets in transit. Also, with DSR a significative share of routing packets are unicasted due to gratuitous route replies and route replies from cache, contrarily to AODV which relies much more on broadcasting. Since broadcast packets are not acknowledged under IEEE 802.11, congested scenarios will cause more of these packets to be lost due to collisions. This fact is also put in evidence by observing the results shown in figure 7.11 relative to the routing overhead.

Figure 7.11 shows that the number of routing control packets transmitted with AODV increases when using IEEE 802.11e. So, IEEE 802.11e has the effect of increasing the robustness of the AODV routing mechanism by making routing related communication more reliable (fewer routing packets dropped). When using the DSR routing protocol the results are the opposite of those found for AODV. Now, assigning routing packets more priority when accessing the medium makes routing related communication between stations much faster. The effect this produces is that fewer timeouts are triggered, and therefore fewer routing control packets are generated.

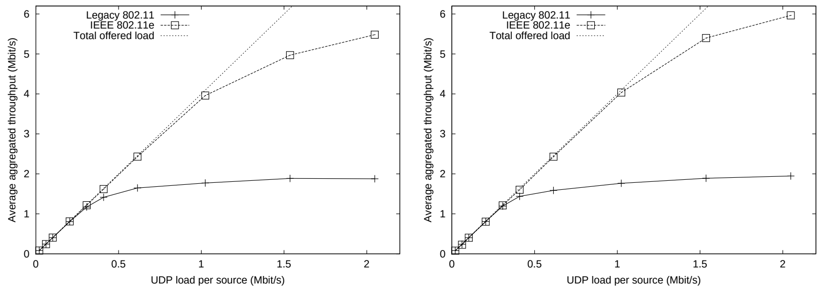
The goodness of routing protocols is usually evaluated in terms of normalized routing overhead, which is defined as routing packets required per data packet arriving to the destination. In our case, the data packets arriving to the destination are both TCP data and ACK packets.

The results relative to AODV (see figure 7.12) show that, in general, the increase in throughput compensates the increase in routing overhead. In fact, with more than 5 stations, results using IEEE 802.11e are significatively better. When using DSR this difference is even more noticeable, and the normalized routing overhead can be decreased by up to 6 times when IEEE 802.11e is used.



a) b)

Figure 7.12: Normalized routing overhead with a) AODV and b) DSR when using different IEEE 802.11 MAC layers



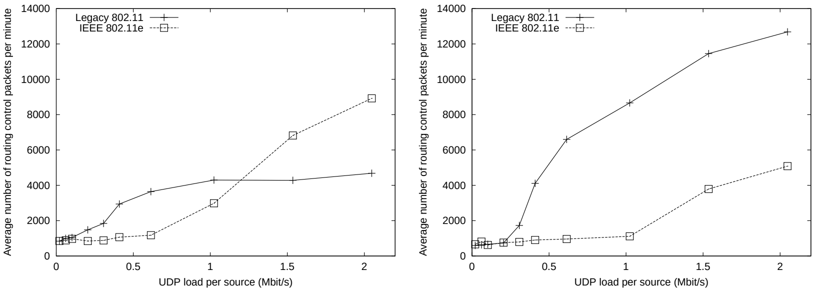
a) b)

Figure 7.13: UDP throughput with a) AODV and b) DSR when using different IEEE 802.11 MAC layers

7.2.2 Improvements on UDP traffic

In this section we analyze the improvements obtained with different UDP traffic loads. One of our purposes is to observe routing misbehavior as the congestion in the network increases.

We start by analyzing the improvements in terms of throughput with increasing source load. Results are shown in figure 7.13. We can see that, again, when using IEEE 802.11e the overall throughput increases, though all UDP traffic is assigned to the best-effort access category (AC_BE). We find that, for a source load up to 0.25 Mbit/s, the difference in terms of throughput between using IEEE 802.11e or not usually does not exceed 1%. For higher source loads the difference becomes quite noticeable. When using UDP traffic there is no loss-dependent behavior as with TCP traffic, and so the difference in terms of throughput can be directly related to the degree of responsiveness of routing mechanisms. In this situation we found that the differences between both routing protocols are not very relevant, though DSR always performs slightly better.



a) b)

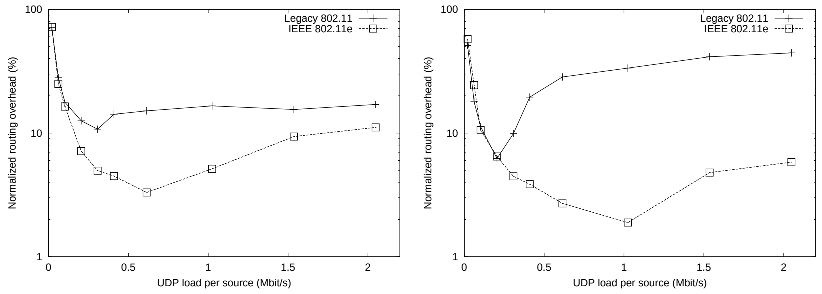
Figure 7.14: Number of routing control packets with a) AODV and b) DSR when using different IEEE 802.11 MAC layers

In figure 7.14 we show results relative to routing overhead when using the two different MAC layers. As the traffic load increases routing protocols need to increase their responsiveness, which means more control packets. If we look at the results for AODV with legacy IEEE 802.11 we find that, after a certain threshold, the number of control packets in the network is kept at constant levels; this means that it is no longer able to increase its responsiveness. When IEEE 802.11e is used the number of control packets increases steadily, “on demand”, as necessary. With DSR the phenomena experienced is different because, as referred in the previous section, IEEE 802.11e makes routing related communication between stations much faster, which provokes fewer timeouts to be triggered. Also, notice that routing packets are always put at the front of the interface queues, independently of using legacy IEEE 802.11 or IEEE 802.11e. This means that routing packets normally will not be lost due to queue overflows, but only due to channel noise or collisions. Since a significant share of DSR’s packets are unicasted (and so acknowledged), they will typically not be lost when flowing through the MANET, though experiencing different degrees of delay. This explains why in figure 7.14 DSR’s control packets depending on legacy IEEE 802.11 increase steadily.

In terms of normalized routing overhead we find, as we did in the previous section, that IEEE 802.11e improves performance (see figure 7.15). It is particularly relevant to notice the difference encountered with DSR as the source load becomes very high. In such cases the normalized routing overhead using legacy IEEE 802.11 achieves very high values, a clear indicator that congestion is making the routing protocol mechanism perform poorly.

7.3 Conclusions

In this chapter we evaluated the effectiveness of the IEEE 802.11e technology in multi-hop ad-hoc networks. We analyzed the performance achieved in a static scenario varying the levels of traffic load, along with the effect of traversing a different number of hops on the theoretical throughput limits. We showed that



a) b)

Figure 7.15: Normalized routing overhead with a) AODV and b) DSR when using different IEEE 802.11 MAC layers

the bandwidth share among different traffic categories depends on the number of hops traversed, and that the *contention-free bursting* mechanism loses most of its effectiveness as the number of hops increases. We also showed that, with IEEE 802.11e, high priority traffic (Voice and Video) is able to maintain a steady throughput independently of the lower priority traffic load (Best effort and Background), and that, in terms of end-to-end delay, the impact is kept within very acceptable bounds.

Our analysis also included a comparative performance evaluation of the static scenario configuration with a mobile scenario configuration for a variable number of traffic source stations. Results show that, compared to the static configuration, the overall capacity is increased due to the spreading of traffic throughout the test area; however, throughput values are always lower than the offered load due to mobility related losses. Overall, we found that the upcoming IEEE 802.11e technology retains most of its effectiveness in both static and mobile multi-hop scenarios.

We finished our study by offering an insight into the interaction of routing protocols and different IEEE 802.11 MAC layer implementations. Our study focused on both TCP and UDP performance improvements in a typical MANET environment when assigning routing packets to the highest priority access category under IEEE 802.11e. We detail the performance differences of two reactive routing protocols - AODV and DSR - relating the traffic throughput and routing overhead results to their internal mechanisms.

Results show that, when routing packets benefit from the prioritization mechanism of IEEE 802.11e, the performance is improved drastically. We find that this improvement is due to an increase in the responsiveness of both routing protocols. In terms of TCP throughput, we achieve an increase of up to 150% with DSR and up to 300% with AODV. Maximum UDP throughput is also increased substantially, up to 200% for both routing protocols. Relatively to normalized routing overhead, which is our reference metric to measure the effectiveness of routing protocols, we find that IEEE 802.11e allows achieving better results. The difference becomes more noticeable as we increase the level of saturation in the network, since saturation causes the routing protocol's mechanisms to malfunction.

Overall, we consider that upgrading the MAC layer of MANET stations to IEEE 802.11e is very important not also for multimedia traffic support, but also to improve the efficency of the routing mechanism used, especially if it is a reactive one.

After the analysis made on this chapter we confirm that the IEEE 802.11e technology is able to offer traffic differentiation, being adequate to deploy QoS-constrained applications in MANET environments. So, the topic of the next chapter is a proposal for a novel admission control system to be deployed in IEEE 802.11e-enabled MANETs.

Chapter 8

Distributed Admission Control for MANET Environments

Signaling mechanisms were first introduced in the field of telephony as the most primitive form of admission control and resource reservation. In the beginning manual signaling systems were used, providing a simple calling signal to the next operator in the call path. Later these mechanisms became automatic (no intervention from a human operator) and, with the introduction of the first PCM systems, they further evolved to digital line signaling systems.

In the Internet we can also find examples of the use of signaling at different levels. The establishment of a TCP session is an example of a signaling procedure that takes place at the transport layer. HTTP request/reply exchanges is an example of a signaling procedure taking place at the application layer. Below the network layer we can also find a variety of signaling mechanisms depending on the underlying network technology used. In chapter 2.4 we presented some QoS models for MANETs. These QoS models offer solutions which approach the traditional concept of signaling in what refers to admission control and reservation of resources.

In this chapter we propose a novel QoS framework for MANETs which represents a shift from the traditional signaling paradigm. Our solution, which we named DACME (Distributed Admission Control for Manet Environments), does not rely on the successive network elements in the path from source to destination nodes to perform any sort of resource acquisition, relying instead on application-level end-to-end resource measurements.

DACME is designed to operate on MANETs which benefit from the MAC-level QoS support offered by the IEEE 802.11e technology. The purpose of DACME is offering a distributed admission control mechanism whose implementation in real-life MANETs is feasible, not too complicated, and does not impose constraints or requirements on intermediate stations participating on traffic forwarding tasks.

This chapter is organized as follows: in the next section we discuss the main problems to achieve QoS support in MANETs, motivating our work by evidencing what was and wasn't achieved in previous works; in section 8.2 we expose the core of our proposal. We then proceed to tune DACME parameters for MANET environments in section 8.3, followed by some conformance tests in section 8.4. Afterwards, we analyze the benefits of DACME under different conditions, and we assess the performance when supporting bandwidth-constrained, delay-constrained and jitter-constrained applications. Finally, we present our conclusions for this chapter.

8.1 Motivation

MANETs do not follow the Internet's Client / Service provider paradigm IntServ and DiffServ were created for. The development of a QoS framework for MANETs requires a much more flexible philosophy of cooperation and resource sharing among users. In fact, the concept of MANET itself implies the existence of users, except in situations where special nodes have been deployed at a particular location to form some sort of infrastructure, offering more reliability to a particular MANET. The "ad hoc" concept, though, is somehow misapplied in that situation.

In MANETs we can devise two main policies for resource management and user control. The first one follows the *centralized control* paradigm, where one person or entity has practically all the control of the devices themselves, both in terms of their components (including hardware and software), as well as control on how users will operate the devices. An example of this can be a military unit or a firemen rescue group, where both devices and users follow strict policies and rules. The *centralized control* paradigm allows optimal operation and much flexibility at all levels in order to achieve the best QoS support possible. However, it can only be deployed in very limited situations.

The second one drops the *centralized control* paradigm, embracing a much softer one of *cooperation among equals*. This cooperation may be based on willingness to achieve mutual benefit, or enforced through punishment of selfish nodes. Anyway, there is always a notion of strong dependency among users and complex interactions in the network, which can be remarkably well described using game theory when nodes behave selfishly. Urpi et al. [AMS03] develop a formal explanation of the characteristics of ad-hoc networks, using the Nash equilibrium to analyze strategies that stimulate cooperation among nodes. We consider that, in such MANET environments of selfish users, reserving resources of others can be quite difficult to accomplish.

In real-world scenarios, only rarely will we have full control of all the stations in a MANET as described in the first paradigm. Also, we consider that only some expert users will manipulate their terminals in order to obtain benefits; most users will simply behave in a more naive manner, and so we can expect that stations will, at most, use standard routing protocols on top of standard MAC/PHY radio interfaces. However, even when we have full control of all devices, making strict QoS reservations is still a very complex issue. Wang et al. [ZJ96] proved that, in a wired environment such as the Internet, if the QoS requirements contain two additive metrics (e.g. cost, delay, delay jitter) or more, QoS routing is an NP-complete problem. In multihop wireless environments, performing resource reservation alone is a more complicate task; in [LPB04], Georgiadis et al. show that link interferences (due to the hidden terminal problem) in multihop wireless networks make the problem of selecting a path satisfying bandwidth requirements an NP-complete problem, even under simplified rules for bandwidth reservation. This means that the per-node local measurements do not offer enough data for end-to-end bandwidth reservation, which makes difficult the implementation of bandwidth reservation schemes for MANETs (e.g. the one proposed in the INSIGNIA [SAXA00] framework).

If we focus instead on less constraining proposals, we also find that there are difficulties to implement them. Taking the SWAN [GAAL02] proposal as an example, we discover that the admission control mechanism used requires all stations to keep track of the MAC's transmission delay for all packets. Also, all stations are required to update the *available bandwidth* field of admission control packets and apply rate control to best-effort traffic. This approach has several drawbacks. An IEEE 802.11 radio performs adaptive rate control when transmitting data according to the Signal-to-Noise (SNR) ratio towards the receiving station. Also, stations may dynamically select different RTS/CTS and fragmentation thresholds. Therefore, the association of a global estimate for transmission delay with a certain bandwidth in the link towards a specific target station is not straightforward at all. Another issue is that MANET stations have heterogeneous characteristics and capabilities, and also different operating systems. So, developing such complex solutions for all available operating systems used by laptops, PDAs, etc. can be a long-lasting task and, in some cases, not possible due to unavailable APIs and proprietary software issues.

To the best of our knowledge, previous proposals for QoS support in MANETs require that all MANET stations implement some sort of resource reservation or admission control mechanism, fitting perfectly in the first paradigm of centralized user control described before. If only a few stations do not implement these mechanisms, then the whole QoS architecture fails or is deeply hindered. Another practical issue that should be taken into account is that terminals typically have low battery and processing power levels, and so reservation of resources, traffic classification, policing and scheduling tasks appear to be excessive for them.

Focusing on our proposal, we wish to develop a technique provide QoS support in MANETs that does not rely on intermediate stations for admission control or signaling purposes, and that does not require them to perform resource consuming tasks such as continuous channel measurements, traffic shaping, resource reservation, etc. We consider that strict QoS reservations do not fit into the underlying philosophy of MANETs, especially those pertaining to the second paradigm described earlier. Rather, we consider that each user should assess what himself and others can/accept to offer, and use it in the best way possible. Iraqi and Boutaba [YR03] introduced the concept of *degree of participation* in ad hoc networks. They argue that stations may participate in the forwarding process with different degrees, possibly performing rate-control or filtering depending on parameters such as power level, number of processes running, memory levels, node capabilities, security and internal resource management policies. Notice that such policies can be chosen with selfish interests. We consider this a very pragmatic approach, and so we wish to find a solution that can accommodate to the degree of participation that each station is willing to offer to the MANET.

Another of our goals is to find a method that can harmonize with the rate-adaptation mechanism of IEEE 802.11, and that can benefit from the MAC-level QoS support of IEEE 802.11e. In chapter 6 we showed that multipath routing can offer serious benefits to multimedia traffic, especially when mobility causes frequent link breaks. So, it would be interesting that the proposed method does not fail when the routing protocol uses multiple paths. It would also be desirable to obtain, apart from available bandwidth estimations, an estimate for the end-to-end delay and delay jitter. These can be particularly useful to support adaptive multimedia applications.

Finally, we also consider important to find a procedure that allows assessing the resources available if a station does not belong to the MANET. This includes not only connections to stations in local area networks (or possibly the Internet), but also stations indirectly connected to a MANET, such as Bluetooth terminals (see [CRP03]), wired devices connected to MANET stations, or terminals belonging to other MANETs.

The solution we consider more adequate to fulfill these requirements relies on distributed admission control. It differs from the previous admission control schemes in the sense that it can cover a very broad range of devices and operating systems, and so does not require that nodes belonging to the MANET implement QoS policies, perform resource reservations, etc. In fact, we consider that the most adequate option is trying to make this process as transparent as possible to the nodes belonging to the MANET.

Distributed admission control, also known as endpoint admission control, is a very scalable technique where hosts (endpoints) probe the network in order to detect its current state. By probing the network with actual traffic we avoid relying on inaccurate, per-node local measurements. This technique is not new, and has already been studied in previous works such as [LES$^+$00], proving adequate for the Internet since it does not have the scalability problems of IntServ, but allows to have a better QoS control than that offered by the DiffServ architecture. Though in [GAAL02] authors argue that probing techniques are not adequate for MANETs since they are hindered by mobility issues, the probing process they refer to was developed specifically for Internet environments; it consists of sending probe packets at the required data rate, being the level of packet losses monitored at the destination. Obviously, such approach can be hindered by mobility since it takes several seconds. However, we consider that merely obtaining the packet loss percentage is not an adequate measurement for admission control in MANETs. We prefer using other kinds of probes to estimate available bandwidth and, optionally, end-to-end delay and delay jitter. With our proposal we expect that the admission time can be strongly reduced, making it suitable for MANET environments.

Relatively to mobility problems, we consider that these can be solved by periodic probing, smart filtering of results and, optionally, awareness of the routing states through packet sniffing or feedback from the routing agent. From our point of view, the benefits of distributed admission control surpass its drawbacks.

With such a distributed access control scheme we seek its transparent integration in a multi-layer QoS framework, where the MAC layer provides QoS support

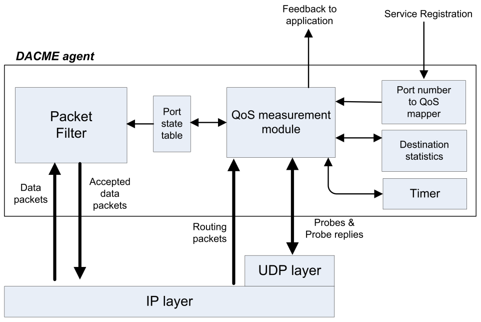


Figure 8.1: Functional block diagram of the DACME agent

using IEEE 802.11e, and the routing layer can also offer QoS support by avoiding congested links (e.g. Q-AODV [CES00] and Q-OLSR [AHKG03]) or by applying multipath routing techniques, such as those exposed in chapter 6.

8.2 The distributed admission control mechanism

DACME is a probe-based admission control mechanism that performs end-to-end QoS measurements according to the QoS requirements of multimedia streams. In order for DACME to operate under optimal conditions in IEEE 802.11-based MANETs, all the radio interfaces should be IEEE 802.11e enabled. However this is not a strict requirement, which means that DACME will still operate correctly in non QoS-compliant MANETs.

In terms of software restrictions on MANET nodes, only the source and destination of a QoS flow must have a DACME agent running. The rest of the nodes will simply treat DACME packets as regular data packets, being unaware of the mechanism itself.

Concerning DACME's components, in figure 8.1 we present the functional block diagram of a DACME agent. The main element of DACME is the *QoS measurement module*. This module is responsible for assessing QoS parameters on an end-to-end path. Another important element is the *Packet Filter*. Its purpose is to block all traffic which is not accepted into the MANET, and also of altering the IP TOS (Type of Service) packet header field in the packets of all accepted flows according to the QoS that has been requested.

An application that wishes to benefit from DACME must register with the DACME agent by indicating the source and destination port numbers, the destination IP address and the required QoS parameters; these data are stored internally in a table indexed using source port numbers.

Once registration is completed successfully the *QoS measurement module* is activated, and will periodically perform path probing between source and destination. The purpose is to assess the current state of the path in terms of available bandwidth, end-to-end delay and jitter. The destination agent, upon receiving

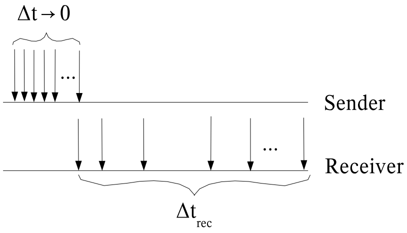


Figure 8.2: Data collecting by the receiver in the bandwidth estimation process

probe packets, will update the *Destination statistics* table where it keeps per source information of the packets received during the current probe. After receiving the last packet of a probe (or if a timeout is triggered) the destination agent will send a reply back to the source DACME agent. The *QoS measurement module*, upon receiving each probe reply, will update the state of the path accordingly. Once enough information is gathered it checks all the registered connections towards that destination, and then decides when a connection should be accepted, preserved or rejected, updating the *Port state table* accordingly (with either accept or drop). If only part of the registered connections can be allowed, preference is given to those which have registered first. This module can also notify applications about service events using a callback function, if requested at service registration.

Actual QoS support is achieved when the *Packet Filter* element, according to the data in the *Port state table*, configures the IP TOS (Type of Service) packet header field on packets belonging to accepted data flows, according to the requested QoS. The IEEE 802.11e MAC must then map the service type defined in the IP TOS packet header field to one of the four MAC Access Categories available: Voice, Video, Best effort and Background.

8.2.1 Support for bandwidth-constrained applications

Relatively to the support for bandwidth-constrained applications, it relies on end-to-end bandwidth measurements, which consist of sending probes from source to destination. Each probe consists of several back-to-back packets; in section 8.3.1 we study the most appropriate number of packets per probe when operating in medium-sized MANET environments (4 hops between source and destination on average).

In figure 8.2 we show the packet generation events at the sender, along with the packet reception events at the receiver. The packets are generated at the source in a bursty manner, so that the inter-packet time approaches zero.

The DACME agent in the destination, upon receiving the probe, will obtain a measure of available end-to-end bandwidth and send it back to the source. This is obtained through the following expression:

$$B_{measured} = \frac{8 \cdot P_{size}}{AIT},$$

Algorithm 2 Probabilistic admission control mechanism for bandwidth-constrained applications

```
After receiving a bandwidth probe reply do {
 correct the bandwidth estimation using all available values
 if (there is a level of confidence of 95% that the available bandwidth is higher that the
requested one)
   then set bandwidth flag to true
 else if (there is a level of confidence of 95% that the available bandwidth is lower that
the requested one)
   then set the bandwidth flag to false
 else if (number of probes used is less than maximum allowed)
   then send a new probe
 else maintain the previous bandwidth flag value }
```

where P_{size} is the size of each probe packet in bytes, and AIT is the Average Inter-arrival Time for probe packets. The Average Inter-arrival Time (AIT) is defined as:

$$AIT = \frac{\triangle t_{rec}}{N-1},$$

where $\triangle t_{rec}$ is the time interval between the the first and the last packet arriving, and N is the number of packets received (not the number of packets sent).

In order to achieve more accurate results, this process can be repeated a certain number of times, though not too many times due to mobility related impediments and also to avoid long startup times, as we will see later in section 8.3.1. Optionally, the receiver can also indicate to the source the number of packets lost.

The DACME source agent, when receiving the probe reply packet, will collect the $B_{measured}$ values sent by the destination agent to be able to reach a decision of whether to admit the connection or not. As we will show in section 8.3.1.2, the source agent must correct the bandwidth estimation value before using it to reach a decision.

The strategy we propose to perform probabilistic admission control is the one described in algorithm 2. Notice that, for applications with bandwidth constraints only, the decision to accept or block traffic will then be taken according to the value of the bandwidth flag.

This algorithm allows reducing the number of probes required to perform a decision to a value as low as two probes; it occurs often in those situations where it becomes quickly evident that the available bandwidth is either much higher or much lower than the requested one. If, after sending the maximum number of probes allowed, still no decision can reached, the chosen criteria consists of maintaining the previous path state. That way, if a connection is waiting for admission it will remain blocked, and if it is active it will remain active. Such criteria aims at reducing the entropy in the MANET.

It should be noticed that the DACME agent or the application itself should always reserve some extra bandwidth to cope with network bandwidth fluctuations, routing data and probes from other sources. By doing so the amount of QoS drops for incoming and out-going video data is reduced and, more important, it avoids routing misbehavior; therefore, it is also each user's best interest.

If the application is only bandwidth constrained, the source will then notify it if the connection can currently be admitted or not. If the application also has requirements on end-to-end delay and delay jitter, the DACME source agent will perform more tests to assess the current end-to-end delay and delay jitter values. These topics will be handled in the next two sections.

Timers

When designing an algorithm for a lossy network environment we should always take care of handling losses in a clear and straightforward manner. In DACME this loss awareness is gained by recurring to timers, being a central element of both source and destination DACME agents.

Each source agent keeps a timer to be able to react in case a probe reply is never received. So, after sending a probe, it sets the timer to go off after 500 ms. If no probe reply is received, causing the timer to be triggered, or in the case that the probing process is completed, the source will schedule a new probing cycle after 3 seconds ±500ms of jitter to avoid possible negative effects due to probe synchronization. This value was chosen from the "Hello"-based version of AODV, where the authors determine that a reaction time of 3 seconds is adequate in the presence of typical topology change rates; moreover, we consider that it offers a balance between the performance drop caused by poor reaction times and the overhead introduced by the probing process itself.

The destination agent must accommodate to the possibility that not all the packets of a probe arrive. So, when the destination receives the first packet, it updates the current sequence number. When the second or the following packets are received it continuously updates an internal timer, setting it go off after:

$$T = \frac{T_{last}-T_{first}}{N_{recv}-1} \cdot (N_{rem} + \varepsilon) + \tau, \tag{3}$$

where T_{last} and T_{first} are the times of arrival of the last and first packet received, N_{recv} is the number of packets currently received, N_{rem} is the number of packets that remain (not received yet), and ε is a fixed number of additional packets used to model a certain degree of tolerance; in our experiments we set this parameter to three packets. While in the first part of the expression we try to accommodate dynamically to the observed network performance, there are situations where we cannot predict the timeout value correctly; an example is a MANET where the routing protocol splits traffic through multiple paths. So, to take into account such situations, we also add τ, a small constant time value; in out experiments it is set to 50 ms since our analysis of MDSR showed that the typical delay differences between different routes is normally less than this value.

Relatively to the maximum packet loss rate allowed, it may occur that, when traffic is split through multiple paths, one of the paths is down. In that situation only a subset of the packets in a probe would arrive. To avoid accepting such measurements as valid we impose that the number of probe packets received should be of more than half in case the routing protocol splits traffic through two different

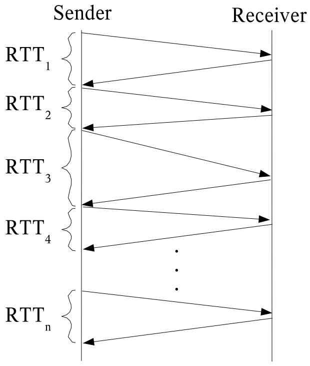


Figure 8.3: Data collecting by the receiver in end-to-end delay estimation processes.

paths (using two alternative paths is the most common case, and also applies to the MDSR routing protocol). If the timer goes up and the destination did not receive enough probe packets, it notifies such event to the source.

8.2.2 Support for delay-constrained applications

When an application has bandwidth and delay requirements, or delay requirements alone, a DACME agent is required to offer a different measurement technique to handle this new constraint.

The technique used to measure end-to-end delay as part of DACME's architecture is similar to the measurements made by a ping application, with the difference that a new *echo request* packet is sent immediately after receiving an *echo reply* packet to reduce as much as possible the time used to perform measurements. Also, the *echo reply* packet should have the same length and the same IP TOS field as the *echo request* one. In figure 8.3 we illustrate this technique.

As we will show in section 8.3.2, we require at least three consecutive round-trip times to obtain a reliable measurement. Therefore, the technique we use to handle applications with delay requirements is the following: we start with several consecutive *probe request/probe reply* rounds to assess the end-to-end delay. The value of the first round is discarded since it is used as a worm-up round to trigger routing and find end-to-end bidirectional paths. The results from the remaining probing rounds are averaged and stored. In case any of the packets is lost, the end-to-end path is considered to be broken and the traffic is blocked.

If the application is also bandwidth-constrained, we then proceed to assess the available bandwidth following the strategy defined in the previous section. The only difference is that, once code from algorithm 2 is executed and a decision is taken relatively to bandwidth, we must then proceed with algorithm 3 to reach a decision based on end-to-end delay also. If the application is delay-constrained alone, it will recur to algorithm 3 immediately after the delay probing process ends.

Algorithm 3 Probabilistic admission control mechanism for delay-bounded applications

```
Execute code from algorithm 2 if appropriate. Then do {
 if (application is bandwidth-constrained && traffic is currently blocked)
     then find worst and best case estimates for delay using both delay and bandwidth measurements;
 else use the measured delay as the best and worst case delay
 if (best case delay > maximum delay allowed)
  then set delay flag to false
 else if (worst case delay < 90% of the maximum delay allowed)
  then set delay flag to true
 else if (application is bandwidth-constrained && number of bandwidth probes used is less than maximum allowed)
  then send a new bandwidth probe
 else maintain the previous delay flag value
}
```

The strategy followed in this algorithm consists of rectifying end-to-end delay by finding worst and best case estimations in case the application is bandwidth-constrained and the traffic is blocked. We will explain this technique further in section 8.3.2. When traffic is flowing, or when the application is delay-bounded only (which suggests that bandwidth requirements are minimal), there is no need to perform adjustments, and the measured value is directly used.

We allow a small margin of uncertainty between 90% and 100% of the maximum delay requested to provoke hysteresis, and so avoid frequent traffic fluctuations.

8.2.3 Support for jitter-constrained applications

In this section we complete DACME's QoS framework by including support for jitter-constrained applications.

Relatively to the jitter measurement process, the source must send packets with the same size, IP TOS field and data rate as the application being served. The receiving end, aware of the source's packet sending rate by explicit notification, calculates the mean and standard deviation values for the absolute jitter and returns them to the source. Measurements made during the jitter measurement phase can also be used to obtain an estimate of network congestion by counting the number of packets lost. This strategy is similar to the one followed by the RTP protocol [HSRV96] for *Sender report* RTCP packets. In figure 8.4 we illustrate the mechanism used to estimate jitter.

These measurements are only performed if the application's traffic is blocked, and they are performed after delay and bandwidth probes if neither test denied the connection. In case that the traffic from the application is flowing through the network there is no need to send jitter probes; this is because the destination agent can measure the jitter of the actual traffic and send it back to the source. Since in our scope of work all the applications with jitter requirements are also delay bounded, we can and will use the first probe reply packet of the delay measurement cycle to carry this information from destination to source. That way

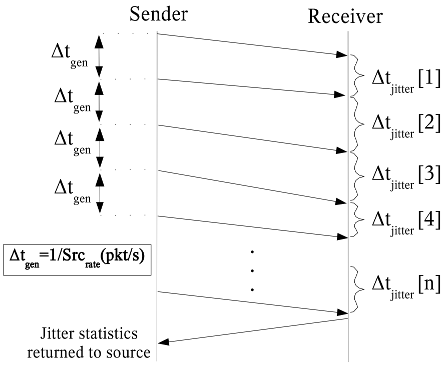


Figure 8.4: Data collecting by the DACME agent at the receiver to estimate jitter

Algorithm 4 Probabilistic admission control mechanism for jitter-bounded applications

```
After receiving a jitter reply do {
  if (2 × standard deviation < maximum jitter)
    then set jitter flag to true
  else if (1.9 × standard deviation > maximum jitter)
    then set jitter flag to false
  else maintain the previous jitter flag value
}
```

we avoid further probing if we find that the jitter requirements are not being met.

Independently of the method used to measure jitter (probes or actual traffic), once the source receives jitter statistics (absolute mean and standard deviation values) it will assess the compliance with the maximum value defined using algorithm 4.

Since jitter follows a normal distribution with a mean value of zero, about 95% of the cases fall between $\pm 2\sigma$. Therefore, in our algorithm, we accept traffic only if 95% of the packets have a jitter value lower than the maximum requested. We also introduce hysteresis by defining a zone between 1.9 and 2σ where the strategy consists of maintaining the previous value. As referred before for delay, this aims at reducing fluctuations on traffic.

After all the required measurements are done, the source agent will decide weather to admit/deny a connection or simply inform the application layer of network status, according to previous negotiations with it. Such interaction is the topic of the next section.

8.2.4 Application level QoS interface

Traditionally, admission control is only required on connection setup (e.g. telephony). Afterwards, QoS is maintained by the network until the connection ends. MANETs do not fit into this paradigm due to mobility issues and because there is no central management entity that can assure that the QoS a flow is experiencing at any given time is maintained at any later time. Therefore, there is the need to perform a periodic check of connection status in order to ensure that both station and radio resources are not being wasted.

An application that wishes to benefit from DACME must first register with the DACME agent. On startup the application identifies the desired destination IP address, the source and destination ports (UDP traffic is assumed) and the priority level of its QoS stream (either Voice or Video); if the application is designed for DACME awareness, it can also define a callback function. The DACME agent will then make a periodic analysis of the state of the end-to-end path through probes, accepting or blocking the application's traffic according to the path's conditions. DACME's actions upon a certain QoS flow can also be sent to the respective application through the user-defined callback function, thereby allowing the application to decide what to do. Such strategy presents some similarities with TCP's congestion control paradigm.

If either the avaliable bandwidth, the end-to-end delay or the delay jitter values do not meet the application's requirements, and if the application has registered a callback function, it can choose to act upon DACME events; for example, it can choose to close the session if it finds that the path is down often, or it can adapt itself to variable network bandwidth conditions. Real-time video streaming sources are an example of applications that can adapt themselves to variable bandwidth conditions by dynamically varying the quality of the video signal. Non-real-time video streaming sources can adapt even more easily to variable bandwidth, end-to-end delay and delay jitter conditions through rate adjustments and delayed playback at the destination node.

A DACME agent can optionally group applications with the same priority requirements and same destination IP address, avoiding this way to perform duplicated probe measurements. Relatively to probe timing, it should try to space different probing tasks in time so that probes do not interfere with each other.

An application that no longer requires the services of DACME must unregister itself with the local DACME agent, thereby avoiding wasting station and network resources. To avoid situations where applications do not unregister from DACME (hanging, etc.), an inactivity timer will detect the unused flows automagically, and as a consequence they will be removed from DACME's registration data.

8.2.5 Interactions with the routing layer

The DACME agent can interact with the routing agent in several different manners. It can use the routing layer information to assess the current state of end-to-end paths, avoiding probe packets when there is no path available, or measuring the QoS of new paths as soon as they become available. The assessment of routing states can be done by communicating directly with the routing agent, or by intercepting routing packets arriving through the wireless interface.

Though the proposed agent does not require any specific functionality from the routing agent, we consider that both DACME and real-time streams would benefit greatly if the routing agent is IEEE 802.11e aware. Such a routing agent would avoid nodes that do not comply with IEEE 802.11e, preferring alternative paths whose nodes' interfaces do comply with that standard.

Concerning the interaction between DACME and the routing protocol, we consider that a DACME agent, and therefore the application that relies on it, can benefit from obtaining awareness of the state of a path as seen by the routing protocol. Therefore, we propose an enhancement to DACME which consists in making the QoS measurement module receive, besides probing packets, routing packets also. This interaction has been represented in figure 8.1 by the arrow tagged "Routing packets". The routing protocol we propose for this enhancement is AODV [CES03]. AODV is a reactive routing protocol that only performs routing tasks when there is actual traffic requiring it. Basically, when a route must be found a RREQ packet is propagated through broadcasting throughout the MANET until the destination is reached. The destination will then send a RREP packet back to the source, and communication can be started. In case the break of a link is detected the node detecting the failure sends a RERR packet to the source, which must then start a new route discovery cycle (for more details on AODV, please refer to section 2.3.3).

Relatively to the integration with DACME, we consider that the packet flow should not be interrupted when the source is notified that a route is not valid since data packets will be enqueued; also, the route discovery cycle is usually not too long, especially when using IEEE 802.11e. Therefore, we consider that the DACME agent should only act when a new path has already been established, so as to assess if the new path can sustain the desired QoS.

There are two situations where the DACME agent will act based on information it gained by listening to routing information. In the first situation the DACME agent is idle because the next probe set is scheduled for a later time. Upon detecting that a RREP packet was received from an active DACME destination, the DACME agent will immediately start a new probing cycle. The purpose is to assess the end-to-end QoS of the new path as soon as it is found. That way, the DACME agent avoids sending data through routes that are possibly congested, improving the overall MANET performance.

In a second (and not common) situation the DACME agent has sent a probe to the destination, and it is waiting for the reply when a RREP from that destination is received. If no probe reply is received and the timer is triggered, the DACME agent will not consider the path to be down; instead, it will send a new probe to the destination to find out if the path has become available during that short period of time.

We consider that the strategy defined above provides enough information from the routing layer, allowing to achieve substantial improvements. The integration of DACME with AODV will be denoted as DACME-AODV from now on.

To assess the effectiveness of the DACME-AODV solution, we devised a very simple mobile scenario with 10 nodes just for illustration purposes. The traffic consists of a single CBR source regulated by a DACME agent, and there is no background traffic. Our purpose is to evaluate only the effects of the interaction between the DACME agent and routing data. We set node mobility at a constant speed of 5 m/s according to the random way-point mobility model. The routing

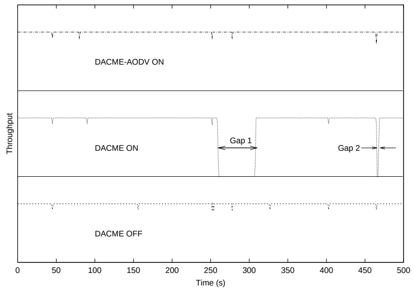


Figure 8.5: Benefits of routing awareness by the DACME agent.

protocol used is AODV and simulation time is 500 seconds.

In figure 8.5 we present the results in terms of throughput when DACME is turned off, and when it is turned on. In terms of DACME behavior, we compare the default DACME agent with the AODV-aware DACME agent. The example presented was chosen among several tests since it is particularly illustrative for our purpose.

The results found show us that, when DACME is turned off, there are sometimes routing-related packet losses which cause throughput levels to decay slightly. When the default DACME solution is used such behavior persists; however, there are at times situations where the lack of routing awareness causes the DACME agent to interrupt the data flow for periods that are usually short (e.g., Gap 2), but that can last longer in worst case situations (e.g., Gap 1). We observe that, when using DACME-AODV, such misbehavior is not prone to occur. We therefore consider that introducing the DACME-AODV agent in mobile environments allows maintaining good performance levels and keeping data flow interruptions to a minimum.

8.2.6 Interactions between DACME and the IEEE 802.11e MAC layer

QoS parameters are typically defined at the application level depending on the requirements of a particular application. The Internet Protocol (IP) supports traffic differentiation mechanisms in the sense that it allows tagging the packets according to QoS requirements, so that successive network elements can treat them adequately. This is achieved using the 8 bits of the “Type of service” field in an IPv4 datagram header or the “Traffic class” field in an IPv6 datagram header. In this work our proposal consists of using the 3 TOS bits, part of both “Type of service” (IPv4) or “Traffic class” (IPv6) fields, to indicate the desired user priority. These shall then be mapped to IEEE 802.11e ACs according to table 1.2 on page 15.

The IEEE 802.11e draft [IEE05] states that stations that benefit from IEEE 802.11e are able to offer differentiated treatment to a packet by negotiating with the IEEE 802.11e MAC Service Access Point. The IEEE 802.11e MAC Service Access Point (MAC_SAP) allows to negotiate QoS specifications in two ways: either directly by setting a traffic category (TC), or indirectly by making a traffic specification (TSPEC) instead. It is the value of the user priority (UP) parameter which indicates to the MAC_SAP the desired choice using values in the range 0 through 15. Priority parameter values 0 through 7 are interpreted as actual user priority values according to table 1.2, and so outgoing MSDUs are therefore marked according to the correspondent access category. Priority parameter values 8 through 15 specify traffic stream identifiers (TSIDs), and allow selecting a TSPEC instead.

The value of the chosen user priority is mapped to packets transmitted by setting the QoS Control field, part of the IEEE 802.11e MAC header, accordingly. The QoS Control field is a 16-bit field that identifies the traffic category or traffic stream (TS) to which the frame belongs and various other QoS-related information about the frame that varies for the particular sender and by frame type and subtype. In particular, it is the TID field (part of the QoS Control field) the one that identifies the TC or TS of traffic for which a TXOP is being requested. The most significant bit of the TID, when set to 0, indicates that the request is for data associated with prioritized QoS and, when set to 1, indicates that the request is for data associated with parameterized QoS. The remaining bits define the UP value or the TSID accordingly.

When receiving a packet, the IEEE 802.11e MAC analyzes the QoS Control field and also offers a differentiated treatment to packets with different QoS requirements when passing them to upper stack layers.

The QoS strategy proposed in DACME's framework requires MANET stations to treat packets according to the priority tagging in their IP header. Probe packets, similarly to data packets, should be handled by the MAC layer according to their priority (set through the IP TOS field), so that MAC layer QoS becomes effective.

The IEEE 802.11e MAC layer distinguishes between four different traffic classes. However, only two of them (Voice and Video) are adequate for real-time services. In our framework we always assign bandwidth probing packets to the Video Access Category (AC_VI). This avoids accepting higher priority (Voice) connections that would affect ongoing connections of a lower priority (Video). Otherwise, we would incur into the stolen bandwidth problem described in [LES$^+$00] (see section 8.3.1 for further details).

Relatively to end-to-end delay and jitter probing packets, these should be assigned according to the Access Category of the application stream itself, so that measurements are correct and meaningful.

As a final remark, the Contention-free Bursting mechanism - part of the IEEE 802.11e framework - should be turned off in order to avoid jitter peaks and, more important, to make the probing measurement process more reliable.

8.2.7 Description of DACME packets

DACME agents make use of UDP over IP for end-to-end information interchange. The information being transmitted follows a strict format to avoid any ambiguities. DACME packets have a header (the DACME header) that follows the UDP header, and where the most important information is. After this header follows a payload with variable size, and whose content can be random - for packets traveling from

TYPE:8	SEQ_NUMBER:16	PKT_ID:8	LAST_PKT_ID:8	*PAYLOAD*

Figure 8.6: Format of a DACME packet

source to destination - or measurement results when available - for packets traveling from destination to source.

We now proceed to define the general format for the DACME packet header. This format is presented in figure 8.6 and, as shown in that figure, has a length of 5 octets.

The type of packet field (TYPE) can take one of the following values: B_PROBE, B_REPLY, E2ED_PROBE, E2ED_REPLY, JIT_PROBE and JIT_REPLY; so, there are 250 values that remain, and which are reserved. The sequence number field (SEQ_NUMBER) consists of a number that is always increasing, used to distinguish DACME packets from the current *probe request/probe reply* cycle from previous ones. The packet ID field (PKT_ID) is a value that differs from packet to packet for the same sequence number, and the last packet ID field (LAST_PKT_ID) is used by the destination DACME agent to calculate the number of packets that remain (not received yet).

Concerning the payload, it is set so that, along with the DACME header, it matches the desired packet size (the same one as the application) for packets traveling from source to destination; it can optionally contain data depending on the type of packet. We will use it in our experiments to carry estimates of available bandwidth, delay and jitter from the DACME agent in the destination to the one in the source.

8.3 Tuning DACME for MANET environments

In the previous section we defined the architecture of DACME, defining the structure of DACME agents and highlighting the relationship between DACME and other network protocols. As explained before, DACME relies heavily on probes to assess available bandwidth, delay and jitter.

In this section we will tune each of these probing processes so that we achieve optimum performance in medium-sized MANET environments, characterized by an average number of four hops from source to destination. Yet, applying the tuning results developed here to other environments will not cause significant degrees of inaccuracy since the strategy developed is as much flexible as possible.

The need for tuning derives from the field of application itself: wireless mobile ad-hoc networks. These networks are characterized by complex interactions between stations in terms of radio resources, making any local measurements insufficient for the calculation of end-to-end QoS parameters. Moreover, any probe-based measurement will have an impact on the traffic traversing the MANET and will be affected by it, which means that all measurements require adjustments when targeting accuracy.

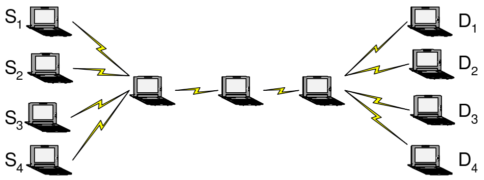


Figure 8.7: Static scenario

8.3.1 Available bandwidth estimation

In this section we will tune DACME's admission control mechanism to achieve reliable bandwidth measurements in a short period of time. We consider that if measurements take too long they can be corrupted by mobility, making them unreliable and possibly useless. We are also aware that longer measurements lead to more reliable and accurate results. So, we need to find the best trade-off.

In order to obtain precise reference values we devised a static scenario (see figure 8.7) which allows us to make measurements in a controlled environment. Such an environment is very important at this initial stage in order to measure the different degrees of accuracy achieved by the different types of probes evaluated.

When choosing the scenario our aim was that contention among stations had effects on performance. This was achieved by aggregating several stations within data communication range of each other (traffic sources and first intermediate station). Our aim was also to create a multi-hop environment, since this is typical of MANETs. That was achieved by including intermediate stations to achieve four-hops paths. With this setting we also experience the effects of hidden terminals and carrier sensing ranges (considerably higher than communication ranges).

To conduct our experiments we used the ns-2 simulator [KK00] with the IEEE 802.11e extentions by Wietholter and Hoene [SC03]. We set up the IEEE 802.11 radio for both IEEE 802.11a and IEEE 802.11g since the simulator's radio model does not differentiate between them. Concerning the IEEE 802.11e MAC, it was configured according to the values presented in section 1.2 (see table 1.3).

Our measurements were made over a period of 60 seconds and averaged over twenty simulation runs. We use static routing so that routing actions do not interfere with data traffic. The purpose is to make reliable measurements in steady-state.

Relatively to the sources of traffic, source/destination pair (S_1, D_1) is used by reference streams of different categories to generate packets at a very high data rate, so that the source's interface queue is always full; it is also used to perform measurements using probes. The three remaining source/destination pairs are used to generate different levels of background traffic by varying the data generation rate. The purpose is to saturate the network gradually. These three background sources generate traffic with negative-exponentially distributed inter-arrival times for all 4 Access Categories of IEEE 802.11e.

Concerning the RTS/CTS and fragmentation mechanisms, they are turned off. We fix the packet size to 512 bytes for all sources, including probing packets. We

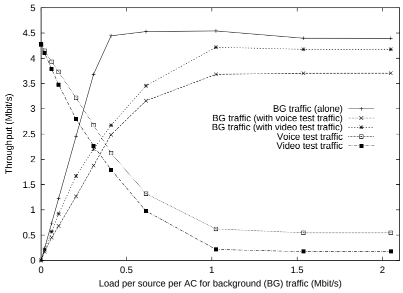


Figure 8.8: Measured available bandwidth varying background traffic

consider that the size of the probe packets should be the same as the ones used by the application wishing to perform a connection. This requirement stems from the fact that the source is unaware of the RTS/CTS and the fragmentation thresholds of the nodes participating in the end-to-end path, which are factors that can clearly affect the available bandwidth. When the source does not use a fixed packet size, it can use an average value for it, though results will not be so accurate.

To obtain reference bandwidth values we perform different simulation experiments where we set source S_1 to generate no traffic, then Voice test traffic and finally Video test traffic. Both Voice and Video test traffics saturate the channel, so that we can measure the long-term end-to-end bandwidth available. In figure 8.8 we show how the aggregated background traffic (BG) and the test traffic change by varying the background load. We have a level of confidence of 99% that the mean value will be within 1% of all points represented.

From figure 8.8 we see that, if all bandwidth is available (background traffic load is null), there is not much difference between the throughput achieved with the Voice and Video test traffics. However, as the background load increases, it becomes evident that, if the test traffic pertains to the highest priority AC (Voice), the bandwidth share obtained is higher (especially in saturation).

The wrong approach in this situation would be to send probes with the same priority as the application's data. This is because, similarly to what was found in [LES+00], we should set bandwidth probes to the same priority always (see section 8.2.6). Therefore, we will perform all our bandwidth measurements using probes set to the Video AC. This aspect will be fundamental to our admission control strategy. Also, remember that all remaining bandwidth will be used by best-effort and routing traffic, and so there is never an actual resource misuse, but rather a trade-off.

8.3.1.1 Probe size tuning

In this section we will find the optimal number of packets per probe taking into account the trade-offs between the accuracy of bandwidth measurements and probing time. We use the values for the Video test traffic depicted in figure 8.8 as our reference, and we test different probes with sizes (number of packets) equal to 2,

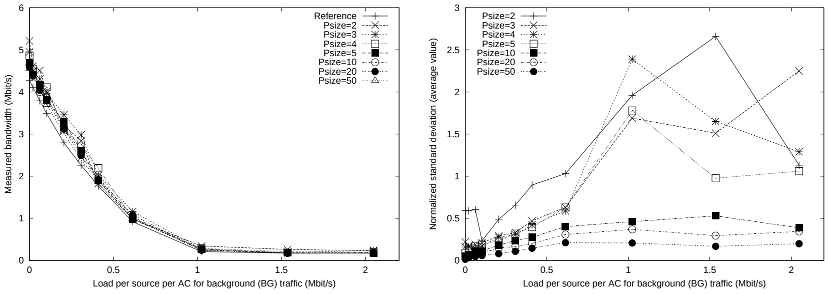


Figure 8.9: Bandwidth estimation (left) and normalized standard deviation(right) for varying background traffic

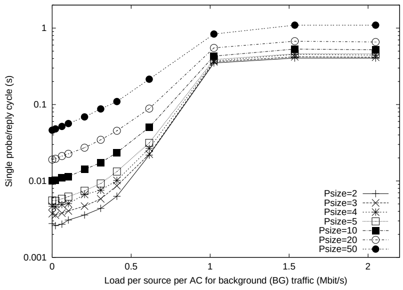


Figure 8.10: Single probing cycle time for varying background traffic

3, 4, 5, 10, 20 and 50. Since each node's queue is limited to 50 packets, the highest value under test is also the upper limit.

In figure 8.9 (left) we show the average values for the measured bandwidth for the different probe sets, as well as the reference bandwidth curve when varying background traffic. As it can be seen, probe measurements tend to over-estimate available bandwidth, and so we find that the mean is a biased estimator for the available bandwidth. Also, as the normalized standard deviation shows, the measurement results tend to spread as the probe size decreases. From figure 8.9 we conclude that tested probe sizes inferior to 10 can lead to significant errors, and so should be avoided. We also conclude that we may require several consecutive probe cycles to achieve an accurate estimate of available bandwidth.

We now proceed by analyzing the time it takes to complete a single probing cycle. Such cycle consists of sending several probe packets by the source, followed by a reply in return.

Figure 8.10 shows that the single probing cycle time increases with probe size, as expected. We can also see that, under saturation, the cycle time can reach relatively high values (more than one second for a probe size of 50). Moreover, there are further problems under saturation, such as the probe becoming unusable due to too many packet losses. In table 8.1 we show the percentage of lost packets

Table 8.1: Probability for a probe to become unusable under saturation for different probe sizes

Number of packets per probe	Lower threshold	Lost packets [saturation]	P(probe unusable) [saturation]
2	2	39 %	57 %
3	2	43 %	40 %
4	3	40 %	54 %
5	3	44 %	40 %
10	6	48 %	49 %
20	11	44 %	42 %
50	26	45 %	32 %

for different probe sizes, as well as the probability of a probe becoming unusable under saturation; in this calculation we take into account the lower threshold for the number of packets arriving to the destination agent. In our choice for the lower threshold we consider that more than half of the packets should arrive for the probe to be meaningful, and that this number should never be less than 2 because, otherwise, no estimate can be calculated at all.

As we can see from that table, very small probe sizes cause slightly fewer packet losses, but may lead to quite high chances for a probe to become unusable. This is due to the lower threshold imposed. Independently of these results, we should take into account that, when the network is saturated, the source will not accept further flows (especially because there is always an additional amount of bandwidth which is reserved).

Taking into account all the results found in this section, we consider that setting the number of packets per probe to 10 is a reasonable choice, and so we will use 10-packet probes from now on.

8.3.1.2 Improving the estimation of available bandwidth

In this section we will refine the bandwidth estimation process by using consecutive probes and a bandwidth correction function.

In the previous section, an analysis of the bandwidth probing results showed that the sample mean was a biased estimator for the available bandwidth. We now present further insight into this phenomena by presenting the discrete probability distribution for the probing process at three distinct background traffic levels; we will analyze the behavior under low, average and high congestion levels. We obtain the distributions by splitting the range of each probing process into fifteen intervals of equal length. Results are shown in figure 8.11.

The arrow/letter pairs refer to available bandwidth measurements made with real traffic, and are used as reference for comparison. As it can be seen, the three probability distributions are not centered around the reference values (H, A and L), which explains why their mean is superior to the real traffic measurements.

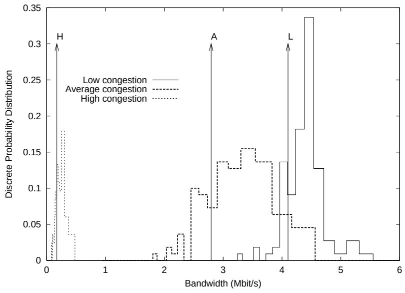


Figure 8.11: Discrete probability distribution for the probing process under low, average and high levels of congestion

Table 8.2: Parameter values and residual values for the three estimators

	B_{e1}	B_{e2}	B_{e3}
α	0.899	0.943	0.940
β	-	-0.379	-0.315
γ	-	-	-0.155
Squares of residuals	1.224e11	1.517e-1	1.495e-1

Also, we notice that average levels of congestion tend to favor lower kurtosis values.

We find that the mode and the median also over-estimate the available bandwidth. So, we need to correct the measured values to make them match the actual ones. With that aim, we propose three different estimators for available bandwidth:

$$B_{e1} = \alpha \cdot \mu_e \tag{1}$$

$$B_{e2} = \alpha \cdot \mu_e + \beta \cdot \sigma_e \tag{2}$$

$$B_{e3} = \alpha \cdot \mu_e + \beta \cdot \sigma_e + \gamma \cdot \frac{\sigma_e^2}{\mu_e} \tag{3}$$

where μ_e is the samples' mean, σ_e is the standard deviation, and parameters (α, β, γ) will be used for curve fitting purposes. Each sample consists of $B_{measured}$ values (bit/s). We have chosen these three relations for study since both mean and standard deviation estimates require all the sample values for their calculation. We have tested other possibilities for the estimators, but we did not find any that offered better results. Starting from these three possible available bandwidth estimators, we use a curve fitting process to find the optimum values for parameters α, β and γ based on least square error regression. The optimum parameter values found, as well as the squares of residuals, are presented on table 8.3. As it can be seen, there is a very significative difference in terms of residuals from B_{e1} to B_{e2}. However, the difference between B_{e2} and $B_{e3}$is only slight.

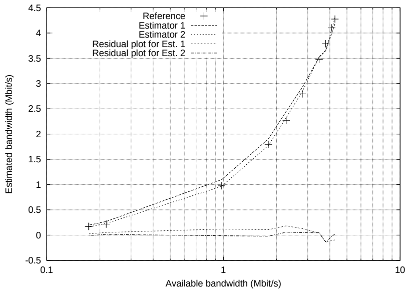


Figure 8.12: Estimation curves and residual plot

In figure 8.12 we show the result of the curve fitting process for estimator 1 (B_{e1}) and estimator 2 (B_{e2}). We do not present results relative to B_{e3} since these are visually indistinguishable from those of B_{e2}. As the residual plots show, the estimation accuracy of B_{e2} is quite better than that of B_{e1}, being kept at very low values. In fact, we find that the maximum relative error found is of 24.54% when using B_{e1}, 6.42% when using B_{e2} and 6.94% when using B_{e3}. Taking into account these results, we consider that the level of accuracy achieved with B_{e2} is good enough, and so it will be our available bandwidth estimator from now on, using B_{e1} for comparison when needed. Also, notice that expression B_{e2} is more simple and clear than the one for B_{e3}, requiring fewer calculations.

8.3.1.3 Tuning the number of probe cycles

In the previous section we showed that by using several probes we can obtain a more accurate estimate of the end-to-end bandwidth available. However, increasing the number of probes used also has drawbacks, such as increasing the bandwidth consumed, increasing the admission control time and depleting more resources.

In this section we will tune the number of probe cycles in order to find a good balance between bandwidth estimation accuracy and admission control time, an issue we consider very important when talking about QoS streams. We shall refer to the number of probing cycles as the probe set size (in contrast to probe size, which refers to the number of packets per probe). Using both bandwidth estimators B_{e1} and B_{e2} developed in the previous section, we study the mean and peak estimation errors when varying the probe set size (see figure 8.13).

That figure shows that estimator 2 always offers the best results, even when the probe set size is as low as 2. Also, we can see that both mean and peak errors tend to reach a steady value after a given probe set size. In fact, we observe that the difference in terms of peak error between probe sets of sizes 5 and 6 decreases by less than 3%, and that the mean error between both decreases by less than 0.03%. In contrast, the peak error decreases by more than 15% between probe sets with size 4 and 5. These results further evidence the goodness of estimator 2, and so we now drop estimator 1.

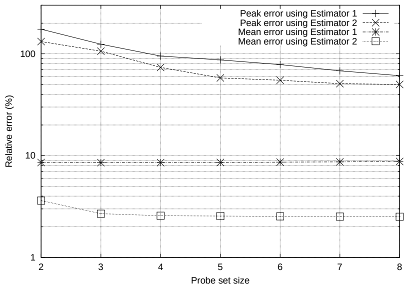


Figure 8.13: Peak and mean relative error for B_{e1} and B_{e2} using different probe set sizes

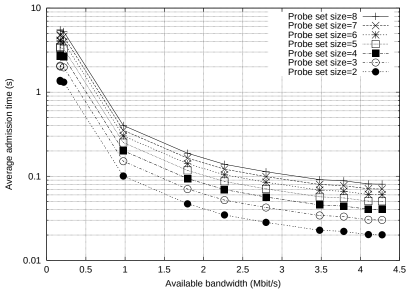


Figure 8.14: Average admission time with different number of probe sets

We also consider the total admission control time using different numbers of probe sets in our study. In figure 8.14 we present the average admission control time when the available bandwidth in the end-to-end path differs. In our calculations we take into consideration the results of table 8.1 relative to the probability for a probe to become unusable, so that our admission time estimates are accurate.

We verify that, under small/moderate congestion, the admission control time in our scenario is very low when using a probe set size of 5 (under 300 ms), never reaching values above 3.5 seconds, even when the network suffers traffic congestion. Taking into consideration the results found, we consider that using probe sets sized 5 is a reasonable choice, offering a good balance between bandwidth estimation accuracy and admission control time.

To complete our study, we will now detail the results achieved with the bandwidth estimation process tuned to use probe sets sized 5; each probing cycle consists of 10 consecutive packets sent by the source, followed by a reply from the destination.

Figure 8.15 shows the accuracy achieved with the bandwidth estimation algorithm proposed. We show the mean and an interval defined by the inter-quartile range which contains the mid 50% of the data. We also show the minimum and maximum values found.

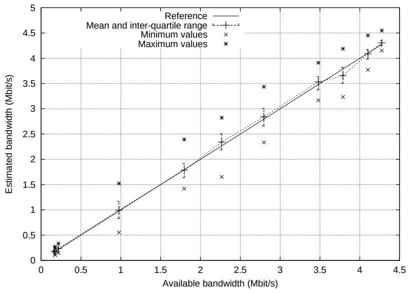


Figure 8.15: Bandwidth estimation accuracy

As it can be seen, the estimation process achieves a good degree of accuracy, and most measurements are kept very close to the reference line. The differences for the minimum and maximum values encountered are expected due to the nature of the stochastic packet generation process of the competing background sources, which also explains why the min-max range is higher for mid-scale values.

The strategy developed in this section aims at providing accurate end-to-end bandwidth measurements. However, there will be many cases where such accurate measurements are not necessary. An example are applications that cannot adapt to available bandwidth, and where a binary decision should be made on weather to admit a flow or not according to algorithm 2 on page 185. In this situation, the agent may use fewer probing cycles if it estimates that available bandwidth is much higher than the one required by the application or if, on the contrary, it is much lower.

8.3.2 End-to-end delay estimation

In this section we will present the algorithm used to estimate end-to-end delay values at admission time. Such measurements can be useful to applications that have strict end-to-end requirements and that, without bounds on delay, cannot meet the purpose they were design for.

We start by finding what is the minimum number of consecutive ping-pong probes that offers an acceptable degree of accuracy on end-to-end delay estimation. With that purpose, we again pick the scenario of figure 8.7 for study. We set sources S_2, S_3 and S_4 to generate a moderate amount of background traffic with negative-exponentially distributed inter-arrival times at 200 kbit/s per AC; source S_1 is used to obtain reference values at different source rates with either Voice or Video traffic, being this traffic sent at a constant bit-rate. The results are presented in figure 8.16. We have a confidence level of 99% that the mean values of the distribution are within an interval of 1% of the values obtained.

According to the measurements performed using the bandwidth estimation tool developed in section 8.3.1, the available bandwidth is of about 2.9 Mbit/s. Since, as exposed before, available bandwidth measurements are always done by setting probes to the Video AC, we can easily relate this value to the change observed

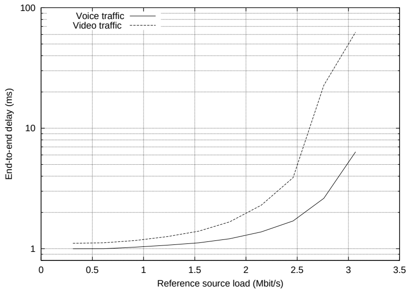


Figure 8.16: Reference end-to-end values obtained with real traffic for different source rates.

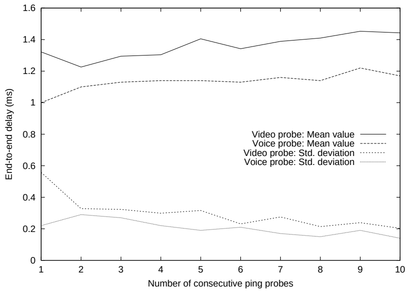


Figure 8.17: Mean and standard deviation values for Voice and Video end-to-end delay probes

close to this source load in figure 8.16.

Using these reference values, we now proceed by performing end-to-end delay estimation through probes. The process consists, as depicted in figure 8.4, of making several consecutive ping-pong tests without waiting for a reply packet to send the next request packet. This is to accelerate the probing process as much as possible.

After the test ends, we use these values to obtain a low-rate estimate for end-to-end delay. We call it low-rate estimate since, as we will discuss next, it needs to be adjusted according to the ratio between requested and available bandwidth. This low-rate estimate can be obtained by taking either the mean or the median of all values, and then diving it by two (since we wish to find the end-to-end delay, and not the round trip time). We found that the median does not provide more accuracy than the mean, and so we choose the mean as our basis for further estimation.

To obtain values directly related to the actual traffic, probe priority is set either to the Voice or Video access categories. In figure 8.17 we show the mean and standard deviation values for an increasing number of consecutive ping probes. As expected, by increasing the number of consecutive ping probes the mean value also tends to increase (congestion related effects) and the standard deviation to decrease. Taking into consideration both accuracy and the time to accomplish this task, we decide to use 3 consecutive ping probes for further study. The total delay in this step using the current configuration is around 7 milliseconds, a very low value.

As referred previously, the end-to-end delay estimation value found through ping-pong probing is a low-rate estimate. This means that it can accurately estimate the delay if the application rate is small when compared to the estimated available bandwidth value. We wish to find a method to estimate the end-to-end delay values for medium/high traffic bandwidths also, and we will accomplish this first by normalizing both the bandwidth required by the application and the end-to-end delay values, and then adjusting the resulting curves using functions of one of the following types:

$$D_{e1}(x) = e^{\alpha \cdot x + \beta} + \gamma \tag{4}$$

$$D_{e2}(x) = e^{\alpha \cdot x + \beta} + \gamma + \eta \cdot x \tag{5}$$

$$D_{e3}(x) = e^{\alpha \cdot x + \beta} + \gamma + \eta \cdot x + \vartheta \cdot x^2 \tag{6}$$

In these expressions x is the normalized bandwidth value, and parameters $(\alpha, \beta, \gamma, \eta, \vartheta)$ are used for curve fitting purposes. We have tested other types of functions, but these were the ones which offered the best results. The value of x (ranging from 0 to 1) is obtained by the ratio between the application's chosen data rate and the bandwidth estimated through the probing process. Relatively to D_e values, these are normalized delay values obtained dividing end-to-end delay values by the low-rate delay estimation (obtained through the ping-pong probing process). The normalization of both bandwidth and end-to-end delay values serves the purpose of finding values for the different parameters that are as independent as possible of the scenario used; that way, we can find the general trend of delay's behavior for both access categories when varying relative bandwidth usage.

Using any of the delay estimators defined before, we can then estimate the delay value on an end-to-end path (for a certain value of x) by multiplying the normalized delay value by the low-rate value obtained in the ping-pong process; this operation is required to de-normalize it.

We use the reference values obtained initially to find the optimum values for vector $v = (\alpha, \beta, \gamma, \eta, \vartheta)$ for both Voice and Video traffic. These different traffic categories require a different treatment since the normalizing factor for x is the same for both, while their behavior for the same source data rate (see figure 8.16) differs. In table 8.3 we show the optimum values for the different parameters, as well as the square of residuals (which is our measure of the goodness of the curve fitting process).

In both cases (Voice and Video traffic) we observe that estimator D_{e3} is considerably more accurate than the remaining ones, and so we choose it as our bandwidth-dependent end-to-end delay estimator. Figure 8.18 allows making a visual comparison between the reference delay values and the estimated delay curves for Voice and Video traffic; we find that the accuracy levels are quite satisfactory.

As a final remark, we should point out that this method is only required when the application traffic is retained at the queue, either because initial admission

Table 8.3: Parameter values and residuals for the different delay estimator functions

	Voice			Video		
	D_{e1}	D_{e2}	D_{e3}	D_{e1}	D_{e2}	D_{e3}
α	10.5083	12.051	13.575	19.2953	21.965	25.476
β	-9.99919	-11.7511	-13.5063	-16.2059	-18.8781	-22.3964
γ	0.955884	0.79969	0.910468	1.09242	0.646413	1.04908
η	-	0.393359	-0.299056	-	1.08634	-1.45776
ϑ	-	-	0.825743	-	-	3.03319
Residuals^2	4.64371e-3	7.45571e-3	4.36604e-4	4.23408e-1	1.02343e-1	1.41473e-2

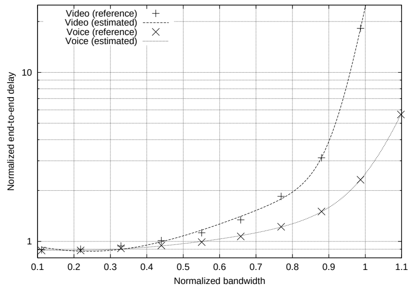


Figure 8.18: Reference and estimated end-to-end delay values

control is being performed, or previous path conditions caused the application to stop transmitting. When transmissions are on-going measurements can still be made, though no bandwidth-dependent adjustments are required.

8.3.3 Jitter estimation

In this section we will detail the jitter estimation process. We wish to obtain not only the mean absolute jitter value, but also the standard deviation for the jitter. Most multimedia applications rely on the RTP protocol for transport which relies on RTCP to provide estimates for transmission parameters such as loss rate and jitter. Therefore, we consider that the method we develop will be useful mostly during initial admission control to verify if the network is able to satisfy the minimum requirements of the application, and also when there are route changes that invalidate previous measurements.

We will use the same scenario configuration of the previous section in order to obtain reference values for the jitter, and also to perform tests using jitter probes. These probes consist, as depicted previously in figure 8.4, of sending some packets at the same rate as the application's traffic, and the collection of jitter statistics by the agent in the destination which are returned to the source in a single packet.

The minimum number of packets required to obtain mean and standard deviation jitter values is 3, though many more should be sent so that the values achieved resemble those of actual traffic. If the application's data rate is too low, this could cause the admission time to increase beyond acceptable bounds. For these situations we propose using higher data rates during probing, and to estimate jitter based on those measurements. Moreover, we consider that most of such applications will not depend heavily on jitter either.

Contrarily to the approaches previously followed for probing available bandwidth and end-to-end delay, in the jitter estimation process we will not find the minimum number of packets that offers good accuracy for all traffic rates; instead, we will find the smallest time interval for probing that offers the desired accuracy, which is the opposite strategy. This way we bound the delay associated with the jitter probing test, making it independent of the application's data rate. An interval range between 50 and 300 ms seems reasonable and is acceptable as admission control delay, and so we will use it for further analysis.

In figure 8.19 we show the values used for reference relative to the mean absolute jitter and the jitter's standard deviation for both Voice and Video access categories. As expected, increasing the source load also increases jitter due to contention with background traffic. The differences found between Voice and Video traffic are expected due to the differences in terms of IEEE 802.11e MAC parameters (see table 1.3), which obviously favor Voice traffic.

Our objective is to find the minimum interval of time that offers a reasonable accuracy when estimating the mean absolute value and the standard deviation for the jitter. The metric we choose to assess the concentration of samples is the inter-quartile range (IQR). In figure 8.20 we show the average inter-quartile range of the distribution for both mean and standard deviation values relative to jitter when performing the probing process.

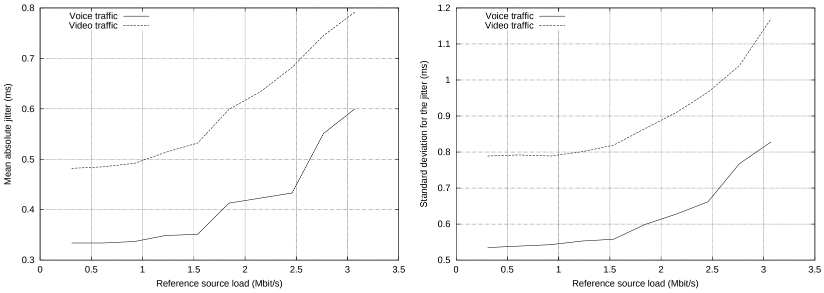


Figure 8.19: Reference values for the mean absolute jitter (left) and the jitter's standard deviation (right) obtained with actual traffic for different source rates.

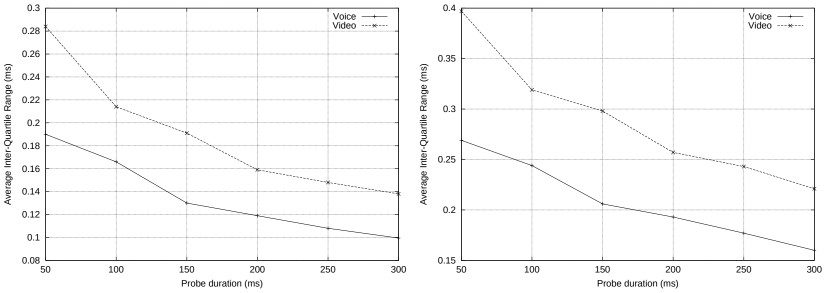


Figure 8.20: Average Inter-Quartile Range estimating the mean jitter (left) and the standard deviation for the jitter (right) for different probe durations

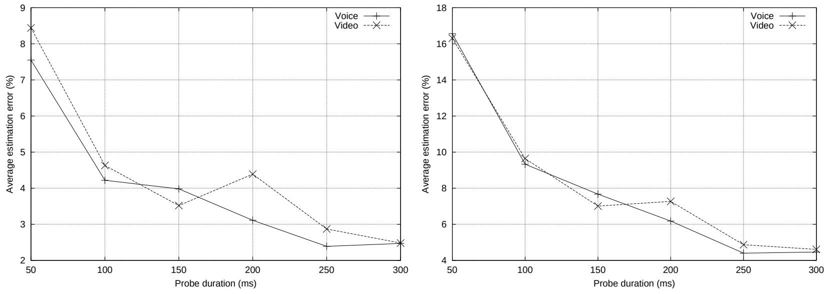


Figure 8.21: Average estimation error for the absolute mean (left) and the standard deviation (right) for Voice and Video traffic varying the probe size

We can see that the degree of peakedness of all four distributions increases (IQR decreases) with longer probe duration times, as expected. We aim at interquartile range values as low as possible so that consecutive measurements achieve similar values.

Another issue to be addressed is the accuracy achieved when estimating jitter compared to the reference values. In figure 8.21 we show the relative error estimating the mean absolute value and the standard deviation for the jitter. Again, longer probe duration intervals are related to smaller estimation errors.

From the results found previously in figures 8.20 and 8.21 we consider that a probe duration of 250 milliseconds is enough to achieve accurate and consistent estimations for the mean absolute jitter value, as well as for the standard deviation. Relatively to measurements when traffic streaming from the source has already started, they are done using actual traffic, being reported to the source using other probe reply packets (e.g. the first delay probe packet).

With this last step we conclude the tuning of the admission control system, having therefore set the complete framework required for DACME agents to be implemented and configured. In the next section we proceed by making conformance tests to assess if DACME is operating as expected.

8.4 Basic functionality conformance testing

After completing the description of DACME's framework, and having tuned DACME for optimal operation in MANET environments, we now proceed to make some conformance tests concerning the bandwidth measurement process. These tests consist of checking if DACME is operating correctly in a static environment; we want to verify if the QoS enhancements experienced by sources is significant, and also if DACME is able to react quickly enough to congestion changes in the MANET.

In the second part of this chapter we make further tests, the main differences being that they are performed in a dynamic environment; we also wish to determine how the performance changes when varying the amount of additional bandwidth reserved, an issue that we already referred to in section 8.2.1.

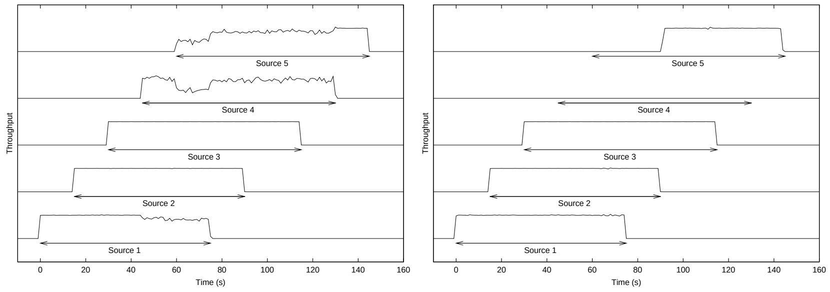


Figure 8.22: Throughput values for different sources without DACME (left) and with DACME (right)

8.4.1 Behavior in static scenario environments

MANETs are characterized by the requirement to adapt, not only to congestion changes, but also to mobility of the nodes that conform it. In this section, however, we study a MANET environment where nodes are not moving; this way we can assess if DACME is functioning adequately both in terms of QoS support and responsiveness to traffic changes. The issue of mobility will be handled in the next section.

To perform the desired evaluations we setup a 1900×400 squared meters scenario with 50 static nodes. All nodes are equipped with an IEEE 802.11g/e interface, and the radio range is of 250 meters. The position of nodes is random, and the average number of hops between nodes is 4. Concerning routing, we use static routing at this stage.

DACME agents handle five CBR sources sending data at a rate of 1 Mbit/s. All packets are set to the Video Access Category. Concerning background traffic, it consists of four sources, each sending negative-exponentially distributed traffic at a rate of 50 packets per second in all four Access Categories. The packet size for both CBR and background sources is 512 bytes.

Concerning DACME CBR sources, the first source starts at the beginning of the simulation, and a new source is started every 15 seconds until all five sources are active. Afterwards, they are turned off in the same order they were turned on.

In figure 8.22 we show the throughput for each source (maximum is 1 Mbit/s); we use an arrow to indicate the period of activity for each QoS source.

We can see that, when DACME is not used, sources 1, 4 and 5 suffer from throughput degradation, which results in QoS degradation if we were in the presence of, for example, video data. What we aim at is, when a flow is admitted into the MANET, to be able to sustain the requested QoS; when that is not possible, the flows should no longer be admitted into the network.

We find that DACME allows us to achieve our purpose successfully. Notice that source 4 is never allowed to transmit since the DACME agent verifies that there is not enough bandwidth at any instant of time while it is active. Relatively to source 5, we verify that it is allowed to transmit as soon as source 2 stops

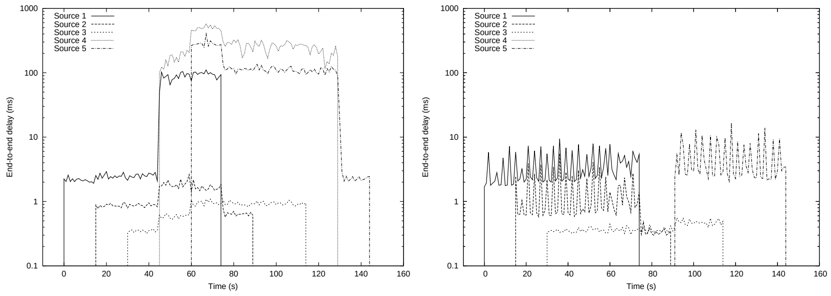


Figure 8.23: End-to-end delay values for different sources without DACME (left) and with DACME (right)

Table 8.4: DACME statistics

	Source 1	Source 2	Source 3	Source 4	Source 5
Probe packets generated	1140	770	720	680	1120
Probe packets lost (%)	0.614	0.000	0.000	3.382	0.738
Num. replies	114	77	72	68	112
Probe replies lost (%)	0.000	0.000	0.000	0.000	0.000
Avg. probe/reply time (ms)	33.804	4.813	3.174	57.891	40.769
Avg. cycle time (ms)	157.414	15.098	8.297	147.015	194.055
Avg. inter-cycle time (s)	2.927	2.887	2.963	3.047	3.035
Num. hops to destination	5	1	1	7	5

transmitting, which indicates that the algorithm proposed for DACME is able to react relatively quick to changing network traffic conditions.

In terms of end-to-end delay, figure 8.23 shows the improvements obtained with DACME. We can see that, when DACME is not used, the average end-to-end delay for some of the sources reaches very high values (close to 500 ms). Such high values are not desirable, and are related to high congestion states. If DACME is used, the end-to-end delay values are always kept low (usually less that 10ms), though we observe a periodic variability. Such variability is directly related to the probing process, periodically repeated every 3 seconds (plus jitter); this is something inherent to the architecture proposed, and so cannot be avoided. We should point out that the magnitude of such variability depends on the path congestion, as well as on the application's data rate.

We now proceed to analyze DACME performance in terms of several parameters. The results are presented in table 8.4. We observe that DACME traffic generated about 32 to 64 kbit/s of overhead. In this scenario only a few probe packets were lost, and therefore probe replies were sent and reached the source successfully every time. We also observe that the average cycle time is intimately

Table 8.5: Probe set size incidence for the different sources

Probe set size	Source 1	Source 2	Source 3	Source 4	Source 5
2	12.50%	28.00%	53.57%	61.54%	7.69%
3	4.17%	52.00%	42.86%	26.92%	3.85%
4	0.00%	16.00%	3.57%	7.69%	7.69%
5	83.33%	4.00%	0.00%	3.85%	80.77%

related to the number of hops, as expected; however, congestion also plays an important role (e.g., see differences between source 2 and 3). Relatively to the inter-cycle time, it is kept close to 3 seconds as desired.

Another important issue is to assess how many probes had to be sent each time in order to reach a conclusion on weather to accept a connection or not. With that purpose in mind, we elaborated table 8.5 were we show the incidence, for each source, of the number of probes required. We can see that DACME agents at sources 2, 3 and 4 were usually able to reach a decision with only 2 or 3 probes. On the contrary, DACME agents at sources 1 and 5 usually found that the available bandwidth was very close to the desired value, which often required using the maximum number of consecutive probes allowed (5).

If we take into account the results of both tables presented, and also the fact that in figure 8.22 activity periods are kept very stable, we conclude that the strategy followed in algorithm 2 (maintain the previous path state when no decision can be taken) is adequate and promotes stability for both users and the MANET as a whole.

8.4.2 Impact of the Reserved bandwidth parameter

Depending on the specific routing protocol used, it will require more or less bandwidth to operate properly. As shown in section 7.2, restricting the amount of radio resources available for the routing protocol will cause it to malfunction. Such malfunctioning usually has an impact on both routing overhead and packet delivery rate.

When using DACME to assess the available bandwidth in the network, we should take into consideration that enough bandwidth should be preserved for the routing tasks. Moreover, most applications do not generate a constant bitrate data stream. This means that we should be able to handle these traffic fluctuations, avoiding running out of resources.

Another issue that we should also take into consideration is that most of the traffic currently flowing in the Internet is best effort, mostly TCP (web, FTP, mail, peer-to-peer, etc.). Assuring that QoS traffic does not consume all the resources available is also important; users will hardly accept any QoS framework which makes the MANET unusable in terms of best effort traffic. So, we also desire to study the impact of varying this additional amount of bandwidth reserved in terms of global throughput for best effort traffic.

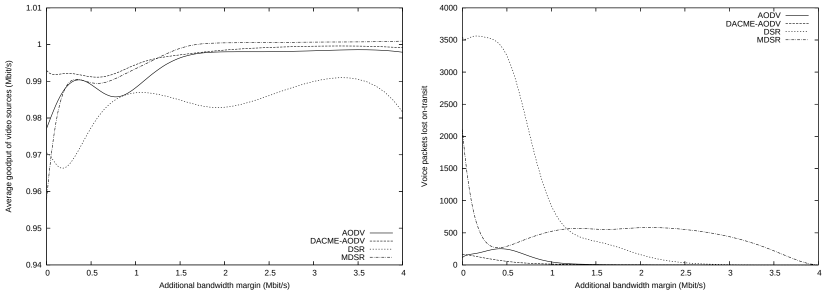


Figure 8.24: Video goodput (left) and voice drops (right) for the different routing protocols tested

With these purposes on mind, we use ns-2 to simulate a MANET environment where 50 nodes move at a constant speed of 5 m/s according to the random waypoint mobility model; the scenario is sized 1900×400 squared meters. Radio interfaces are IEEE 802.11g/e enabled, and the radio range is 250 meters; this leads to an average of 4 hops between nodes.

The protocols tested are AODV, DSR and MDSR (exposed in section 6). We also include DACME-AODV, a combination between DACME and AODV that improves performance, as described in section 8.2.5.

Relatively to traffic, we have four FTP/TCP background sources that are active during the entire simulation period. Concerning the data sources under study (regulated by DACME), these consist of four video streams and three voice streams. The video sources send CBR/UDP traffic at 1 Mbit/s using 512 byte packets. Voice sources are VoIP streams simulated using a Pareto On/Off distribution with both burst and idle time set to 500 ms. The shaping factor used is 1.5, and the average data rate is of 100 kbit/s. Relatively to start and end times for the different sources, the first video source is started at the beginning of the simulation, and then every 15 seconds a new data source becomes active, alternating between voice and video sources. Each source is active for two minutes.

In our experiments we vary the amount of additional bandwidth reserved in a range from 0 to 4 Mbit/s. In terms of video goodput, figure 8.24 shows that the impact is minimum. The decay in terms of bandwidth for low values of the additional bandwidth margin is not very significant: below 4% for all routing protocols, and below 1% for DACME-AODV. Notice that, contrarily to the DSR routing protocol, MDSR achieves a stable operation point close to the maximum value with low additional bandwidth margins, while DSR performs worse.

We should also point to the fact that, as the amount of additional bandwidth reserved increases, the amount of traffic admitted into the MANET decreases. Such behavior is also responsible for the performance improvements experienced when the additional bandwidth margin approaches 4 Mbit/s.

In terms of voice packet drops, we observe that both DSR and MDSR are prone to cause much more losses than AODV or DACME-AODV. By taking a look at this picture we find that AODV-based protocols should operate with an additional

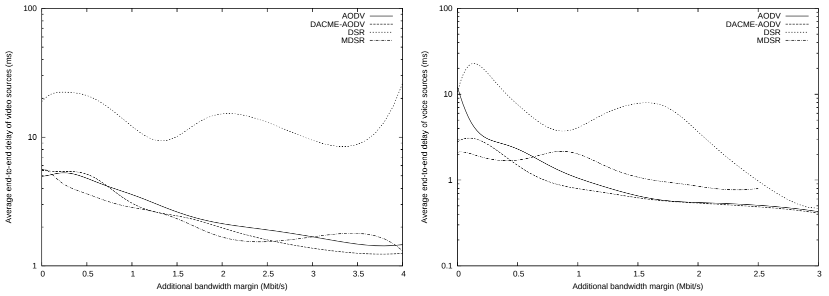


Figure 8.25: End-to-end delay for video (left) and voice data (right) for the different routing protocols tested

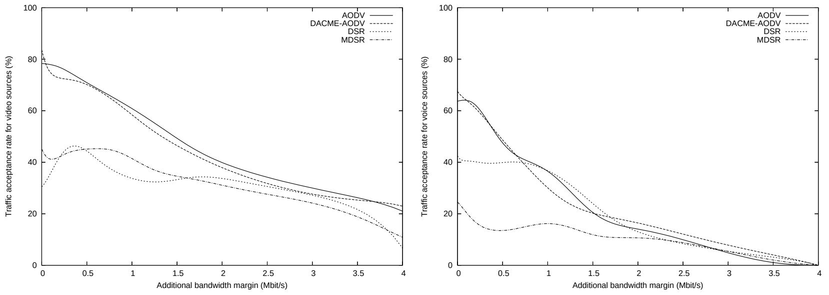


Figure 8.26: Percentage of admitted traffic relative to video (left) and voice (right) traffic for the different routing protocols tested

bandwidth margin above 0.75 Mbit/s. The DSR routing protocol requires a higher bandwidth margin, and we consider that the minimum value should be of about 1.25 Mbit/s.

Concerning end-to-end delay, we verify that, again, both AODV versions and MDSR perform better than DSR (see figure 8.25). This shows that the changes performed on DSR to obtain MDSR were relevant from the point of view of QoS streams, and that the interoperability with DACME is higher for MDSR than for DSR.

Relative to the results for the traffic acceptance rate, figure 8.26 clearly puts in evidence the impact of choosing different values for the additional bandwidth margin reserved. It is interesting to notice that, despite AODV and DACME-AODV admit more traffic into the network, the quality of service experienced by that traffic is higher than with DSR and similar to MDSR's. That fact, along with the voice loss rate results presented before, gives us a hint on which routing protocols offer better performance when operating in conjunction with DACME.

If we now analyze the amount of background TCP traffic flowing in the MANET (see figure 8.27) we observe that DSR and MDSR, compared to both AODV versions, reduce the available bandwidth for best effort traffic. Though the trend

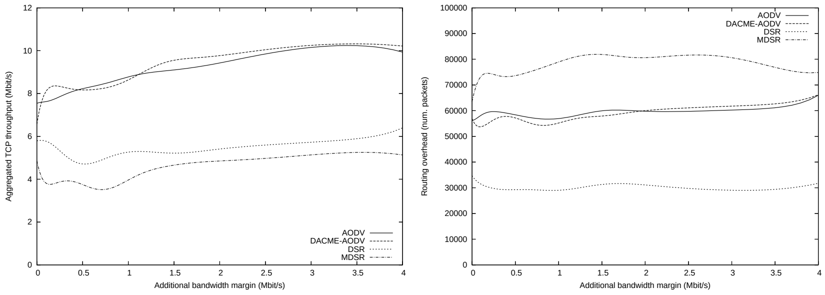


Figure 8.27: Aggregated TCP throughput (left) and routing overhead (right) for the different routing protocols tested

is to increase with increasing reserved bandwidth values, we find that MDSR is the routing protocol which penalizes more heavily the bandwidth available for best-effort traffic, though TCP traffic is not being split through different routes.

To conclude this analysis we now study the results in terms of routing overhead. Figure 8.27 (right) shows that DSR is very efficient, generating a much lower overhead than the rest. AODV and DACME-AODV have a very similar performance, since it is actually the same routing protocol (the interaction consists of DACME interpreting AODV traffic, and not the opposite). Concerning MDSR, it generates a greater overhead than the rest as expected (see section 6.4).

Overall, the results found in this section are somehow unexpected since the different routing protocols do not misbehave ever, even when the additional bandwidth margin reserved is reduced to zero. This can be explained taking into account two different facts. The first one has to do with the MAC Access Category used; since DACME's probes are sent with Video priority, the measurements made are always conservative (remember that, as exposed in section 7.1, data sources using the Voice AC are able to achieve a higher bandwidth than sources using other ACs for a same period of time). Moreover, the chances that QoS data sources fully occupy the wireless medium on a certain area is very low - taking video sources as an example, each data source generates either 1 Mbit/s or nothing, which is a coarse level of granularity.

In summary, the analysis made in this section has shed some light into basic but important issues, such as the effectiveness of the different routing mechanisms tested, and the impact of varying the reserved bandwidth parameter. We studied the effects of having different reserved bandwidth values on the QoS experienced by applications, the amount of QoS-traffic admitted into the network and the overall capacity remaining for best effort traffic sources. We now proceed with a different evaluation, assessing the effectiveness of DACME when the amount of QoS traffic increases.

8.5 DACME's support for bandwidth constrained applications

In this section we will assess the effectiveness of DACME supporting applications with bandwidth constraints in a typical MANET environment, and under different degrees of background congestion. With that purpose we take the scenario described in section 8.4.2 for studying.

Relatively to DACME-regulated sources of traffic, we again use the same setup as before; it consists of four CBR/UDP video sources generating a data rate of 1 Mbit/s, and three VoIP sources generating and average data rate of 100 kbit/s.

In addition to DACME-regulated sources of traffic, we also use four background sources that generate negative-exponentially distributed traffic. Contrarily to section 8.4.1, we do not set the same data rate to the different MAC Access Categories. The difference is due to the need to perform routing tasks; since routing traffic is set to the Voice AC (highest), we limit the amount of traffic in that AC to avoid routing misbehavior. So, 50% of the data generated belongs to the Video AC, and the Best effort and Background ACs receive a share of 25% each.

In the next section we will perform these experiments with the AODV routing protocol and with DACME-AODV. We compare the results obtained to a no-DACME solution (no source benefits from DACME). Later we make similar experiments for the DSR and MDSR routing protocols.

8.5.1 Results for the AODV routing protocol

In section 8.4.2 we found that the amount of reserved bandwidth had an impact on the overall performance, as well as on the amount of bandwidth available for background traffic. So, for the experiments made on this section, as well as for later experiments using AODV, we set the amount of additional bandwidth reserved to 0.75 Mbit/s since we consider that it offers a good trade-off between performance and session blocking.

Figure 8.28 shows the improvements - in terms of video goodput and voice packets dropped - obtained by using DACME or DACME-AODV; we compared these results to a solution where DACME is not used (turned off). We can observe that, when DACME is not used, the average goodput for the different video sources drops steadily with increasing congestion. By using DACME the average goodput is maintained higher (close to maximum). This occurs because sources are only allowed to transmit if the DACME agent verifies that the available bandwidth is enough.

From figure 8.28 we can also observe that DACME-AODV offers a better performance compared to the default DACME implementation in terms of voice packet drops, while performance in terms of video throughput is maintained at similar levels.

We now proceed to evaluate the performance achieved in terms of end-to-end delay. The results are shown in figure 8.29. As expected, end-to-end delay values tend to increase with increasing background traffic.

In the scenario presented we see that, when using either standard DACME

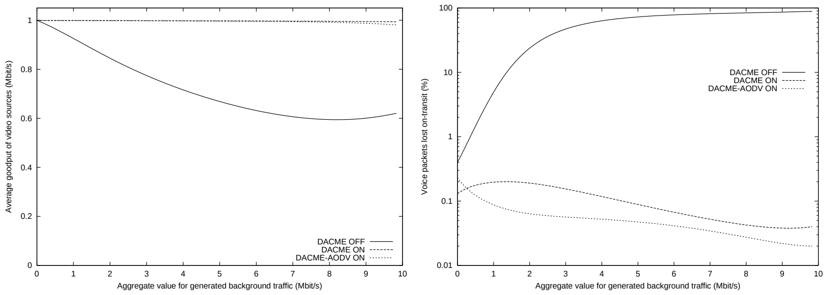


Figure 8.28: Improvements on video goodput (left) and voice drops (right) by using DACME and DACME-AODV

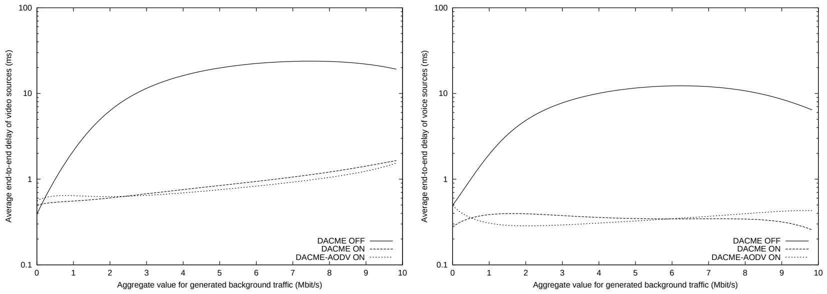


Figure 8.29: Average end-to-end delay values for video (left) and voice (right) sources.

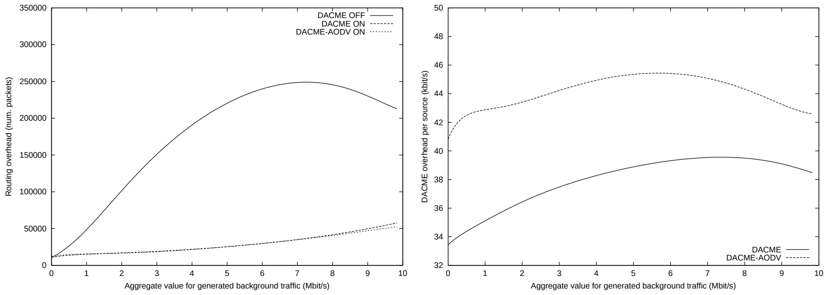


Figure 8.30: Routing overhead and DACME overhead

or DACME-AODV, the end-to-end delay values for both video and voice sources were lower than with DACME turned off. We also observe that the AODV-aware version of DACME tends to cause slightly higher delay values that its non-aware counterpart. This is mainly due to an increase of the average probing frequency, which interferes with data traffic by increasing delay.

An interesting way to gain further insight on the benefits of DACME in MANET environments is to analyze the stability in terms of routing overhead, or the lack of it. In figure 8.30 we show the variation in terms of total routing packets when varying the amount of background traffic generated. It shows that, without the admission control mechanism offered by DACME, the routing protocol misbehaves due to congestion related effects. Such problem is not new, and we have discussed it in detail in section 7.2.

Relatively to DACME's overhead, we observe that it is maintained at low levels. In fact, the average DACME overhead per source does never reach 50 kbit/s, a very reasonable value taking into consideration that we are following a probe-based approach.

The results presented until now allow us to conclude that both AODV-aware and unaware versions of DACME allow achieving performance improvements compared to a solution without DACME. In terms of the differences between these two solutions, we observe that the AODV-aware version offers greater stability at the cost of slightly higher end-to-end delay values. To further evidence the differences between both DACME versions, we show in figure 8.31 the results achieved in terms of percentage of admitted traffic. We can see that the main differences occur when the congestion levels are low. In these situations a quick reaction to newly found routes avoids blocking traffic for large periods if the new route can sustain the desired traffic rate.

It is interesting to notice that, as congestion increases, the amount of video traffic admitted decreases at a steady rate, contrarily to voice traffic. This is due to the fact that video streams require much larger bandwidth shares.

We now proceed to study in more depth the behavior under low, moderate and high congestion levels. In terms of aggregated value for generated background traffic, these congestion levels map into the values: 0.65 Mbit/s, 2.3 Mbit/s, and 6.5 Mbit/s respectively.

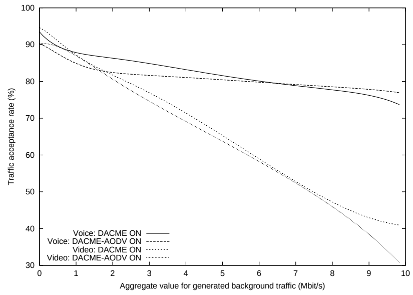


Figure 8.31: Percentage of admitted traffic using both DACME versions at different congestion levels

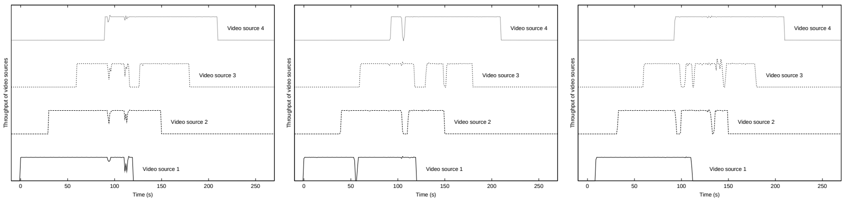


Figure 8.32: Throughput variation with time for the video sources using default, DACME and DACME-AODV solutions under low congestion

8.5.1.1 Low congestion environment

At this congestion level we observe that all sources, both voice and video, enjoy of enough bandwidth for transmission most of the time (see figure 8.32). Therefore, the DACME agent blocks traffic only occasionally; also, routing tasks cause very few packet drops.

In such an environment it is more meaningful to observe the performance in terms of end-to-end delay. In figure 8.33 we show the end-to-end delay results for the video sources. We can observe that results are very similar, with DACME-AODV performing better than the default DACME agent. Relatively to the Voice sources, figure 8.33 shows that the DACME-AODV agent offers the best results essentially by eliminating some peaks on the end-to-end delay.

Overall we can conclude that, by introducing DACME or DACME-AODV agents in low-congestion environments, they will not cause performance to decrease even though the need for them is minimal. On the contrary, we observe that performance is maintained and in some cases improved.

8.5.1.2 Moderate congestion environment

We will now analyze what occurs when the available bandwidth on the MANET environment is not enough for all video sources to transmit uninterruptedly, though it is enough for some of them to do so. In this situation the performance should be

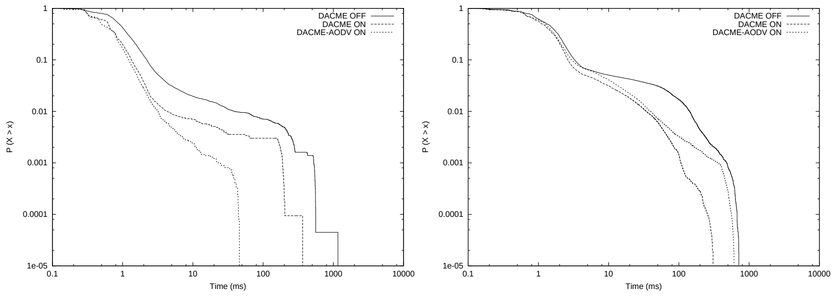


Figure 8.33: End-to-end delay variation with time for the voice (left) and video (right) sources using default, DACME and DACME-AODV solutions under low congestion

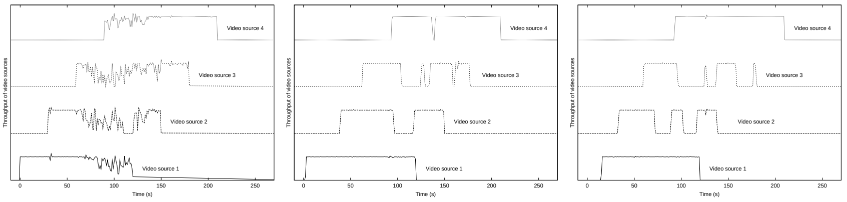


Figure 8.34: Throughput variation with time for the video sources using default, DACME and DACME-AODV solutions under moderate congestion

affected when there is a route change, and so the DACME agent must act according to the new network conditions. As shown in figure 8.34, the throughput of video sources under moderate congestion is constantly suffering changes if DACME is not used.

When using DACME, though, we find that video sessions (when active) achieve very steady throughput values. The benefits of DACME are further evidenced if we take a look at the number of packets dropped in the network (see table 8.6); it becomes clear that using DACME offers much better results, avoiding unnecessary resource consumption.

If we now study the behavior in terms of end-to-end delay, we see that we obtain significant improvements by using DACME. DACME-AODV offers the best results in terms of video traffic (see figure 8.35), while for voice traffic the results are very similar for both DACME versions, and significantly better than the no-DACME solution.

8.5.1.3 High congestion environment

When the MANET environment is highly congested, sources generating high data rates should have few chances of transmitting. If no type of admission control is performed, though, the network congestion will increase even more and these

Table 8.6: Number of video packets dropped in the network

	DACME off	DACME	DACME-AODV
Video source 1	4295	19	40
Video source 2	11176	66	48
Video source 3	14079	118	41
Video source 4	4945	10	35
Total loss	34495	213	164
Loss (%)	29,43	0,92	0,78

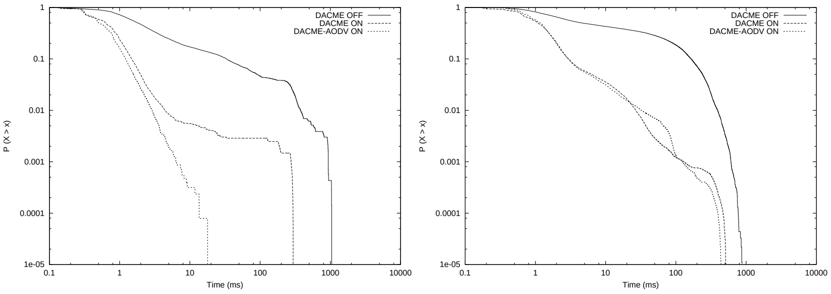


Figure 8.35: End-to-end delay variation with time for the voice (left) and video (right) sources using default, DACME and DACME-AODV solutions under moderate congestion

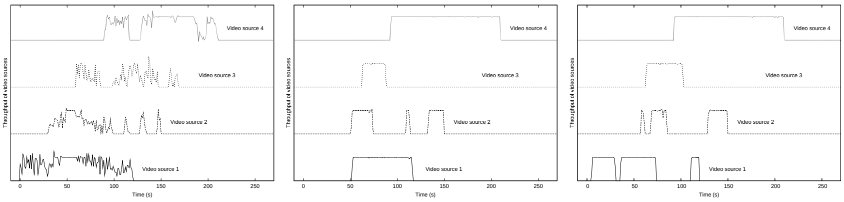


Figure 8.36: Throughput variation with time for the video sources with DACME turned off under high congestion

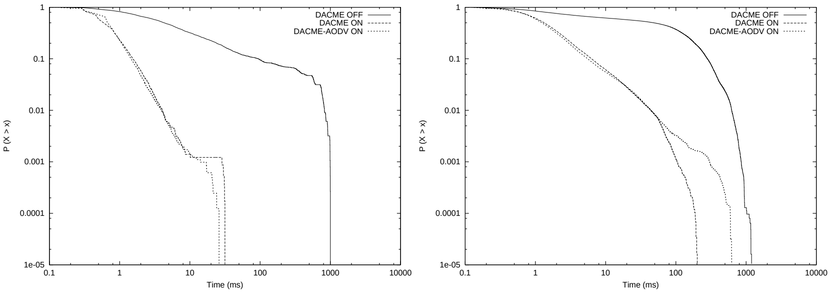


Figure 8.37: End-to-end delay variation with time for the voice (left) and video (right) sources using default, DACME and DACME-AODV solutions under high congestion

high data rate sources will consume resources unnecessarily. This is what occurs in the current situation, where we observe (see figure 8.36) that the different video sources can barely succeed in transmitting any data to the destination.

When DACME is used (either default and AODV-aware implementations) we find that video sessions, though often blocked, are able to transmit with very high throughput values when active.

Concerning end-to-end delay results, these are depicted in figure 8.37.

We can see that, by introducing DACME or DACME-AODV, the end-to-end delay values for both voice and video traffic are significantly improved. These results also show that, even by introducing probe tests in environments where saturation is already high, the performance of voice streams does not suffer degradation.

The results found in this section lead us to conclude that using DACME can clearly avoid the waste of resources by interrupting communication when the minimum QoS requirements are not met. We also conclude that, when MANET mobility is high and the congestion level is such that not all QoS streams can be accommodated, we should expect to find an on/off behavior often. Such behavior is intimately related to the chosen policy of not performing any kind of resource reservation, but instead to adapt to whatever resources available at each measurement action.

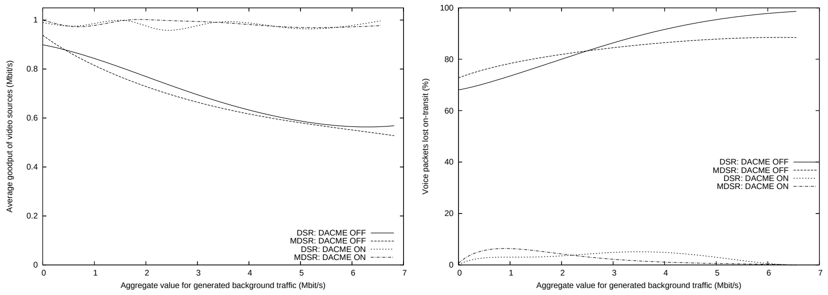


Figure 8.38: Improvements on video goodput (left) and voice drops (right) by using DACME

8.5.2 Results for the DSR and MDSR routing protocols

In this section we present the results obtained by doing similar experiments with the DSR and MDSR routing protocols. As referred in section 8.4.2, we need to choose the extra amount of bandwidth reserved to allow room for routing traffic and DACME probes from different sources; for the routing protocols studied in this section (DSR and MDSR) we found that an adequate value for the additional bandwidth reserved should be about 1.25 Mbit/s to achieve good performance.

Figure 8.38 shows the improvements in terms of video goodput and voice packets dropped by using DACME. We observe that, when DACME is not used, the average goodput for the different video sources drops steadily with increasing congestion. By using DACME the average goodput is maintained much higher for both DSR and MDSR; in fact, we verify that when DACME is active MDSR performs even better than DSR, which is a strong indication that the admission control strategy adopted for DACME can operate in conjunction with multipath routing protocols without performance decay. Relatively to the improvements introduced by DACME, these occur because sources are only allowed to transmit if the DACME agent verifies that the available bandwidth is enough.

From figure 8.38 we can also observe that, by using DACME, the number of voice packets lost is greatly reduced. Again, the combination of MDSR and DACME is the best one, achieving a very low packet loss rate always.

We now proceed to evaluate the performance achieved in terms of end-to-end delay. The results are shown in figure 8.39.

Under these conditions we see that, by using DACME, the end-to-end delay values for both video and voice sources were lower using either DSR or MDSR. In terms of video traffic, it is interesting to notice that MDSR performs better than DSR with and without DACME; this shows that the traffic splitting strategy used in MDSR offers advantages in terms of end-to-end delay despite the fact that sometimes part of the traffic traverses paths with more hops. Concerning voice traffic, the end-to-end delay results also show that both DSR and MDSR clearly benefit from DACME. The difference of curve shapes between DSR and MDSR is related to the degree of voice traffic accepted into the network, to the contention

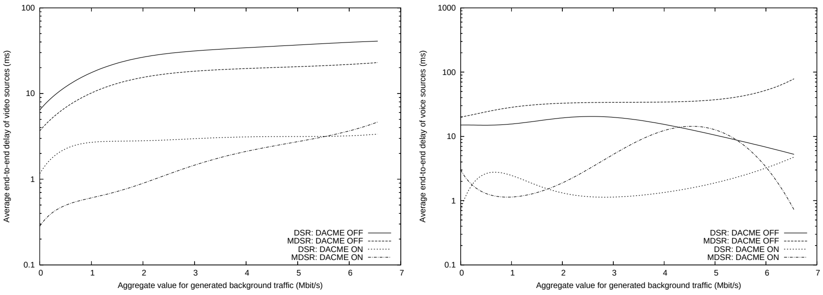


Figure 8.39: Average end-to-end delay values for video (left) and voice (right) sources.

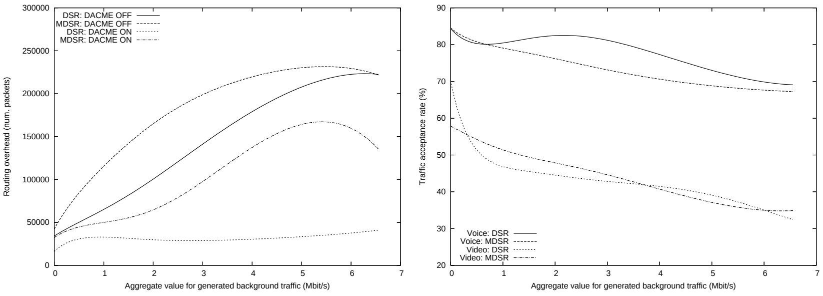
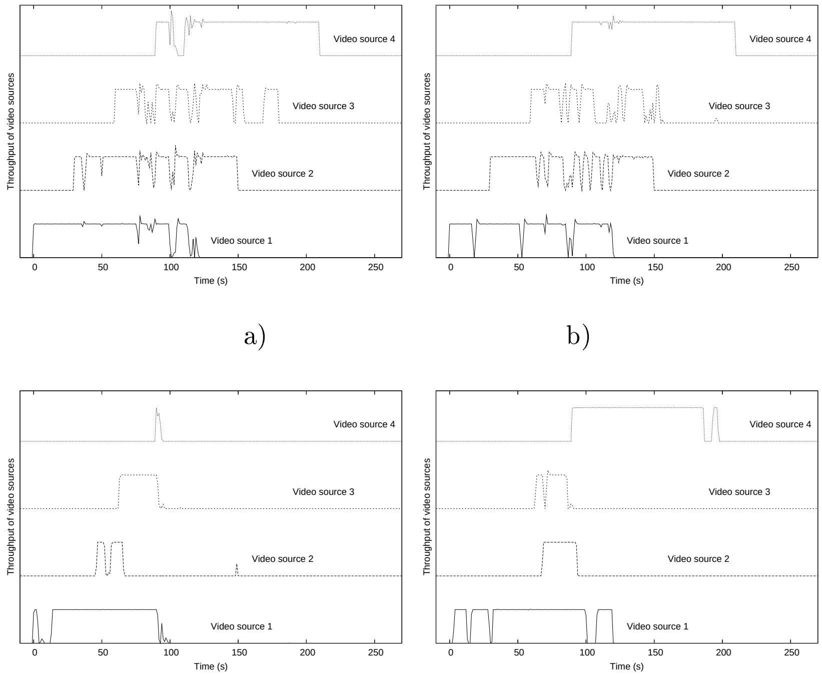


Figure 8.40: Routing overhead (left) and percentage of admitted traffic (right) at different congestion levels

between data packets and routing packets, and to the fact that MDSR splits traffic through different paths.

One of the main differences between DSR and MDSR is related to the amount of routing overhead generated. MDSR's route discovery mechanism and, to a lesser extent, traffic splitting through different routes result in an increased routing overhead. Therefore, we expect to observe this difference when analyzing the routing overhead generated in our experiments. In figure 8.40 we show the variation in terms of total routing packets when varying the amount of generated background traffic. We observe that MDSR does in fact generate a higher amount of routing traffic than DSR with or without DACME. However, it is important to notice that by using DACME we are able to maintaining the routing overhead stable when congestion increases, avoiding the routing misbehavior problem we discussed in [CPM05b].

Another issue that deserves attention is related to the acceptance rate experienced by voice and video traffic. Voice sources generate much lower data rates, and so we expect the amount of voice traffic admitted into the MANET to be higher than the amount of video traffic. In figure 8.40 we show the differences between both when using either DSR or MDSR. We can see that, effectively, the higher-



a)

b)

c)

d)

Figure 8.41: Throughput variation with time for the video sources using a) DSR, b) MDSR, c) DSR+DACME and d) MDSR+DACME under low congestion

rate video sources are more penalized by congestion, experiencing more frequent cut-offs than voice traffic which is less bandwidth demanding.

We will now proceed to study in more depth the behavior under low, moderate and high congestion levels.

8.5.2.1 Low congestion environment

Under low congestion the amount of background traffic is relatively low, which results in a greater interaction between the distinct DACME-regulated sources. In figure 8.41 we show that in this situation DACME already offers benefits, maintaining steadier levels of throughput for all video sources with both DSR and MDSR. It is interesting to notice that when DACME operates in conjunction with DSR it behaves in a conservative manner, blocking connections often; the MDSR plus DACME solution offers better results by increasing the total time of activity.

In terms of end-to-end delay, figure 8.42 shows that DACME offers important improvements for both DSR and MDSR routing protocols, even when the congestion on the MANET is relatively low. We also observe that the best performing solution for the highest share of traffic is DSR with DACME; we consider that this is due to the increased number of hops that part of the traffic has to go through

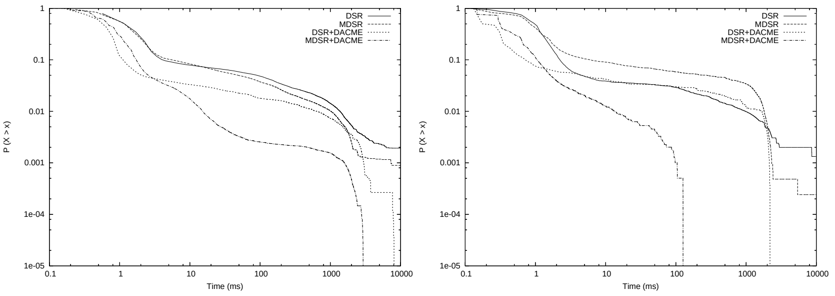


Figure 8.42: End-to-end delay variation with time for the video (left) and voice (right) sources under low congestion

during some periods when relying on MDSR. However, the MDSR plus DACME solution is more effective in reducing the amount of packets that reach the destination with very high delay values, especially for Voice data where none of the packets arrives with a delay above 150 ms.

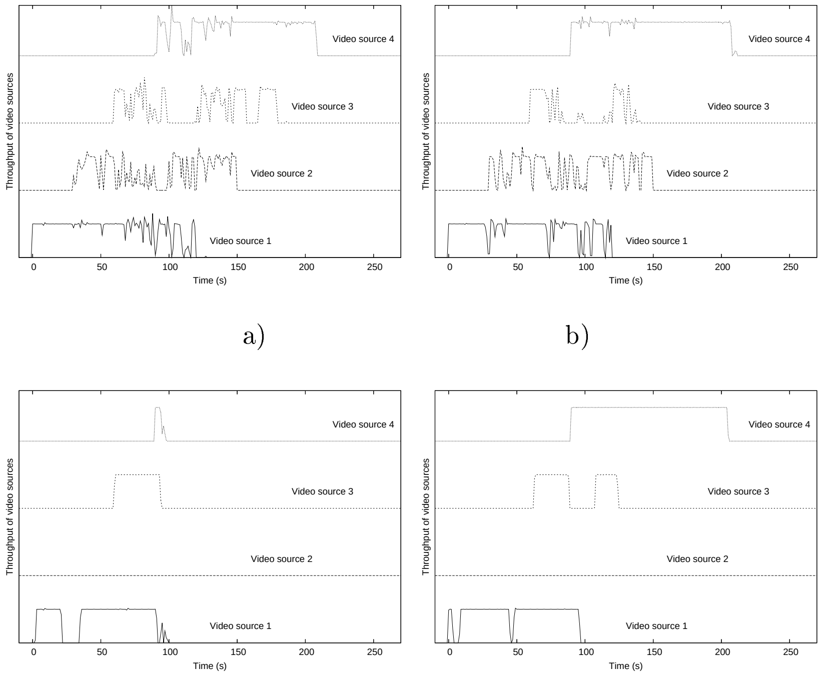
The results found until now show that DACME is also effective when used in conjuction with a multipath routing protocol. We will now proceed with our evaluation under moderate congestion, and verify if the effectiveness of DACME with MDSR persists.

8.5.2.2 Moderate congestion environment

In this environment the level of background congestion is enough to cause important losses to both video and voice data streams (see figure 8.43). In this situation the use of DACME brings even more benefits than in the previous scenario with low congestion, not only because the video throughput is maintained at much steadier levels, but also because the number of packets lost on-transit is greatly reduced.

In table 8.7 we show the number of packets lost in the network due to multiple factors such as routing, full queues or MAC related drops. We observe that MDSR performs much better than DSR, reducing losses by an order of magnitude. Such improvement is essentially related to the traffic splitting algorithm that MDSR uses which, as shown in chapter 6, is able to reduce mobility-related packet losses. These loss levels are quite acceptable for a MANET environment and, compared to the non-DACME results, show that by using DACME we also save energy resources on MANET nodes by only forwarding packets when the chances of reaching the destination are high.

If we now study the behavior in terms of end-to-end delay (see figure 8.44), we can again notice DACME's effectiveness. In terms of video traffic we see that DSR and MDSR combined with DACME offer similar results, though MDSR performs slightly better. Relatively to audio traffic, the performance achieved by using DSR plus DACME slightly surpasses that achieved with MDSR plus DACME, though it is prone to generate more packets with very high delay values.



a) b) c) d)

Figure 8.43: Throughput variation with time for the video sources using a) DSR, b) MDSR, c) DSR+DACME and d) MDSR+DACME under moderate congestion

Table 8.7: Number of video packets dropped in the network

	DSR	MDSR	DSR+DACME	MDSR+DACME
Video source 1	3036	3292	488	60
Video source 2	11249	9229	0	1
Video source 3	13745	21821	0	7
Video source 4	3039	614	144	0
Total loss	31069	34956	632	68
Loss (%)	26,5	29,82	2,17	0,11

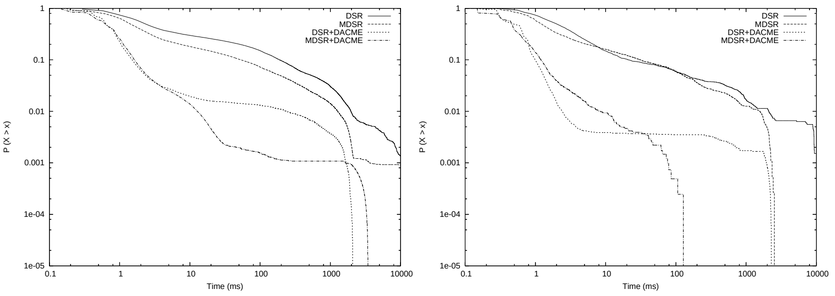


Figure 8.44: End-to-end delay variation with time for the video (left) and voice (right) sources under moderate congestion

The results found in this section further sustain the applicability of the DACME admission control algorithm with a multipath routing protocol, showing no signs of misbehavior or poor performance. We now proceed to analyze in detail the performance under high congestion.

8.5.2.3 High congestion environment

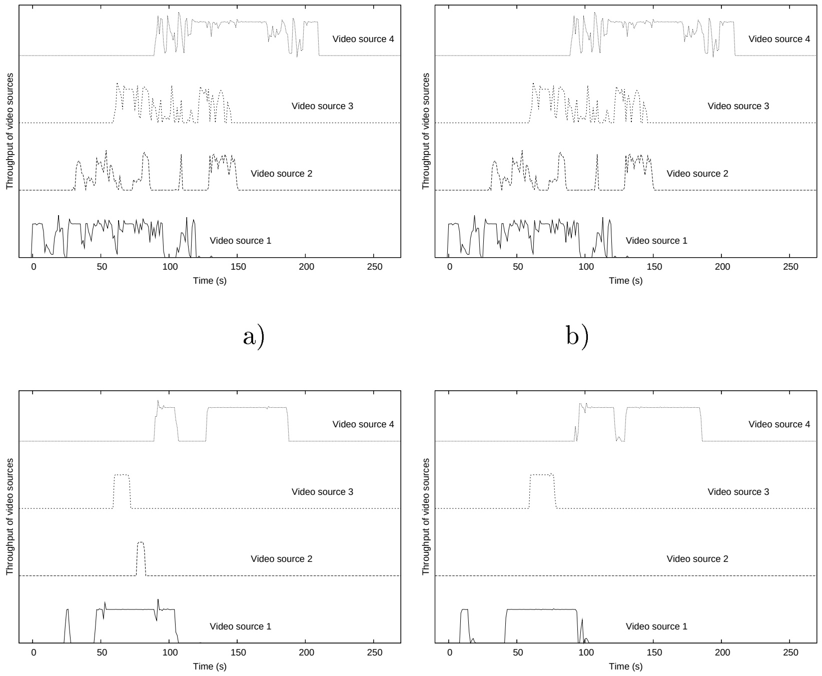
When the MANET environment is highly congested, it is especially important for sources generating high data rates to avoid transmitting. If no type of admission control is performed, though, the network congestion will increase even further and these high data rate sources will consume resources unnecessarily. This is what occurs in the current situation where we observe (see figure 8.45) that the different video sources can barely succeed in transmitting data to the destination when DACME is not used.

When DACME is used we see that the situation changes: most of the time the video sources are not allowed to transmit, and when they are allowed to do so the throughput is maintained reasonably steady.

In terms of end-to-end delay figure 8.46 shows that, in a similar manner to what was observed in the previous section, the performance for DSR and MDSR with DACME relative to video traffic is similar, and clearly superior compared to the non-DACME results.

We can also see that introducing DACME provides much better performance to the voice flows, especially with MDSR where the end-to-end delay does not surpass 20 ms.

The results found in this section lead us to conclude that using DACME can clearly avoid the waste of resources by interrupting communication when the minimum QoS requirements are not met. Comparing DSR to MDSR we observe that the effect they have on QoS streams differs; yet, we can definitely affirm that the proposed admission control mechanism is adequate for using with both single path and multipath routing protocols.



a)

b)

c)

d)

Figure 8.45: Throughput variation with time for the video sources using a) DSR, b) MDSR, c) DSR+DACME and d) MDSR+DACME under high congestion

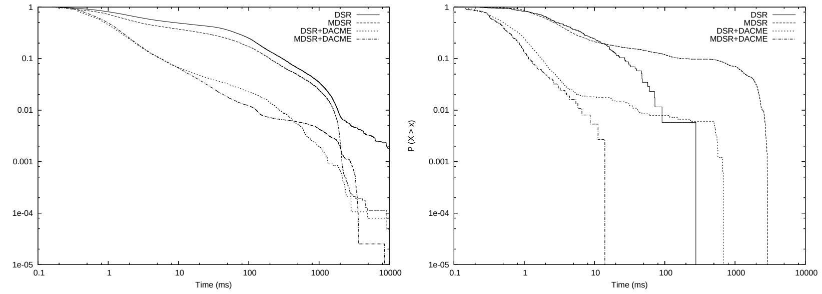


Figure 8.46: End-to-end delay variation with time for the video (left) and voice (right) sources under high congestion

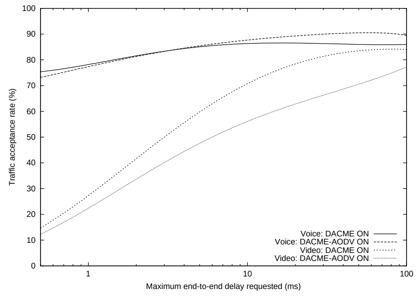


Figure 8.47: Traffic acceptance rate values when varying the maximum end-to-end delay requested

8.6 DACME's support for delay-bounded applications

In this section we will assess the effectiveness of DACME in supporting delay-bounded applications. Our tests will focus on the AODV routing protocol since it was found to offer the best results in previous sections. Also, DACME is not capable of performing delay measurements when the routing protocol splits traffic through different paths, which excluded MDSR from the tests performed in this section.

For our study we set the aggregate background congestion to a fixed value to proceed with our experiments. The chosen value is of 2.3 Mbit/s, and it is maintained throughout the simulations.

The simulations made are similar to those performed for bandwidth-constrained applications. The only difference is that we now notify the DACME agent that the applications are also delay bounded, setting different values for the maximum end-to-end delay. In our experiments these values vary between 0.1 and 100 ms.

In the course of our simulations we observed that, when applying a maximum delay threshold of 0.1 ms, no video or voice traffic was accepted into the network. In figure 8.47 we present the traffic acceptance rate curves when varying the maximum delay settings. We observe that the impact of imposing delay requirements is more pronounced on video sources, being that the voice traffic only varies slightly; as expected, when demanding relatively high values for end-to-end delay (100 ms) the amount of traffic accepted for both video and voice sources is close to the one found when applying bandwidth-constraints only.

We now proceed to measure the average end-to-end delay experienced by the video and voice sources. In figure 8.48 we present the results found; we observe that the average end-to-end delay values are always well below the threshold defined, as desired. We find that the average end-to-end delay experienced by the video sources increases steadily with increasing delay thresholds; for the voice sources, though, we only appreciate slight variations. This phenomena occurs because the MAC layer parameters associated with the Voice Access Category do not allow

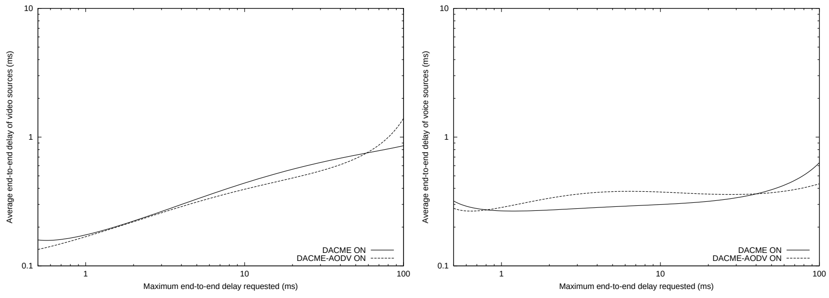


Figure 8.48: Average end-to-end delay variation when varying the maximum end-to-end delay requested for video (left) and voice (right) traffic

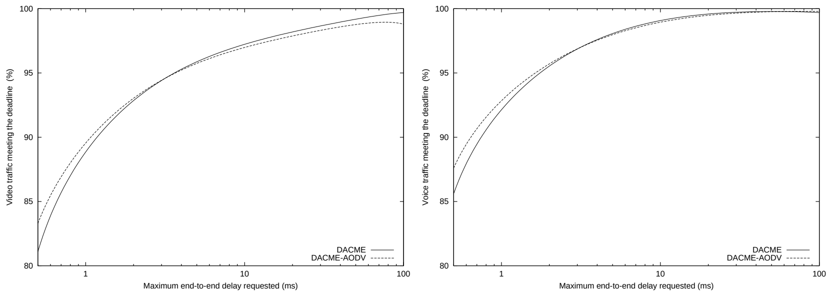


Figure 8.49: Percentage of traffic meeting the end-to-end delay deadline for video (left) and voice (right) traffic

much margin for such variations.

If we now take into consideration the percentage of traffic that meets the predefined maximum value for end-to-end delay, we observe that voice traffic meets the requirements more strictly than video traffic (see figure 8.49). These results also allow measuring the effectiveness of DACME's architecture in complying with the end-to-end requirements imposed. We find that, although DACME agents only re-assess the end-to-end delay values every 1.5 seconds when traffic is flowing, this strategy offers good results even when the scenario is characterized by an important degree of mobility: more than 80% of the accepted traffic meets the deadline always.

To conclude the evaluation we now analyze the average overhead per source introduced by DACME. In section 8.5 we found that, at the selected degree of congestion (the aggregated background traffic is 2.3 Mbit/s), DACME's overhead was found to be around 37 and 43 kbit/s for DACME and DACME-AODV, respectively. In figure 8.50 we observe that introducing additional probes to measure end-to-end delay does not have a significant impact on overhead. In fact we find that, when the requested end-to-end delay is low, DACME's overhead is inferior to the one found without delay constraints. This occurs because sometimes the delay

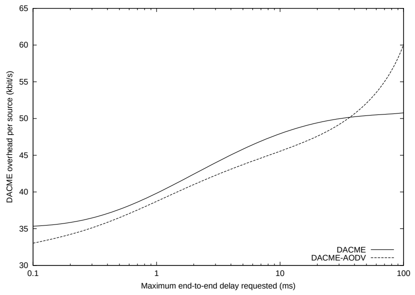


Figure 8.50: DACME overhead when varying the maximum end-to-end delay requested

requirements allow reaching a *deny flow* decision without requiring any measurements to decide about bandwidth.

Once we reach relatively high values for the requested end-to-end delay we find that the overhead, compared to the bandwidth-constrained solution, is increased by 14 and 17 kbit/s respectively; we consider that these are quite acceptable values.

We will now proceed by doing a similar analysis in the scope of jitter bounded applications.

8.7 DACME's support for jitter-bounded applications

In the previous section we could appreciate the effectiveness of DACME to support applications with both bandwidth and end-to-end delay constraints. In this section we take a final step to evaluate the effectiveness of DACME in supporting applications with bandwidth, delay and also jitter requirements. With this purpose we maintain all the simulation parameters used in the previous section, fixing the value for the maximum end-to-end delay requested at 10 ms. We now impose different requirements for jitter, with values ranging from 0.1 ms up to a maximum value of 10 ms.

Relatively to traffic, we have video sources generating CBR traffic and voice sources generating traffic according to a Pareto on-off distribution. Since, from the destination point of view, it is only meaningful to assess the jitter of traffic with a constant packet arrival rate, the results presented in this section refer only to video traffic; voice sources are also included as before, but they only have bandwidth and delay constraints.

In figure 8.51 we show the variation in terms of accepted video traffic as the maximum jitter allowed increases. We observe that 0.1 ms is a cut-off value for jitter and that, when the maximum jitter allowed is of 10 ms, the traffic acceptance rate is similar to the one found without jitter constraints.

In terms of mean absolute jitter values figure 8.51 shows that, when plain DACME is used, the mean value is below the limit; however, for DACME-AODV

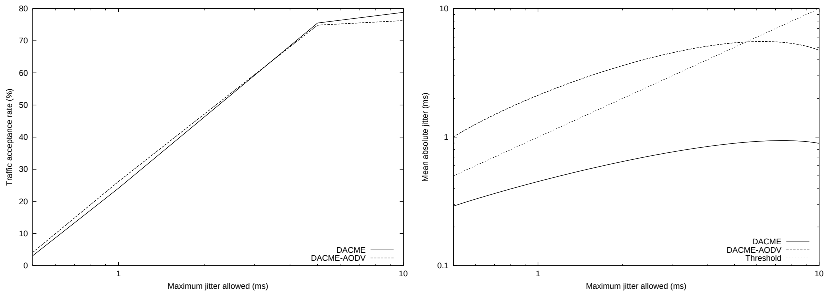


Figure 8.51: Traffic acceptance rate growth (left) and mean absolute jitter variation (right) by increasing the maximum jitter allowed

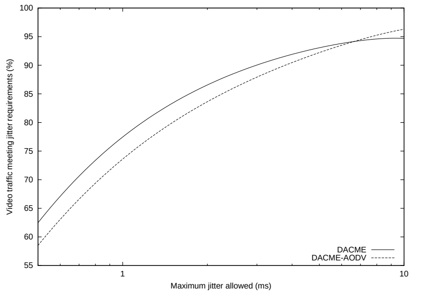


Figure 8.52: Percentage of traffic meeting the maximum jitter value requested

the situation is the opposite: the mean value is higher except when the maximum jitter reaches higher values. This difference can be explained by the fact that DACME-AODV performs more frequent measurements, which interfere with video traffic (especially bandwidth probes). We consider that, if the traffic belongs to the Voice category, the impact of DACME on the application stream's jitter will be much lower.

We now proceed by analyzing the amount of video traffic that meets the jitter requirements imposed. The results of figure 8.52 show that, when the jitter limits are too low, only about 2/3 of the traffic meets these limits. As we relax the jitter constraints the percentage of traffic meeting the requirements increases significantly.

It is interesting to notice that, despite the differences found between DACME and DACME-AODV in terms of mean absolute jitter values (right side of figure 8.51), in terms of percentage of traffic meeting the jitter requirements there is not much difference between both, though plain DACME still outperforms DACME-AODV.

To conclude this section we now study the overhead of DACME when jitter probes are also included. Figure 8.53 presents the results found through simulation; it shows that, when the maximum jitter allowed is very low, the overhead generated

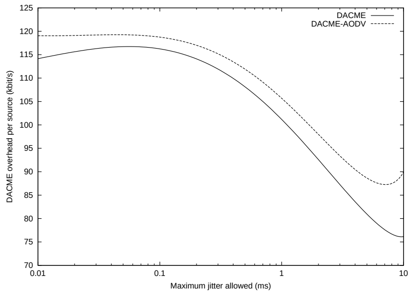


Figure 8.53: DACME overhead by increasing the maximum jitter allowed

by DACME is relatively high. This occurs because the DACME agent will be sending jitter probes every 3 seconds after the delay and bandwidth probes, which consume a considerable amount of bandwidth. As we increase the maximum jitter allowed we find that DACME's overhead is reduced. In fact, as jitter constraints are relaxed, more traffic is admitted into the network; this enables performing jitter measurements based on actual traffic instead of probes. This technique explains the reduction observed.

Relatively to the absolute overhead values, we found in the previous section that DACME and DACME-AODV required and overhead of 47 and 45 kbit/s respectively for a maximum end-to-end delay of 10 ms. Adding jitter constraints causes these values to increase by 76 and 90 kbit/s respectively for the best case, which is a significative but tolerable increase.

8.8 Conclusions

In this chapter we presented DACME, a novel QoS architecture for MANET environments which enables real-time multimedia communication among peers. Our proposal can be easily deployed since it uses distributed admission control techniques and imposes very few requirements on MANET nodes. In fact, intermediate MANET stations only need to have IEEE 802.11e capable interfaces and to handle packets according to the TOS field in their IP header. With our technique we expect to solve some of the problems encountered in previous proposals, accommodating to different paradigms of user cooperation in MANETs.

We described the general functionality of the distributed admission control mechanism proposed, evidencing its relation with the different layers of the network stack.

The core of our architecture consists of DACME, a probe-based distributed admission control mechanism capable of supporting bandwidth, delay and jitter QoS requirements. The probe-based measurements are used by DACME agents to decide whether to admit traffic from an application or not based on its QoS demands and the estimated available resources. Results show that our distributed admission control technique is able to offer reliable end-to-end measurements, and that the time spent in that process is typically low.

By simulating DACME in both static and mobile MANET environments we observed that it behaves in the manner it should, offering clear performance improvements. Simulation results show that the probabilistic admission control technique used in DACME is effective at different levels of congestion, and that delay and jitter constraints are met with a good level of accuracy. We also proved that DACME can be used in conjunction with multipath routing protocols when supporting bandwidth-constrained applications; this was done by testing it in conjunction with the multipath-enabled version of DSR developed in chapter 6.

Overall, we found that DACME improves the performance experienced by users and also avoids wasting MANET resources. We observe that enhancing DACME with routing awareness further improves the performance achieved. Relatively to the overhead introduced by the DACME's mechanism, we found it to be quite low: between 30 and 60 kbit/s, except when jitter support is also required, in which case it can reach values up to 120 kbit/s.

Chapter 9

Overall evaluation of the proposed framework

The purpose of this chapter is to offer an overview of the framework proposed in this thesis. We will show how, starting from currently available technology, we were able to enhance a MANET so as to achieve a system with full QoS support. That system is accomplished by combining IEEE 802.11e technology (MAC layer) with DACME (admission control layer). At the routing layer we offer the possibility of using either the AODV routing protocol, or our enhanced version of the DSR routing protocol if we seek to split traffic through multiple paths, thereby avoiding the effects of mobility.

Our focus is on video streaming applications, and so the results we present always put in evidence the benefits achieved from the point of view of a real-time video stream.

The results will be presented in three steps. In the first step we will show how an adequate use of the IEEE 802.11e technology is able to differentiate QoS traffic from best effort traffic at the MAC level, simultaneously achieving an increase in terms of routing responsiveness. In the second step we will show how we were able to mitigate the effects of node mobility in MANETs by improving the DSR routing protocol by means of enhanced route discovery mechanisms and traffic splitting strategies. In the third and final step we will show how the use of distributed admission control is able to improve the overall QoS support in MANETs. This QoS improvement is mainly related to avoiding packet drops, avoiding routing misbehavior and reducing the delay experienced by applications.

For our experiments we will again focus on a MANET environment simulated with the aid of the ns-2 simulator. For all of our tests we inject into the MANET traces of actual video streaming traffic. Our video sequence of choice is the well known Foreman sequence (see 4.1 on page 62) in the CIF format (352×288 pixels), which is adequate for video-conferencing. We concatenate this Foreman sequence (10 seconds long) several times to obtain a 300 seconds long sequence. The frame rate used is of 30 Hz, and so the total number of frames is 9000. As proposed in chapter 4, we again split each frame into seven slices, which map into seven

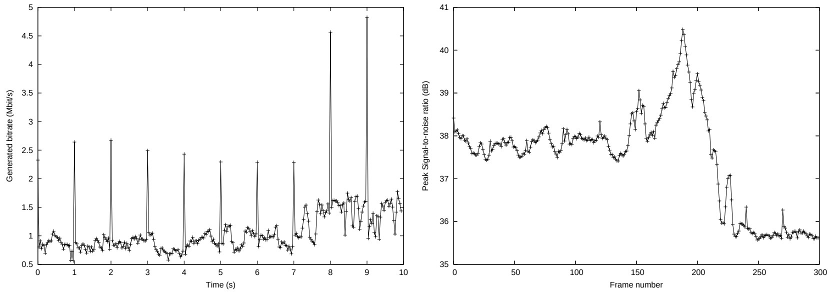


Figure 9.1: 10-second snapshot of the bitrate (left) and PNSR (right) for the H.264-encoded Foreman sequence

packets per frame; therefore, the number of packets generated per second is 210. The global quantization parameters for the sequence where adjusted to a target bitrate of 1 Mbit/s.

On the left side of figure 9.1 we show with great detail the bitrate generated by the H.264 codec for the Foreman sequence during a 10-second period. As it can be seen, every second there is a peak on the instantaneous bitrate generated. This is because we set one I frame to be generated every second to reset error propagation, and so the GOP size is of 30 frames. The remaining frames are predictively coded (P frames). Also, we randomly choose 40 macroblocks of each frame for intra-updating, which means about 10% of the total number of macroblocks per frame (40 out of 396 macroblocks). The remaining codec parameters were tuned according to choices found to be more appropriate in chapter 4.

On the right side of figure 9.1 we show the PSNR variation for this 10-second video sequence. We observe a slight decay of the PSNR value close to the end of the sequence, which is related to a higher motion degree for the sequence during that period. The average PSNR value is of 38.2 dB.

9.1 Improvements obtained through MAC level QoS support

To assess the improvements offered to MANETs by the IEEE 802.11e technology, we devise a set of experiments that will evidence how this technology is able to differentiate QoS traffic from best effort traffic. We use standard IEEE 802.11 technology, whose MAC layer is not QoS-enabled, for comparison.

As referred at the beginning of the chapter, our focus is on the support for real-time video streaming applications. We therefore inject into the MANET the trace of a single video stream with the characteristics referred before. Concerning best effort traffic, we inject a variable number of FTP/TCP sources (bandwidth greedy) and CBR/UDP sources generating data at a rate of 1 Mbit/s. We study the performance variations when increasing the number of best effort sources from 0 to 18. We increase the number of sources with a granularity of three, maintaining

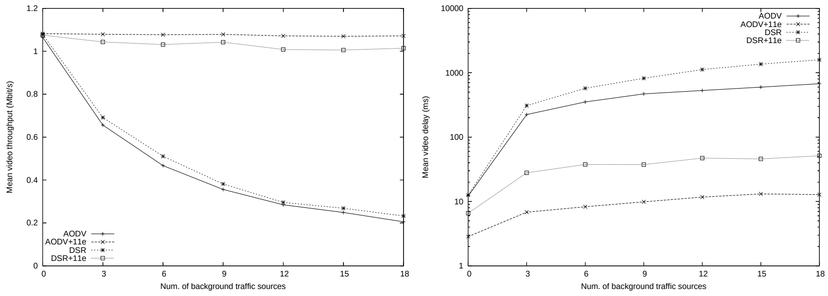


Figure 9.2: Mean values for the video throughput (left) and the video delay (right) varying the number of background traffic sources

a 2 to 1 relationship between the number of TCP and UDP sources, respectively. This means that there are twice as many TCP sources compared to UDP sources in all tests.

Simulations are conducted in a square area sized 870×870 m where the radio range is set to 250 meters. The number of nodes used is 50, and each of them has an IEEE 802.11g radio interface and a routing agent running (either AODV or DSR).

Concerning mobility, it is generated according to the random way-point mobility model. We configure the mobility generation process so that all MANET nodes are constantly moving at a speed of 5 meters per second (no pause times allowed).

Relatively to the simulation process itself, we begin with a 100 second period during which routes between traffic sources and destinations are found; also during that period we start background traffic so that, when the video streaming session begins, it encounters a steady-state MANET environment. After the 100-second warm-up period we start injecting QoS traffic, and each experiment runs for 300 additional seconds. The results obtained are actually drawn from this 300-second period.

For each degree of congestion being evaluated we experiment with 10 distinct scenarios. All the values depicted are, therefore, mean values for these 10 scenarios analyzed.

In figure 9.2 we show the performance experienced by our reference video stream. We can observe that, when using either AODV or DSR, the throughput it maintained close to the maximum with IEEE 802.11e when increasing the number of background traffic sources. If IEEE 802.11e is not used the throughput decays gradually, with loss values up to 80%.

In terms of end-to-end delay we observe similar performance improvements; now, the reduction in terms of delay surpasses one order of magnitude. Also, notice that AODV offers a slightly better performance than the DSR routing protocol.

Concerning the quality of the video session as experienced by the user, we obtain values relative to the expected PSNR values, along with a confidence interval; the degree of confidence used is of 95%. Figure 9.3 shows that PSNR values are

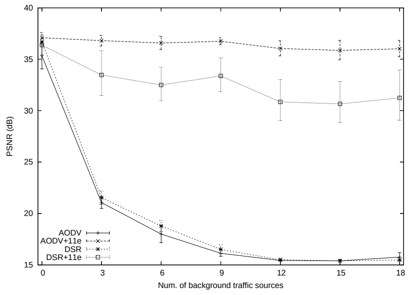


Figure 9.3: PSNR confidence intervals when varying the number of background traffic sources

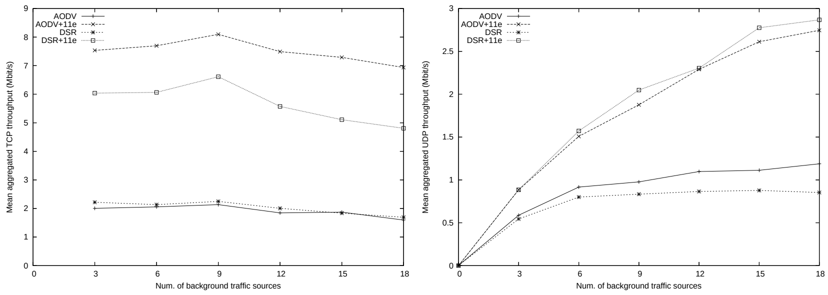


Figure 9.4: Mean values for the TCP throughput (left) and UDP throughput (right) varying the number of background traffic sources

kept at very good quality levels if IEEE 802.11e is used. When using legacy IEEE 802.11 technology, though, PSNR values drop below 25 dB when the first three sources of background traffic are started; this clearly puts into evidence that offering MAC-level QoS support is a *sine qua non* condition to achieve a global QoS framework.

In terms of background traffic (see right side of figure 9.4) we find that, despite its MAC Access Category (best effort) penalizes it relatively to video traffic, the aggregated value is significantly increased for both TCP and UDP sources. This means that there is a win-win situation where all traffic sources benefit from IEEE 802.11e technology. One of the causes of this improvement is related to routing; since routing traffic is assigned to the Voice MAC Access Category, the routing tasks are performed quicker. In figure 9.5 we find that the routing overhead for the AODV routing protocol decreases by using IEEE 802.11e technology, while for DSR the opposite occurs.

Besides the improvements in terms of routing efficiency, both Voice and Video MAC Access Categories of IEEE 802.11e, when used, allow the channel utilization to be improved; ultimately, this means that more traffic per unit of time can be transmitted in the MANET.

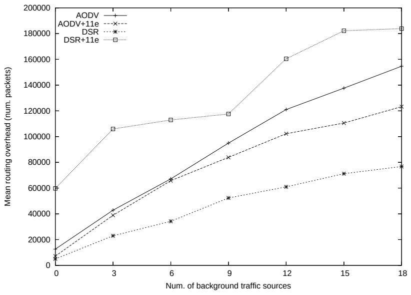


Figure 9.5: Mean routing overhead varying the number of background traffic sources

9.2 Improvements obtained with multipath routing techniques

In this section our focus goes to video streaming gaps. These are significant interruptions of a video streaming session which, in MANET environments, are typically caused by mobility and related re-routing processes. We build upon the results of the previous section, and so we will use a QoS-enabled MAC layer for our tests. We also include several best effort sources (six, to be exact) as background traffic.

As occurred before, our focus goes to a single real-time video stream inserted into the MANET using a real trace of an H.264-encoded video sequence.

The simulation settings are very similar to those of the previous section. The main differences are related to the background traffic (fixed instead of variable), the mobility settings and the routing protocols used; we now compare the DSR routing protocol to our enhanced version of that same protocol (MDSR), which was proposed in section 6. We wish to assess the effectiveness of the later in reducing streaming gaps compared to the former. Concerning mobility, we again use the random way-point mobility model to generate movement traces. Since our focus is on how multipath routing techniques benefit video streams in the presence of mobility, we vary the degree of mobility in a range between 1 and 9 m/s.

As shown in figure 9.6, the MDSR routing protocol we propose outperforms the DSR routing protocol in terms of both video throughput and end-to-end delay as node mobility increases; the only exception are very low speeds, where route maintenance is scarce. In fact, the difference between both in terms of throughput can be up to 65 kbit/s; in terms of delay the MDSR routing protocol shows high stability, with values below DSR for speeds above 1 m/s.

In terms of the performance of the real-time video stream, where our actual focus is on, figure 9.7 shows the mean PSNR values obtained, along with a confidence interval for the mean; the degree of confidence is of 90%. We find that MDSR presents a greater stability in terms of PSNR values, despite that those for DSR are good enough too. However, if we analyze the frame loss pattern at a moderate speed of 5 m/s (see right side of figure 9.7), we find that the user

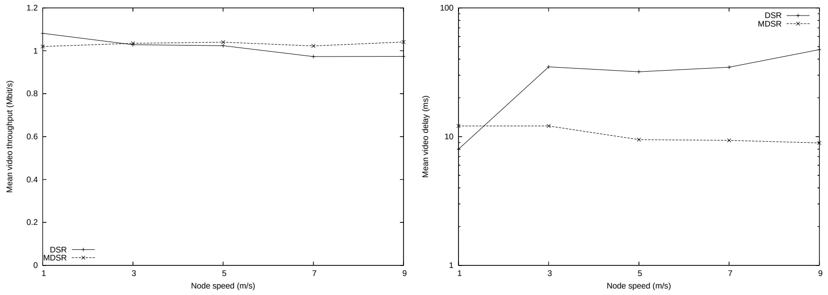


Figure 9.6: Mean values for the video throughput (left) and the video delay (right) varying the speed of nodes

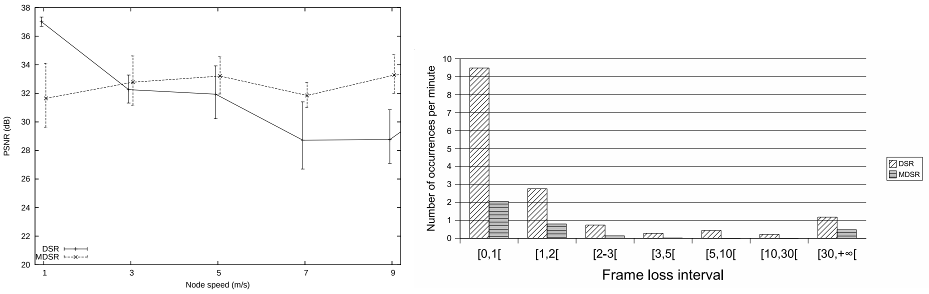


Figure 9.7: PSNR when varying the speed of nodes (left) and video gap histogram at a speed of 5 m/s (right)

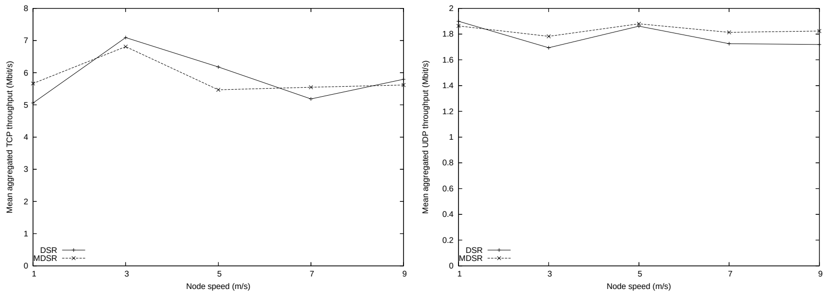


Figure 9.8: Mean values for the TCP throughput (left) and UDP throughput (right) varying the number of background traffic sources

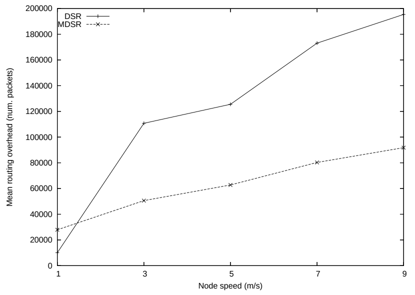


Figure 9.9: Mean routing overhead varying the number of background traffic sources

experience is greatly improved by using MDSR. Notice that such improvements are clearly not put in evidence by the 1.2 dB difference between both in terms of PSNR values.

Relatively to background traffic, figure 9.8 shows that mobility has a stronger impact on TCP traffic, which experiments an increase and then a decrease of its throughput; on the contrary, UDP background traffic tends to maintain its throughput values, being the slight changes observed directly related to the variations experienced for TCP traffic.

In terms of routing overhead, figure 9.9 shows that DSR requires an excessive routing overhead to offer a performance similar to that of MDSR at moderate and high speeds; interestingly we find that MDSR, which in theory imposes more routing overhead than DSR due to its enhanced route discovery mechanism, actually requires fewer routing packets than the latter.

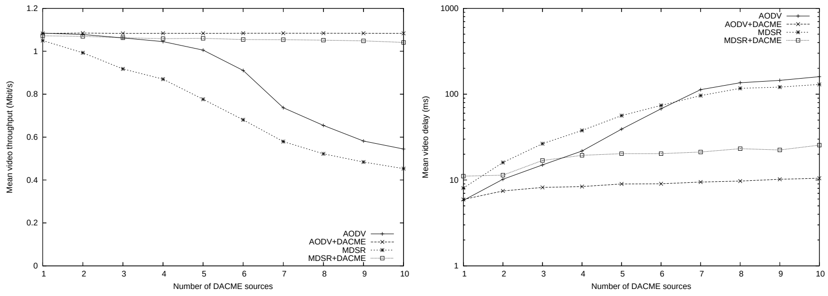


Figure 9.10: Mean values for the video throughput (left) and the video delay (right) varying the speed of nodes

9.3 Improvements obtained with distributed admission control techniques

In the two previous sections we showed how a MANET environment can be enhanced in terms of QoS support for video streaming applications. We first showed the benefits of using the IEEE 802.11e technology to differentiate traffic at the MAC layer, and we then proceeded with an analysis of routing-level techniques to mitigate the effects of node mobility in a MANET.

In this section we again build upon the findings of previous sections by analyzing the benefits of distributed admission control in supporting multiple real-time video streams in the MANET. So, in our experiments, we will study how we can regulate congestion and increase the QoS support of H.264 video sources using DACME agents (see chapter 8). With this purpose we will experiment with an increasing number of video sources, beginning with 1 and testing with up to 10 video sources.

The simulation setup is very similar to the one of the previous sections; we now compare the best-performing single-path routing protocol (AODV), with our enhanced version of the DSR routing protocol (MDSR) for testing, and we fix the number of best effort background sources at six (four TCP sources and two UDP sources). Concerning node mobility, it is also fixed at 5 m/s.

In figure 9.10 we observe that the admission control mechanism proposed maintains its effectiveness with VBR video sources, offering steady throughput values when the number of video sources increases; in case DACME is not used, the mean throughput for the video sources drops steadily, reaching arrivals rates below 60% for AODV and below 50% for MDSR.

In terms of end-to-end delay, the admission control strategy allows maintaining it at lower values for both routing protocols. Moreover, increasing the number of QoS sources under DACME only causes the delay to increase slightly.

In figure 9.11 we show the results in terms of PSNR, including confidence intervals for the mean; the degree of confidence used is of 95%. We find that DACME-regulated sources are able to maintain very good values for video distor-

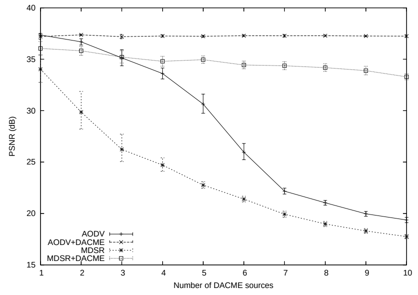


Figure 9.11: PSNR confidence intervals when varying the number of background traffic sources

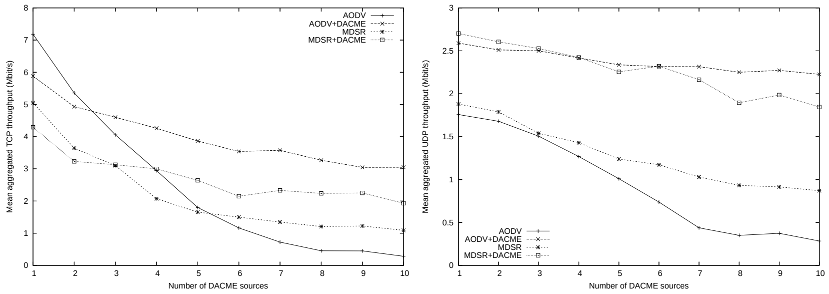


Figure 9.12: Mean values for the TCP throughput (left) and UDP throughput (right) varying the number of background traffic sources

tion (above 33dB), while the video distortion values without DACME drop to very low quality (below 25 dB) or even noise levels (below 20 dB).

Relatively to background traffic, the increase of video sources causes a decrease of both TCP and UDP traffic as expected (see figure 9.12). However, since DACME agents block video traffic when resources are not enough, we find that both TCP and UDP tend to achieve higher throughput when DACME is active. We therefore conclude that DACME makes resource usage more efficient for both DACME and non-DACME traffic.

Concerning the traffic acceptance rate, figure 9.13 shows that DACME restricts traffic admittance to values beginning at 80%, and reaching about 35% of the injected traffic when the number of sources is ten (for this particular scenario). The difference experienced using both routing protocols is rather slight, and in general the overall admission rate tends to increase as the number of sources increases. In terms of aggregated QoS traffic, though, we find that it increases almost linearly with an increasing number of sources, meaning that more sources do not cause the admission control mechanism to misuse the available radio resources.

Focusing now on routing overhead, figure 9.14 shows that DACME has a stabilizing effect on routing mechanisms, avoiding that routing traffic increases too

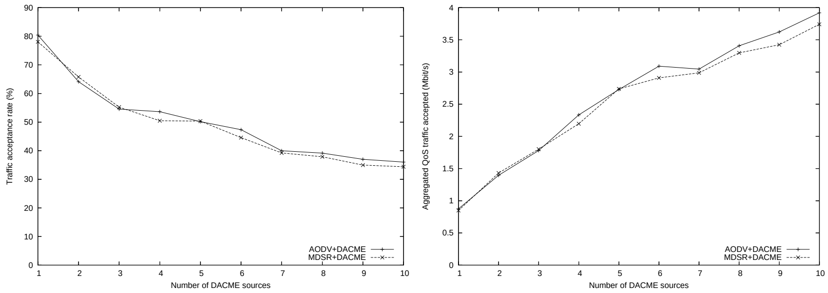


Figure 9.13: Traffic acceptance rate (left) and aggregated QoS traffic (right)

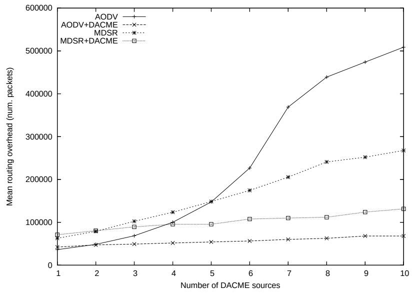


Figure 9.14: Mean routing overhead varying the number of background traffic sources

much due to congestion; this is especially noticeable for the AODV routing protocol.

9.4 Conclusions

In this last chapter we offered the summary of the main contributions of this thesis, demonstrating through simulation how we were able to evolve from legacy Wi-fi technology to a system with full QoS and admission control support. Such an enhanced system is able to offer a superior performance to video streams, solving most of the problems these suffer in MANET environments, namely TCP-related congestion, node mobility and unregulated QoS traffic.

Chapter 10

Conclusions, publications and future work

10.1 Conclusions

Throughout this thesis several contributions have been made to the area of wireless mobile ad-hoc networks and digital video. Our focus was on the optimum integration of real-time H.264 video streams with MANET environments; the purpose was to design a QoS framework that was able to enhance currently available technology in a very significant manner. We have shown, based on experimental results, that the different goals have been achieved.

We now proceed to summarize the most relevant contributions of this work:

- A general purpose performance analysis of the H.264 codec, where the different video codec parameters were evaluated in terms of their influence on rate/distortion values, as well as on encoding time.

- A study of the error-resilience tools available in the H.264 framework, including the improvements they provide under both random and bursty packet loss scenarios.

- A performance evaluation of real-time H.264 streaming over mobile ad-hoc networks using currently available, state-of-the-art technology. In that study we showed that MANETs using legacy technology offer a very poor performance to H.264 video streams, despite the different error-resilience mechanisms that the H.264 technology makes available. We concluded that improvements were required in terms of bandwidth, MAC-level QoS support, routing algorithms and admission control techniques to deploy a robust video transmission system for MANETs.

- A novel end-to-end path model for MANETs based on Hidden Markov Models whose purpose is to accelerate the process of video codec tuning; it avoids the time-consuming task of repeating MANET simulation experiments several times before conclusions can be drawn in terms of codec parameter choices by modeling the uncommon loss and arrival patterns typical of MANET environments.

- Extensions to the route discovery and forwarding mechanism of the DSR routing protocol so as to minimize the impact of station mobility on real-time streams, namely H.264 real-time video streams.
- A study on the adequacy and performance of the IEEE 802.11e technology under multi-hop MANET environments.
- An analysis of the impact of the IEEE 802.11e technology on reactive routing protocols, namely DSR and AODV.
- DACME, a novel distributed admission control system for MANET environments based on end-to-end resource measurement which imposes minimum requirements on intermediate stations. DACME is able to support adequately applications with bandwidth, delay and jitter requirements, besides increasing the overall stability of the MANET by avoiding that congestion causes routing protocols to misbehave.
- An incremental analysis of the benefits of IEEE 802.11e, MDSR and DACME when evolving from legacy MANET technology to a high-performance, QoS-enabled MANET environment.

Having accomplished all of our pre-defined goals, we consider that the final purpose of this thesis have been achieved successfully, and so we conclude this dissertation.

10.2 Publications related to the thesis

The research work related to this thesis has resulted in eighteen publications; among them we have two journal articles, fifteen conference papers and one research report. We now proceed by presenting a brief description of each of them; we have organized the different publications based on the chapter were the contents have been discussed.

Publications related with chapter 4:

[CM03b] Carlos T. Calafate, Manuel P. Malumbres, "Evaluation of the H.264 codec". *DISCA/60-2003*, UPV, Spain.

In this preliminary work we analyze the performance of the new H.264 video coding technology. Our analysis focuses on different issues such as video coding efficiency, error resilience and encoding times.

[CM03c] Carlos T. Calafate, Manuel P. Malumbres, "Testing the H.264 Resilience on Wireless Ad-hoc Networks", in *4th EURASIP Conference focused on Video / Image Processing and Multimedia Communications*, Zagreb, Croatia. July, 2003.

In this paper we demonstrate the error-resilience capabilities of the H.264 codec when facing both random and bursty packet losses, conditions prone to occur in ad hoc network environments.

[CMP04d] Carlos T. Calafate, Manuel P. Malumbres, Pietro Manzoni, "Performance of H.264 compressed video streams over 802.11b based MANETs", *International Conference on Distributed Computing Systems Workshops (ICDCSW)*, Tokyo, Japan. March, 2004.

[CMP] Carlos T. Calafate, Manuel P. Malumbres, Pietro Manzoni, "Performance issues of H.264 compressed video streams over IEEE 802.11b based MANETs", *International Journal of Wireless and Mobile Computing (IJWMC) - special issue on 'Wireless Ad Hoc Networking' (Invited article).* To appear in 2006.

In both this paper and this journal article we make a performance analysis of streaming H.264 sequences in a simulated ad hoc network environment. We show the effectiveness of the H.264 error resilience tools when combined with different routing protocols. This work puts in evidence the difficulties to achieve a reliable video transmission system in MANETs, also offering an insight on those problems which can not be solved by the video codec alone.

[CM03a] Carlos T. Calafate, Manuel P. Malumbres, "A step-by-step tuning of H.264 for unreliable dynamic networks", *XIV Jornadas de Paralelismo*, Universidad Carlos III de Madrid, Leganés, Spain. September, 2003.

This paper details the process to follow when we desire to tune an H.264 codec for adequate operation in networks characterized by low reliability.

Publications related with chapter 5:

[CPM04b] Carlos T. Calafate, Pietro Manzoni, Manuel P. Malumbres, "Speeding up the evaluation of multimedia streaming applications in MANETs using HMMs", in *the Seventh ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM'04)*, Venice, Italy. October, 2004.

In this paper we propose a novel end-to-end model for MANETs based on Hidden Markov chains. We show how to adapt the model to the behavior of both reactive and proactive routing protocols, and we show the benefits of using such models taking the process of video codec tuning as an example.

Publications related with chapter 6:

[CMP04a] Carlos T. Calafate, Manuel P. Malumbres, Pietro Manzoni, "A flexible and tunable route discovery mechanism for on-demand protocols", *12th*

Euromicro Conference on Parallel, Distributed and Network based Processing, La Coruña, Spain. February, 2004.

This paper analyzes different variations to DSR's route discovery technique in order to find more routes without provoking an excessive routing overhead.

[CMP04e] Carlos T. Calafate, Manuel P. Malumbres, Pietro Manzoni, "Route stability techniques for enhanced video delivery on MANETs", in the *Sixth IFIP IEEE International Conference on Mobile and Wireless Communication Networks*, Paris, France. October, 2004.

This paper analyzes the benefits of using enhanced routing techniques, evidencing the impact on a video stream of using an improved route discovery algorithm and of using traffic splitting through different routes.

[CMP04b] Carlos T. Calafate, Manuel P. Malumbres, Pietro Manzoni, "Improving H.264 real-time streaming in MANETs through adaptive multipath routing techniques", *IEEE Workshop on Adaptive Wireless Networks, Globecom 2004*. Dallas, Texas, USA. December, 2004.

[CMP04c] Carlos T. Calafate, Manuel P. Malumbres, Pietro Manzoni, "Mitigating the impact of mobility on H.264 real-time video streams using multiple paths", *Journal of Communications and Networks*, Volume 6, Number 4. December 2004.

This paper and this journal article present the MDSR routing protocol, discussing the origin of the improvements experienced by real-time video streams when stations rely on MDSR to perform routing and packet forwarding tasks.

Publications related with chapter 7:

[CPM04a] Carlos T. Calafate, Pietro Manzoni, Manuel P. Malumbres, "Assessing the effectiveness of IEEE 802.11e in multi-hop mobile network environments", in *IEEE/ACM International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS'04)*. Volendam, Netherlands. October, 2004.

In this paper we study the viability of using the IEEE 802.11e technology to achieve QoS support in multi-hop ad hoc networks.

[CPM05b] Carlos T. Calafate, Pietro Manzoni, Manuel P. Malumbres, "On the interaction between IEEE 802.11e and routing protocols in Mobile Ad-hoc Networks", in the *13th Euromicro Conference on Parallel, Distributed and Network-based Processing (PDP)*, Lugano, Switzerland. February, 2005.

[CJPM04] Carlos T. Calafate, Juan Carlos Cano, Pietro Manzoni, Manuel P. Malumbres, "Achieving enhanced performance in MANETs using IEEE 802.11e", *XV Jornadas de Paralelismo*, Almeria, Spain. September, 2004.

In these two papers we focus on the interaction between the IEEE 802.11e technology and two reactive routing protocols for MANETs (AODV and

DSR), showing how the former can significantly improve routing responsiveness.

Publications related with chapter 8:

[CPM05c] Carlos T. Calafate, Pietro Manzoni, Manuel P. Malumbres, "Supporting soft real-time services in MANETs using distributed admission control and IEEE 802.11e technology", in the *Tenth IEEE Symposium on Computers and Communications (ISCC'2005)*, La Manga del Mar Menor, Cartagena, Spain. June, 2005.

In this paper we make a mathematical and statistical analysis of the most adequate and accurate probe-based techniques to measure end-to-end bandwidth, delay and jitter in a MANET environment.

[CPM05d] Carlos T. Calafate, Pietro Manzoni, Manuel P. Malumbres, "Using distributed admission control to support multimedia applications in MANET environments", in the *31st EUROMICRO Conference on Software Engineering and Advanced Applications*, Porto, Portugal. August 30th - September 3rd, 2005.

[CPM05a] Carlos T. Calafate, Pietro Manzoni, Manuel P. Malumbres, "A framework to deploy bandwidth constrained applications in IEEE 802.11-based ad hoc networks", *Simposio de Computación Ubicua e Inteligencia Ambiental (UCAmI 2005), I Congreso Español de Informática*, Granada, Spain. September, 2005.

These two papers introduce DACME, a novel distributed admission control system for MANETs. We prove through simulation that relying on probe-based bandwidth measurements can improve significantly the QoS experienced by both voice and video streams.

[CJPM05] Carlos T. Calafate, Juan Carlos Cano, Pietro Manzoni, and Manuel P. Malumbres, "A QoS architecture for MANETs supporting real-time peer-to-peer multimedia applications", in the *7th IEEE International Symposium of Multimedia (ISM2005)*, Irvine, California, USA. December, 2005.

In this paper we extend DACME's framework to also include support for applications that not only have bandwidth constraints, but also delay and jitter constraints.

[CPM06] Carlos T. Calafate, Pietro Manzoni, Manuel P. Malumbres, "A novel QoS framework for MANETs supporting multipath routing protocols", in the *11th IEEE Symposium on Computers and Communications (ISCC'2006)*, Pula-Cagliari, Sardinia, Italy. June, 2006

This last paper shows the required steps to take in order to achieve a reliable system combining both multipath routing and distributed admission control techniques.

To obtain the "Advanced Studies Degree", a preliminary study on the performance of ad hoc networks in real environments was made, along with an implementation of the OLSR routing protocol. The publications related with that work are:

[CP03] Carlos T. Calafate and Pietro Manzoni, "A multi-platform programming interface for protocol development", in *11th Euromicro Conference on Parallel Distributed and Network based Processing*, Genoa, Italy. February, 2003.

In this publication we offer details about the PICA library, a library created to simplify the creation and porting of routing protocols for ad hoc networks to different platforms. The platforms we focused on were Linux, Windows NT and Windows CE.

[CRP03] Carlos T. Calafate, Roman Garcia Garcia, Pietro Manzoni, "Optimizing the implementation of a MANET routing protocol in a heterogeneous environment", in *The 8th IEEE Symposium on Computers and Communications (ISCC'2003)*, Kemer, Antalya, Turkey. July, 2003.

This work presents some performance results obtained in a real ad hoc network testbed. The testbed used a version of the OSLR routing protocol based on the PICA library referred in the previous work. MANET stations employed for experiments are heterogeneous in terms of hardware (desktop and laptop PCs, as well as PDAs), operating systems used (Linux, Windows 2000 Professional, Windows CE 3.0), and wireless technologies employed (Wi-Fi and Bluetooth).

During the development of this thesis there have also been other works, somehow related to the contents of this thesis, that we also consider relevant to refer:

[JCMP04a] Juan Carlos Cano, Carlos T. Calafate, Manuel P. Malumbres, Pietro Manzoni, "On The Use Of Mobile Ad Hoc Networks For The Support Of Ubiquitous Computing", *UPGRADE, The European Journal for the Informatics Professional.* ISSN 1684-5285. February, 2004.

[JCMP04b] Juan Carlos Cano, Carlos T. Calafate, Manuel P. Malumbres, Pietro Manzoni, "Redes Inalámbricas Ad Hoc como Tecnología de Soporte para la Computación Ubicua", *Novática, in Revista de la Asociación de Técnicos Informáticos.* ISSN 0211-2124. February, 2004.

In these two journals we refer to the relevance of ad hoc networks in ubiquitous computing environments, offering as an example an application for an Ubiquitous Museum.

[JCPM04] Juan Carlos Cano, Carlos T. Calafate, Manuel P. Malumbres, Pietro Manzoni, "Evaluating the Performance Impact of Group Mobility in MANETs", in *XV Jornadas de Paralelismo*, Almeria, Spain. September, 2004.

In this paper we take a look to the controversial topic of mobility models for ad hoc networks, proposing new models for mobility based on groups of users and assessing the performance achieved for different study cases.

10.3 Future work

In the development of this thesis several issues emerged which deserve further scrutiny in a future. The ones we consider most relevant are the following:

- To test the MDSR routing protocol in a real ad hoc network environment, validating the goodness of that proposal. This requires enhancing any of the DSR implementations available, and so the resulting code would also become a contribution to the scientific community.

- To develop a new admission control system based on DACME, improving the results by forcing all the MANET stations to actively participate on all QoS tasks.

- To design an implementation of DACME for different Operating System platforms (e.g. Linux, Windows, Windows CE). This could include the deployment of a real testbed based on IEEE 802.11e technology to build a complete QoS framework.

- To study the applicability of the Hidden Markov Models developed under different MANET environments, thereby assessing their usefulness and applicability in real scenarios.

- To implement a bi-directional audio-visual communication system based on DACME, MDSR and IEEE 802.11e, using H.264 technology for efficient and error-resilient video coding.

Bibliography

[AHKG03] A. Munaretto Fonseca, H. Badis, K. Al Agha, and G. Pujolle. QoS-enhanced OLSR protocol for Mobile Ad Hoc Networks. In *1st International ANWIRE Workshop*, Glasgow, UK, April 2003.

[AMS03] A. Urpi, M. Bonuccelli, and S. Giordano. Modelling cooperation in mobile ad hoc networks: a formal description of selfishness. In *WiOpt'03: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, Sophia-Antipolis, France, March 2003.

[ARS01] Asis Nasipuri, Robert Castaneda, and Samir R. Das. Performance of multipath routing for on-demand protocols in mobile ad hoc networks. *ACM/Baltzer Mobile Networks and Applications (MONET) Journal, vol. 6*, pages 339–349, 2001.

[AT99] George Aggelou and Rahim Tafazolli. RDMAR: A bandwidth-efficient routing protocol for mobile ad hoc networks. In *Proceedings of the WOWMOM*, pages 26–33, 1999.

[Bas99] S. Basagni. Distributed clustering for ad hoc networks. In *Proceedings of the IEEE International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN)*, pages 310–315, Perth, Western Australia, June 1999.

[Blu02] IEEE 802.15.1(tm) IEEE Standard for Information technology–Telecommunications and information exchange between systems– Local and metropolitan area networks–Specific requirements Part 15.1: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs(tm)), 2002.

[BR99] Bhargav Bellur and Richard G. Ogier. A Reliable, Efficient Topology Broadcast Protocol for Dynamic Networks. Proceedings of the IEEE INFOCOM, The Conference on Computer Communications, New York, USA, March 1999.

[C. 97] C. K. Toh. Associativity-Based Routing for Ad-Hoc Mobile Networks. *Wireless Personal Communication*, 4(2):1–36, March 1997.

[CES00] Charles E. Perkins, Elizabeth M. Belding-Royer, and Samir R. Das. Quality of service in ad hoc on-demand distance vector routing. IETF Internet Draft, draft-ietf-manet-qos-00.txt, July 2000. Work in progress.

[CES03] Charles E. Perkins, Elizabeth M. Belding-Royer, and Samir R. Das. Ad hoc on-demand distance vector (AODV) routing. Request for Comments 3561, MANET Working Group, http://www.ietf.org/rfc/rfc3561.txt, July 2003. Work in progress.

[Chi97] C.-C. Chiang. Routing in clustered multihop, mobile wireless networks with fading channel. In *Proc. IEEE SICON 97*, pages 197–211, April 1997.

[CJPM04] Carlos T. Calafate, Juan-Carlos Cano, Pietro Manzoni, and Manuel P. Malumbres. Achieving enhanced performance in MANETs using IEEE 802.11e. In *XV Jornadas de Paralelismo*, Almeria, Spain, September 2004.

[CJPM05] Carlos T. Calafate, Juan-Carlos Cano, Pietro Manzoni, and Manuel P. Malumbres. A QoS architecture for MANETs supporting real-time peer-to-peer multimedia applications. In *IEEE International Symposium of Multimedia (ISM2005)*, Irvine, California, USA, December 2005.

[CM03a] Carlos T. Calafate and Manuel P. Malumbres. A step-by-step tuning of H.264 for unreliable dynamic networks. In *XIV Jornadas de Paralelismo*, Universidad Carlos III de Madrid, Leganés, Spain, September 2003.

[CM03b] Carlos T. Calafate and Manuel P. Malumbres. Evaluation of the H.264 codec. Technical Report DISCA/60-2003, UPV, Spain, 2003.

[CM03c] Carlos T. Calafate and Manuel P. Malumbres. Testing the H.264 Error-Resilience on Wireless Ad-hoc Networks. In *4th EURASIP Conference focused on Video / Image Processing and Multimedia Communications*, Zagreb, Croatia, July 2003.

[CMP] Carlos T. Calafate, Manuel P. Malumbres, and Pietro Manzoni. Performance issues of H.264 compressed video streams over IEEE 802.11b based MANETs. *International Journal of Wireless and Mobile Computing (IJWMC) - special issue on 'Wireless Ad Hoc Networking'.* (Invited paper). To apear.

[CMP04a] Carlos T. Calafate, Manuel P. Malumbres, and Pietro Manzoni. A flexible and tunable route discovery mechanism for on-demand protocols. In *12-th Euromicro Conference on Parallel, Distributed and Network based Processing*, La Coruña, Spain, February 2004.

[CMP04b] Carlos T. Calafate, Manuel P. Malumbres, and Pietro Manzoni. Improving H.264 real-time streaming in MANETs through adaptive multipath routing techniques. In *IEEE Workshop on Adaptive Wireless Networks, Globecom 2004*, Dallas, Texas, USA, December 2004.

[CMP04c] Carlos T. Calafate, Manuel P. Malumbres, and Pietro Manzoni. Mitigating the impact of mobility on H.264 real-time video streams using multiple paths. *Journal of Communications and Networks*, 6(4):387–396, December 2004.

[CMP04d] Carlos T. Calafate, Manuel P. Malumbres, and Pietro Manzoni. Performance of H.264 compressed video streams over 802.11b based MANETs. In *International Conference on Distributed Computing Systems Workshops (ICDCSW'04)*, Hachioji - Tokyo, Japan, March 2004.

[CMP04e] Carlos T. Calafate, Manuel P. Malumbres, and Pietro Manzoni. Route stability techniques for enhanced video delivery on MANETs. In *Sixth IFIP IEEE International Conference on Mobile and Wireless Communication Networks*, Paris, France, October 2004.

[CP94] C. E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. *ACM Computer Communication Review*, 24(2):234–244, October 1994.

[CP03] Carlos T. Calafate and Pietro Manzoni. A multi-platform programming interface for protocol development. In *11-th Euromicro Conference on Parallel Distributed and Network based Processing*, Genova, Italy, February 2003.

[CPM04a] Carlos T. Calafate, Pietro Manzoni, and Manuel P. Malumbres. Assessing the effectiveness of IEEE 802.11e in multi-hop mobile network environments. In *12th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS'04)*, Volendam, Netherlands, October 2004.

[CPM04b] Carlos T. Calafate, Pietro Manzoni, and Manuel P. Malumbres. Speeding up the evaluation of multimedia streaming applications in MANETs using HMMs. In *The Seventh ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM'04)*, Venice, Italy, October 2004.

[CPM05a] Carlos T. Calafate, Pietro Manzoni, and Manuel P. Malumbres. A framework to deploy bandwidth constrained applications in IEEE 802.11-based ad hoc networks. In *Simposio de Computación Ubicua e Inteligencia Ambiental (UCAmI), I Congreso Español de Informática*, Granada, Spain, September 2005.

[CPM05b] Carlos T. Calafate, Pietro Manzoni, and Manuel P. Malumbres. On the interaction between IEEE 802.11e and routing protocols in Mobile

Ad-hoc Networks. In *13th Euromicro Conference on Parallel, Distributed and Network-based Processing (PDP)*, Lugano, Switzerland, February 2005.

[CPM05c] Carlos T. Calafate, Pietro Manzoni, and Manuel P. Malumbres. Supporting soft real-time services in MANETs using distributed admission control and IEEE 802.11e technology. In *The 10th IEEE Symposium on Computers and Communications*, June 2005.

[CPM05d] Carlos T. Calafate, Pietro Manzoni, and Manuel P. Malumbres. Using distributed admission control to support multimedia applications in MANET environments. In *31st EUROMICRO Conference on Software Engineering and Advanced Applications*, Porto, Portugal, September 2005.

[CPM06] Carlos T. Calafate, Pietro Manzoni, and Manuel P. Malumbres. A novel QoS framework for MANETs supporting multipath routing protocols. In *The Eleventh IEEE Symposium on Computers and Communications (ISCC'2006)*, Pula-Cagliari, Sardinia, Italy, June 2006.

[CQS98] X. Chen, L. Qi, and D. Sun. Global and superlinear convergence of the smoothing Newton method and its application to general box constrained variational inequalities. *Mathematics of Computation*, 67(222):519–540, 1998.

[CRP03] Carlos T. Calafate, Roman Garcia Garcia, and Pietro Manzoni. Optimizing the implementation of a manet routing protocol in a heterogeneous environment. In *The 8th IEEE Symposium on Computers and Communications (ISCC'2003)*, Kemer - Antalya, Turkey, July 2003.

[DDY04] David B. Johnson, David A. Maltz, and Yih-Chun Hu. The dynamic source routing protocol. Internet Draft, MANET Working Group, draft-ietf-manet-dsr-10.txt, July 2004. Work in progress.

[DRWT96] Rohit Dube, Cynthia D. Rais, Kuang-Yeh Wang, and Satish K. Tripathi. Signal stability based adaptive routing (ssa) for ad-hoc mobile networks. Technical report, 1996.

[G. 98] G. Malkin. RIP Version 2. IETF RFC 2453, November 1998.

[GAAL02] G-S. Ahn, A. T. Campbell, A. Veres, and L. Sun. Supporting service differentiation for real-time and best effort traffic in stateless wireless ad hoc networks (SWAN). *IEEE Transactions on Mobile Computing*, September 2002.

[GCNB01] J. Gomez, A. Campbell, M. Naghshineh, and C. Bisdikian. Conserving transmission power in wireless ad hoc networks, 2001.

[GLAS99] J. J. Garcia-Luna-Aceves and Marcelo Spohn. Source-tree routing in wireless networks. In *ICNP*, pages 273–282, 1999.

[GN99] Gavin Holland and Nitin H. Vaidya. Analysis of TCP performance over mobile ad hoc networks. In *5th annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 219–230, Seattle, Washington, USA, 1999.

[GSM97] Digital cellular telecommunications system (Phase 2+); Terminal Equipment to Mobile Station (TE-MS) multiplexer protocol. ETSI GSM 07.10 version 6.3.0, 1997.

[H2695] Video coding for low bitrate communication. ITU-T Recommendation H.263, 1995.

[H2603a] Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC), March 2003.

[H2603b] JM Reference Software, version 3.9a. http://iphome.hhi.de/suehring/tml, 2003.

[HKWA01] Hannan Xiao, Kee Chaing Chua, Winston K.G. Seah, and Anthony Lo. On service prioritization in mobile ad-hoc networks. In *IEEE International Conference on Communications (ICC 2001)*, Helsinki, Finland, June 2001.

[HSRV96] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. IETF RFC 2205, January 1996.

[HWAK00] Hannan Xiao, Winston K.G. Seah, Anthony Lo, and Kee Chaing Chua. A flexible quality of service model for mobile ad-hoc networks. In *IEEE 51st Vehicular Technology Conference Proceedings*, volume 1, pages 445–440, Tokyo, 2000.

[IEE05] IEEE 802.11 WG. 802.11e IEEE Standard for Information technology-Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements, 2005.

[IrD01] IrDA Physical Layer Standard version 1.3. http://www.irda.org, March 2001.

[ITU] ITU Telecommunication Standardization Sector (ITU-T). http://www.itu.int/ITU-T.

[J. 98] J. Moy. OSPF Version 2. IETF RFC 2328, April 1998.

[JCMP04a] Juan-Carlos Cano, Carlos T. Calafate, Manuel P. Malumbres, and Pietro Manzoni. On The Use Of Mobile Ad Hoc Networks For The Support Of Ubiquitous Computing. *UPGRADE, The European Journal for the Informatics Professional*, V(1):56–62, February 2004.

[JCMP04b] Juan-Carlos Cano, Carlos T. Calafate, Manuel P. Malumbres, and Pietro Manzoni. Redes Inalambricas Ad Hoc como Tecnologia de Soporte para la Computacion Ubicua. *Novática, la Revista de la Asociación de Técnicos Informáticos*, XXX(167):33–38, January-February 2004.

[JCPM04] Juan-Carlos Cano, Carlos T. Calafate, Pietro Manzoni, and Manuel P. Malumbres. Evaluating the Performance Impact of Group Mobility in MANETs. In *XV Jornadas de Paralelismo*, Almeria, Spain, September 2004.

[JDP03] Juan Carlos Cano, Dongkyun Kim, and Pietro Manzoni. CERA: Cluster-based Energy Saving Algorithm to Coordinate Routing in Short-Range Wireless Networks. The International Conference on Information Networking (ICOIN) 2003, Jeju Island, Korea, February 2003.

[JM96] David B Johnson and David A Maltz. Dynamic source routing in ad hoc wireless networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.

[JTC] ISO/IEC JTC1: Joint Technical Committee for Information Technology. http://www.jtc1.org.

[KK00] K. Fall and K. Varadhan. ns notes and documents. The VINT Project. UC Berkeley, LBL, USC/ISI, and Xerox PARC, February 2000.

[LES$^+$00] Lee Breslau, Ed Knightly, Scott Shenker, Ion Stoica, and Hui Zhan. Endpoint Admission Control: Architectural Issues and Performance. In *Proceedings of ACM Sigcomm 2000*, Stockholm, Sweden, September 2000.

[LG01] S. Lee and M. Gerla. Split multipath routing with maximally disjoint paths in ad hoc networks. In Proceedings of the IEEE ICC, pages 3201–3205, 2001.

[LPB04] Leonidas Georgiadis, Philippe Jacquet, and Bernard Mans. Bandwidth Reservation in Multihop Wireless Networks: Complexity and Mechanisms. In *International Conference on Distributed Computing Systems Workshops (ICDCSW'04)*, Hachioji - Tokyo, Japan, March 2004.

[LYM$^+$01] Lei Wang, Yantai Shu, Miao Dong, Lianfang Zhang, and O.W.W. Yang. Adaptive multipath source routing in ad hoc networks. ICC 2001. IEEE International Conference on Communications; Page(s): 867-871 vol.3, 2001.

[M. 01] M. Sánchez. Adaptive Power Control for Ad-hoc Networks. 5th International Conference on Systemics, Cybernetics and Informatics (SCI 2001), Orlando, Florida, July 2001.

[MD01] M. Marina and S. Das. On demand multipath distance vector routing in ad hoc networks. In Proceedings of IEEE International Conference on Network Protocols (ICNP), pages 14–23, 2001.

[MGLA96] Shree Murthy and J. J. Garcia-Luna-Aceves. An efficient routing protocol for wireless networks. *Mobile Networks and Applications*, 1(2):183–197, 1996.

[MJ97] M. Budagavi and J. D. Gibson. Error Propagation in Motion Compensated Video over Wireless Channels. *Proceedings of the IEEE International Conference on Image Processing, Santa Barbara, USA*, pages 89–92, October 1997.

[MJ01] M. Budagavi and J. D. Gibson. Multiframe video coding for improved performance over wireless channels. *IEEE Transactions on Image Processing, Volume: 10, Issue: 2*, pages 252–265, February 2001.

[mpe01] ISO/IEC IS,Coding of Audio-Visual Objects, part 2: Visual (MPEG-4). Information Technology, November 2001.

[PGC00] Guangyu Pei, Mario Gerla, and Tsu-Wei Chen. Fisheye state routing: A routing scheme for ad hoc wireless networks. In *ICC (1)*, pages 70–74, 2000.

[PGH00] G. Pei, M. Gerla, and X. Hong. Lanmar: Landmark routing for large scale wireless ad hoc networks with group mobility, 2000.

[Phi90] Phil Karn. MACA - A New Channel Access Method for Packet Radio. Proceedings of the 9th ARRL Computer Networking Conference, London, Ontario, Canada, 1990.

[PR99] Charles E. Perkins and Elizabeth M. Royer. Ad hoc On-Demand Distance Vector Routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA*, pages 90–100, February 1999.

[Rab89] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[RDS94] R. Braden, D. Clark, and S. Shenker. Integrated Services in the Internet Architecture - an Overview. IETF RFC 1633, June 1994.

[RLS$^+$97] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification. IETF RFC 2205, September 1997.

[S. 66] S. W. Golomb. Run-Length Encoding. *IEEE Transactions on Information Theory, IT-1*, pages 399–401, December 1966.

[S. 98] S. Blake. An Architecture for Differentiated Services. IETF RFC 2475, December 1998.

[SAXA00] S.B. Lee, A. Gahng-Seop, X. Zhang, and Andrew T. Campbell. INSIGNIA: An IP-Based Quality of Service Framework for Mobile Ad Hoc Networks. *Journal of Parallel and Distributed Computing (Academic Press) , Special issue on Wireless and Mobile Computing and Communications*, 60(4):374–406, April 2000.

[SC03] Sven Wietholter and Christian Hoene. Design and Verification of an IEEE 802.11e EDCF Simulation Model in ns-2.26. Technical Report TKN-03-019, Telecommunication Networks Group, Technische Universitat Berlin, November 2003.

[SCM99] S.-J. Lee, C.-K. Toh, and M. Gerla. Performance Evaluation of Table-Driven and On-Demand Ad Hoc Routing Protocols. In *Proceedings of IEEE PIMRC'99, Osaka, Japan*, pages 297–301, September 1999.

[SK99] J. L. Sobrinho and A. S. Krishnakumar. Quality-of-Service in ad hoc carrier sense multiple access networks. *IEEE Journal on Selected Areas in Communications*, 17(8):1353–1368, August 1999.

[SWR98] Suresh Singh, Mike Woo, and C. S. Raghavendra. Power-aware routing in mobile ad hoc networks. In *Mobile Computing and Networking*, pages 181–190, 1998.

[TDT02] Thomas Stockhammer, Dimitrios Kontopodis, and Thomas Wiegand. Rate-Distortion Optimization for H.26L Video Coding in Packet Loss Environment. *12th International Packet Video Workshop (PV 2002), Pittsburg, PY*, May 2002.

[TM02] Till Halbach and Mathias Wien. Concepts and performance of next-generation video compression standardization. *5th Nordic Signal Processing Symposium (NORSIG-2002)*, October 2002.

[TP03] T. Clausen and P. Jacquet. Optimized link state routing protocol (OLSR). Request for Comments 3626, MANET Working Group, http://www.ietf.org/rfc/rfc3626.txt, October 2003. Work in progress.

[TPA$^+$01] T. Clausen, P. Jacquet, A. Laouiti, P. Muhlethaler, A. Qayyum, and L. Viennot. Optimized link state routing protocol. *International Multi Topic Conference, Pakistan*, 2001.

[TR01] T. Dyer and R. Boppana. A comparison of TCP performance over three routing protocols for mobile ad hoc networks. In *ACM Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc)*, Long Beach, California, USA, October 2001.

[TS02] Thomas Stockhammer and Stephan Wenger. Standard compliant enhancements of jvt coded video over fixed and wireless ip. *2002 International Tyrrhenian Workshop on Digital Communications (IWDC 2002), Capri (Italy)*, September 2002.

[VS00] V. Park and S. Corson. Temporally-ordered routing algorithm (TORA) version 1 - functional specification. Internet Draft, MANET Working Group, draft-ietf-manet-tora-spec-03.txt, November 2000. Work in progress.

[WG99] IEEE 802.11 WG. International Standard for Information Technology - Telecom. and Information exchange between systems - Local and Metropolitan Area Networks - Specific Requirements - Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications, ISO/IEC 8802-11:1999(E) IEEE Std. 802.11, 1999.

[Wu02] Jie Wu. An Extended Dynamic Source Routing Scheme in Ad Hoc Wireless Networks. 35th Annual Hawaii International Conference on System Sciences (HICSS'02)-Volume 9, Big Island, Hawaii, January 2002.

[YMV$^+$02] Ye-Kui Wang, Miska Hannuksela, Viktor Varsa, Ari Hourunranta, and Moncef Gabbouj. The error concealment feature in the H.26L test model. *IEEE 2002 International Conference on Image Processing (ICIP'2002), Rochester, New York, USA*, September 2002.

[YN98] Y.B.Ko and N.H.Vaidya. Location aided routing (lar) in mobile adhoc networks. In *The Annual International Conference on Mobile Computing and Networking (MOBICOM)*, Dallas, Texas, USA, October 1998.

[YR03] Y. Iraqi and R. Boutaba. The degree of participation concept in ad hoc networks. In *IEEE Symposium on Computers and Communications (ISCC 2003)*, pages 197–202, Antalya, Turkey, 2003.

[ZJ96] Z. Wang and J. Crowcroft. Quality-of-Service Routing for Supporting Multimedia Applications. *IEEE Journal on Selected Areas in Communications*, 14(7):1228–1234, September 1996.

[ZM99] Z. Haas and M. Pearlman. The zone routing protocol (ZRP) for ad hoc networks. Internet Draft, MANET Working Group, draft-ietf-manet-zone-zrp-02.txt, June 1999. Work in progress.