*INSTITUT POLYTECHNIQUE DE GRENOBLE*

*N° attribué par la bibliothèque*

|__|__|__|__|__|__|__|__|__|__|

***THÈSE***

pour obtenir le grade de

**DOCTEUR DE l'Institut polytechnique de Grenoble**

***Spécialité : "Informatique : Systèmes et Logiciels"***

Préparée au Laboratoire d'Informatique de Grenoble

Dans le cadre de **l'École Doctorale "Mathématiques, Sciences et Technologies de l'Information, Informatique"**

préparée et soutenue publiquement par


# Mohammad NASSIRI


**Le 24 Novembre 2008**

***TITRE :***

# Amélioration des performances MAC dans les réseaux sans-fil

*Directeurs de thèse :*
***Andrzej* DUDA**
***Martin* HEUSSE**

**JURY**

| | |
|---|---|
| Pr. Guy MAZARÉ | *Président* |
| Pr. David SIMPLOT-RYL | *Rapporteur* |
| Pr. André-Luc BEYLOT | *Rapporteur* |
| Pr. Andrzej DUDA | *Directeur de thèse* |
| Dr. Martin HEUSSE | *Co-encadrant* |
| Pr. Fabrice VALOIS | *Examinateur* |

*To*

*my dear wife, Fatemeh*

*and*

*my loving daughter, Nava*

## Acknowledgements

I would like to express my gratitude to all those who gave me the possibility to complete this thesis. First of all, I owe my warmest thanks to my supervisors Prof. Andrzej Duda and Dr. Martin Heusse for valuable advice and criticism and for introducing me to scientific research. They both were available when I needed their help and I will never forget their kindness.

I am grateful to Prof. David Simplot-Ryl and Prof. André-Luc Beylot for reviewing the present manuscript and providing constructive comments and criticism. I would like to thank Prof. Guy Mazaré the president of jury and Prof. Fabrice Valois for the time they spend over my thesis. I also want to thank them for attending my PhD defense.

I extend special thanks to my colleagues from Grenoble Informatics Laboratory (LIG) and particularly members of Drakkar Research Group with whom I spent memorable moments. I want to thank them for all their help, support, interest and valuable hints. I am especially obliged to Fabrice Theoleyre for his collaboration in contributions of chapter 5 of this thesis.

I wish to thank my parents, my brother, and my sisters for their support and encouragement throughout my life. I would like to express my loving thanks to my wife Fatemeh, without whom this work could never have been successful and whose patience and love have made this possible. Finally, I thank my loving daughter, Nava, who joined us during the second year of my thesis, for giving me unlimited happiness and pleasure.

This work was partially supported by the Iranian Ministry of Science, Research and Technology (MSRT), the European Commission project WIP under contract 27402, the French Ministry of Research projects AIRNET under contract ANR-05-RNRT-012-01, and ARESA under contract ANR-05-RNRT-01703.

# Abstract

In this thesis, we study the important performance problems that arise at the MAC layer in 802.11 wireless networks when they are used in the context of ad hoc or multi-hop. We proposed several mechanisms to overcome these performance issues. First, we enhance performance of a single wireless cell by accommodating several classes of services. Our mechanism is a prioritized access method that supports both proportional throughput allocation and absolute priority. Second, we propose an efficient localized way to solve the problem of blocked stations in large-scale wireless networks. Finally, we propose a more global approach that combines topology constraints and a MAC layer mechanism to improve the performance of wireless ad hoc or mesh networks in terms of throughput and fairness. We use simulation to evaluate our proposals and also compare them with some state-of-the-art solutions.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

In 1895, a few decades after the telephone was invented, Marconi demonstrated the first radio transmission from the Isle of Wight to a tugboat 18 miles away. This was the birth of radio communications. Radio technology, then advanced rapidly to enable transmissions over larger distances with better quality, less power, and smaller, cheaper devices. Cellular networks, satellite networks, and television are all the results of radio technology. Generally the term of *wireless network* refers to any telecommunication network with interconnections between nodes implemented without wires.

With the advent of computers, the need to communicate with each other and to share resources and information led to the design of modern computer networks. The ARPANET (Advanced Research Projects Agency Network) developed in 1972 by United States Department of Defense was the first wide-area packet switching network and the predecessor of the Internet. Stations exchange data in the Internet on the principle of packet switching: packets are routed between nodes over shared data links. The Internet can be considered as a worldwide infrastructure that enables information sharing in a transparent manner in the end-user perspective. Just several decades after their introduction, computer networks and especially the Internet play a very important role in our daily life.

The ALOHAnet, a pioneering computer networking system developed in 1970 at the University of Hawaii, was the first infrastructure-based wireless network. In 1973, ARPA initiated another research on the feasibility of using packet-switched radio communications to build computer networks. The ARPA Packet Radio Network (PRnet) has evolved through several years (1973-1987) to become a robust, reliable, and operational experimental network. The PRnet was the earliest wireless ad hoc network.

Although the first wireless networks such as PRnet were used for military purposes, today an increased use of laptop computers within enterprises and university campuses, as well as the increase of people mobility have fueled the demand for wireless computer networks. Moreover, a wireless network is an inexpensive and quick way to get connected to the Internet. Today many of public places offer Wi-Fi access to the Internet. Users can use a laptop, Wi-Fi phone, or other suitable portable device to benefit from the wireless link to the Internet. Also, one of the current research trends in the Internet consists of the following idea: connect to the Internet anytime everywhere. Due to the utilization of a radio channel as the communication medium, wireless networks can provide the possibility of ubiquitous connectivity.

In this thesis, we consider wireless computer networks based on IEEE 802.11 standard [44] that cover a small geographical areas like a home, a group of buildings, a conference hall, or a school as well as those that cover a campus, a metropolitan area or even a wider area. The 802.11 standard specifies medium access control (MAC) and physical layer (PHY) in wireless LANs. Although 802.11 technology was proposed at the beginning for wireless LANs, we extend its channel access method to adapt it to large-scale wireless networks.

## 1.1 Motivations and Objectives

Our work concerns the MAC layer for which, it proposes efficient solutions to several issues raised by the use of wireless LANs in a dense or mesh environment. First, we enhance performance of a single wireless cell by accommodating several classes of services. Second, we propose an efficient localized way to solve the problem of blocked stations in large-scale wireless networks. Finally, we propose a more global approach that combines topology constraints and a MAC layer mechanism to improve the performance of wireless ad hoc or mesh networks in terms of throughput and fairness.

### 1.1.1 Quality of Service in Wireless LANs

Traditionally, real-time multimedia data were transferred through circuit switched networks. Today, with the development of the Internet, it also serves to convey voice and video. However, multimedia traffic requires some quality of service (QoS) such as a minimal throughput, a maximal delay, and a bounded jitter. Therefore, fulfilling QoS requirements in packet switched networks has been an active area of research. Supporting QoS guarantees in wireless networks is much more difficult because of physical characteristics of the radio channel, such as variable throughput, higher bit error rate, and interference.

In addition to physical characteristics, the medium access mechanism also affects the QoS parameters. More specifically, when channel conditions are good and communicating devices are within the range of good reception, QoS characteristics mainly depend on the channel access method. 802.11e [48] standard has been proposed to provide prioritized access and service differentiation in wireless LANs. However, 802.11e does not perform well when the number of contending stations increases. Therefore, we think that an access method compatible with 802.11 wireless LANs is needed that would be able to provide priority access to the medium without sacrificing throughput.

Many researchers have addressed the problem of QoS differentiation in 802.11 wireless networks, however no method proposed so far features all desirable properties: high aggregate throughput even for a large number of contending stations, fair allocation to all stations in the same class, fast adaptation to changing conditions, and support for absolute priorities. If we consider the IEEE 802.11e standard, its EDCA (Enhanced Distributed Channel Access) access method suffers from an increased collision rate when the number of stations increases. In Chapter 3, we propose a novel access method that supports both *relative proportional* throughput allocation and *absolute priorities* in 802.11 wireless networks. The method is efficient, scalable, and fair. It builds on the idea of the *Idle Sense* method that provides near optimal throughput and fairness for 802.11 WLANs [40]: each station adjusts its contention window based on the observed average number of idle slots before each transmission. All active stations attempt collectively to maintain this number of idle slot near a given target value that correspond to near optimal channel utilzation. We achieve absolute priority differentiation by setting the target value for the number of idle slots to a small value, so that the contention window of the absolute priority class converges to a small value. Consequently, it captures the medium. The method also supports relative proportional throughput allocation in which several classes share the available throughput according to desired ratios. We also evaluate the efficiency of our prioritized access method via simulations.

### 1.1.2 Fairness in Wireless Ad-hoc Networks

When several nodes contend for transmission of their data packets on the shared medium, ideally they need to obtain the same throughput (throughput-fairness) or the time-share of the channel (time-fairness). The IEEE 802.11 DCF (*Distributed Coordination Function*) access method does not support fairness in some spatial configurations, in which some nodes are not in the receiving range of each other.

In Chapter 4, we consider the problem of *blocked stations* that appears in some spatial configurations of multi-hop wireless networks based on the 802.11 DCF. The problem leads to starvation of at least one station caused by the presence of neighbor stations within its carrier sensing range that do not hear each other. When a station is almost always blocked by other nodes, the performance of DCF is unsatisfactory in terms of fairness. We propose *Forced Transmissions*, a simple and efficient solution to this problem. It consists of detecting that a station is blocked by other stations and forcing a transmission. This results in a collision that increases the contention windows of blocking stations and leaves some channel time to the blocked station for transmitting. The blocked station forces transmission only with some probability adjusted in function of the time spent waiting for the channel.

We evaluate our MAC solution in terms of fairness and the throughput of the blocked station. The *Forced Transmissions* method guarantees a minimum throughput for the blocked station and consequently increases the fairness.

### 1.1.3 Throughput in Multi-hop Wireless Networks

DCF works relatively well in a wireless LAN in which all stations can directly communicate with each other. However, its performance dramatically degrades in multi-hop wireless networks due to interference between contending traffic flows. We will study the main problems that lead to the performance degradation of DCF in Chapter 2.

Although all variants of 802.11 LANs provide several non-interfering channels, the legacy DCF uses only one channel and all nodes in the network have to use a common channel to communicate. However, we can extend 802.11 networks to use multiple channels. In this way, two neighbor flows can be transmitted simultaneously on two separated channels. These parallel transmissions highly increase throughput of wireless networks. Deploying multi-channel could be particularly useful in a multi-hop wireless network in which the probability for interfering traffic flows is quite high. Nevertheless, the main problem in the design of an efficient multi-channel protocol is that a station wanting to communicate with a second station has to know the channel currently used by its counterpart. Otherwise transmitted frames do not reach the destination. In Chapter 2 of the thesis, we study specific problems of multi-channel MAC protocols. Furthermore, a multi-channel protocol for multi-hop 802.11 networks has to be scalable, which means that the performance of the network should not dramatically degrade for an increasing number of active stations or number of hops.

We consider wireless mesh networks (WMN) that provide cheap connectivity over an interconnection of wireless links. The infrastructure of this networks is formed by stationary mesh routers. In Chapter 5, we propose a multi-channel MAC mechanism for wireless mesh networks. We present a novel view on packet forwarding in wireless mesh networks by adopting a molecular analogy—mesh routers can be nuclei or electrons in an atom. Our mechanism, *Molecular MAC*, takes advantage of dynamic channel switching at neighbor mesh routers to efficiently forward packets over multiple hops. Molecular MAC defines how nuclei *notify* electrons about pending packets and how electrons *pull* them from nuclei for further forwarding. We evaluate the proposed scheme through simulation and compare with other proposals.

## 1.2 Thesis Outline

The next chapter is dedicated to the state of the art. We explore the principal concepts necessary to understand our contributions in the thesis. We review the IEEE 802.11 standard for wireless LANs, the DCF access method, and the performance problems of DCF in a multi-hop network. Supporting quality of service at the MAC layer in a wireless LAN is also discussed in this chapter. As the use of multiple non-interfering channels results in higher throughput, we review different multi-channel MAC protocols already proposed for 802.11 networks. We present our contributions in Chapters 3, 4, and 5. Finally, Chapter 6 concludes the thesis and explores the perspectives and future work inspired by this thesis.

# Part I

# Context and State of the Art

Chapter 2

# IEEE 802.11 Wireless Networks: *overview and some challenges*

## 2.1 Introduction

Wireless networking refers to technology that enables two or more computers to communicate over radio. However, wireless networks are challenging in many respects. Compared to their wired counterparts, they offer a less reliable communication medium and lower bandwidth. In addition, the realized throughput of wireless communications—after accounting for the effects of multiple access, fading, noise, and interference conditions, etc.— is often much less than a radio's maximum transmission rate.

In this chapter we present the principles as well as the challenges of wireless networks, which are necessary to better understand our contributions in the next chapters of the thesis. In the next section, we briefly describe the IEEE 802.11 standard for wireless networks. In Section 2.3 we present the most well-known architectures for wireless networks such as infrastructure-based wireless LANs, wireless ad-hoc networks, and wireless mesh networks. Section 2.4 is dedicated to the MAC layer of 802.11 networks. We explain the basic concepts of 802.11 DCF as well as the performance problems associated with the access method. In the same section, we also review the most well-known proposals for improving performance of DCF in a single-cell wireless network. We describe the *Idle Sense* access method in more details, because our contribution that deals with priority access builds upon it. The *Idle Sense* method modifies the behavior of DCF in order to make it more efficient and much more fair.

In Section 2.5 we review the IEEE 802.11e [48] standard that provides QoS in wireless LANs. Considering this aspect is important for multimedia applications that require some QoS support such as guaranteed bandwidth, delay, jitter, and error rate. Guaranteeing QoS requirements in 802.11 WLANs is especially challenging due to the noisy and variable physical layer characteristics. Quality of service has been already explored in wireless networks. In this thesis we are especially interested in studying the solutions that provide quality of service at the MAC layer. In addition to the 802.11e standard, we also discuss several related works.

In Section 2.6 we investigate the behavior of DCF in the context of multi-hop wireless networks. When these networks use 802.11 wireless links, several problems may arise due to the spatial positions of stations. We describe several well-known spatial problems: *hidden, exposed, blocked*, and *masked* stations that lead to a drastic degradation of performance.

To increase network capacity, a thoroughgoing solution is to enable a station to operate on multiple non interfering channels rather than on a single fixed channel. In Section 2.7 we study specific problems related to the use of multiple channels in wireless networks. Moreover, we classify already proposed solutions based on their characteristics. Finally, we conclude the chapter.

## 2.2 IEEE 802.11 Standards

Although many technologies and standards for wireless LANs were developed in the 1990's, the IEEE 802.11 standard set [44] has clearly emerged as the winner. The original standard proposed in 1997 specifies the medium access control (MAC) and physical layer (PHY) in wireless LANs. IEEE 802.11 was developed by the IEEE LAN/MAN Standards Committee (IEEE 802) in the 5 GHz and 2.4 GHz public spectrum bands.

In the following, at first we briefly describe the functionality of the physical layer in 802.11 networks. We then review various amendments to 802.11 standard with respect to their physical characteristics.

### 2.2.1 Physical Layer Functionality

The physical layer provides the MAC layer with an interface to the wireless media that transmits and receives signals. The PHY layer:

1. provides a frame exchange between the MAC and PHY under the control of the physical layer convergence procedure (PLCP) sublayer. This sublayer simplifies provision of a PHY service interface to the MAC services;

2. uses signal carrier and spread spectrum modulation to transmit data frames over the media under the control of the physical medium dependent (PMD) sublayer;

3. provides a carrier sense indication back to the MAC to verify activity on the media and trigger the frame reception procedure if needed.

During transmission, a MAC data frame is appended to a PLCP preamble and a header to create a physical layer frame. PLCP preamble (SYNC + Start of Frame Delimiter) allows the receiver to acquire the signal and synchronize itself with the transmitter. The PLCP header contains information about modulation scheme, transmission rate, length of payload, etc. For example, Figure 2.1 shows the format for the long physical frame in 802.11b LANs. As shown in the figure, the PLCP preamble and header are sent at 1Mbit/s.

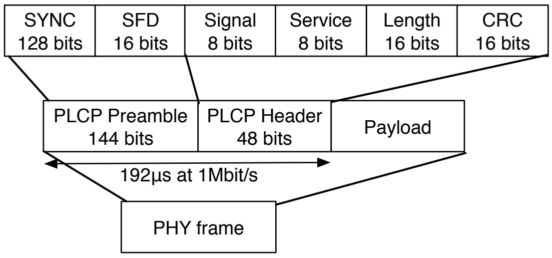



Figure 2.1: Physical frame format in 802.11b

### 2.2.2 Physical Characteristics of 802.11a/b/g LANs

After 802.11-1997 [44], the first wireless networking standard, several amendments such as 802.11a [45], 802.11b [46], and 802.11g [47] were added to improve the performance of 802.11 LANs . A new amendment, 802.11n [49], is under development. Table 2.1 shows the characteristics of 802.11 family standards.

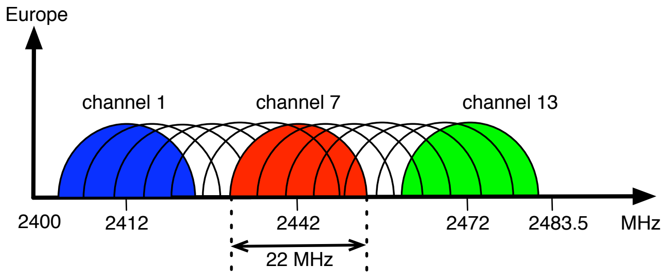
Table 2.1: Summary of physical characteristics of IEEE 802.11 variants

| Standard | Frequency Range (GHz) | Data Rate (Mbit/s) | Date |
|---|---|---|---|
| 802.11 | 2.4-2.4835 | up to 2 | 1997 |
| 802.11a | 5.1-5.8 | up to 54 | 1999 |
| 802.11b | 2.4-2.4835 | up to 11 | 1999 |
| 802.11g | 2.4-2.4835 | up to 54 | 2003 |
| 802.11n | 5.1-5.8 and/or 2.4-2.4835 | up to 300 | in progress |

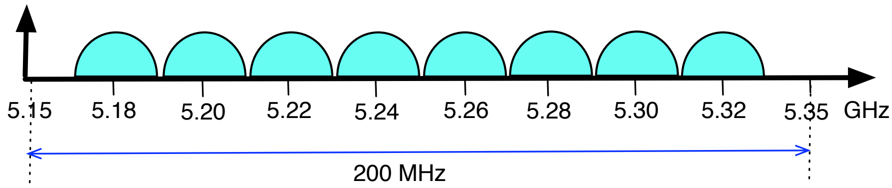
The 802.11b/g LANs use the 2.4 GHz ISM (Industrial, Scientific and Medical) band. The 802.11 standard divides this frequency band into 14 channels each of width 22 MHz, spaced only 5 MHz apart with channel 1 centred on 2412 MHz and 13 on 2472 MHz (cf. Figure 2.2(a)). The 14th channel of width 12 MHz is a special one only used in Japan and it is placed above channel 13.

As orthogonal channels require 25 MHz of channel separation, adjacent channels overlap and will interfere with each other. Therefore to be totally non-overlapping, channels 1, 7 and 13 (in Europe) are typically used for communication in current implementations [46]. However, in a large-scale building or campus-wide installation, using only three channels can increase the interference with neighboring 802.11b/g networks.

802.11a uses the 5 GHz U-NII (Unlicensed National Information Infrastructure) band that offers 12 non-overlapping channels: 8 in the lower part of the frequency band for indoor uses and 4 in the upper part for outdoor uses [45]. Figure 2.2(b) shows the 8 non-overlapping channels for 802.11a LANs.




(a) 802.11b/g channels in 2.4GHz band




(b) 802.11a channels in 5GHz lower band

Figure 2.2: Channel spacing for 802.11b/g and 802.11a LANs respectively in 2.4GHz and 5GHz frequency band.

The fist IEEE 802.11 standard provided for three PHY specifications including infrared (IR), 1-2 Mbit/s frequency hopping spread spectrum (FHSS), and 1-2 Mbit/s direct sequence spread spectrum (DSSS). The DSSS method uses baseband modulations of differential binary phase shift keying (DBPSK) and differential quadrature phase shift keying (DQPSK) to provide 1 and 2 Mbit/s data rates, respectively. 802.11b uses a higher rate extension of DSSS (HR/DSSS) to provide 5.5 and 11 Mbit/s in addition to 1 and 2 Mbit/s data rates. The HR/DSSS incorporates a more efficient coding scheme known as complimentary code keying (CCK) to attain higher data rates. The PLCP preamble and header, however are usually sent at a lower rate compared to the data payload.

802.11a utilizes a multi-carrier modulation technique known as orthogonal frequency division multiplexing (OFDM). The OFDM provides data rates of 6, 9, 12, 18, 24, 36, 48, and 54 Mbit/s. As mentioned earlier, a 802.11g LAN operates in the same lower frequency band as 802.11b. However, it provides the higher-speed transmission rates of 802.11a. Moreover, to be backward compatible with 802.11b, 802.11g also provides the transmission rates of 802.11b. To achieve these goals, 802.11g LANs adopt HR/DSSS and OFDM modulation technics.

An issue for the 802.11a LANs is that by operating at a higher frequency, they have a shorter transmission distance for a given power level and suffer more from multi-path propagation. On the other hand, the 802.11b/g LANs compete for frequency spectrum with microwaves oven and bluetooth devices.

IEEE 802.11n is an under-development amendment that improves upon the previous 802.11 standards by adding multiple-input multiple-output (MIMO) and Channel-Bonding operation to the physical layer and frame aggregation to the MAC layer. MIMO uses multiple transmitter and receiver antennas to improve performance. Channel Bonding increases the amount of data to transmit by simultaneously using two non-overlapping adjacent channels. 802.11n is expected to be finalized in November 2009 [50], although many "Draft N" products are already available.

## 2.3 Wireless Network Architectures

Today, wireless networks have various architectures based on user requirements and network deployment schemes such as infrastructure-based WLANs, ad-hoc networks, wireless mesh networks, and wireless sensor networks (we do not consider the latter in this thesis).

### 2.3.1 Infrastructure-based Wireless LANs

According to the IEEE 802.11-1999 standard, the *basic service set* (BSS) is the main building block of an IEEE 802.11 wireless LAN. A BSS contains one or more wireless stations (STA) and a central base station, known as an *access point* (AP). Figure 2.3 shows the AP in each of two BSSs connecting to a router, which in turn leads to the Internet. Here, the access point acts as a bridge . However it can also be the router at the same time.

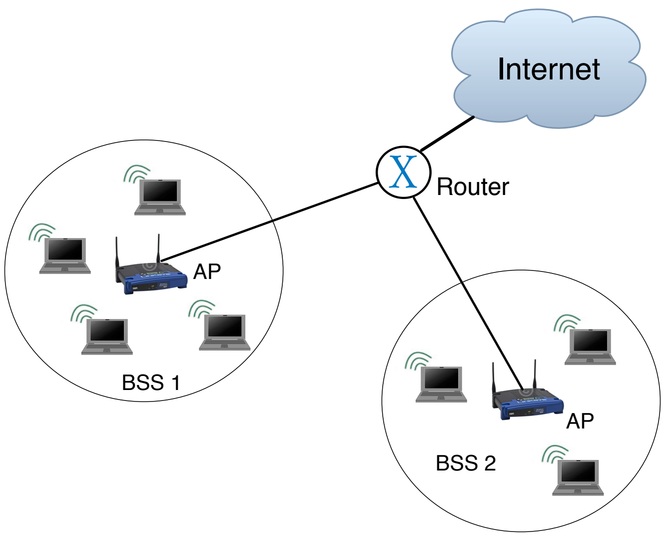



Figure 2.3: IEEE 802.11 Infrastructure WLAN

Most wireless LANs have a similar centralized architecture: stations communicate directly with an access point connected to the wired network. Two stations in such a wireless LAN cannot directly exchange data. Instead, they forward their frames to the AP while in turn sends the frame to the intended receiver. However, in 802.11e [48] the direct link protocol (DLP) provides a mechanism to allow direct station-to-station communication in the case where two communicating stations are within the transmission range of each other.

To access all the service of an infrastructure BSS, a STA at first has to associate with the corresponding AP. To achieve this goal, it needs to get some information from the AP. The station can get it by one of two means:

- Passive Scanning: In this case the station just waits to receive a beacon frame from the AP. The beacon is a periodic frame sent out by the access point containing informations like: service set identifier (SSID), time stamp, beacon period, MAC address of AP, etc.

- Active Scanning: In this case, the station tries to discover an access point by transmitting *probe request frame*s and waits for *probe response* from the AP.

In 802.11, an Extended Service Set (ESS) is a set of infrastructure BSSs, where the APs communicate with each other to forward traffic from one BSS to another and to facilitate the movement of mobile stations from one BSS to another. An abstract medium known as Distribution System (DS) enables the wireless interconnection among access points in 802.11 networks. Such an architecture is also called *wireless distribution system* (WDS). To network equipment outside of the WDS, the WDS and all of its mobile stations appears to be a single MAC-layer network where all stations are physically stationary. Thus, the WDS hides the mobility of the mobile stations from everything outside of the WDS.

### 2.3.2 Ad-hoc Wireless Networks

An ad hoc wireless network is a collection of wireless nodes that self-configure to form a network without the need for any established infrastructure, as shown in Figure 2.4. Without infrastructure, the stations handle the necessary control and networking tasks by themselves, generally through the use of distributed control algorithms [36].

Connectivity is the fundamental aspect of any network. A network is connected if every node is able to communicate with every other node. A single-hop wireless network in which all stations are within the transmission range of each other is connected. Figure 2.4 shows an ad hoc network in which some nodes are not within the transmission range of each other. As a result, each node cannot directly communicate with all other nodes. However, a node can forward data on behalf of other nodes. For example, node A forwards its frame with destination of C to node B. Node B then redirects the frame to its final destination C. Such a wireless network is called *multi-hop*, because it uses two or more wireless hops to convey data from a source to a destination.

When stations are mobile, a multi-hop ad-hoc wireless network is known as Mobile Ad-hoc NETwork (MANET [29]). One of the most noticeable characteristics of a MANET is its dynamic topology. As nodes are free to move arbitrarily, the network topology may

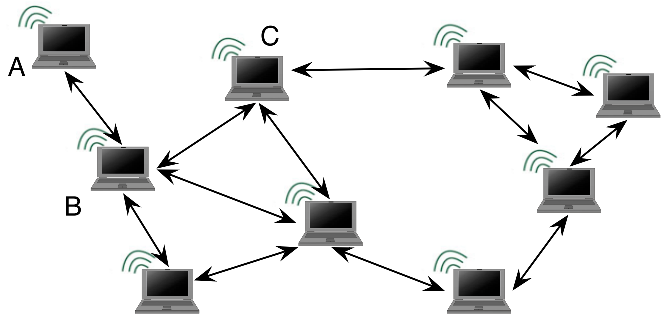



Figure 2.4: Ad-hoc Wireless Network

change randomly and rapidly at unpredictable times and may consist of both bidirectional and unidirectional links.

The decentralized nature of wireless ad hoc networks makes them suitable for a variety of applications where central nodes cannot be relied on and may improve the scalability of wireless ad hoc networks compared to infrastructure-based wireless networks. However, theoretical [63] and practical [38] limits to the overall capacity of such networks have been identified. The lack of infrastructure is highly desirable for military systems and emergency situations like natural disasters, where communication networks must be configured quickly as the need arises, often in remote areas.

The presence of a dynamic and adaptive routing protocol will enable ad hoc networks to be formed quickly. The Ad-Hoc On-Demand Distance Vector (AODV) [77] and Dynamic Source Routing (DSR) [53] are two examples of reactive routing protocols that are used in MANETs. Reactive protocols seek to set up routes on-demand. If a node wants to initiate communication with a node to which it has no route, the reactive routing protocol will try to establish such a route. The main disadvantages of reactive algorithms is high latency time in finding routes.

The Optimized Link State routing (OLSR) [28] is a proactive approach to MANET routing. Proactive protocols seek to maintain a constantly up-to-date topology knowledge. Every node learns about other nodes in the network and ways to reach them. As time goes on, each node knows about all other nodes and one or more ways to reach them. The main advantage of such a protocol is that routes will always be available beforehand. To maintain the up-to-date routing information, topology information needs to be exchanged between the nodes on a regular basis, leading to relatively high overhead on the network.

### 802.11 ad-hoc networks

In 802.11 networks, the ad hoc mode is accomplished using the Independent Basic Service Set (IBSS). With an IBSS, there are no access points and stations communicate directly with each other Contrary to infrastructure-based wireless LANs, all stations that join the ad hoc network must periodically send a beacon. Stations use these beacons to discover their neighborhood. Hence, wireless devices within the range of each other can discover and directly communicate without involving central access points. This method is typically used by two computers so that they can connect to each other to form a network.

### 2.3.3 Wireless Mesh Networks

Wireless mesh networks (WMNs) consist of mesh routers and mesh clients, where mesh routers are stationary and form the infrastructure of WMNs. They provide network access for mesh clients. The infrastructure part of a WMN has a relatively stable topology except for the occasional failure of nodes or addition of new nodes.

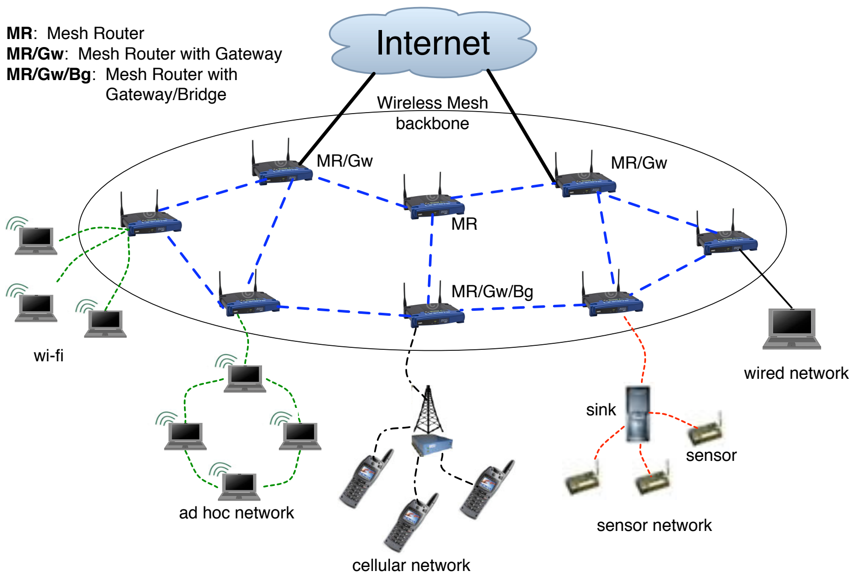



Figure 2.5: Wireless Mesh Network Architecture

A wireless mesh network can be seen as a type of wireless ad hoc network in which the objective is to extend the coverage range without sacrificing channel capacity [5]. Mesh architecture sustains signal strength by breaking long distances into a series of shorter hops. Intermediate nodes not only boost the signal, but cooperatively make forwarding decisions based on their knowledge of the network, i.e. perform routing. Such an architecture, if designed carefully, may provide promising bandwidth, spectral efficiency, and economic advantage over the coverage area.

A typical architecture for WMN is shown in Figure 2.5 , where dash and solid lines indicate wireless and wired links, respectively [5]. Wireless mesh routers form an infrastructure for clients to connect to them. The WMN infrastructure can be built using various types of radio technologies, in addition to the mostly used IEEE 802.11 technologies. Mesh routers form a mesh of self-configuring, self-healing links among themselves. Moreover, mesh routers can be connected to the Internet.

Mesh routers can be equipped with multiple radios to perform routing and access functionalities. This enables separation of two main types of traffic in the wireless domain. While routing and configuration are performed between mesh routers, the access to the network by end users can be carried out on a different radio. This significantly improves the capacity of the network.

However, in this thesis we consider a wireless mesh network of mesh routers and in Chapter 5 we propose an efficient MAC to improve performance of such a network.

## 2.4 Medium Access Control

The Medium Access Control (MAC) is a mechanism that makes it possible for several stations connected to the same physical medium to share it and to communicate with each other. Since 802.11 networks use a random access method, in this section we focus on this kind of MAC protocols.

In a random access protocol, users attempt to access the channel in an uncoordinated manner. A transmitting node always uses the entire capacity of the channel during its transmission. When there is a collision, each node involved in the collision retransmits its frame until the frame gets through without collision. However, the transmitter does not necessarily retransmit its collided frame immediately, but after a independently chosen random delay. The node that chooses the least random delay will at first transmit its frame.

The ALOHA [2, 3] and the Carrier Sense Multiple Access (CSMA) [56, 68] protocols are the most popular random access mechanisms. The first ALOHA protocol [2], also known as pure ALOHA is fully decentralized: the transmitting node sends the frame on the channel upon its arrival to the MAC layer. If collision occurs, the node will immediately retransmit the frame with probability $p$. Otherwise, the node waits for a frame transmission time and then it transmits a new frame with probability $p$ or waits for another frame transmission time with probability $1 - p$. In slotted Aloha protocol [3], all nodes synchronize their transmissions to start at the beginning of a slot.

In ALOHA protocols, a node starts to transmit regardless of whether other nodes are transmitting. However, before speaking, a polite person *listen*s if someone else is speaking and waits until they are finished. CSMA protocols propose more civilized mechanisms by following the polite human conversation rules. CSMA is a MAC mechanism in which a node verifies the absence of another transmission trying to access the shared medium. *Carrier Sense* means that a transmitter listens for a carrier signal before trying to send. If a carrier is sensed, the node waits for the end of outgoing transmission before initiating its own transmission.

Another rule in conversation is that if someone else starts talking at the same time, you should stop talking. In the network world this rule is known as *collision detection* that is embodied in CSMA/CD (Collision Detection). Ethernet [68] uses a CSMA/CD protocol. Collision detection improves the performance of CSMA by terminating transmission as soon as a collision is detected, which reduces the probability of a second collision on retry. A transmitting data station that detects another signal while transmitting a frame, stops transmitting that frame, transmits a jam signal, and then waits for a random time interval before trying to retransmit its frame.

CSMA/CA (Collision Avoidance) is also one of the well-known enhancement to the pure CSMA. Collision avoidance is used to improve the performance of CSMA by attempting to be less *greedy* on the channel. If the channel is sensed busy before transmission, then the transmission is deferred for a random interval. This reduces the probability of collisions on the channel. Collision avoidance also refers to use of Request-To-Send (RTS) and Clear-To-Send (CTS) control frames [55]. Before data transmission, a sender transmits an RTS and the receiver replies with an CTS. These control frames are significantly smaller than the data frames. collisions between RTS frames are much less costly than the collisions that would otherwise occur between data frames.

In wireless LANs, collision detection is difficult, because even if a sender can receive and transmit at the same time, it receives its proper signal with a much higher power than the signal of a remote transmitter. Inspired by the huge success of the Ethernet and its random access protocol, the designers of IEEE 802.11 chose a random access protocol based on CSMA/CA for 802.11 wireless LANs.

### 2.4.1 802.11 DCF

The 802.11 standard defines an access method –DCF, a derivative of CSMA/CA. DCF is defined as mandatory in subclause 9.2 of the IEEE 802.11 standard and is the defacto default setting for Wi-Fi hardware. The MAC layer is the same for all variants 802.11a/b/g, but parameters are different.

Wireless links are prone to errors and collisions: when a station sends a frame, the frame may be incorrectly received for a variety of reasons like attenuation of signal strength, collisions, multipath propagation, etc. On the other hand, high packet loss rate is detrimental to transport-layer performance. Therefore, mechanisms are needed to reduce packet loss rate experienced by upper layers. DCF uses *Automatic Repeat re-Quest*(ARQ), an error control method for data transmission based on acknowledgments and timeouts to achieve reliable data transmission. The DCF ARQ is a positive acknowledgment scheme: the destination acknowledges with a control frame a correctly received frame. If the sender does not receive an acknowledgment before a timeout, it retransmits the frame or abandons if it exceeds a predefined number of retransmissions.

Figure 2.6 illustrates the basic operation of DCF. As shown in the figure, we assume that initially the channel is idle. DCF requires a station wishing to transmit to listen for the channel status for the Distributed Inter-frame Space (DIFS) interval. If the channel is sensed busy during the DIFS interval, the station defers its transmission. Otherwise, it transmits its data frame. If the receiver is able to correctly decode the received frame, it sends back an ACK frame after a short period of time known as Short Inter-frame Space (SIFS). Note that DCF acknowledgment is only used for the unicast frames and not for the broadcast ones.

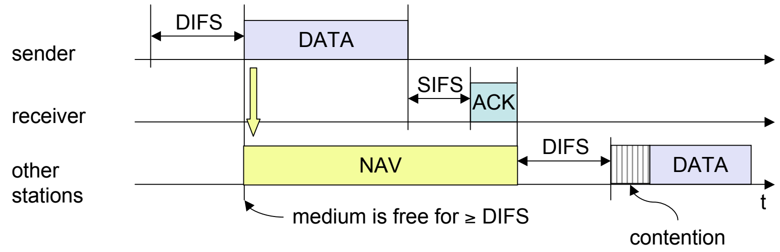



Figure 2.6: Basic operation of IEEE 802.11 DCF

The PLCP header of each frame contains a *duration* field that specifies the transmission time required for the frame during which the medium will be busy. Stations listening to the wireless medium read the *duration* field and set their Network Allocation Vector (NAV), which is an indicator for the station of how long it must defer from accessing the medium. This mechanism, also known as *virtual carrier sensing*, can limit the need for physical carrier sensing at the radio interface in order to save power.

In a network with a number of stations contending for the channel, when several stations sense the channel busy and defer their access, they will try to seize the channel after transmission. As a result, collisions may occur. In order to avoid such collisions, DCF specifies a random backoff that forces a station to defer its access to the channel for an extra period (cf. Figure 2.6).

Each host maintains a contention window ($CW$) that is initially set to $CW_{\text{min}}$. A node wishing to transmit will wait until the channel becomes idle for more than DIFS. It then chooses a random backoff between 0 and $CW$ and starts decrementing the counter. The counter is decremented by one after each *time slot*, as long as the channel is idle (cf. Figure 2.7). If the channel becomes busy, the node will freeze the counter until the channel is free again. When the backoff counter reaches zero, the node will start to transmit. After each successful transmission, a transmitter resets its contention window to $CW_{\text{min}}$. On the other hand, after each unsuccessful transmission attempt the transmitter applies the exponential backoff algorithm:

$$CW = min(2 \times (CW + 1) - 1,\ CW_{\text{max}}). \tag{2.1}$$

$CW$ returns to its minimal value of $CW_{\text{min}}$ after a successful transmission. Moreover, the number of retransmissions for each frame is bounded to a limit, *retry_limit*. If a sender has failed to successfully transmit a frame after *retry_limit* attempts, it throws the frame away and resets its $CW$ to $CW_{\text{min}}$ for a new data frame.

If the data frame is corrupted as shown in Figure 2.8, it is not received by the destination and therefore no ACK is sent back to the sender. At the end of ACK timeout, the sender doubles its $CW$ and chooses a new backoff. The sender, then waits for a DIFS before contention to access the medium. However, any station listening to the medium and perceiving a corrupted frame, waits for a period of time known as Extended Inter-frame Space (EIFS) which is equal to DIFS + ACK_Timeout. Figure 2.9 shows another

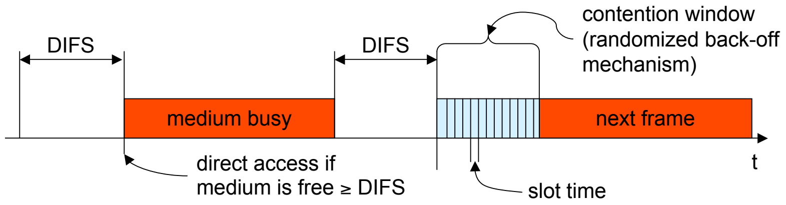



Figure 2.7: Contention in DCF

scenario, in which the ACK is corrupted. Although the data frame is received correctly at the destination, due to the failure of ACK, the sender cannot correctly receive the ACK. As a result, the sender will retransmit the frame after backoff from doubled $CW$. However, the receiver will drop this duplicated frame, because it has already received it.

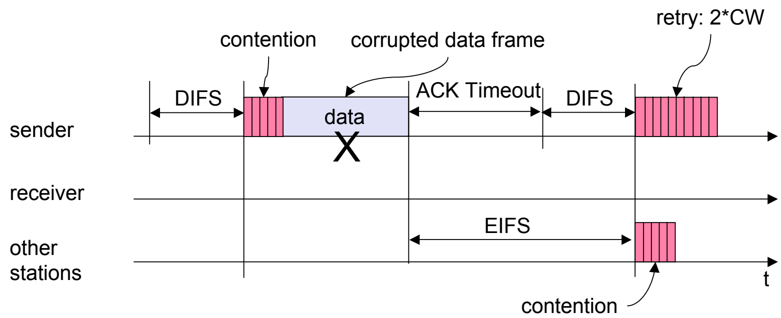



Figure 2.8: The behavior of DCF when a data frame is corrupted.

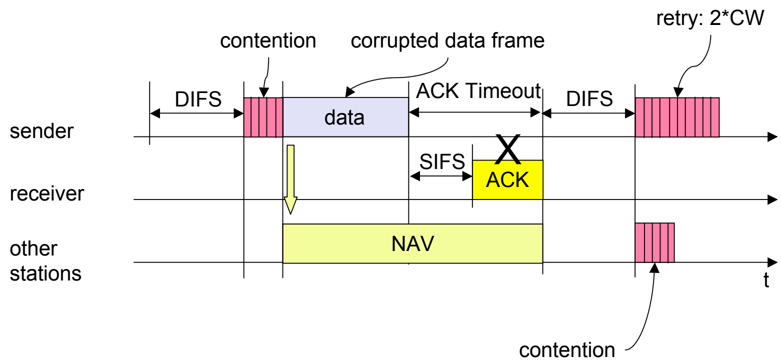



Figure 2.9: The behavior of DCF when an ACK frame is corrupted.

Figure 2.10 shows the behavior of DCF in a wireless LAN with 5 active stations. Due to the lack of space, ACK and SIFS are not shown in the figure. At the beginning the medium is idle. Station $S_3$ receives a packet from the upper layer and starts to sense the medium. A short while after, $S_1$ also starts to sense the medium in order to transmit its arriving packet. $S_3$ start to transmit after DIFS forcing $S_1$ to defer its transmission, because the medium becomes busy. At the end of $S_3$'s transmission there are three contending stations: $S_1$, $S_2$, and $S_5$. Each station draws a backoff and starts to decrement its backoff counter after DIFS. By choosing the lowest backoff, $S_2$ gains access to the medium and starts to transmits its data frame. Other stations, meanwhile, freeze their backoff counter. For the next turn, $S_1$ and $S_5$ continue to decrement their residual backoff counter while $S_4$ draws a new backoff which is the same as the residual backoff of the $S_5$. Thus, both of $S_4$ and $S_5$ start to transmit at the same time that leads to a collision. According to the exponential backoff mechanism in DCF, both stations double their $CW$s. $S_1$, not being able to decode the collided frames, waits for a EIFS and then continue to decrement its residual backoff. It gains access to the medium and transmits its frame.

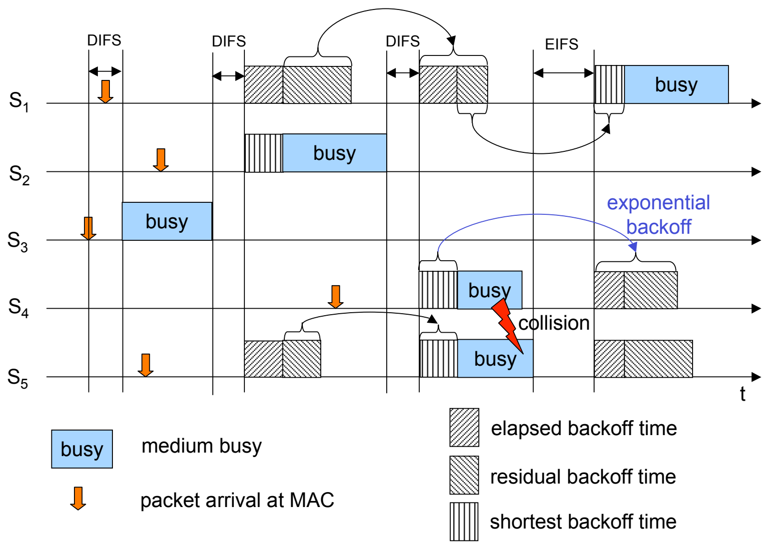



Figure 2.10: An example scenario that explains the operation of IEEE 802.11 DCF

Before transmitting any frame, a node has to wait for a small duration of time even if the channel is idle. This is called inter-frame spacing. Four different intervals enable each frame to have different priority when contending for the channel. SIFS, PIFS (Point coordination IFS), DIFS, and EIFS are the four inter-frame spacings in the order of their increasing length (cf. Figure 2.11). A node waits for a DIFS before transmitting a data frame, but waits for a SIFS before sending an ACK. Thus, an ACK frame will win the channel when contending with DATA frames because the SIFS duration is smaller than a DIFS. Table 2.2 gives the MAC parameters for 802.11 variants. The PIFS is used by the access point to gain access to the medium before any other station.

The original 802.11 MAC defines another coordination function called the Point Coordination Function (PCF): this is available only in the infrastructure mode, where stations are connected to the network through an AP. This mode is optional and only few APs or Wi-Fi adapters actually implement it. APs send beacon frames at regular intervals (usually every 0.1 second). Between these beacon frames, PCF defines two periods: the

Table 2.2: Summary of IEEE 802.11 MAC Parameters

| Variant | $CW_{\min}$ | $CW_{\max}$ | DIFS | SIFS | $T_{\text{slot}}$ |
|---|---|---|---|---|---|
| 802.11a | 15 | 1023 | 34 $\mu s$ | 16 $\mu s$ | 9 $\mu s$ |
| 802.11b | 31 | 1023 | 50 $\mu s$ | 10 $\mu s$ | 20 $\mu s$ |
| 802.11g | 15 | 1023 | 28 $\mu s$ | 10 $\mu s$ | 9 $\mu s$ |

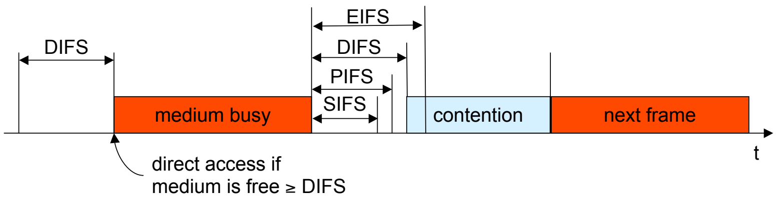



Figure 2.11: Different inter-frame spaces in IEEE 802.11 DCF and PCF.

Contention Free Period (CFP) and the Contention Period (CP). In CP, the DCF is simply used. In CFP, the AP sends Contention Free-Poll (CF-Poll) frames to each station, one at a time, to give them the right to send a frame. As the AP is the coordinator that decides which station can send a frame, this method supports quality of service.

### 2.4.2 DCF Problems in a Wireless LAN

The 802.11 DCF suffers from several performance problems. In this section, we explore some of the most well-known problems that may arise in a single-cell wireless LAN. Later, in Section 2.6, we will study performance problems of DCF in the context of multi-hop networks.

#### Rate Diversity and Performance Anomaly

In IEEE 802.11 standard the data rate (a physical level parameter) is available for tuning at the side of the wireless network interface card. In 802.11b, for example, it can be set to 1, 2, 5.5, and 11Mbit/s. Each rate corresponds to a different modulation scheme with its own trade-off between data throughput and signal quality. Several rate adaptation technics have been already proposed for wireless networks [42, 54].

Heusse *et al.* [41] analyzed the performance of the 802.11b wireless LANs and showed that rate diversity in wireless LANs leads to performance anomaly: when some hosts use a lower bit rate than the others, the performance of all hosts is considerably degraded.

Such a situation is a common case in wireless local area networks in which a host far away from an access point is subject to important signal fading and interference. To cope with this problem, the host changes its modulation type, which degrades its bit rate to some lower value. Typically, 802.11b products degrade the bit rate from 11 Mb/s to 5.5, 2, or 1 Mb/s when repeated unsuccessful frame transmissions are detected. In such a case, a host transmitting for example at 1 Mb/s reduces the throughput of all other hosts transmitting at 11 Mb/s to a low value below 1 Mb/s. The basic CSMA/CA channel access method is at the root of this anomaly: it guarantees an equal long term channel access probability to all hosts. When one host captures the channel for a long time because its bit rate is low, it penalizes other hosts that use the higher rate.

#### Bad Day Effect

In a 802.11 network, a sender cannot distinguish between a corrupted frame and a collision. Therefore, if a host looses frames due to bad transmission conditions, performs frequent exponential backoffs. An increase in contention window lowers the transmission attempt probability. This host then has lower chance to gain access the medium and to transmit its frame which results in unfairness among stations. This problem occurs when the MAC method adjusts the contention window after a collision.

#### Physical Layer Capture Effect

Consider 3 stations in a wireless LAN as shown in Figure 2.12. Assume that the distance A—B is shorter than C—B. Therefore, if all devices transmit with the same power, the A's signal at B is stronger than C's signal at B. Also, all stations are in the range of each other, i.e., there is no hidden station. Stations A and C transmit to B. Kochut *et al.* [57] showed that A's throughput is consistently higher than C's in the long term. A and C achieve equal performance when the other is not transmitting. Thus, the imbalance can only be caused by the interaction between A and C when both are transmitting simultaneously. In other words, the imbalance is due to the fact that collisions are resolved in favor of A. The phenomenon of the stronger frame in a collision being received successfully is known as *physical layer capture* effect.

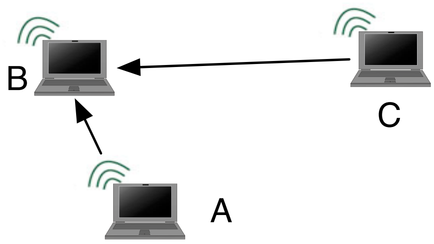



Figure 2.12: Physical layer capture effect in DCF.

### 2.4.3 Enhancements to the DCF

Several authors have extensively analyzed the performance of 802.11 DCF [16, 17]. Besides, much work proposes various enhancements to 802.11 DCF. Some of them consist of dynamic adjustment of $CW$. Bianchi *et al.* define a method for estimating the number of active hosts by means of a Kalman filter to set suitable values for *CW* [14]. Ma *et al.* proposed a way to overcome the complexity of such solutions by using a centralized approach: an access point measures the number of contending hosts and broadcasts the optimal value of *CW* [66].

Bononi *et al.* [15] proposed an improvement to 802.11 DCF called *Asymptotically Optimal Backoff* (AOB). AOB limits the impact of collisions in presence of many active stations by maintaining the transmission probability above a given threshold. Any station independently measure two load factors: the slot utilization and the average size of transmitted frames. AOB keeps the exponential backoff mechanism of DCF, so it does not completely decouple collision detection from load control.

Aad *et al.* have introduced a *Slow Decrease* method [72]. Their mechanism modifies the backoff algorithm of DCF and it consists in dividing *CW* by 2 instead of resetting it to initial value $CW_{\min}$ after a successful transmission. Although the method was intended for improving efficiency and fairness, it presents much worse fairness as shown in Figure 2.14(a). Kwon *et al.* have defined *Fast Collision Resolution* [61] which also modifies the backoff algorithm. The main idea is to double the contention window *CW* of any host that either experiences a collision or loses a contention; it then draws a new backoff counter. To decrease the time spent in backoff, hosts can exponentially decrease their backoff timer after observing a number of empty slots. This method presents a significant improvement of throughput compared to the standard 802.11 DCF method. However, as only the host that has just succeeded a transmission decreases its *CW* to the minimal value, the method causes high short-term unfairness. The *Fairly Scheduled Fast Collision Resolution* (FS-FCR) variant [61] addresses this issue by setting a limit on the number of successive retransmissions that a host may perform: when a station reaches the limit, it sets its *CW* to $CW_{\max}$.

The *Binary Countdown Method* [103] can reduce collision overhead. As collisions significantly limit throughput, the method is more efficient than the standard 802.11 DCF. However, it requires a control channel for transmitting management messages to schedule each transmission.

[9] proposed a backoff control mechanism based on the MILD (*Multiplicative Increase, Linear Decrease*) principle: the backoff counter is increased by a factor upon a collision and decreased by 1 after a successful transmission. The method uses the same value of backoff counters for all hosts, which is distributed in the packet header and copied by the receiver into its own counter. Song *et al.* proposed a backoff algorithm based on the EIED (*Exponential Increase Exponential Decrease*) principle: *CW* is increased by a factor upon a collision and decreased by another factor after a successful transmission [89]. Their method compares favorably with MILD and the standard exponential backoff of 802.11 DCF.

The main problem with all these proposals is related to the core principle of the dynamic control of channel access probability: in all the methods a host increases its contention window after it experiences a collision, which is considered as a signal to decrease the rate of transmission attempts. This leads to degraded performance, because the methods cannot distinguish collisions from corrupted frames nor can they handle the capture effect [57] correctly: when a method adjusts the contention window after a collision, its operation is not optimal.

A different approach is represented by Tan *et al.* who proposed placing a regulator above the MAC layer in an access point to control the cell and set rates for hosts according to some performance objective (throughput or time-fairness) [91]. Although the idea is interesting from the point of view of providing equal time shares to hosts, it relies on a central coordinator.

### 2.4.4 Idle Sense Access Method

The main source of short-term unfairness in the DCF access method is the exponential backoff algorithm applied after a collision [40]: colliding hosts double their contention windows and have higher probability of choosing a larger backoff during which other hosts will benefit from channel access. This also means increased delay for hosts that doubled their $CW$. In this way, the standard DCF method controls the load on the channel by reducing the number of contending hosts, because the hosts that have failed their transmission are likely to attempt to access the channel later on. Moreover, hosts consider all failed transmissions as collisions, whereas only a part of them are really collisions. So, DCF bases its load control on a biased indicator, which leads to lower performance and increased unfairness.

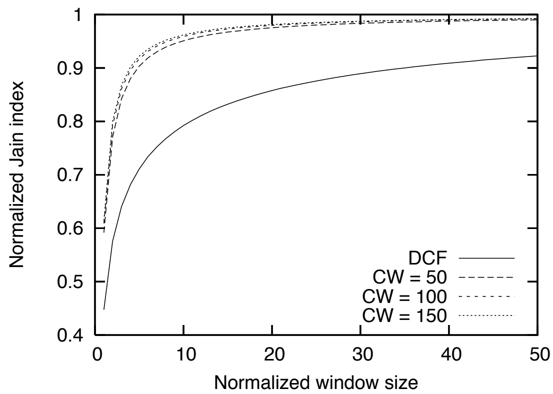



Figure 2.13: Fairness of 802.11 DCF with equal contention windows in a wireless LAN of 8 active stations.

Heusse *et al.* evaluated the short-term fairness of a modified 802.11 DCF with $CW$ equal for all hosts by using the sliding window method. The method considers the patterns of transmissions and computes the average Jain fairness index in a window of an increasing size [59]. It is defined as follows: let $\gamma_i$ be the fraction of transmissions performed by host $i$ during window $w$; the fairness index is the following:

$$F_J(w) = \frac{(\sum_{i=1}^{N} \gamma_i)^2}{N \sum_{i=1}^{N} \gamma_i^2}. \tag{2.2}$$

Perfect fairness is achieved for $F_J(w) = 1$ and perfect unfairness for $F_J(w) = 1/N$.

The definition of window $w$ also should take into account $N$, the number of competing hosts. The authors propose to normalize the window size with respect to the number of hosts and compute the Jain index for the window sizes which are multiples of $N$. $m$ is a *normalized window size* such that $w = m \times N, m = 0, 1, 2, ....$ The Jain index will be computed as $F_J(m)$.

Figure 2.13 compares the Jain fairness index for the standard 802.11 DCF access method with the case in which $CW$ is kept equal for all hosts. We can observe that equal $CW$ results in significant improvement of short-term fairness. Moreover, the short-term fairness of 802.11 becomes worse for an increasing number of stations due to the exponential backoff mechanism.

*Idle Sense* optimizes 802.11 DCF for high throughput and fairness: contending stations do not perform the exponential backoff algorithm after collisions or failed transmissions, rather they make their contention windows dynamically converge in a fully distributed way to similar values solely by tracking the number of idle slots between consecutive transmissions. The method works as follows: each station measures $n_i$, the number of consecutive idle slots between two transmission attempts. Every $maxtrans$ transmissions, it estimates $\hat{n}_i$, the average of observed values of $n_i$. Then, it uses $\hat{n}_i$ to adjust its contention window to the target value $n_{\mathrm{i}}^{\mathrm{target}}$, computed numerically for a given variant of IEEE 802.11 PHY and MAC parameters—its value is 5.68 for IEEE 802.11b and 3.91 for IEEE 802.11g. When stations adjust their $CW$ so that $n_i$ converges to ${n_i}^{\mathrm{target}}$, their throughput is optimal.

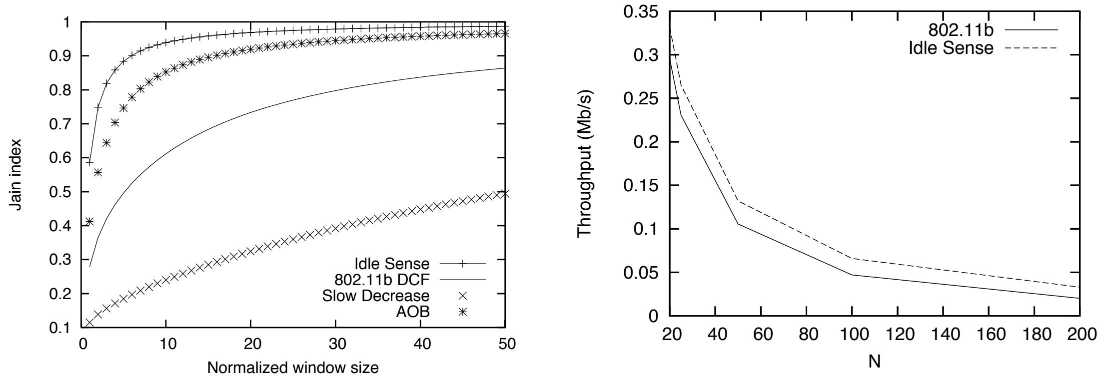
The *Idle Sense* adaptation algorithm makes $\hat{n}_i$ converge to $n_i^{target}$ by applying AIMD (*Additive Increase Multiplicative Decrease*) [27] to the contention window $CW$ as follows:

- If $\hat{n}_i \geq n_i^{target}$ then $CW \leftarrow \alpha \cdot CW$
- If $\hat{n}_i < n_i^{target}$ then $CW \leftarrow CW + \epsilon$

where $\epsilon$ and $\alpha$ are some adaptation parameters. If a station observes too many idle slots compared to the target, it needs to increase $CW$ additively, which in turn will decrease $n_{\mathrm{i}}$, whereas if it observes too few idle slots, it needs to decrease $CW$ in a multiplicative way, which in turn will increase $n_{\mathrm{i}}$. For more details about *Idle Sense* we refer the reader to other papers [40, 65, 37].

As mentioned in the previous section, when the physical layer capture effect takes place, the host that detects a collision performs exponential backoff thus reducing its transmission opportunity. The other host succeeds on two fronts —it transmits its frame and continues to operate using the initial contention window. The *Idle Sense* access method alleviates this effect, because a host does not adjust its $CW$ when perceiving a collision. For the same reason, bad day effect is also avoided by *Idle Sense*. It is straightforward to solve the "performance anomaly problem" by scaling the CW according to the transmission rate. This is a principle that is also at the core of our subsequent proposal for proportional channel access priorities.

As *Idle Sense* relies on only observing idle periods in channel activity, it is insensitive to all the problems that arise in methods based on inferring the channel load from collisions. Heusse *et al.* evaluated short-term fairness by using the normalized Jain fairness index [52, 59]. The simulation result [40] from Figure 2.14(a) shows that *Idle Sense* provides much better short-term fairness than 802.11b and other proposed modifications of 802.11 DCF such as *Slow Decrease* and *AOB*. Another advantage of *Idle Sense* is its improvement in throughput for an increasing number of hosts (cf. Figure 2.14(b)).




(a) Fairness comparison for $N = 50$ competing hosts

(b) Throughput comparison

Figure 2.14: Comparison of Idle Sense with other MAC protocols [40].

## 2.5 Quality of Service in 802.11 Wireless LANs

Since wireless LANs are the successors of wired LANs, they are expected to support the same applications as the wired LANs they are replacing. However, e.g. the MAC mechanisms of the original IEEE 802.11 standard do not provide a service differentiation mechanism to guarantee a lower bound on throughput or an upper bound on delay. In DCF all stations have almost equal probability to access the channel and share it according to equal frame rate and not according to equal throughput. This offers no support for priority access to the channel for time-sensitive traffic. Even though the PCF can offer some sort of priority to an overloaded station, it cannot differentiate between traffic types or sources. Therefore, it cannot tell which stations have long queues of time-sensitive traffic, and which only hold best-effort traffic.

Due to substantial demand for the transmission of delay sensitive video and audio data, 802.11E task group proposed 802.11e [48] amendment to 802.11 standard. The QoS features of the 802.11e are beneficial to prioritize for example voice and video traffic over more elastic data traffic. The 802.11e enhances the DCF and the PCF through a new coordination function: the Hybrid Coordination Function (HCF). Within the HCF, there are two methods of channel access similar to those defined in the legacy 802.11 MAC: HCF Controlled Channel Access (HCCA) and Enhanced Distributed Channel Access (EDCA). Both EDCA and HCCA define traffic classes. For example, emails could be assigned to a low priority class, and voice could be assigned to a high priority class.

### 2.5.1 802.11e EDCA

EDCA extends the CSMA/CA-based medium access scheme of IEEE 802.11 DCF by introducing prioritized medium access for different traffic categories. In order to allow prioritized and separate handling of traffic, 802.11e EDCA uses eight IEEE 802.1D User Priorities (UPs). Arriving traffic of these eight UPs is mapped to four different Access Categories (ACs)—for voice, video, best-effort, and background traffic. Each access category corresponds to a single transmit queue. For each AC, an access function contends separately for the right to initiate one or more transmissions during the contention phase.

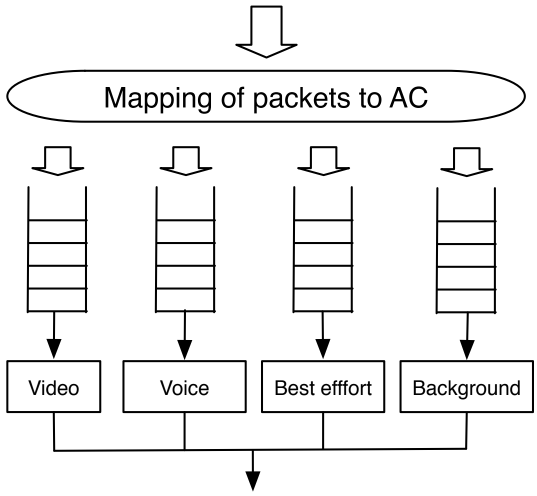



Figure 2.15: EDCA: an access function for each priority queues

The basic reference model is shown in Figure 2.15. Each access function has an own parameter set. It consists of the upper and lower bounds of the contention window (abbreviated as $CW_{min}$ respective $CW_{max}$ ), inter-frame spaces, and Transmit Opportunity (TXOP) Limit values. A TXOP is a bounded time interval during which a station can send as many frames as possible (as long as the duration of the transmissions does not extend beyond the maximum duration of the TXOP). If a frame is too large to be transmitted in a single TXOP, it should be fragmented into smaller frames. The use of TXOPs reduces the problem of low rate stations gaining an disproportionate amount of channel time in the legacy 802.11 DCF MAC. A TXOP time interval of 0 means it is limited to a single MAC layer frame.

With EDCA, high priority traffic has a higher chance of being sent than low priority traffic: a station with high priority traffic waits a little less before it sends its packet, on average, than a station with low priority traffic. Giving priority to different traffic category is based on their value of inter-frame space and contention window.

EDCA extends DIFS of legacy DCF by enlarging the duration that the channel has to be idle prior initiation of a transmission dependent of AC. These Arbitration Inter-frame

Spaces (AIFS) compute as follows:

$$AIFS[AC] = SIFS + AIFSN[AC] \times T_{slot} \tag{2.3}$$

where $T_{slot}$ is the duration of a slot and $AIFSN[i]$ is the $AIFS[i]$ represented in number. Figure 2.16 shows the impact of AIFS. Thus, lower priorities have to defer longer prior transmission initiation or backoff decrement.

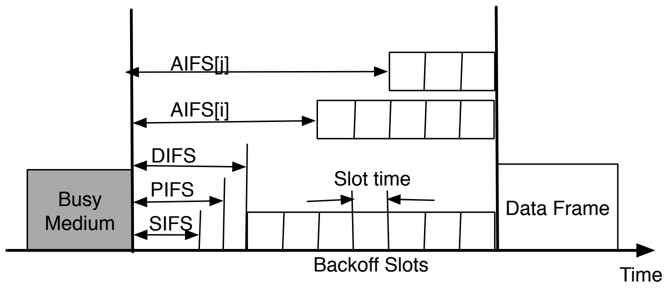



Figure 2.16: Inter-frame spaces with 802.11e EDCA

Similarly to DCF, a transmission failure leads to an increased contention window of an access function according to the Binary Exponential Backoff (BEB) algorithm, which doubles CW as long as it does not exceed $CW_{\max}$ . After a successful transmission, CW is reset to the minimum value for the corresponding AC. Internal collisions between two access functions within one station are resolved such that the higher-prioritized function gains access to the wireless medium, while the lower-prioritized function performs actions as if an external collision happened.

The purpose of QoS is to protect high priority data from low priority data, but there can be scenarios in which the data which belongs to same priority needs to be protected from data of same priority. Example being a network that can only accommodate 10 data calls and an eleventh call is made. Admission Control in EDCA address this type of problems. The AP publishes the available bandwidth in beacons. The clients can check the available bandwidth before adding more traffic in the network that cannot be handled.

### 2.5.2 802.11e HCCA

The HCCA works a lot like the PCF. However, in contrast to PCF, in which the interval between two beacon frames is divided into two periods of contention-free and contention period, the HCCA allows for contention free periods being initiated at almost anytime during a contention period. This kind of contention free period is called a Controlled Access Phase (CAP) in 802.11e. A CAP is initiated by the AP, whenever it wants to send a frame to a station or receive a frame from a station in a contention free manner. During a CAP, the access point controls the access to the medium. During the contention period, all stations function in EDCA. The other difference with the PCF is that traffic class and traffic streams are defined. This means that the AP is not limited to per-station queuing and can provide a kind of per-session service. Also, the AP can coordinate these streams or sessions in any fashion it chooses (not just round-robin). Moreover, the stations give info about the lengths of their queues for each traffic class. The AP can use this info to give priority to one station over another or better adjust its scheduling mechanism. Another difference is that stations are given a TXOP: they may send multiple packets in a row, for a given time period selected by the AP. During the contention period, the access point allows stations to send data by sending CF-Poll frames.

HCCA support is not mandatory for 802.11e APs. In fact, few (if any) APs currently available are enabled for HCCA.

### 2.5.3 Other QoS Proposals

There is a considerable amount of research on QoS differentiation for wireless LANs and ad hoc networks. We discuss here only a small subset of closely related work, especially we are interested in solutions at the MAC layer that aim at enhancing the 802.11e type of wireless networks. We focus on distributed access methods, because they can work in the infrastructure mode as well as in an ad hoc setting.

Pattara-Atikom *et al.* proposed a survey of distributed MAC schemes that support QoS in 802.11 networks [76]. They categorize access methods into two groups: priority oriented and fair scheduling based. The priority schemes [48, 31, 1] provide service differentiation by allowing privileged access to the channel for traffic classes with higher priority while mechanisms based on fair scheduling aim at allocating weighted fair share of throughput among different demands [96, 8]. In all the above mechanisms, QoS is usually provided by tuning access parameters like IFS (Inter-Frame Spaces) and contention windows. The authors also compared three approaches based on fair scheduling: Distributed Weighted Fair Queueing (DWFQ) [8], Distributed Fair Scheduling (DFS) [96], and their own proposal—the Distributed Deficit Round Robin (DDRR) [74] based on the concept of the Deficit Round Robin (DRR) [87]. In DDRR, each traffic class determines the allotted service quantum rate based on its throughput requirements and maintains a deficit counter of accumulated quanta. The deficit counter decreases by the size of a transmitted frame and a traffic class can transmit only when the counter is positive.

Qiao *et al.* proposed a Priority-based Fair Medium Access Control protocol (P-MAC) to maximize the wireless channel utilization subject to weighted fairness among multiple traffic flows [78]. The basic idea of P-MAC is to optimize the value of the contention window for each wireless station to reflect the relative weights among traffic flows as well as the number of stations contending for the wireless medium. The proposed method is sensitive to the convergence to a desirable operating point. Hu *et al.* have shown that P-MAC may not converge even in the presence of only a single traffic class [43].

Hu *et al.* proposed *MAC Contention Control* (MCC) to achieve proportional throughput allocation [43]. The proposal does not modify the original IEEE 802.11e EDCA, but rather it adds a thin layer above MAC that adjusts the rate of dequeuing frames in function of relative priorities. MCC dequeues frames from layer 3 according to an AIMD algorithm based on one of two MAC layer channel state indicators: the average number of collisions between successful transmissions and the number of idle slots between transmission attempts. The first indicator has major drawback of being dependent on the channel error rate (recall that stations cannot distinguish between a collision and a failed transmission). Moreover, the authors admit that it significantly varies with changing traffic characteristics, which makes it a poor reference. The second indicator has already been used as the basis of the Asymptotically Optimal Backoff (AOB) [15] and *Idle Sense* access methods [40]. The adaptation algorithm of MCC also uses the AIMD algorithm for adjusting the transmission attempt rate similarly to *Idle Sense*. As MCC operates above an unchanged 802.11e layer, it fails to maintain both high reactivity and good channel efficiency: performance evaluation done by the authors [43] shows that the adaptation time is of the order of seconds and the aggregated throughput decreases in function of the number of stations. In particular, the simulation results show two cases: for small AIMD parameters, MCC does obtain a better aggregate throughput than 802.11e, but responds very slowly to the network dynamics. For large AIMD parameters, MCC becomes more responsive to the network dynamics, but it does not attain high aggregate throughput.

Although many authors have addressed the problem of QoS differentiation in 802.11 wireless networks, no method proposed so far offers all the desirable characteristics at the same time: high aggregate throughput for an increasing number of contending stations, fair allocation to all stations in the same class, fast adaptation to changing conditions, and support for absolute priorities.

In Chapter 3, we propose a novel access method that supports both relative proportional throughput allocation and absolute priorities in 802.11 wireless networks. We build our access method upon the idea of the *Idle Sense* [40] access method that provides near optimal throughput and fairness for 802.11 WLANs.

## 2.6 Spatial Problems in Wireless Networks

As it was mentioned earlier, DCF suffers from several well-known problems such as performance anomaly, bad day effect, and physical layer capture effect. However, some enhancements to this MAC protocol like AOB or *Idle Sense* improve the performance in terms of throughput and fairness when all stations are within the carrier sense range of each other. Thus, an enhanced version of DCF could be a good choice for this kind of wireless LANs. However, problems may arise in an ad-hoc wireless network. In this section, we study the behavior of DCF in ad-hoc networks. Various problems may arise due to spatial positions of stations. We describe several well-known spatial problems in wireless networks: *hidden, exposed, blocked*, and *masked* station problem.

Before talking about different spatial problems, we recall the definition of transmission and carrier sensing range. The transmission range of a node A is modeled as the area inside which other nodes are able to correctly receive A's frames. On the other hand, the carrier sense range of A is the area encompassing those nodes whose transmission A can perceive (carrier sense) while not necessarily being able to receive the transmitted frames. Figure 2.17 illustrates transmission and carrier sensing range for node A.

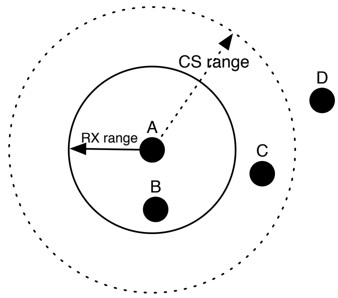



Figure 2.17: Transmission and carrier sensing range: A emits a frame: B being within the transmission range of A is able to decode the frame, C overhears it but cannot decode it while D does not even overhear transmissions from A.

### 2.6.1 Hidden Station Problem

The hidden station problem is the most familiar performance problem that may arise in a 802.11 wireless network. Assume the configuration in Figure 2.18, whereby B is within the transmission range of A and C, while C is outside of the carrier sensing range of A. In such a case, C will not be able to detect the ongoing transmission of A to B by carrier sensing and, consequently, it can inadvertently interfere with B's reception of A's frame. The problem could also happen when there is an obstacle between A and C.

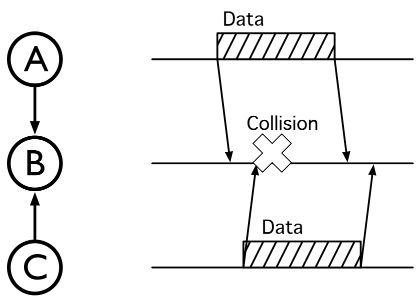



Figure 2.18: Problem of hidden stations: A and C are hidden from each other.

To alleviate the hidden node problem, Karn [55] proposed a two-way handshake involving short frames whose exchange should precede the actual transmission. The sender starts by transmitting a Request-To-Send (RTS) frame. After receiving RTS, the intended recipient sends a Clear-To-Send (CTS) frame to the sender (cf. Figure 2.19). RTS and CTS frames include the expected duration of time for which the channel will be in use. Other hosts that overhear these frames must defer their transmission for the duration specified in the frames. For this reason, each host records, in its NAV, the duration of time it must defer its transmission. This whole process is called virtual carrier sensing, which allows the area around the sender and receiver to be reserved for communication, thus avoiding the hidden terminal problem [93]. A station can enter a power-saving mode for a duration of time according to its NAV counter. The station then decrements its NAV counter and awakes to sense the medium again as soon as its NAV counter reaches zero.

The RTS/CTS option has been standardized into the IEEE 802.11 family of access schemes. The complete exchange involves four frames: RTS/CTS/DATA/ACK, with the first pair taking care of the hidden nodes, and the final ACK providing for reliable delivery.

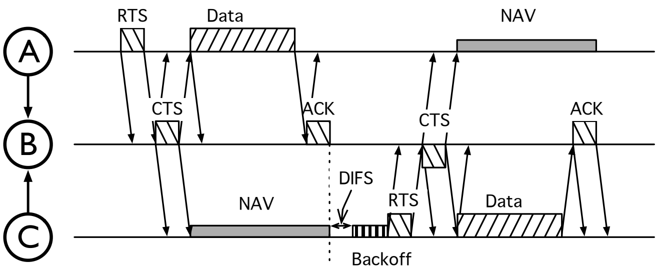



Figure 2.19: RTS/CTS handshaking alleviates the problem of hidden station.

Note that an increase in performance using RTS/CTS is the net result of extra overhead of RTS/CTS control frames and reducing number of retransmissions. Hence, in a configuration without hidden terminals the use of RTS/CTS will only increase the overhead, which reduces throughput. A insignificant hidden station problem may also result in performance degradation in case of using RTS/CTS. In this case, the additional RTS/CTS frames cost more in terms of overhead than what we gain by reducing retransmissions. Chaudet *et al.* [22] showed that in a chain topology the use of RTS/CTS results in a lower throughput compared to the basic DCF.

In case of the RTS/CTS option, the node will try to reserve the channel by sending an RTS to the target node. Since two nodes can pick the same backoff counter, the RTS frame may be lost because of a collision. Since the probability of a collision gets higher as the number of nodes increases, a sender will interpret the absence of a CTS as a sign of congestion. In this case, the node will double its contention window to lower the probability of another collision.

Finally, the RTS/CTS option may alleviate hidden node problem, but it only works if a hidden station is within the transmission range of the receiver that allows to correctly decode the RTS or CTS control frames. However, this is not the case for example for station $D$ in Figure 2.20 (the solid arrow corresponds to two stations within their transmission range while the dashed one represents communication between two stations within their carrier sensing range).

### 2.6.2 Exposed Station Problem

In wireless networks, the exposed node problem occurs when a node is prevented from sending frames to other nodes due to a neighboring transmitter. Consider the example

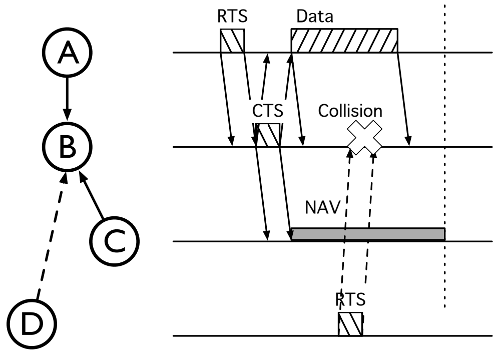



Figure 2.20: RTS/CTS handshaking cannot resolve the hidden station problem completely.

illustrated in Figure 2.21. Here, if a transmission between C and D is taking place, node B is prevented from transmitting to A as it concludes after carrier sense that it will interfere with the transmission by its neighbor C (cf. the left part in timing diagram of Figure 2.21). However note that A could still receive the transmission of B without interference because it is out of range from C. This problem was studied by Karn [55]. Node B is said to be *exposed*. Note that only if B and C start to transmit their DATA frames with the same size at the same time, as shown in the right part of Figure 2.21, neither B nor C is exposed.

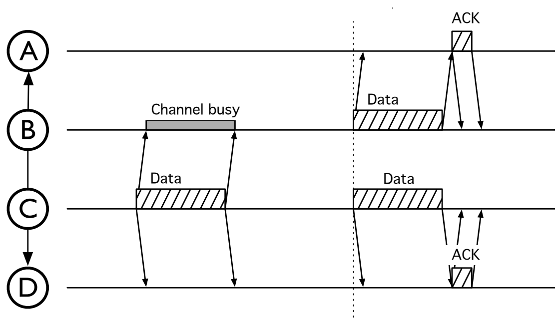



Figure 2.21: Problem of exposed stations: B is exposed by C.

MACA [55] proposes to solve the problem with the RTS/CTS option. When a node hears an RTS from a neighboring node, but not the corresponding CTS, it can deduce that it is exposed and is permitted to transmit to other neighboring nodes. However, the solution relies on several assumptions (symmetrical links, only the CTS frame set the NAV and the RTS frame does not) and only works for a specific placement of stations with respect to their transmission and carrier sensing ranges. As illustrated in the left part of Figure 2.22, when $C$ sends its RTS frame, the exposed station $B$ will not receive the CTS frame sent by $D$. In this case if according to MACA, RTS frame does not set the NAV, B can send RTS to A. The problem is that it cannot decode the corresponding

CTS, because of DATA transmission from C. So it cannot proceed with its transmission (cf. left part of Figure 2.22). The RTS/CTS handshaking helps to solve this problem only if the nodes are synchronized (cf. right part of Figure 2.22).

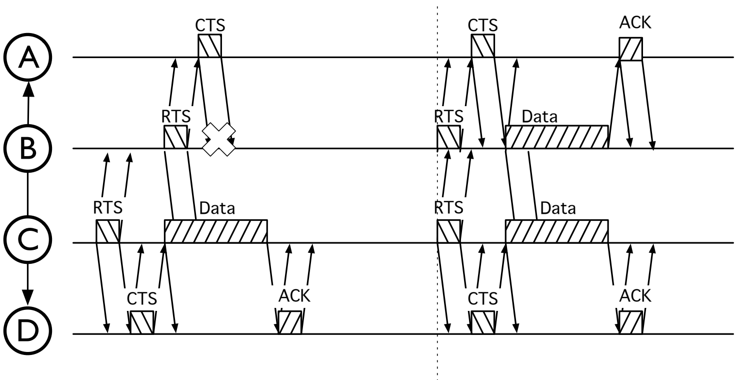



Figure 2.22: Problem of exposed stations with RTS/CTS handshaking.

### 2.6.3 Blocked Station Problem

The use of the RTS/CTS option may also lead to the problem of *blocked stations* [51, 83] in which some stations become *blocked*, because they conform to the NAV of a neighbor station transmitted in the RTS and CTS frames. Figure 2.23 illustrates an example of blocked stations: C sends RTS to D in order to initiate a new transmission while E is in the middle of transmitting its DATA to station F. D is blocked by E and cannot reply to the RTS frame sent by C. Thus $C$ becomes also blocked; in the same way $B$ is blocked, because it respects the NAV of the RTS frame sent by $C$ and similarly $A$ becomes blocked too. In this example, $A$ and $C$ see false collisions, so they double their contention windows, which further amplifies performance degradation.

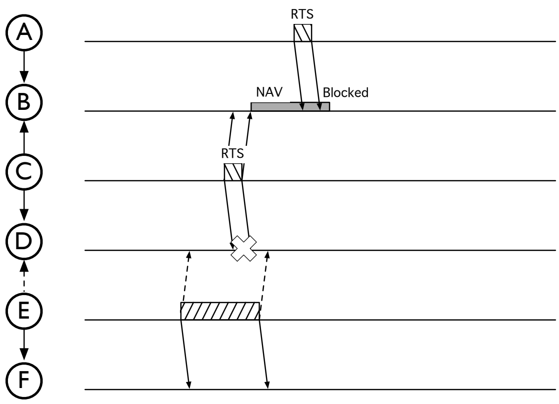



Figure 2.23: Problem of blocked station

In Chapter 4 we will focus on an exacerbated problem of exposed stations that we file into the category of *blocked stations* (cf. Figure 4.1). In this problem there are three pairs of stations that communicate in parallel. The middle pair is always starved from transmission by the exterior pairs. In the same chapter, we present the *Forced Transmissions* mechanism that guarantees a minimum throughput for the central pair in this configuration.

### 2.6.4 Masked Station Problem

The main idea behind using an RTS/CTS handshake is that nodes within the transmission range of either a sender or a receiver will hear at least one of the control frames and hence defer their transmission. However, it is incorrect to assume that all nodes within the sender's transmission range can hear the RTS and all nodes within the receiver's transmission range can hear the CTS, even under perfect operating conditions.

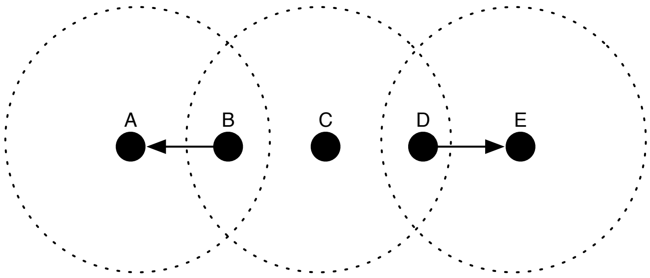



Figure 2.24: Problem of masked station: node B transmits a frame to A and D transmits a frame to E at the same time. C cannot decode either of these frames. Node C is said to be *masked*.

In Figure 2.24, node B transmits a frame to node A. Shortly thereafter, node D starts transmitting a frame to node E. Since node C receives signals from two different sources at the same time, it cannot decode either of the frames [1]. Node C is unable to hear the transmission of node D to node E, because it is *masked* by the on-going transmission from node B to node A (and vice-versa). Rey *et al.* [82] referred to this problem as the *masked station problem*. Masked nodes do not necessarily cause frame collisions. However, if one of the frames a masked node was supposed to receive is a CTS or an RTS, then the masked node may subsequently cause a collision. The scenario is illustrated in Figure 2.25.

Assume that initially all the nodes are idle and none of them is prevented from transmitting. Now, node D and node E exchange an RTS/CTS dialog successfully and node D starts sending a DATA frame to node E. Node C receives the RTS sent by node D and updates its NAV consequently. After node D starts transmitting its DATA frame,

[1]If a node receives two or more frame simultaneously, the physical layer capture effect (already studied in Section 2.4.1) may occur. However, most of the time a receiver cannot decode two simultaneous signals of the almost same power.

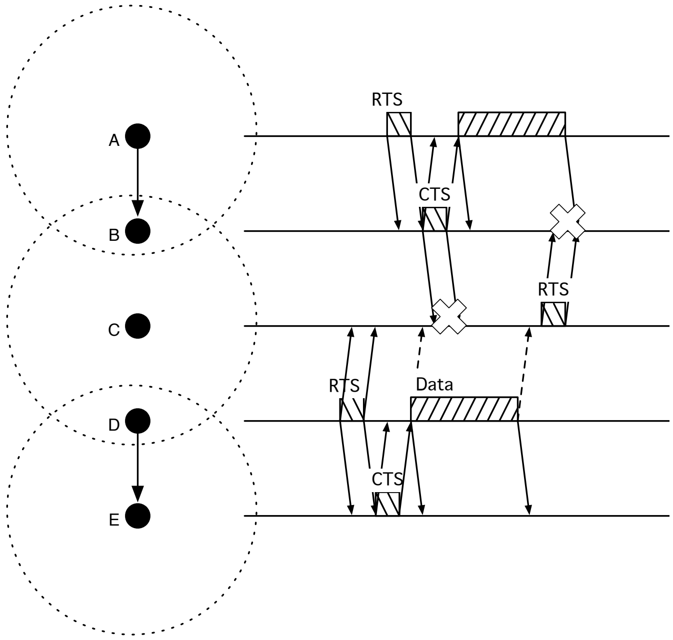



Figure 2.25: Collision caused by a masked node: node B transmits a CTS to C while D is transmitting a DATA frame to E. C cannot decode CTS from B and hence does not update its NAV. Node C may create a collision when initiating a new transmission.

node A sends an RTS to node B. Since node B is not within node D's transmission range, it does not sense any carrier and responds with a CTS. This CTS should reach node C. However, node C is masked by the signal from node D. Thus, node C cannot decode the CTS frame. Node A, on the other hand, does receive the CTS and thus starts sending its DATA frame. In the mean time, nodes D and E complete their communication and node C becomes free to transmit. Node C now transmits an RTS destined for one of its neighbors. This RTS reaches node B and destroys the data frame node B is receiving. Another possibility is that after nodes D and E complete their communication, an RTS is sent by node D, or by another node, to node C . Since node C is free to transmit as per its NAV, it responds with a CTS which collides with the DATA frame that node B is currently receiving.

### 2.6.5 DCF Behavior in Multi-hop Wireless Networks

In the previous subsections, we have shown that spatial positions of stations in a multi-hop wireless network may lead to collisions or missed transmission opportunity. Chaudet *et al.* studied the behavior of DCF in 802.11 ad-hoc networks for some common topologies (line, parallel pairs) [22] and showed that these networks suffer from severe performance problems. Figure 2.26 shows one of their simulation results that compares throughputs achieved by line topologies with different lengths when RTS/CTS option is enabled. We can see that the throughput decreases as the number of hops increases. Increasing the number of hops from two to three results in a throughput loss of 50 percent. The greatest gap occurs when increasing to four hops and results in a further decrease of 80 percent due to the carrier sense dependencies.

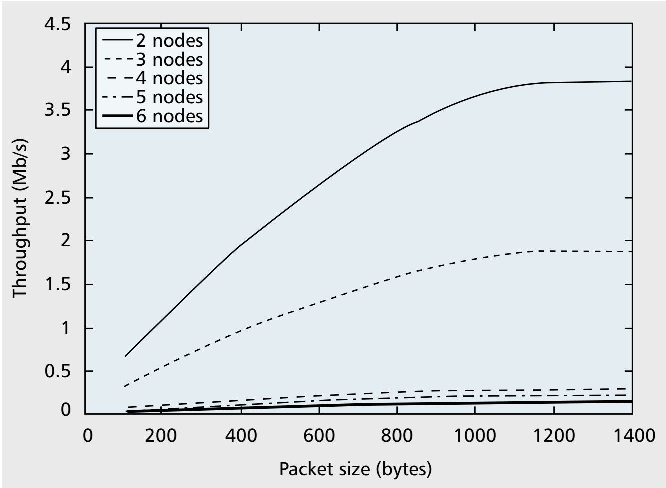



Figure 2.26: Throughput obtained for line topology: one CBR flow from first node to the last one [22].

### 2.6.6 MAC Enhancements for Multi-hop Wireless Networks

To improve performance of the MAC mechanism in multi-hop wireless networks, some authors proposed solutions before the emergence of the 802.11 standard. The *hidden terminal problem* motivated MACA (*Multiple Access Collision Avoidance*) to use RTS/CTS for collision avoidance on the shared channel [55]. The objective of MACAW (*MACA for Wireless*) was to achieve high throughput and fair channel allocation [9]. It suggested the use of link-layer ACKs with RTS/CTS and an additional DS (Data Sending) frame. Talucci *et* al. [90] have defined the MACA-BI (MACA By Invitation) access method that modifies the MACA method [55] to make it receiver oriented by keeping only the CTS part of the RTS/CTS handshake. However, the destination must be aware of current active flows to decide when to send a CTS. The Receiver-Initiated Multiple Access (RIMA) mechanism proposes a similar variant of this approach in which a polling node can also send a request to an inactive polled node [35].

Several MAC protocols have been proposed for multi-hop ad hoc networks by enhancing the CSMA/CA protocol [17, 79]. These schemes usually adjust parameters of CSMA/CA such as the contention window size and modify backoff procedures. They may improve throughput for single-cell wireless LANs. However, for multi-hop cases such as WMNs, these solutions still reach a low end-to-end throughput, because they cannot significantly reduce the probability of contention between neighboring nodes. As long as contention frequently occurs, whichever method is taken to modify backoff and contention resolution procedures, the end-to-end throughput will be significantly reduced due to the accumulating effect on the multi-hop path.

### MAC Mechanisms based on Busy Tone

In another attempt to provide a more efficient access mechanism for ad-hoc networks, several researches have proposed to use busy tones to protect control and data frames. A busy tone is a narrow-bandwidth signal with enough spectral separation on the shared channel. Tobagi *et al.* introduced busy tone multiple access (BTMA) that uses a busy tone to address the hidden terminal problem in a network with base stations [93]. When a base station senses the transmission of a terminal, it broadcasts a busy tone signal to all terminals, keeping them (except the current transmitter) from accessing the channel.

Wu *et al.* [100] proposed the Receiver-Initiated BTMA, in which a frame preamble is sent to the intended receiver by the transmitter. Once the preamble is received correctly, the receiver sets up a busy tone and waits for the data frame. The transmitter, upon sensing the busy tone, sends the data frame to the destination. The busy tone serves two functions: to acknowledge the channel access request and to prevent transmissions from other nodes. RI-BTMA was proposed to be used in the slotted manner. The correct operation of RI-BTMA largely depends on the synchronization of slots, which is usually difficult to achieve in ad-hoc networks.

Haas *et al.* [39] proposed the dual busy tone multiple access (DBTMA) protocol. In DBTMA, the RTS frames are used to initiate channel request. Two out-of-band busy tones are then used to protect the RTS frames and the data frames, respectively. The *transmit busy tone*, which is set up by the RTS transmitter, is used to protect the RTS frames. The *receive busy tone*, which is set up by the receiver, acknowledges the RTS frame and provides continuous protection for the in-coming data frames. Nodes sensing any busy tone defer from sending their RTS frames on the channel. With the use of the RTS frame and the receiver busy tone, the exposed terminals are able to initiate data frame transmissions. Furthermore, the hidden terminals can reply to RTS requests and initiate data frame reception, while data frame transmission is taking place between the transmitter and the receiver.

The disadvantage of such a solution is that extra hardware is required and busy tone transceivers need to be incorporated into each communication node.

## 2.7 Multi-Channel Access Protocols

In order to fundamentally solve the issue of low end-to-end throughput in a multi-hop environment such as wireless mesh networks, innovative solutions are necessary. For example, revisiting the design of MAC protocols based on TDMA or CDMA can be a research topic although developing a distributed and cooperative MAC with TDMA or CDMA is complex [5].

To further improve network performance and also increase network capacity especially for mesh networks, a promising solution is to enable a network node to work on multiple channels instead of only on a single fixed channel. Multiple channels present several advantages. One is to enable a single wireless interface to provide simultaneous connectivity to distinct networks. MultiNet [18] was designed to allow a wireless interface to periodically hop between two channels, enabling a single wireless interface to connect to two logically distinct networks, such as an infrastructure-based network and an ad-hoc network.

However, the most important functionality of multiple channels is capacity improvement. By exploiting multiple channels, we can achieve a higher throughput, because multiple transmissions can take place in parallel without interfering with each other. Although all variants of 802.11 LANs provide several non-interfering channels, the legacy DCF uses only one channel and all nodes in the network have to use a common channel to communicate. Thus, there have been several works on developing MAC protocols that use multiple channels in ad-hoc and wireless mesh networks. Maheshvari *et al.* [67] classified multi-channel protocols into two different categories: *static* and *dynamic*.

In a *static* multi-channel protocol, the channel of a radio interface remains fixed or changes infrequently. Each node has multiple radio interfaces and the emphasis is on assigning frequency channels to these radio interfaces such that two nodes that communicate directly in the resulting topology have at least one channel in common. The static approaches are based on topology control. The channel assignment must be done in a way that interference is minimized. As this approach is necessarily static, the approach is often graph-theoretic and is based on models of interference or protocol behavior, and assumptions on average traffic. There are several papers in literature that take this approach [80, 81, 4, 6, 58]. In all of these works, the channel assignment is essentially considered as an optimization problem.

The *dynamic* approaches rely on the capability of the radio interface to switch channels on the fly with negligible delay. Here, multiple channels can be used even with a single radio interface. Two neighbor nodes wanting to communicate have to switch to the same channel. Designing a dynamic MAC protocol that exploits multiple channels is not a trivial problem, because when a host is listening on a particular channel, it cannot hear communication taking place on a different channel. In this thesis we are especially interested in dynamic multi-channel protocols.

### 2.7.1 Multi-Channel Specific Problems

Figure 2.27 shows a scenario in which A and B exchange their frames on channel 1 while stations C and D use channel 2 to communicate. Consequently, A and C can simultaneously transmit their frames to B and D, respectively. However, under single-channel DCF these two parallel communication result in a collision. Now, assume that B has a data frame to C. If B sends the frame on its current channel, i.e. channel 1, this will cause the transmission attempt to fail, because C is currently on channel 2. This phenomenon is referred to as *deafness*. Station B will retransmit the failed data frame based on the exponential backoff mechanism in 802.11 DCF until it reaches the retry threshold. After *retry_limit* failed transmissions, C throws away the frame and resets its $CW$ to $CW_{\min}$ for the next data frame. Thus, the deafness problem leads to the wastefulness of the network resources and also causes unfairness.

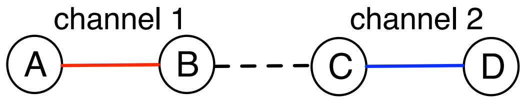



Figure 2.27: Stations use different channels in a line topology

The deafness problem arises when the sender is not aware of the current channel used by the receiver and transmits its frame on the wrong channel. It is the most important problem in a multi-channel network to be considered when designing a multi-channel protocol. Any dynamic multichannel protocol must ensure that the transmitter and the receiver are on the same channel before communicating. To achieve this, it either ensures that they switch to a predetermined channel at a predetermined time or it uses a separate control channel and interface to perform a channel negotiation. This either requires time synchronization or an additional radio interface and channel. However, in Chapter 5 we propose a new multi-channel MAC protocol that does not need synchronization nor a dedicated control channel.

Another problem that may arise in a multi-channel protocol is a new form of the hidden terminal problem. Assume that node A has its interface tuned to the channel $i$ and wants to send a data frame to node B on channel $j$. So, it at first switches the channel from $i$ to $j$. Since A has no prior information about the state of channel $j$ (it has not received the recent RTS/CTS frame on channel $j$), its NAV value is not up to date with respect to the the new channel. Therefore, in case A transmits an RTS or a data frame to B, its transmission may collide with an ongoing transmission from a third node C that is hidden from A. This problem presents a new form of the well-known hidden terminal problem that called *multi-channel hidden terminal problem* and was firstly studied by So *et al.* [88].

A simple solution to this problem would be for A to wait for the longest frame transmission time before attempting transmission after switching channel, but this is clearly inefficient.

In the following subsections, we classify different dynamic multi-channel protocols and we discuss the advantages as well as the drawbacks of each category. Our classification is partially inspired from [70].

### 2.7.2 Receiver-Oriented Mechanisms

In receiver-oriented mechanisms, each node is assumed to have a single interface. Every node also selects a well-known *quiescent* channel for itself. This is the channel the node always listens to when idle. To transmit a data frame, a transmitter switches its interface to the quiescent channel of the intended receiver and then transmits using a regular single channel MAC protocol such as 802.11 DCF. Following a successful transfer, the sender switches its interface back to its quiescent channel.

The protocol assumes that the quiescent channel selection and distribution of this information to the neighboring nodes are done via a separate mechanism. In other words, it is no longer needed that a communicating pair of nodes negotiate a channel beforehand.

One of the earliest work that uses dynamic channel switching was the Receiver-Directed Transmission (RDT) protocol [86]. However, a straightforward use of RDT with 802.11 MAC results in several serious performance issues. This protocol does not handle the multi-channel hidden terminal problem. Moreover, deafness arises when the intended receiver is not on its quiescent channel, as the receiver could be in transmission on the channel of a third node. Extended Receiver-Directed Transmission (xRDT) [67] alleviates the performance problems of RDT using busy tones and further control messages. To handle hidden station problem, it assumes that for each data channel $c$ there is a different busy tone $b_c$. A receiver when receiving a data frame on channel $c$ tunes its $b_c$ tone that in turn forces the potential neighboring transmitters on channel $c$ to defer their transmissions. Although xRDT removes the multi-channel hidden station problem, it does not remove deafness. When node $A$ is not on its own channel, any transmission attempt to $A$ is failed and the contention window of the sender is doubled. However, to alleviate deafness problem, a deaf node broadcasts a notification message when it returns to its quiescent channel. This message will inform potential transmitters that the receiver just came back to its own channel. Now, they can escape from their backoff (that has been exponentially increased because of consecutive transmission failures) and start a new contention process to transmit their data frames. Thus, this notification mechanism does not prevent deafness from occurring, but only helps the potential transmitters to escape from their long backoffs caused by the deafness problem.

### 2.7.3 Dedicated Control Radio

Each device implementing *dedicated control radio* has one radio dedicated to control messages and the second one is used to exchange data frames between stations. All control radios throughout the network are tuned to a common channel and the data radio can be tuned to any other channel. On one hand, a common channel is assigned to all control radios and it remains fixed. On the other hand, channel switching is frequently performed for data radios. Thus, it is more exact to classify this kind of multi-channel protocols as hybrid (static and dynamic).

In principle, before exchanging data frames, the sender negotiates the channel to be used for the next transmission with the corresponding destination. The negotiation mechanism is done on the control channel and then both source and destination tune their data radio to the selected channel. The channel with the least utilization is usually selected during the negotiation.

All devices can overhear all the agreements made by other devices, even during data exchange. The efficiency of this system is limited only by the contention for the control channel and the number of available data channels.

Figure 2.28 illustrates the operations of Dedicated Control Radio. In the figure, channel 0 is the control channel that is assigned to the control radio and channels 1, 2, and 3 are for data transmission. When device A wants to send to device B, it transmits a RTS frame on the control channel. That RTS specifies the lowest numbered free channel. Upon receiving the RTS, B responds with a CTS frame on the control channel, confirming the data channel suggested by A. The RTS and CTS frames also contain a NAV field, as in 802.11, to inform other devices of the time during which the sender, the receiver, and the chosen data channel are busy. Since all devices listen to the control channel all the time, they can keep track of the busy status of other devices and channels, even during data exchange. Devices avoid busy channels when selecting a data channel. Therefore the multi-channel hidden terminal problem is highly alleviated if it is not completely resolved. As the data channel is negotiated between a pair of nodes, the deafness problem is completely resolved.

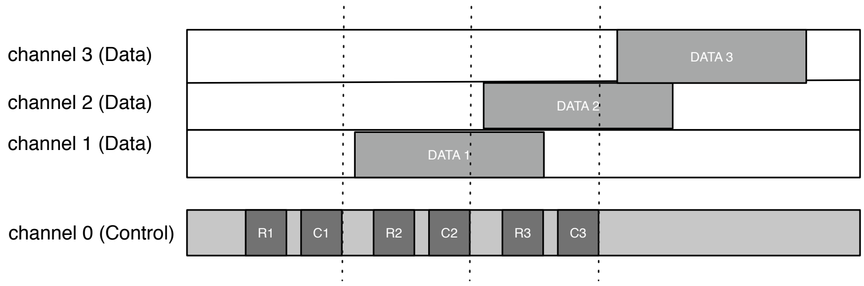



Figure 2.28: Dedicated Control Radio

Examples of this approach include Dynamic Channel Allocation (DCA) [101] and DCA with Power Control (DCA-PC) [64]. The major advantage of Dedicated Control Channel is that it does not require time synchronization: rendezvous always happen on the same channel. The main disadvantage of this protocol is that it requires a separate dedicated control radio and a dedicated channel, thereby increasing cost and decreasing spectral efficiency when few channels are available.

Dedicated Control Radio can be considered as a simple example of a multi-radio multi-channel MAC protocol. Generally, in a multi-radio wireless network, a node may have multiple radios each with its own MAC and physical layers. Communications on these radios are totally independent. Thus, a virtual MAC protocol such as the multi-radio unification protocol (MUP) [4] is required on top of MAC to coordinate communications in all channels.

### 2.7.4 Common Channel Hopping

In this approach devices not exchanging data, hop through all channels synchronously. A pair of devices stop hopping as soon as they make an agreement for transmission and rejoin the common hopping pattern subsequently after transmission ends. The *common channel hopping* protocol improves on *dedicated control radio* in two respects: *i*) it uses all the channels for data exchange and *ii*) it requires only one transceiver per device.

As shown in Figure 2.29, the hopping pattern cycles through channels 0, 1, 2, and 3. When device A wants to send to device B, it sends an RTS to B on the current common channel. If B receives the RTS properly, then it returns a CTS on the same channel. Devices A and B then pause hopping and remain on the same channel during data transfer, whereas the other idle devices continue hopping. When they have finished, they rejoin the *common channel hopping* sequence with all the other idle devices. It is possible that the *common channel hopping* sequence wraps around and visits the channel that A and B are using before they finish data exchange. Idle devices sense the carrier and refrain from transmitting if it is busy.

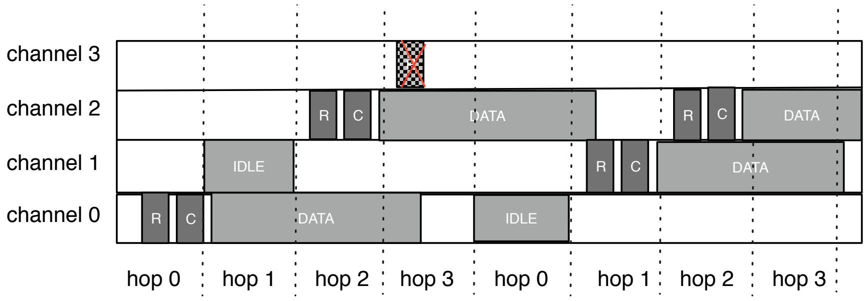



Figure 2.29: Common Channel Hopping

While A and B are exchanging data, they are unaware of the busy status of other devices. Hence, it is possible that a sender sends an RTS to a device that is currently busy on a different channel. In this case, the deafness problem may arise for the RTS frames. For the same reason, multi-channel hidden terminal problem may also occur. The approach also requires devices to have tight synchronization. Another issue with this approach is that devices hop more frequently than other mechanisms. State-of-the-art integrated circuit implementations of trimode 802.11a/b/g radios require only about 30 $\mu s$ for its voltage-controlled oscillator (VCO) to settle [104], however no hardware constraint justifies that this delay cannot be further reduced.

Examples of this design approach include Hop-Reservation Multiple Access (HRMA) [92], Channel Hopping Multiple Access (CHMA) [94], and Channel Hopping multiple Access with packet Trains (CHAT) [95].

### 2.7.5 Control-Data Phase

In this approach, time is divided into an alternating sequence of control and data exchange phases as shown in Figure 2.30. During a control phase, all devices tune to the control channel and attempt to make agreements for channels to be used during the following data exchange phase. The control channel is common and so all devices in the vicinity of each other overhear all the agreements made by other devices during the control phase.

In the second phase, devices tune to the agreed channel and start data transfer. The protocol allows multiple pairs of devices to choose the same channel because each pair might not have enough data to use up the entire data phase. As a result, the different pairs must either schedule themselves or contend during the data phase.

The advantage of this approach is that it requires only one radio per device. However, it requires time synchronization among all devices, though the synchronization can be looser than in *common channel hopping*, because devices hop less frequently. Examples of this approach are Multichannel MAC (MMAC) [88] and Multichannel Access Protocol (MAP) [24]. Their main difference is that the duration of the data phase is fixed in MMAC, whereas it is variable in MAP and depends on the agreements made during the control phase. We describe MMAC protocol in more details, because we will compare it with our contribution for the multi-channel MAC mechanism in Chapter 5. We chose MMAC, because it is one of the most well-known multi-channel MAC protocols. Furthermore, it solves the multi-channel hidden station as well as the deafness problems.

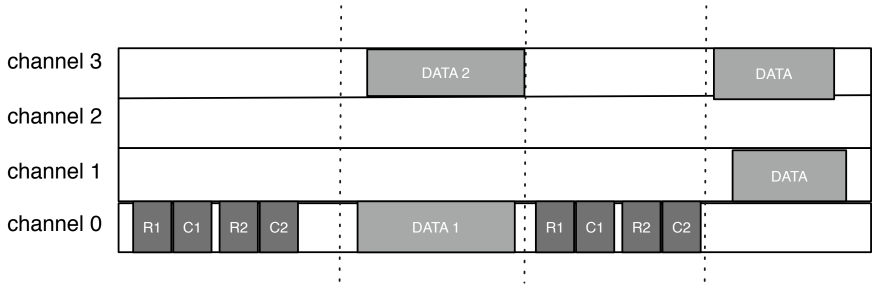



Figure 2.30: Control-Data Phase Mechanism

### MMAC

In MMAC, periodically transmitted beacons divide time into beacon intervals. A small window called the ATIM (Announcement Traffic Indication Message) window is placed at the start of each beacon interval[1]. Reservations are published through the ATIM window. The nodes that have frames to transmit negotiate channels with the destination nodes during this window. In the ATIM window, every node must listen to the default channel. The default channel is one of the multiple channels, which is predefined so that every node knows the default channel. During the ATIM window, all nodes listen on the default channel and beacons as well as ATIM frames are transmitted on this channel. Note that outside the ATIM window, the default channel is used for sending data similarly to other channels.

The process of channel negotiation and data exchange in MMAC is illustrated in Figure 2.31. During the ATIM window, A sends ATIM to B and B replies with ATIM-ACK indicating to use channel 1. This ATIM-ACK is overheard by C. When D sends ATIM to C, C selects channel 2 because channel 1 is already reserved by A and B. After

[1]The term ATIM is used as in IEEE 802.11 PSM (Power-Saving Mode) [44], although it is used for a different purpose in MMAC.

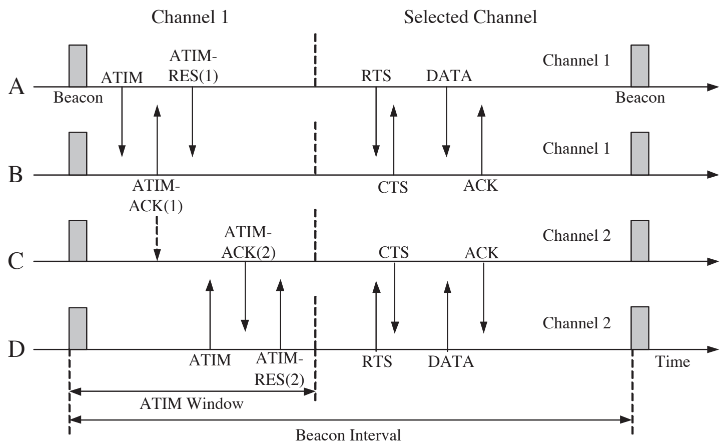



Figure 2.31: Process of channel negotiation and data exchange in MMAC [88].

the ATIM window, two communications (between A and B, and C and D) can take place simultaneously.

Each node maintains a data structure called the Preferable Channel List (PCL) that indicates which channel is preferable to use for the node. PCL records the usage of channels inside the transmission range of the node.

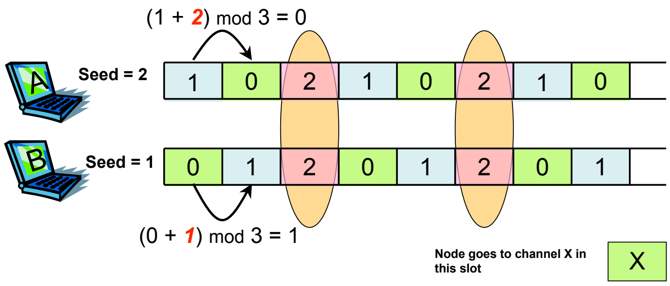
If node S has buffered frames destined for D, it will notify D by sending an ATIM frame. S includes its PCL in the ATIM frame. D, upon receiving the ATIM frame, selects one channel based on the sender's PCL and its own PCL. The receiver's PCL has higher priority in selecting the channel. After D selects a channel, it includes the channel information in the ATIM-ACK frame and sends it to S. When S receives the ATIM-ACK frame, it sees if it can also select the channel specified in the ATIM-ACK. S can select the specified channel only except when S has already selected another channel. If S selects the channel specified in the ATIM-ACK, S sends an ATIM-RES (ATIM-Reservation) frame to D, with S's selected channel specified in the frame. ATIM-RES frame notifies the nodes in the vicinity of S which channel S is going to use, so that the neighboring nodes can use this information to update their PCL. Similarly, ATIM-ACK frame notifies the nodes in the vicinity of D. After the ATIM window, S and D will switch to the selected channel and start communicating by exchanging RTS/CTS.

If S cannot select the same channel as D, because it has already selected another channel, it cannot send frames to D during the beacon interval. It has to wait for the next beacon interval to negotiate channels again. Another issue is that the method requires node synchronization and it is difficult to find the optimal duration for the rendezvous slot.

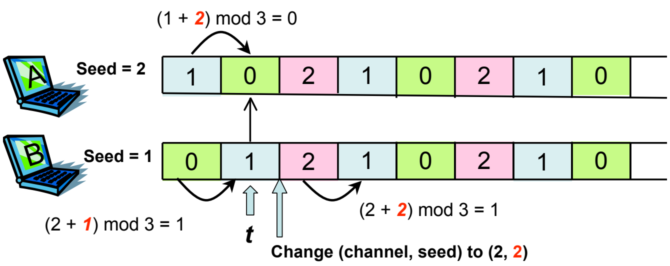
Chen *et al.* [23] further extended this method to allow for broadcast and multicast transmissions and presented a channel allocation algorithm to maximize channel utilization. Then, the same authors proposed a rendezvous negotiation by dynamically increasing the ATIM window [25]. Finally, Benveniste *et al.* [12] adopted a similar method, but with burst reservation to reduce overhead.

### 2.7.6 Home Hopping - Parallel Rendezvous

These protocols differ from the *dedicated control radio* in that multiple device pairs can simultaneously make agreements on distinct channels. The main motivation is to overcome the single control channel bottleneck. However, since there are multiple rendezvous channels, special coordination is required so that two devices can rendezvous on the same channel. One solution is for each idle device to follow a *home* hopping sequence and for the sending device to transmit on that channel to find the intended receiver. The term *home* means that each device follows its own channel hopping sequence. An examples of this approach is Slotted Seeded Channel Hopping (SSCH) [7].




(a) A and B can only communicate when they overlap




(b) B wants to start a flow to A at time $t$, so it adopt the A's hopping sequence for the next slot.

Figure 2.32: Two nodes communicate according to SSCH mechanism

In SSCH, each hopping sequence is uniquely determined by the seed of a pseudo-random generator. Each device picks four sequences and follows them in a time-multiplexed manner. When device A wants to talk to B, A waits until it is on the same channel as B. If A frequently wants to talk to B, then A adopts one or more of B's sequences, thereby increasing the time they spend on the same channel. For this mechanism to work, the sender learns the receiver's current sequences via a seed broadcast mechanism.

Figure 2.32 illustrates the main idea of channel scheduling in SSCH. Time is divided into super-frames composed of slots. Each node switches to a new channel at the beginning of a slot according to its hopping sequence. Channel is scheduled by the following rule:

$$ch_{\text{new}} \leftarrow (ch_{\text{old}} + seed) \bmod n, \tag{2.4}$$

where $ch$ represents the channel and $n$ is the number of channels (3 for this example). Each node iterates through all the channels switching to the channel designated by this rule in each new slot. If B decides to transmit to A, it can wait until two nodes overlap on the same channel (cf. Figure 2.32(a)). However, if B has a traffic flow for A, it has to follow its hopping sequence by adjusting its $(channel, seed)$ to that of A (cf. Figure 2.32(b)). In SSCH, nodes adapt their hopping sequences over time to the traffic, but are not allowed to deviate from their hopping sequences.

To prevent deafness from occurring, SSCH needs that nodes are aware of each other's channel hopping schedules at the beginning of each time slots. Moreover, since a node does not maintain prior information about other channels, multi-hidden terminal problem may occur. Another issue is that SSCH is based on synchronization.

Schedules can be synchronized to improve throughput, but the performance of SSCH is fairly low in the line topology, because neighbor nodes can only occasionally tune to the common channel for packet forwarding.

## 2.8 Conclusion

Our thesis is placed at the MAC layer of 802.11 wireless networks. Thus, we briefly have reviewed the IEEE 802.11 standard and its various amendments. We have discussed the 802.11 DCF access method and its limitations in infrastructure-based as well as in ad hoc wireless networks. These limitations lead to performance degradation in a wireless network. Therefore, many research work has proposed solutions to improve the performance of DCF. They mostly consider a single cell wireless network and try to enhance MAC performance in terms of throughput and fairness.

We have also discussed the provision of quality of service in a wireless LAN. In this regard, we have briefly reviewed the MAC mechanism proposed by the IEEE 802.11e standard as well as some other related work.

We also discussed spatial problems such as *hidden*, *exposed*, *blocked*, and *masked* that may arise in a multi-hop wireless network. These problems result in a drastic degradation of performance. Moreover, in a wireless mesh network with a relatively high volume of traffic the probability of interference between active flows also increases, leading to still more degradation in such networks. In this regard, we explored the possibility of using multiple non interfering channels to transmit interfering parallel flows on different channels. We also classified already proposed multi-channel protocols in wireless networks.

# Part II

# Contributions

Chapter 3

# Providing Priority Access in 802.11 Wireless LANs

## 3.1 Introduction

For many multimedia communication applications, fair sharing of the radio channel is not enough. To achieve a good level of perceived quality, multimedia traffic requires some QoS (Quality of Service) support that guarantees parameters such as a minimal throughput, a maximal delay, and a bounded jitter. Supporting QoS guarantees in wireless networks is not easy, because air medium presents a complex and time-varying behavior: path loss, signal fading and interference, time dispersion, which result in high throughput variability and significant error rate. When channel conditions are good and communicating devices are within a sufficient radio range, QoS characteristics mainly depend on the channel access method. To deal with multimedia traffic, many authors have proposed various priority access schemes to guarantee some QoS parameters [48, 78, 96, 76, 43].

As mentioned before, the IEEE 802.11e defines an extension to the standard 802.11 operation to provide channel access differentiation for several traffic classes through new coordination functions [48]. HCF enhances PCF by allowing a centralized coordinator to allocate TXOP periods at any time in a polling based way. Such an access method can be implemented in an access point to operate in the infrastructure mode, however it would be difficult to use it in the ad hoc mode, because of the need for centralized coordination.

EDCA keeps the contention based operation of DCF, but extends it to give more transmission opportunity to higher priority classes. The principle of EDCA is to set different access parameters (interframe period, contention window, persistency factor, TXOP duration) for different traffic classes. Even if EDCA provides a first valuable support for QoS differentiation in 802.11 networks, it suffers from several performance problems largely reported in the literature [76, 75, 43].

Usually, we can achieve QoS differentiation in several ways. The first one is *relative*—we can allocate different parts of the available throughput to several traffic classes. In *proportional throughput allocation*, one class benefits from a greater throughput than another one, the proportion of the throughputs being defined by a given ratio. Such relative priorities are suitable for assigning relative weights to different types of traffic, e.g. interactive sessions such as instant messaging or remote login with respect to Web access or bulk data transfers. The advantage of relative priorities is that they do not lead to the starvation of low traffic classes when high priority traffic becomes important. However, applications that use relative priorities need to live with a possible degradation of their QoS parameters when the traffic in a given class increases.

The second way of defining QoS differentiation is *absolute*—a high priority class always benefits from the available throughput even if there is some low priority traffic. Voice-over-IP, videoconferencing, or live video broadcast are examples of applications that may require absolute priorities. In this approach, high priority traffic captures all the available throughput so that the low priority class may starve. To guarantee some QoS parameters, e.g. a minimum throughput, absolute priorities require some admission control to limit the high priority traffic so that it does not exceed the available capacity.

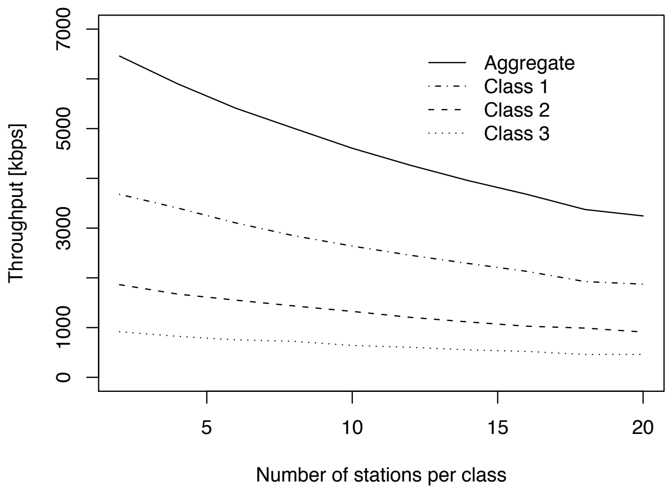



Figure 3.1: Aggregate throughput per class for 802.11e EDCA, 802.11b PHY/MAC parameters.

EDCA suffers from several performance problems [76, 75, 43]. In particular, it does not perform well when the number of competing stations increases, because the collision rate also increases so that the total network throughput drops. Figure 3.1 presents an example of such a bad performance—we can observe a significant decrease in the aggregate throughput per class in a simulation of proportional differentiation with three classes (minimum and maximum contention window are $CW_1 \in [16, 48]$ for class 1, $CW_2 \in [31, 93]$ for class 2, and $CW_3 \in [61, 183]$ for class 3).

One may think that 802.11e EDCA can provide an absolute differentiation in addition to the relative one. However, it is not the case: assigning a short AIFS to the absolute priority class and a long AIFS to the lower priority one does not result in absolute differentiation, because AIFS is followed by the contention backoff that still gives the lower priority class some transmission opportunity. Using short AIFS may only work, if the absolute priority class benefits from a very small $CW_{\min}$, but in this case, collision rate increases significantly for a larger number of absolute priority stations, which leads to a sharp drop in the aggregated throughput.

In this chapter, we propose a novel access method that supports both relative proportional throughput allocation and absolute priorities in 802.11 wireless networks. It has all desirable properties: high aggregate throughput even for a large number of contending stations, fair allocation to all stations in the same class, fast adaptation to changing conditions, and support for absolute priorities in addition to relative proportional allocation. We build upon the idea of the *Idle Sense* method that provides the optimal throughput and fairness for 802.11 WLANs [40]. We recall that in *Idle Sense*, each station adjusts its contention window based on the observed average number of idle slots. The value of contention windows increases with the number of active stations, which results in less collisions. The original version of *Idle Sense* provides fair sharing of the radio channel with better fairness compared to 802.11 DCF. The method proposed in this chapter achieves absolute priority differentiation by setting the target value for the number of idle slots to a small value, so that the absolute priority class gains all the available throughput. The proposed method also supports relative proportional throughput allocation in which several classes share the available throughput according to desired ratios. We define how stations need to adjust their contention windows to achieve relative differentiation. We keep the definition of traffic classes compatible with the IEEE 802.11e standard.

We validate our method with ns-2 simulations that show how the proposed method achieves its objectives of relative and absolute differentiation both with respect to the aggregated throughput and the speed of convergence. Unlike 802.11e EDCA, it presents very good scalability—the throughput remains almost constant in function of the number of contending stations.

The remainder of this chapter is organized as follows. We present our method that supports *proportional throughput allocation* and *absolute priorities* in Sections 3.2 and 3.3. Section 3.4 shows simulation results and comparisons with 802.11e EDCA. Finally, Section 3.5 concludes the chapter.

## 3.2 Proportional Throughput Allocation

In our proposed method, each station manages one class with an absolute priority (denoted by index 0) and $M$ lower priority classes (with an index from 1 to $M$) that benefit from relative proportional throughput allocation. Proportionally means that throughput shares follow some predefined ratios. In this section, we focus on proportional allocation, so we assume at the beginning that there is no traffic with absolute priority.

Our first goal is thus to proportionally allocate throughput to different classes. The mechanism needs to achieve this goal even in the presence of a large number of contending stations. The good properties of *Idle Sense* [40] suggest that by building on its principles and defining a differentiation method, we can benefit from its optimal behavior for a large range of contending stations. The idea for the differentiation method comes from the relation between the channel access probability and the contention window: if class $i$ wants to obtain for example a double throughput of class $j$, it needs to benefit from channel access probability, which is double compared to the probability of class $j$. This condition translates into a relation between contention windows: $CW$ of class $i$ should be a half of $CW$ for class $j$. Unlike 802.11e, all traffic classes in our method use the same inter-frame space (DIFS) before choosing a backoff from the contention window.

To derive the contention windows leading to proportional allocation, we consider that each station uses the *Idle Sense* access method to adjust $CW_{\text{ref}}$, the reference contention window. $CW_{\text{ref}}$ corresponds to the fair throughput share each station obtains when there is no absolute priority traffic. Every *maxtrans* transmissions, the *Idle Sense* access method adjusts $CW_{\text{ref}}$ to the predefined target value $n_i^{\text{target}}$ as in the original *Idle Sense* method. The value of $CW_{\text{ref}}$ varies in function of the number of active stations so that they obtain their fair shares of throughput. This part of throughput per station is then distributed over $M$ traffic classes according to desired ratios: class $i$, resp. class $j$, uses contention window $CW_i$, resp. $CW_j$ to obtain throughput $X_i$, resp. $X_j$, so that

$$\frac{X_i}{r_i} = \frac{X_j}{r_j}, \tag{3.1}$$

where $r_i$ and $r_j$ are the throughput ratios of traffic classes $i$ and $j$, $0 \leq r_i, r_j < 1$ for $i \in \{1, ...M\}$ and $j \in \{2, ...M\}$. We assume $r_1 = 1$ for the highest relative priority class.

We show below how to compute $CW_j$ to obtain the required allocation defined by Eq. 3.1. If all contending entities send frames of the same size, Eq. 3.1 translates into the following relation between the probabilities of a successful transmission $P_{\text{e}}$ of class $i$ and $j$:

$$\frac{P_{\text{t}}^i}{r_i} = \frac{P_{\text{t}}^j}{r_j}. \tag{3.2}$$

A transmission of class $i$ succeeds, if a station attempts to transmit with probability $P_{\text{e}}^i$ and no collision occurs, which is the case if no other station attempts to transmit in the same slot. Thus, we have:

$$P_{\text{t}}^i = P_{\text{e}}^i \, \frac{1}{1 - P_{\text{e}}^i} \, \prod_{k=1}^{M} (1 - P_{\text{e}}^k)^{N_k},$$

where $N_k$ is the number of stations in class $k$ and $\prod_{k=1}^{M}(1 - P_{\text{e}}^k)^{N_k}$ is the probability that a slot is idle. Thus, from Eq. 3.2, we obtain:

$$\frac{P_{\text{e}}^i}{r_{\text{i}}} \, \frac{1}{1 - P_{\text{e}}^i} = \frac{P_{\text{e}}^j}{r_{\text{j}}} \, \frac{1}{1 - P_{\text{e}}^j}$$

Typically, the slot time is as small as possible and transmissions are rare to avoid collisions, so that $\forall i, P_{\mathrm{e}}^i \ll 1$ and thus:

$$\frac{P_{\mathrm{e}}^i}{r_i} \approx \frac{P_{\mathrm{e}}^j}{r_j}. \tag{3.3}$$

When stations do not perform exponential backoff mechanism after collisions, the transmission attempt probability is as follows according to the Bianchi model [13]:

$$P_e = \frac{2}{CW+1}. \tag{3.4}$$

The reference contention window $CW_{\mathrm{ref}}$ corresponds thus to the aggregate transmission attempt probability of multiple traffic classes in one station $P_e^{\mathrm{ref}}$ that we want to distribute over $M$ classes:

$$\sum_{i=1}^{M} P_e^i = P_e^{\mathrm{ref}} \tag{3.5}$$

This yields the following expression for $P_{\mathrm{e}}^1$:

$$P_{\mathrm{e}}^1 = \frac{P_{\mathrm{e}}^{\mathrm{ref}}}{\sum_{i=1}^{M} r_i} \tag{3.6}$$

and we obtain:

$$P_{\mathrm{e}}^j = \frac{r_j P_{\mathrm{e}}^{\mathrm{ref}}}{\sum_{i=1}^{M} r_i}. \tag{3.7}$$

Knowing $P_e^j$ we can derive $CW_j$ to achieve the allocation defined by Eq. 3.1:

$$CW_j = \frac{\sum_{i=1}^{M} r_i}{r_j}(CW_{\mathrm{ref}}+1) - 1. \tag{3.8}$$

The proposed access method updates the values of $CW_j$ when it computes the new value of $CW_{\mathrm{ref}}$. Algorithm 1 in the next section specifies more formally the proposed method after the discussion of the absolute priority part.

Note that in addition to providing relative differentiation between several traffic classes, the proposed method can provide good fairness to the traffic in the same priority class: stations just behave as in the original *Idle Sense* method with contention values $CW_j$ converging to the same target value thus inheriting the good fairness property of *Idle Sense*.

## 3.3 Absolute Priority Access

Besides proportional throughput allocation, we want to provide *absolute priority* access that enables a high priority class to always benefit from the available throughput even if there is some low priority traffic. In other words, we want that as long as there are packets to send in the class with the absolute priority, low priority traffic is suspended. Moreover, traffic of absolute priority coming from different stations benefits from fair allocation of the channel capacity. If there is no traffic with the absolute priority, $M$ lower priority classes share the available throughput proportionally.

Our idea is to exploit another possibility offered by the *Idle Sense* access method: assign a small value of $n_i^{\text{target}}$ to the absolute priority class. Thus, to offer both proportional throughput allocation and absolute priorities, we define two target values:

- $n_i^{target} = n_i^a$ for the absolute priority traffic class,
- $n_i^{target} = n_i^p, n_i^p < n_i^a$ for all classes that benefit from proportional throughput allocation.

The superscripts $a$ and $p$ refer here to the absolute and proportional priority access, respectively. If $n_i^a$ is smaller than $n_i^p$, the contention window of the absolute priority traffic converges to a much smaller value than the contention window of low priority classes. At the same time, the adaptation mechanism of *Idle Sense* adjusts the small contention window of the absolute priority traffic so that all stations with this priority class benefit from a fair share of the available throughput.

There is still one issue left: how to choose a suitable target value of $n_i^a$ and $n_i^p$? Proportional throughput allocation needs to obtain the whole available capacity in the absence of the absolute traffic. So the target of proportional priorities $n_i^p$ should be the target of the basic version of *Idle Sense*, that is 5.68 for IEEE 802.11b and 3.91 for IEEE 802.11g. For the absolute priority class, we need to choose a smaller target value, but not too small, because of an increasing collision rate.

To choose the right value of $n_i^a$, we analyze the conditional probability $P[\text{coll}|\dagger\text{transmission}]$ that a transmission attempt results in a collision, which corresponds to the collision probability. This analysis is fairly general and holds for any variant of IEEE 802.11. Recall that for a large number of stations $N$, the network attains the optimal throughput when the transmission attempt probability in a given slot $P_{\text{e}}$ is such that $NP_{\text{e}} = \zeta$, where $\zeta$ is a constant [40]. Furthermore, we have $\zeta = \log(\frac{1}{n_{\text{i}}} + 1)$; $P_{\text{t}} = NP_{\text{e}}(1 - P_{\text{e}})^{N-1} \approx \zeta e^{-\zeta}$; $P_{\text{i}} = (1 - P_{\text{e}})^N \approx e^{-\zeta}$; $P_{\text{c}} = 1 - P_{\text{t}} - P_{\text{i}} \approx 1 - (\zeta - 1)e^{-\zeta}$ [40]. Finally, the collision probability is the following:

$$P[\text{coll}|\dagger\text{transmission}] = \frac{P_{\text{c}}}{P_{\text{c}} + P_{\text{t}}} \approx 1 - n_{\text{i}} \log(\frac{1}{n_{\text{i}}} + 1) \tag{3.9}$$

Figure 3.2 presents the collision probability in function of the target number of idle slots. We can observe that it steeply increases for low values of $n_i$. We have observed in our simulations that the choice of $n_i^a = 3$ for 802.11b leads to a high network throughput,

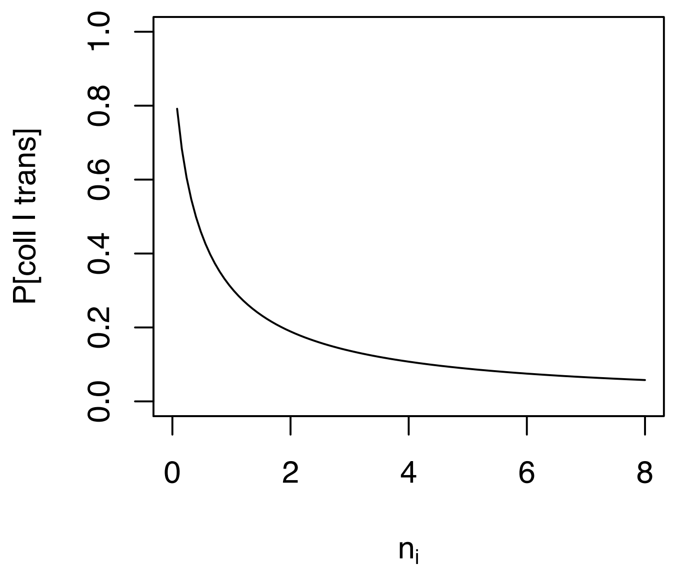



Figure 3.2: Conditional collision probability as a function of the target number of idle slots.

while the collision probability is still low as it can be seen in Figure 3.2. For 802.11g, our simulations show that the best choice is $n_i^a = 2.3$.

Algorithm 1 formally specifies our access method that provides both proportional throughput allocation and absolute priority access. The adaptation mechanism adjusts the contention window of absolute priority class $CW_a$ to the target value $n_i^a$ and the reference contention window $CW_{\text{ref}}$ to $n_i^p$. The effect of this control is a fast increase of the contention window for lower priority classes when the absolute traffic appears. Conversely, the contention window needs to decrease fast when the absolute traffic disappears. To obtain the fast decrease, we need to limit the upper bound $CW_{\max}$ of lower priority classes. This limitation is not presented in the pseudo-code of the algorithm for the sake of conciseness. Notice also that the adaptation of contention window (lines A and B) follows the AIMD mechanism described in section 2.4.4.

Similarly to proportional allocation, our method for absolute differentiation also can provide good fairness when several stations generate traffic with absolute priority. This results from the properties of the *Idle Sense* method—the station adjust their contention windows $CW_0$ to the same target value so that their long-term average converge to the same value, which results in almost the same throughput.

In this thesis we focus on providing a single absolute priority class, although the method can be extended to accommodate several absolute priority levels using smaller and smaller $n_i^a$ target values. Nevertheless, this may lead to an increasing collision rate and needs further investigation.

**Algorithm 1** *Priority Idle Sense*

$maxtrans \leftarrow 5$ ; $sum \leftarrow 0$ ; $ntrans \leftarrow 0$
$sumratio \leftarrow r_1 + ... + r_M$
After each transmission {
/* Station observes $n_i$ idle slots before a transmission */
$sum \leftarrow sum + n_i$
$ntrans \leftarrow ntrans + 1$
**if** $(ntrans >= maxtrans)$ **then**
  /* Compute the estimator */
  $\hat{n}_i \leftarrow sum/ntrans$
  /* Reset variables */
  $sum \leftarrow 0$
  $ntrans \leftarrow 0$
  A. Adjust $CW_0$ to $n_i^a$ with respect to $\hat{n}_i$
  B. Adjust $CW_{\text{ref}}$ to $n_i^p$ with respect to $\hat{n}_i$
  /* Update contention window of low priority classes */
  **for** $j = 1$ to $M$ **do**
    $CW_j \leftarrow \frac{sumratio}{r_j}(CW_{\text{ref}} + 1) - 1$
  **end for**
**end if**
}

## 3.4 Performance Evaluation

In this section, we evaluate our access method by means of simulation. We have implemented it in the NS2 network simulator version 2.29 [73] and compare to the 802.11e EDCA implementation by Wiethoelter *et al.* [99]. We report on comparisons with 802.11e, because we have designed our method to be its enhancement. We do not compare with MCC, because it places its functionality above the standard 802.11e MAC layer and its reported performance is not on a par in terms of convergence speed [43].

Our numerical examples use the MAC and PHY parameters of IEEE 802.11b. We assume ideal channel conditions, i.e. frame losses are only due to collisions. We analyze saturated conditions in which an active station has always a data frame of 1500 bytes to send. We set the transmission and carrier sensing ranges to 160 m and 400 m, respectively. We place stations randomly in a rectangle of 100 m by 100 m so they are all in the transmission range. To evaluate the performance only at the MAC layer, we use a static routing agent without any other traffic sources such as ARP. We present simulation results without the RTS/CTS exchange. Each simulation runs for 30 seconds during which approximately 15000 frames are transmitted.

We use several performance metrics to evaluate our method: *aggregate throughput*, *per class throughput* and *collision rate*.

### 3.4.1 Proportional Throughput Allocation

In this section, we evaluate our method with respect to proportional throughput allocation in several scenarios. To only focus on the properties of the relative priority scheme, we assume that there is no traffic with absolute priority.

#### One Active Class in Half of the Stations

In this scenario, we consider two traffic classes and set the desired throughput ratios to $r_1 = 1$, $r_2 = 0.5$. Only one class is active in a given station: a half of the stations generate traffic of class 1 while the other half sends class 2 traffic. To obtain this allocation, we set the following parameters of 802.11e: $CW_1 \in [16, 48]$ for class 1, $CW_2 \in [31, 93]$ for class 2.

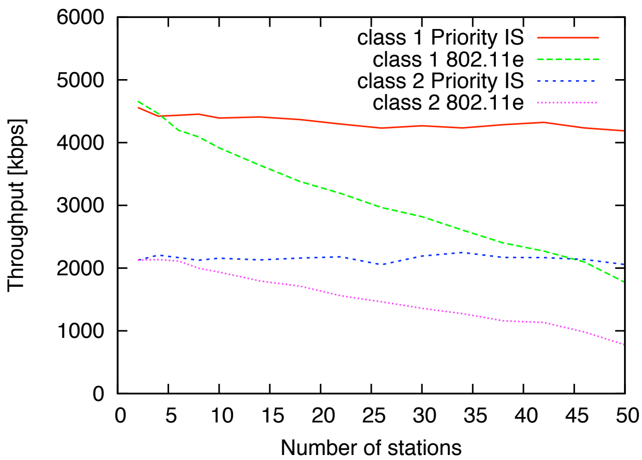



Figure 3.3: Aggregate throughput per class *vs.* number of stations, proportional throughput allocation. Half of stations generate class 1 traffic while the other half send class 2 traffic.

We can expect that the aggregate throughput of stations with class 2 traffic being a half of the class 1 throughput, because of the chosen throughput ratios and the equal number of sources in each class. We can see in Figure 3.3 that this is effectively the case. Moreover, we can observe that the aggregate throughput does not depend on the number of contending stations, which is the sign of good efficiency. 802.11e presents an opposite behavior—the aggregate throughput decreases.

#### Three Active Classes per Station

In this scenario, all stations generate traffic of three classes: 1, 2, and 3. We set the desired throughput ratios for different traffic classes in a station to $r_1 = 1$, $r_2 = 0.5$, and $r_3 = 0.25$, so that the aggregate throughput of class 2 should be a half of what obtains

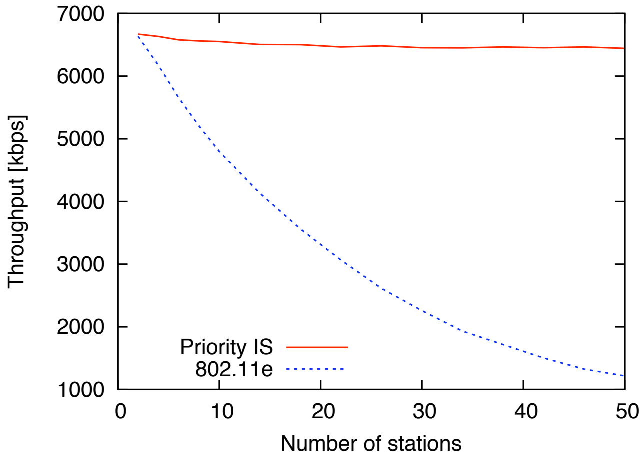



Figure 3.4: Aggregate throughput *vs.* number of stations. Proportional throughput allocation to stations with three active classes.

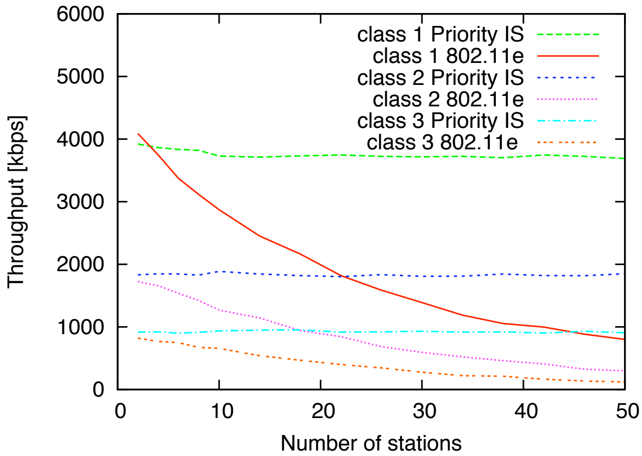



Figure 3.5: Aggregate throughput per class *vs.* number of stations. Proportional throughput allocation to stations with three active classes.

class 1, while class 3 traffic needs to obtain one quarter of the class 1 throughput. Figures 3.4 and 3.5 show simulation comparisons with 802.11e in this scenario. To achieve proportional throughput allocation in 802.11e, we set the appropriate values of the minimum and maximum contention window for different access categories: $CW_1 \in [16, 48]$ for class 1, $CW_2 \in [31, 93]$ for class 2, and $CW_3 \in [61, 183]$ for class 3 to obtain throughput allocation in the proportions defined for our method.

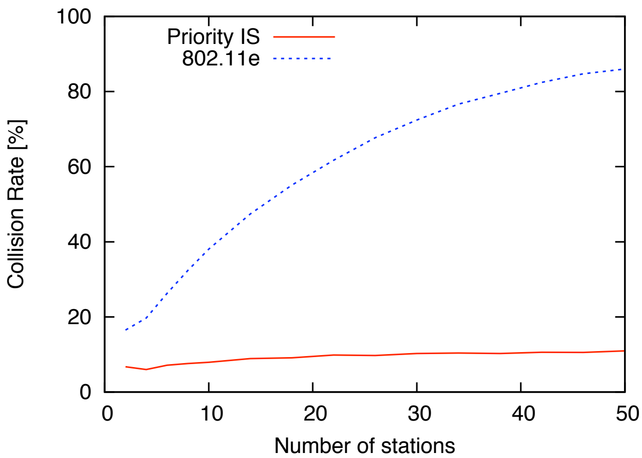



Figure 3.6: Collision rate *vs.* number of stations. Proportional throughput allocation to stations with three active classes.

Figure 3.4 shows that our method provides consistently high aggregated throughput independently of their number. For 802.11e EDCA, the aggregated throughput quickly drops with the number of competing stations. We can also see in Figure 3.5 that for our method, the ratio of class 2 throughput to class 1 is 1/2 regardless of the number of active stations, which is our objective. The EDCA access method of 802.11e keeps the ratios right, but fails to achieve high aggregated throughput for an increasing number of stations. We can also see that the collision rate for 802.11e significantly increases with the number of stations (cf. Figure 3.6), which explains the degradation in aggregated throughput.

We recall that our mechanism is based on *Idle Sense* access method, so it provides good fairness for traffic of the same priority. In other words flows in the same priority level in different stations have the same chance to gain access to the medium.

Figure 3.7 shows how the average contention window changes when the number of active stations increases. For the proposed mechanism, we just show the reference contention window $CW_{ref}$ in the figure, because the contention window for each class is a linear function of $CW_{ref}$. We also present the average contention windows for all three 802.11e classes. We can observe that $CW_{ref}$ adapts fairly well to the number of competing stations. The average contention windows in 802.11e do not increase sufficiently, which results in an increased collision rate shown in Figure 3.6.

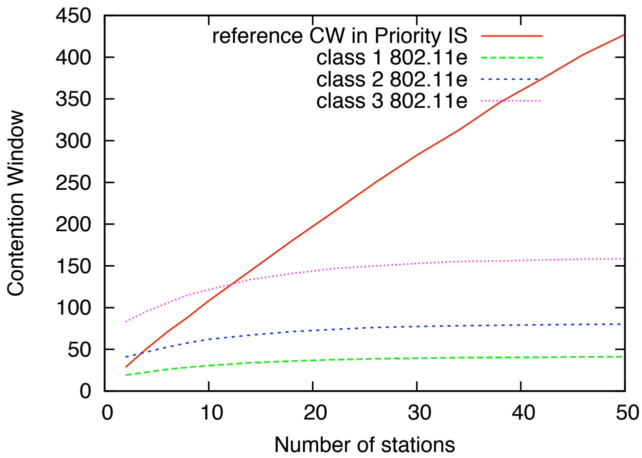



Figure 3.7: Contention window *vs.* number of stations. Proportional throughput allocation to stations with three active classes.

Another well known aspect of EDCA is its rather poor short-term fairness. Figure 3.8 presents the Jain fairness index [52]. We use the sliding window method that observes the patterns of transmissions to compute the Jain index in a window of an increasing size. The smaller the window, the index represents the shorter term fairness with the perfect fairness being 1 (we normalized the window size with respect to the number of stations so that the window sizes are multiples of $N$).

### 3.4.2 Absolute Priority Access

To evaluate our support for absolute priority access, we use a simulation set up of 10 active stations each with two classes: one with an absolute high priority and the other with a lower priority. We are not only interested in observing the desired property of absolute traffic differentiation, but we also want to assess the speed of convergence of our proposed method. Thus, instead of only observing the stationary throughput obtained by the respective priority classes, we consider a dynamically varying workload: 10 stations generate low priority traffic all of the time; one station with absolute priority class joins the network every 10 seconds beginning at instant 30 s and starts sending absolute priority traffic. Figure 3.9 shows the evolution of the aggregate throughput in time. We can observe that even if there is only one active station with absolute priority traffic, it obtains almost 90% of the channel throughput (5.1 Mbps) while all ten low priority classes share the remaining 10% of the throughput. When the second absolute class starts generating traffic, the low priority class becomes almost starved with very low throughput. At instant 130, the absolute priority traffic stops, so that the lower priority class quickly regains the full capacity of the network.

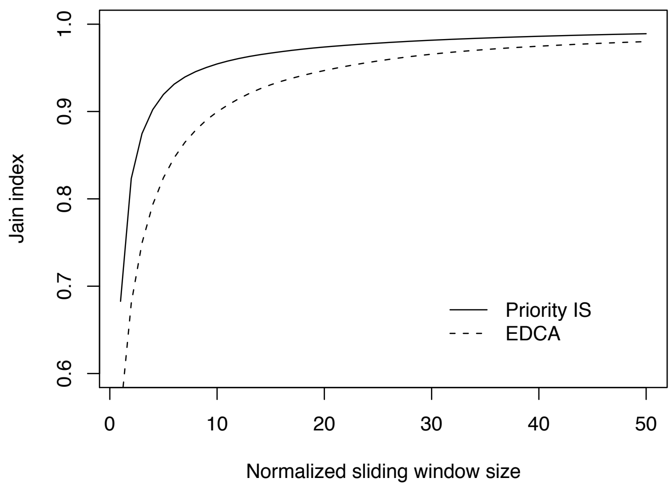



Figure 3.8: Jain fairness index for 10 stations, each station sending traffic of all 3 classes (we only show the Jain index for the highest priority traffic, because it is almost the same for other classes).

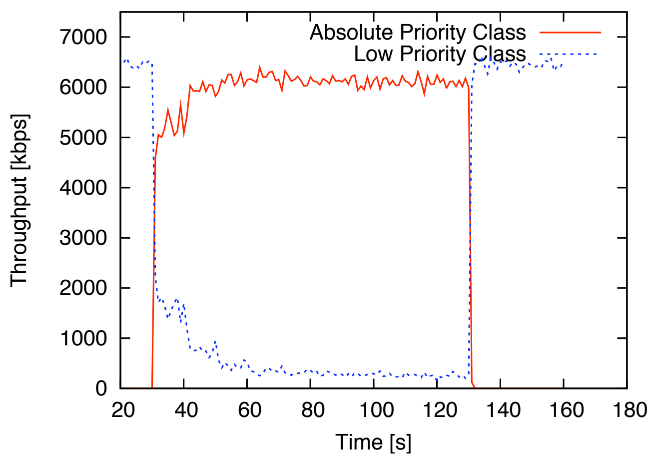



Figure 3.9: Evolution of the aggregate throughput per class in time. 10 stations generate low priority traffic all of the time while one station joins the network every 10 seconds and starts sending absolute priority traffic.

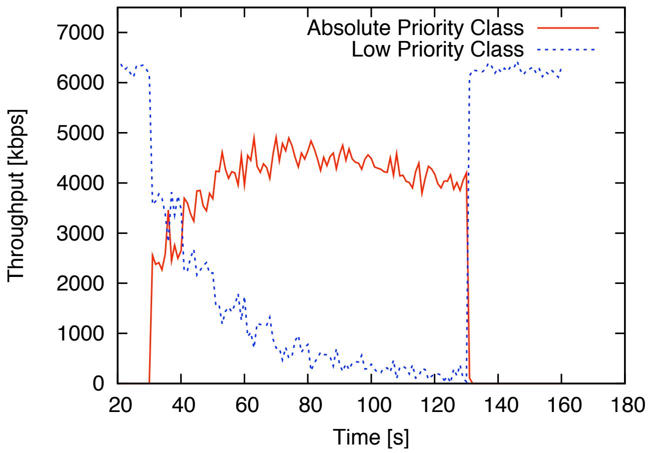



Figure 3.10: Evolution of the aggregate throughput per class in time for 802.11e EDCA. The simulation scenario as in Figure 3.9.

Figure 3.10 shows the results of the same simulation for the 802.11e EDCA. We can see that the absolute priority class obtains much lower throughput compared to our method.

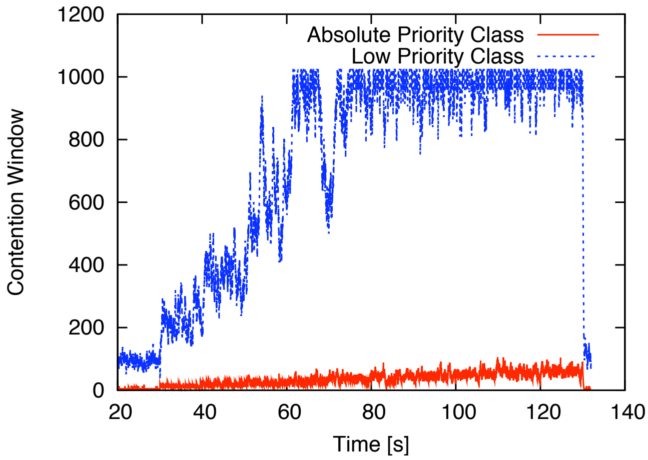



Figure 3.11: Evolution of the contention window for the absolute and low priority classes in time.

Figure 3.11 shows the evolution of the contention window for the absolute and relative low priority classes at a chosen station. We can see that when the station starts to send absolute priority traffic, the contention window of the relative class sharply increases up to the limit of 1024, so that it cannot compete with the absolute priority class. Note also that the contention window of the absolute priority class increases to adapt to an increasing number of contending absolute priority stations. Figure 3.12 presents a zoom

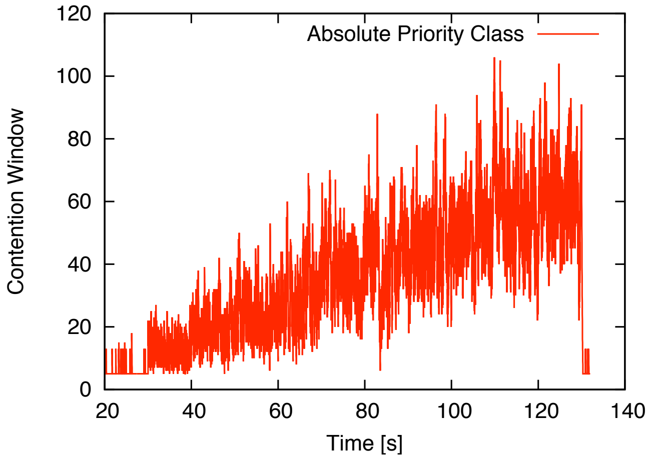



Figure 3.12: Evolution of the contention window for the absolute class in time.

into the behavior of the contention window for the absolute priority class to show its increase in function of the number of contending stations in this class.

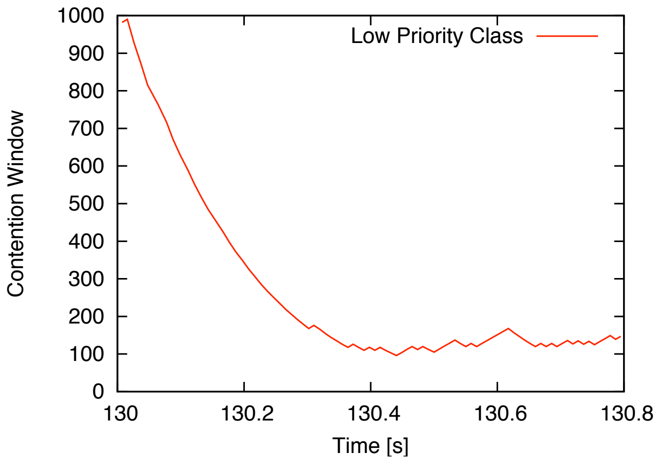



Figure 3.13: Zoom into the behavior of the low priority class at instant 130.

An important feature of our method is its rapid convergence. Figure 3.13 shows how quickly the contention window of the low priority class converges after the end of the absolute priority traffic: it takes roughly 300 ms to adjust to the appropriate value.

Finally, we have analyzed the collision rate when stations use our access method offering absolute priority (cf. Figure 3.14). It increases a little bit, because the target number of idle slots ($n_i^a = 3$) is smaller than the corresponding value used in *Idle Sense* (5.68). Nevertheless, it always increases very slowly with the number of active stations, which results in a very good aggregated throughput per class (cf. Figure 3.9).

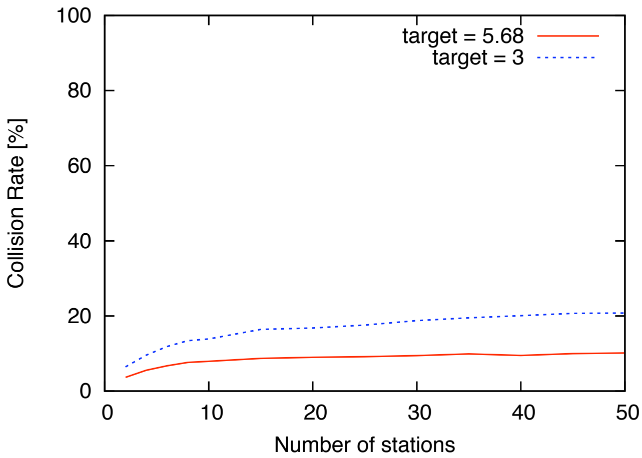



Figure 3.14: Collision rate *vs.* number of stations for different values of the target number of idle slots $n_i^{\text{target}}$.

By selecting different target values for absolute and proportional priorities, there is no need for using different AIFS as in 802.11e, because as our simulation results have shown, it is possible to proportionally allocate throughput as well as to provide absolute priority access only by adjusting contention windows. In particular, it is not necessary to use a larger AIFS for the low priority classes, because it leads to a lower channel utilization in the absence of absolute priority traffic.

Note that we have set the frame size of the absolute priority traffic to a large value of 1500 bytes to exhibit its contention with low priority traffic. For short frame traffic such as VoIP, we can observe a similar behavior with the absolute priority traffic gaining most of the channel capacity, but with a lower throughput due to an increased overhead. For proportional throughput allocation, we can adapt the expressions in Section 3.2 to take the difference in frame sizes into account.

## 3.5 Conclusion

In this chapter, we have proposed an access method that supports both relative proportional throughput allocation and absolute priorities in 802.11 wireless networks. It builds on the idea of the *Idle Sense* method that provides the optimal throughput and fairness for contending stations: each station adjusts its contention window based on the observed mean number of idle slots. We achieve absolute priority differentiation by setting the target value of the number of idle slots to a small value. In this way, the contention window of the absolute priority class becomes small too so that it gains more throughput compared to the low priority class. In relative proportional allocation, the low priority classes use the target value of the number of idle slots defined in *Idle Sense* and share the available throughput according to desired ratios.

Our simulations show that the proposed method achieves its objectives of relative and absolute differentiation both with respect to the aggregated throughput and the speed of convergence. Unlike 802.11e EDCA, it presents very good scalability—the throughput remains almost constant in function of the number of contending stations. As our mechanism is based on *Idle Sense* access method, it provides good fairness for traffic of the same priority.

# Chapter 4

# Handling Blocked Stations in Ad-hoc Networks

## 4.1 Introduction

In this chapter we consider the IEEE 802.11 [44] wireless networks in the ad hoc mode. When stations with omnidirectional antennas use the mandatory DCF access method to the radio channel, several performance problems may arise due to spatial positions of stations. We have already discussed some well-known spatial problems in chapter 2 such as hidden, exposed, blocked, and masked station problems.

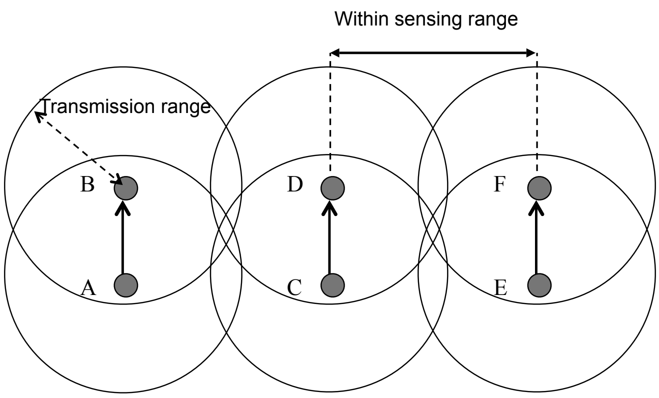
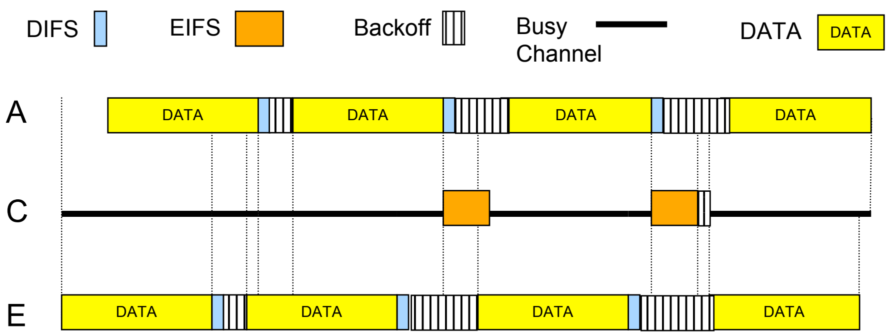
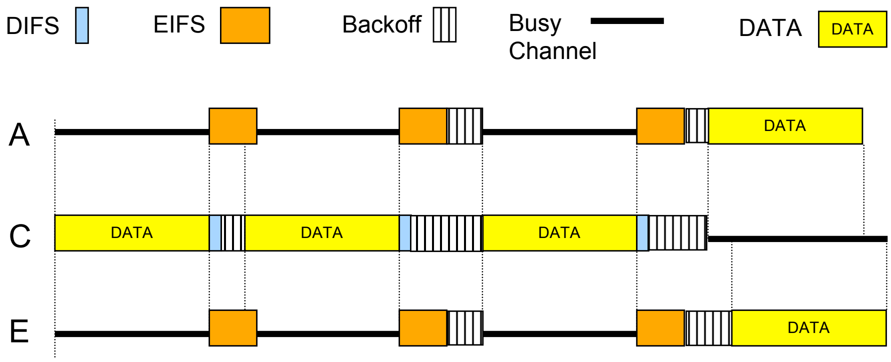



Figure 4.1: Three parallel pairs.

We focus on an exacerbated problem of exposed stations that we class into the category of *blocked stations*. In this problem, three pairs of stations communicate in parallel (pairs are within the transmission range): $A \rightarrow B$, $C \rightarrow D$, $E \rightarrow F$ (cf. Figure 4.1). Immediate neighbors like $A$ and $C$ are within the carrier sensing range of each other, but not in the transmission range and therefore they cannot directly communicate. If sending stations are greedy (they always have a frame to send), the external pairs gain much higher throughput than the internal pair $C, D$, which obtains almost null throughput. Such a spatial configuration of stations under 802.11 DCF leads thus to the starvation of the internal pair [32, 22, 21, 20]. The problem of blocked stations considered in this chapter is a fairly recent one. Dhoutaut identified and studied the configuration of three parallel pairs in his PhD thesis [32]. He has shown that the spatial configuration leads to long-term unfairness between stations and to the starvation of the internal pair. Further analysis and solutions have followed [21, 20, 22, 19].




(a) Starvation of the interior emitter




(b) Starvation of the exterior emitters

Figure 4.2: Starvation of emitters in the configuration of three parallel pairs

The performance problem comes from the fact that $C$ contends with two independent senders $A$ and $E$, while $A$ and $E$ only need to compete with one sender $C$. Moreover, any successful transmission of an external sender forces $C$ to wait during the EIFS interval because the corresponding packet cannot be decoded by $C$ and so it is seen as the noise. In this way, the external senders monopolize the channel and $C$ sees a permanently busy channel. We can observe this effect in Figure 4.2(a) (to simplify, we only show contention backoffs and transmissions, data and ACK frames included). When stations $A$ and $E$

send their frames, the channel observed by station $C$ is almost always busy, so $C$ cannot transmit. If we want to give more throughput to blocked station $C$, the total throughput will decrease anyway, because during the time it sends a frame, the external stations may send two frames in parallel. Figure 4.2(b) shows a scenario in which $C$ transmits several frames consecutively. In fact, if ever $C$ gains access to the channel and transmits its first packet successfully, it will defer for DIFS before the next transmission while the external emitters $A$ and $E$ have to wait for EIFS. This means that, in this scenario, $C$ has more chance to transmit the next packet compared with $A$ and $E$. Note that any transmission from $C$ is considered by both external senders as noise because they cannot decode it. These two scenarios show the asymmetrical behavior of DCF for three pair configuration.

With the development of *wireless mesh networks*, performance problems due to spatial positions of nodes become increasingly important. Mesh networks aim at covering large areas with a high capacity communication infrastructure to convey user traffic between access points providing service to mobile nodes and gateways to the wired Internet. To meet their objectives, they need to exploit multiple parallel paths and efficiently use the available capacity of the wireless medium. The topology of a wireless mesh network usually depends on available places for suitable deployment and may vary from some regular structures such as grids to more irregular graphs, but the existence of dense topologies with multiple parallel paths leads to the performance problems due to spatial positions of nodes [106, 69] that we try to address in this chapter.

In particular, we look for a solution to the problem of blocked stations that closely sticks to the standard 802.11 DCF access method, so that required modifications are minor. We propose a simple and efficient method called *Forced Transmissions*. It consists of detecting that a station is blocked by other stations and forcing a transmission. This results in a collision that increases the contention windows of blocking stations and leaves some channel time to the blocked station for transmitting. The blocked station forces a transmission according to some probability adjusted in function of the time spent waiting for the channel. The price for fixing the fairness problem is an increased number of collisions and a slightly lower overall throughput—we observe that our method increases the minimal throughput of the blocked station at the expense of the average and the total throughput of all stations. Unlike other solutions, our method gives the control of channel access to the blocked station whereas in other approaches, it is up to the blocking stations to decide when to yield the channel to the blocked station.

Our method also improves performance in a similar configuration with four pairs shown in Figure 4.3. In this asymmetric configuration, it is more difficult for the previously proposed solutions [19, 84] to sufficiently improve the throughput of the blocked station. Any proposal that tries to solve the problem of blocked stations needs to test this configuration and show that it provides sufficient throughput to the blocked station.

This chapter is organized as follows: we start with the discussion of two related works in the next section, which both are based on the principle of occasionally giving to a blocked station an opportunity to access the channel. We describe our proposal in Section 4.3. In Section 4.5, we will report on simulation experiments to compare our

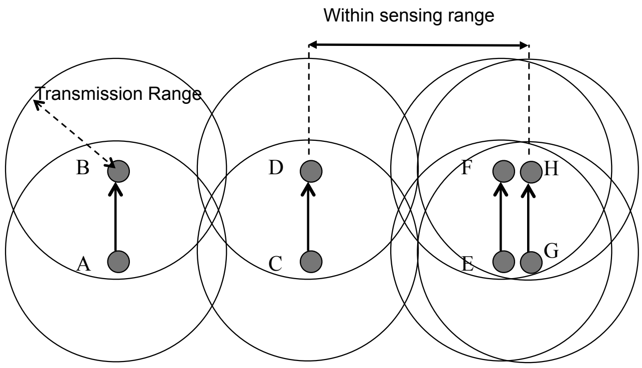


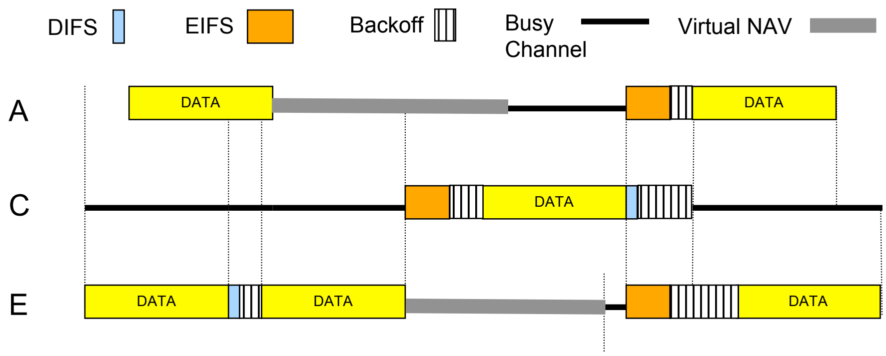

Figure 4.3: Four parallel pairs: stations E, F, G, and H are in transmission range of each other

method with DCF and PNAV, a previously proposed solution. The last section concludes the chapter and presents some future work.

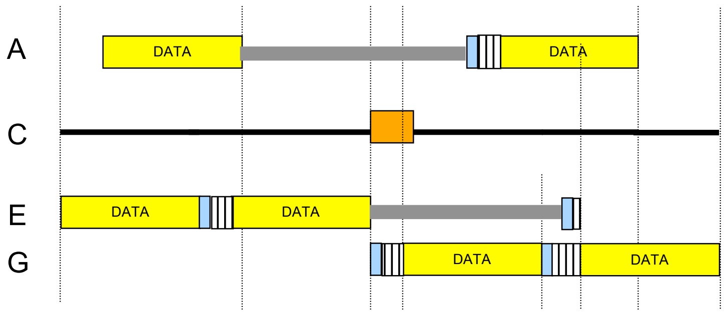
## 4.2 Granting Transmission Opportunity

Chaudet *et al.* have proposed the first solution to this problem called *Probabilistic NAV* or PNAV [19]. It is based on the principle of occasionally giving to a blocked station an opportunity to access the channel. With an adjustable probability $p_{nav}$, a station sets a NAV of duration $\delta$ after a transmission to allow other stations to access the channel. $p_{nav}$, the probability of yielding the channel depends on its utilization by stations. The method solves the problem of three pairs, but in a more general situation of the asymmetric four parallel pairs shown in Figure 4.3, it does not provide a satisfactory solution. In this configuration, the internal pair $C \rightarrow D$ is almost always starved by the external senders, because $E$ and $G$ are unlikely to set the NAV at the same instant, so that $A$, $E$, or $G$ may still monopolize the channel. Figure 4.4 shows the behavior of PNAV in three and four parallel pair configurations.

MADMAC proposes to address the spatial unfairness problem while maximizing global throughput [84]. The main idea is that an active station periodically senses the channel and when the channel is busy, it reduces its MAC throughput by introducing a waiting time before each frame to send. To cope with the problem of blocked stations in the configuration of three parallel pairs, it introduces the following adaptation: after $x$ consecutive successful transmissions, the contention window for the $(x + 1)$th frame is set to $2CW_{\min}$ ($x$ being a parameter of the protocol), to $4CW_{\min}$ for the $(2x + 1)$th transmission, and so on. According to this mechanism, a blocking station increases its contention window in order to give the opportunity to blocked stations to transmit their data frames. Similarly to PNAV, the proposed method solves the problem of the three pairs, but does not provide a satisfactory solution for the asymmetric configuration in Figure 4.3 for similar reasons.




(a) PNAV's behavior in a three parallel pair: A and E defer their transmissions for a virtual NAV. Meanwhile, C transmits a packet.




(b) Starvation of the interior emitter in a four parallel pair configuration: Although A and E set their virtual NAV but G has more chance to gain access to the medium.

Figure 4.4: The behavior of PNAV in three and four parallel pair configurations.

## 4.3 Forced Transmissions

We propose *Forced Transmissions*, a simple and efficient solution to the problem of blocked stations. When a station detects that it is blocked by others stations, it forces a transmission and causes a collision. This is an operation violating the principal rule of CSMA/CA stating that a station cannot transmit when there is an ongoing transmission on the channel. We are aware of this violation, however our simulations show that it results in an improved fairness of stations, i.e. the blocked station gains some channel access and blocking stations slightly lower their throughput.

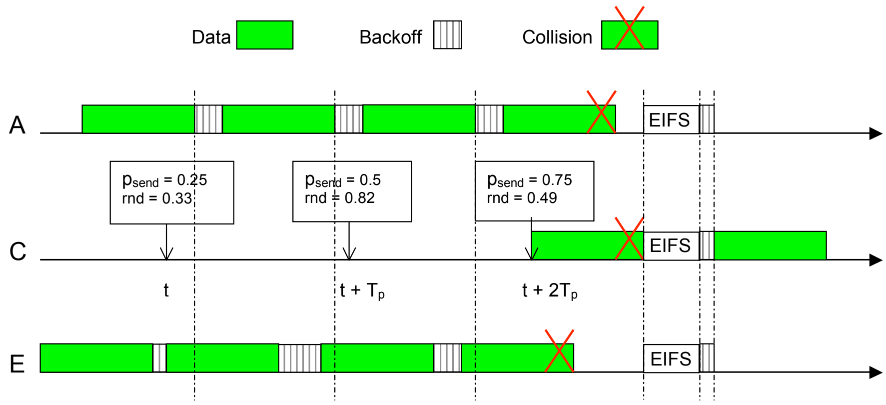



Figure 4.5: Transmission forced by blocked station $C$.

The *Forced Transmissions* mechanism is simple—the blocked station monitors during a period of $T_p$ whether the channel is monopolized by other stations or not. If so, it transmits during the current transmission of a blocking station with an adjustable probability $p_{\text{send}}$. The ongoing transmission will be thus corrupted and perceived by other stations as a collision. Figure 4.5 presents a timeline of events that occur when blocked station $C$ forces a transmission (to simplify we do not show the DIFS intervals). As long as node $C$ is transmitting its packet, nodes $E$ and $A$ cannot access the channel. When node $C$ terminates its data transmission, they will wait for an ACK during interval ACK_TIMEOUT. Then, the next backoff will start after ACK_TIMEOUT + DIFS = EIFS—as nodes $E$ and $A$ cannot interpret the data frame sent by $C$, they should wait for the EIFS interval rather than DIFS before the next backoff.

Two cases may then arise: first, the blocked station transmits during the data frame of a blocking station. Thus, all receiving stations enforce the EIFS interval (364$\mu$s for 802.11b) after the end of the transmission as presented in Figure 4.5. Second, if the blocked station transmits during the ACK frame, the transmitting blocking station defers during the interval equal to the ACK_TIMEOUT set to EIFS - DIFS. In both cases, all stations will synchronize after these long intervals and then they will contend for the channel. As blocking stations double their contention windows after the collision while in our method the blocked station uses $CW_{\text{min}}$ for the next transmission, it has a greater probability of gaining access to the channel.

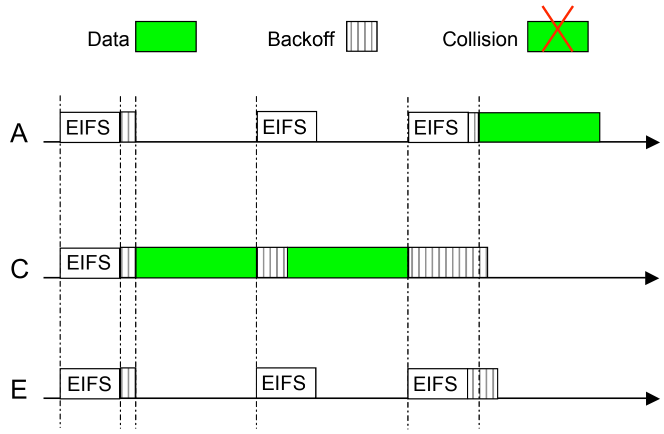



Figure 4.6: Transmissions after a forced collision.

Figure 4.6 presents what may happen next. After having successfully sent its first frame, the blocked station is likely to capture the channel for the next transmissions: as the blocking stations are too far away from the blocked station to interpret its transmission, they will differ access by enforcing the EIFS intervals while the blocked station only waits for the DIFS interval and the backoff chosen from the contention window of $CW_{\min}$. This may even happen several times until the blocked station chooses a backoff longer than the sum of the EIFS interval and the backoff of the external stations (cf. Figure 4.6).

Note that as long as central sender is transmitting, the exterior senders are blocked as well. In this case the overall throughput of the network is lower than when two exterior senders transmit. The number of successive transmission for the central sender can be limited in order to guarantee the short-term fairness within the network.

To maintain an acceptable collision rate, a blocked station uses a probabilistic mechanism to decide on the transmission that generates a collision: it transmits with probability $p_{\text{send}}$ that depends on the interval during which the station has been blocked. To update $p_{\text{send}}$, a station periodically verifies if it is blocked or not. If it is blocked, $p_{\text{send}}$ is increased by $p_{\text{step}}$, otherwise it is decreased by $p_{\text{step}}$. Algorithm 2 illustrates our MAC mechanism to deal with the blocked stations.

The last detail of the proposed method is to define how a station detects whether it is blocked or not. One way is to observe the channel during a sufficiently long time to see if a station contends with at least two other stations. If during the interval of the transmission time plus DIFS, the idle time is less than DIFS, this means that the station is blocked. The *Idle Sense* access method [40] is particularly well adapted to detect such situations. If the contention window of a station does not oscillate around a value that depends on the network load, but rather it raises indefinitely, this means that the station is blocked. In our simulations, we use the first mechanism that tracks busy periods longer than the one corresponding to the transmission of the Maximum Transmission Unit (MTU) plus DIFS.

**Algorithm 2** *Forced Transmission*

$T_p \leftarrow DIFS + T_{MTU}$
During each $T_p$ {
$T_{idle} \leftarrow$ the channel idle time during $T_p$
**if** $(T_{idle} < DIFS)$ **then**
  /* The station was blocked during $T_p$*/
  $p_{\text{send}} \leftarrow p_{\text{send}} + p_{\text{step}}$
**else**
  /* The station was not blocked during $T_p$*/
  $p_{\text{send}} \leftarrow p_{\text{send}} - p_{\text{step}}$
**end if**
/* Forced transmission is probabilistic.*/
**if** $(random(0,1) < p_{\text{send}})$ **then**
  forced_transmit($pkt$)
**end if**
}

Compared to other solutions such as PNAV, the advantage of our method is that the decision of reacting to bad spatial conditions is left to the blocked station. A station may use PNAV in two ways. Either it occasionally defers access for a NAV independently of the spatial situation of its neighbors or it uses DCF by default and switches to PNAV if there is a blocked station nearby. In the first way, the performance of the station is degraded even if there are no blocked stations in the neighborhood. The second way requires a signaling protocol through which the station learns that it is blocking another one and starts using PNAV. As the blocked station can hardly access the channel for communication, such signaling may even be impossible. When using *Forced Transmissions*, a station detects that it is blocked and reacts to gain the channel independently of what other stations do. If there are no blocked stations, there is no performance penalty.

## 4.4 Performance Analysis of Forced Transmissions

In this section, we analyze our mechanism and derive the throughput of the blocked station in the configuration of three parallel pairs. We assume that the station in the middle is always blocked by the external stations. We also assume that simultaneous transmissions from two neighbor pairs never succeed, so that the collision is symmetrical.

To analyze our mechanism, we model the behavior of the blocked station $C$ with the Markov chain shown in Figure 4.7. The station checks each $T_p$ whether the channel is monopolized by other stations or not. So, at the end of each period $T_p$, the station increases $p_{\text{send}}$ by $p_{\text{step}}$ and waits for another period of $T_p$ ($W$ transition) or sends a frame ($S$ transition) to generate a collision. To simplify the model, we reset $p_{\text{send}}$ after the $S$ transition, whereas in simulations presented in the next section, it is decreased by $p_{\text{step}}$. We consider data frames of a fixed size.

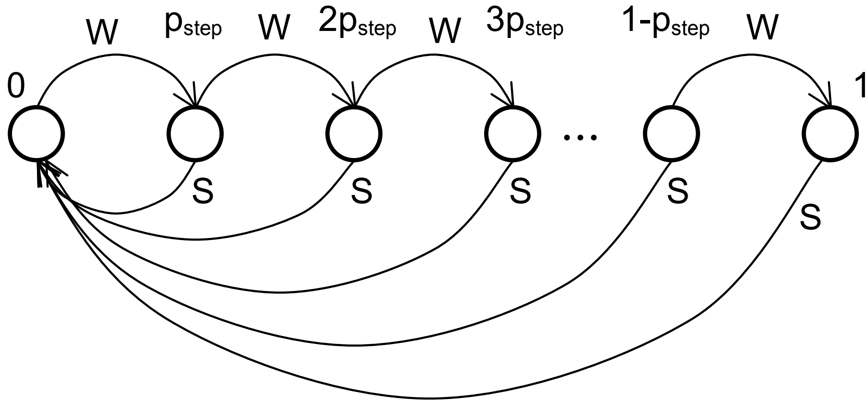



Figure 4.7: Markov chain model for the forced transmission mechanism.

We can find $T_{\text{avg}}$, the average waiting time of the blocked station before a forced transmission as:

$$\begin{aligned} T_{\text{avg}} &= T_p + (1 - p_{\text{step}})\Big[T_p + \\ &\quad +(1 - 2p_{\text{step}})[...[T_p + p_{\text{step}}T_p]..]\Big]. \end{aligned} \tag{4.1}$$

Thus, its throughput is the following:

$$\begin{aligned} X &= \tfrac{p_{\text{win}} N_c S_d}{T} \\ T &= T_{\text{avg}} + T_{\text{col}} + T_{\text{timeout}} + \\ &\quad + p_{\text{win}} N_c(\bar{T}_{\text{bkoff}} + DIFS + T_t), \end{aligned} \tag{4.2}$$

where $N_c$ is the average number of transmissions done by the blocked station after a collision, $S_d$ is the size of a data frame, $T_{\text{col}}$ is the duration of a collision, $T_{\text{timeout}}$ is the time interval between the end of the data frame transmission and the end of the ACK frame, $p_{\text{win}}$ is the probability that the blocked station access the channel just after the collision (it is the probability that its backoff is shorter than the backoffs of the external stations taken from the contention windows of $2CW_{\text{min}}$), $T_t$ is the transmission time of a data frame (ACK included):

$$T_t = T_d + \Delta + SIFS + \Delta + T_{\text{ack}}, \tag{4.3}$$

$\Delta$ being the maximum propagation delay and $T_d$ the transmission time of a data frame. If we assume that the contention window of the blocked station is the half of CW for external stations, $p_{\text{win}}$ is equal to $\frac{9}{16}$.

Figure 4.8 presents the throughput of the blocked station in function of $p_{step}$ and $N_c$, the average number of consecutive transmissions after gaining the channel. We can see that it increases monotonically with probability $p_{step}$, because the blocked station generates an increasing number of collisions and access the channel more often.

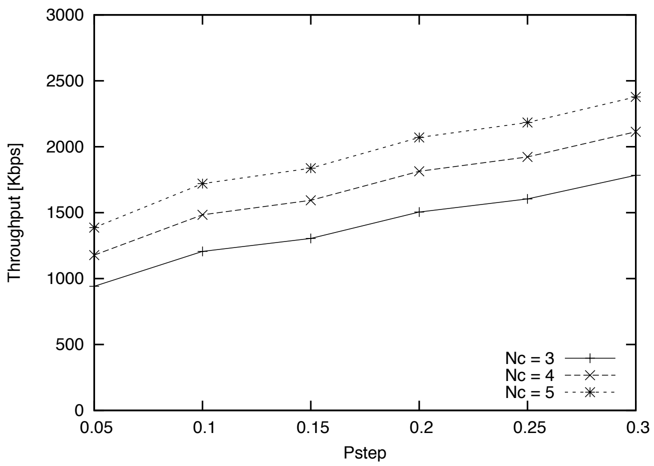



Figure 4.8: Throughput of the blocked station in function of $p_{step}$ and $N_c$.

## 4.5 Simulation and Results

We have implemented the proposed method by modifying the standard implementation of the 802.11 DCF in NS2 (version 2.29) [73]. The PHY and MAC parameters are those of IEEE 802.11b. We have considered a configuration of $n$ parallel sender-receiver pairs as a generalization of the basic three parallel pairs. In this case, when a station uses DCF even pairs are starved and odd pairs monopolize the channel, for instance in the configuration of five parallel pairs, the throughput of the second and the fourth senders is almost null.

We assume that each sender behaves like a greedy CBR source sending 1000 bytes frames. A blocked station sends a data frame of size 1000 bytes to force transmission (it may also send a small dummy frame, but in our simulations, we have assumed the same frame size). We log the amount of data successfully received during the simulation process. The distance between the sender and the receiver is set to 150 m while the distance between neighbor senders is 350 m. The transmission and carrier sensing ranges are 160 m and 400 m, respectively. Each point reported in figures represents the average of 10 simulation runs, each run taking 30 seconds. We present simulation results for transmissions without RTS/CTS exchange, because the results with RTS/CTS are not significantly different.

The objective of our method is to guarantee a minimum throughput for blocked stations. Thus, we use the *minimum throughput* as the main metric for comparing our method with DCF and PNAV. To evaluate the degradation of global network performance, we also report the *average throughput* and the *total throughput* of all stations. The *maximum throughput* allows us to see how the blocking stations decrease their throughput to increase the throughput of blocked stations. We also use the Jain index to evaluate the fairness of the throughput obtained by stations [52].

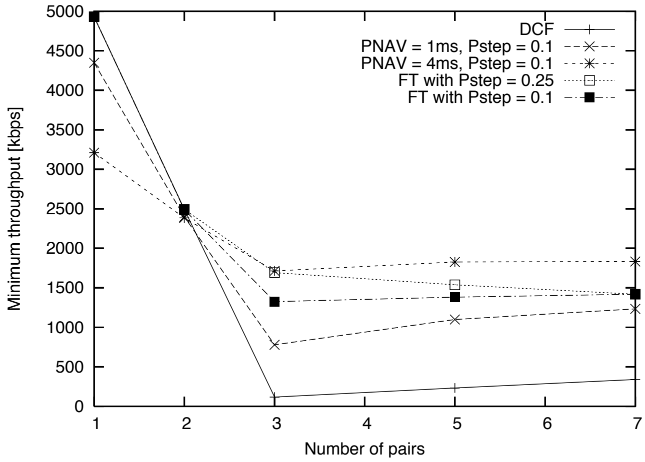



Figure 4.9: Minimum throughput, $n$ parallel pairs.

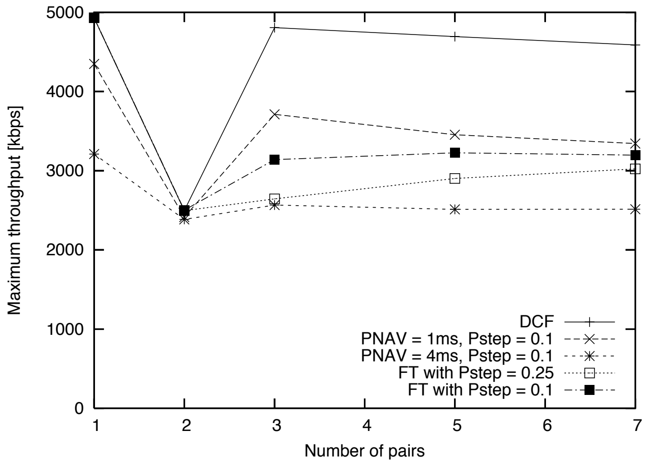



Figure 4.10: Maximum throughput, $n$ parallel pairs.

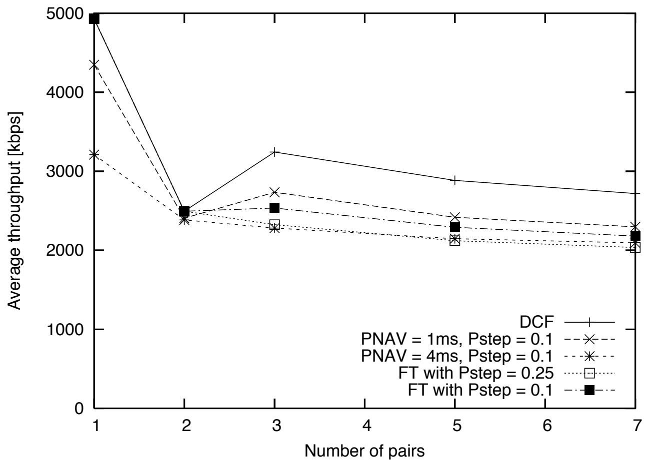



Figure 4.11: Average throughput, $n$ parallel pairs.

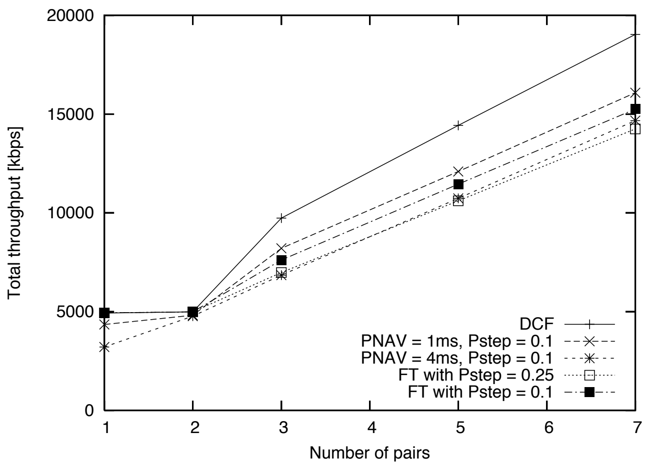



Figure 4.12: Total throughput, $n$ parallel pairs.

Figure 4.9–4.12 show the simulation results for the configuration of $n$ parallel pairs, $n = 1, 2, 3, 5, 7$. First, we can see that when the problem of blocked stations does not arise, i.e. for $n = 1, 2$, the performance of our method is exactly the same as for the standard DCF, because when all the stations in a wireless network can sense each other, our method degenerates to DCF. Unlike this desired behavior, PNAV incurs performance degradation, because a station occasionally relinquishes the channel so that potentially blocked stations can use it.

Our method is only active for blocked nodes. As a result, in a wireless LAN, in which all stations are within the mutual carrier sensing range, our proposal does not change the behavior of the standard DCF access method because there is no blocked station in such a network.

For the case of $n = 3, 5, 7$, when the problem of blocking stations does appear, we can observe that the minimum throughput under DCF is almost null and the blocking stations monopolize the channel by obtaining almost 4.9 Mb/s. Our method guarantees the minimum throughput of the blocked stations between 1.4 and 1.7 Mb/s depending on probability $p_{\text{step}}$, a value that barely varies with $n$ (cf. Figure 4.9). Blocked stations obtains this improved minimal throughput at the expense of only a small decrease in the average throughput compared to DCF (e.g. for $n = 3$, 2.5 Mb/s vs. 3.2 Mb/s for DCF, cf. Figure 4.11). Recall that any method giving some transmission opportunity to the blocked stations decreases the total throughput. In the case of our method, the decrease remains reasonable (e.g. for $n = 3$, 7 Mb/s vs. to 9.5 Mb/s for DCF, cf. Figure 4.12). PNAV obtains the best results for a NAV of 4 ms—in this case, the minimal throughput is important along with the total throughput. Compared to our method, PNAV with a NAV of 4 ms presents a similar or better minimal throughput with almost equal total throughput.

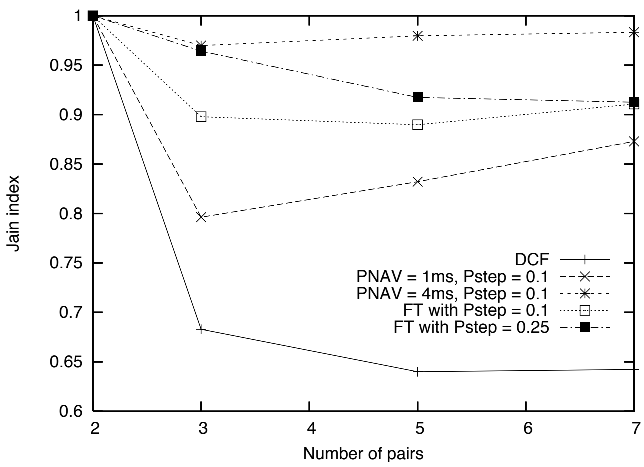



Figure 4.13: Jain index for $n$ parallel pairs.

Figure4.13 presents the Jain index (for the definition cf. Chapter 2) of the throughput obtained by stations in the configuration of $n$ parallel pairs. Note that here we study the long-term fairness and so the window size $w$ is set to the period of simulation that is 30 seconds. Also, $\gamma_i$ represents the throughput obtained by station $i$. We can see that for $n = 3$, the Jain index for DCF is around 2/3, because one value of the throughput is close to zero. We can see that the index is between 0.9 and 1 for our method while the total throughput is still high. PNAV obtains even slightly better results, especially for $n = 5$ and 7.

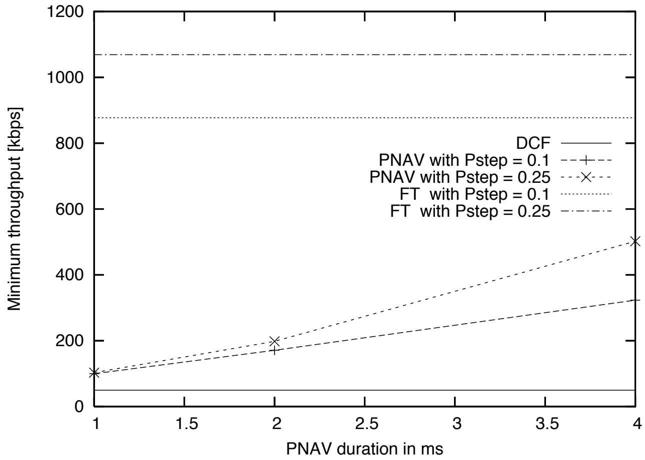



Figure 4.14: Minimum throughput, four pairs.

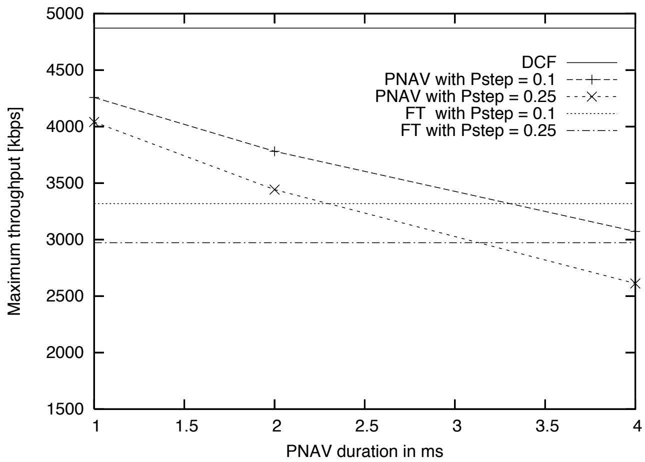



Figure 4.15: Maximum throughput, four pairs.

Figure 4.14–4.17 show the simulation results in the asymmetric configuration of four parallel pairs (cf. Figure 4.3): the minimum throughput experienced by the blocked station $C$, the maximum throughput obtained by one of the blocking stations ($A$), the average and the total throughput. As previously, we can observe that the minimum throughput under DCF is almost null and blocking station $A$ monopolizes the channel

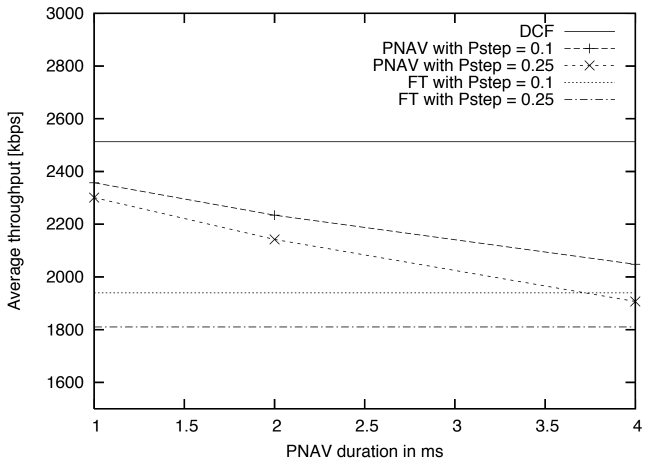



Figure 4.16: Average throughput, four pairs.

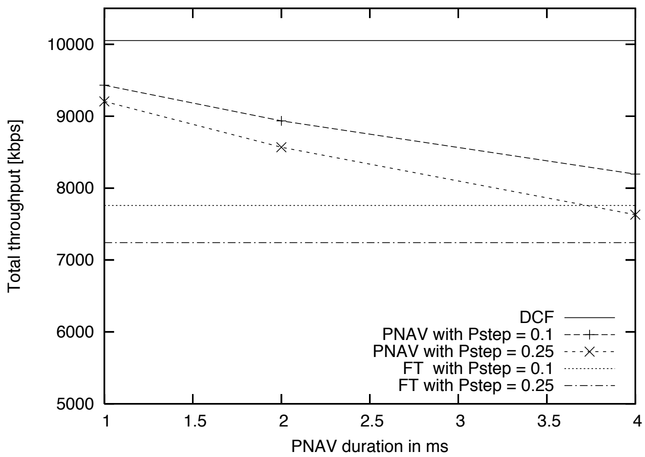



Figure 4.17: Total throughput, four pairs.

by obtaining almost 4.9 Mb/s. Under our method the blocked station obtains much better minimal throughput than in the case of PNAV with NAV of 4 ms (cf. Figure 4.14). This is achieved at the expense of only a small decrease in the average throughput compared to DCF as shown in Figure 4.16 (between 1.8 and 2 Mb/s vs. 2.5 Mb/s for DCF). The total throughput remains important though less than that for DCF (e.g. 7.2 Mb/s vs. 10 Mb/s for DCF, cf. Figure 4.17). In this configuration, PNAV does not perform as well as for $n$ parallel pairs: the minimal throughput is lower than in our method while the total throughput is almost the same for our method and for PNAV with a NAV of 4 ms.

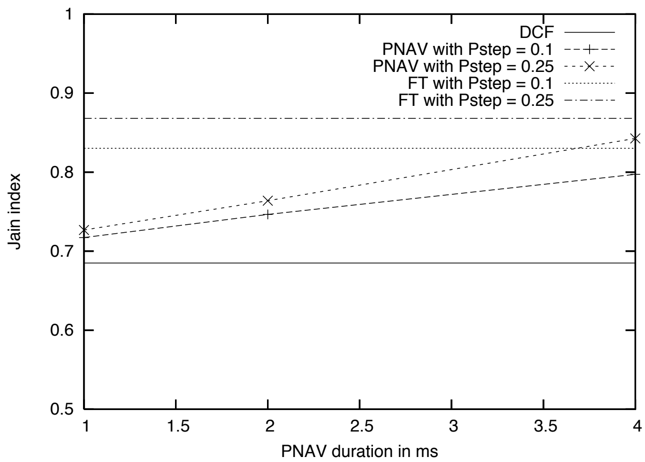



Figure 4.18: Jain index for asymmetric four pairs.

Figure4.18 presents the Jain index in the asymmetric configuration of four pairs. We can see that the index is better for our method than that for DCF and for PNAV with NAV of 4 ms while the total throughput remains similar to PNAV (cf. Figure4.17).

## 4.6 Conclusion

In this chapter, we have proposed *Forced Transmissions*, a simple and efficient solution to the problem of blocked stations. It consists of detecting that a station is blocked by others stations and forcing a transmission. This results in a collision that increases the contention windows of blocking stations and leaves some channel time to the blocked station for transmitting. The blocked station forces transmission only with some probability adjusted in function of the time spent waiting for the channel to become idle.

Our simulations show that the proposed method increases the minimal throughput of the blocked stations in the configuration of $n$ parallel pairs. The price for fixing the fairness problem is an increased number of collisions and a slightly lower overall throughput. PNAV provides a good solution to the problem of blocked stations only if its NAV duration is short, so that the total throughput remains high. For the configuration of $n$ parallel pairs PNAV with a NAV of 4 ms provides slightly better overall performance than our method. However, in a particular configuration of four asymmetric parallel pairs, our method outperforms PNAV.

Our method presents a nice feature: the decision on reacting to bad spatial conditions is left to the blocked station. As described previously, a station operating under PNAV needs to defer access periodically even if there is no blocking stations nearby, thus its performance is lower; or it requires a signaling protocol so that when a blocked station detects its situation, it notifies its neighbors asking for using PNAV. When using *Forced Transmissions*, a station decides itself that it is in an abnormal situation and reacts independently of what other stations do. If there are no blocked stations, then there is no performance penalty.

The proposed method is fairly general and can be deployed in any multi-hop environment (e.g. wireless sensor networks, wireless mesh networks, or wireless ad-hoc networks) in which the problem of blocked stations and their starvation may appear.

Chapter 5

# Molecular MAC for Multi-Channel Wireless Mesh Network

## 5.1 Introduction

In a single-cell wireless LAN all stations are within the range of each other or within that of a base station. 802.11 devices based on DCF perform fairly well for this wireless networks. Moreover, as we discussed before in chapter 2, much research work has already been proposed to improve the performance of the channel access method in a single-cell 802.11 WLAN. Some enhancements to this access method, e.g. *Idle Sense* [40] provide very efficient solutions in terms of throughput as well as fairness.

However, in the simplest case of a wireless mesh network, in which stations have only one network interface and use a single radio channel, the performance of packet forwarding over multiple links quickly degrades with the number of hops due to channel contention and spatial problems such as hidden, exposed, masked, and blocked nodes [22, 82, 51, 83, 19, 71].

In a nutshell, performance can be greatly improved if transmissions can occur in parallel instead of being locally mutually exclusive. This involves using orthogonal communication channels to multiplex data transmissions: we can improve performance if mesh routers take advantage of parallel transmissions over neighbor links, which can be done according to different schemes:

1. neighbor mesh routers use multiple network interfaces tuned to different radio channels,
2. they use multiple network interfaces with directional antennas,
3. they use a single network interface able to switch radio channels on a per frame basis.

We consider the last case for which several authors have already proposed solutions aiming at improving the performance of packet forwarding [7, 88, 23, 25, 12]. Although our proposal mainly addresses the case of nodes with single network interfaces, we will show that it is straightforward to take advantage of the availability of multiple interfaces.

In this chapter, we propose a *Molecular MAC*, a slight modification of the standard IEEE 802.11 access method that takes advantage of dynamic channel switching at neighbor mesh routers to efficiently forward packets over multiple hops. Molecular MAC requires a specific organization of the whole mesh network to form a desired topology and assign specific roles to each mesh router.

We adopt a *molecular* analogy to deal with the organization of the wireless mesh required by the proposed MAC scheme. Mesh routers are either *nuclei* or *electrons*. A nucleus with one or several electrons forms an *atom* and dynamically chooses a radio channel to be used for all communications inside its atom. The nucleus and electrons of an atom contend for the channel by means of an access method such as the 802.11 DCF so, in this respect, this proposal benefits from the advances in wireless access methods. Packet forwarding relies on Molecular MAC that defines how nuclei *notify* electrons about pending packets and how electrons *pull* them from nuclei for further forwarding. To limit interference, adjacent atoms use different channels and electrons belonging to them switch between channels so that packets conveyed across several atoms benefit from parallel communications. Such a molecular mesh network constrains the graph of nodes: by eliminating some links, the graph becomes bipartite with one electron between any two nuclei. In this chapter, we only formulate the requirements for the construction algorithm of the molecule mesh network. Defining an efficient distributed algorithm for molecule construction is out of the scope of this thesis and concerns our future work.

The contribution of this chapter is twofold. First, we present a new efficient MAC mechanism for forwarding packets in a molecular mesh network. In this mechanism, electrons initiate all frame exchanges with a nucleus by explicitly *pulling* a frame. We base our work on 802.11 and the DCF access method, although we can similarly extend any other channel access method. Second, we evaluate the proposed scheme through simulation and compare with other proposals. We show that Molecular MAC obtains much better performance in terms of throughput, packet delivery rate, end-to-end delay, and fairness.

This chapter is organized as follows: we start by describing our approach and its basic concepts (Section 5.2). Section 5.3 defines the Molecular MAC mechanism. We then discuss the construction of a molecular mesh, neighbor discovery, and channel assignment (Section 5.4). Note that, our work in this thesis does not constitute the definition of an efficient mechanism to construct the molecule, but we only discuss the conditions that such an mechanism has to fulfill. We mainly focus on MAC aspects of our molecular architecture. Section 5.5 presents our simulation results. The last section concludes the current thesis chapter.

## 5.2 Approach

We assume at first that each mesh router has a single wireless network interface able to switch channels. We adopt a *molecular* analogy to deal with the organization of a

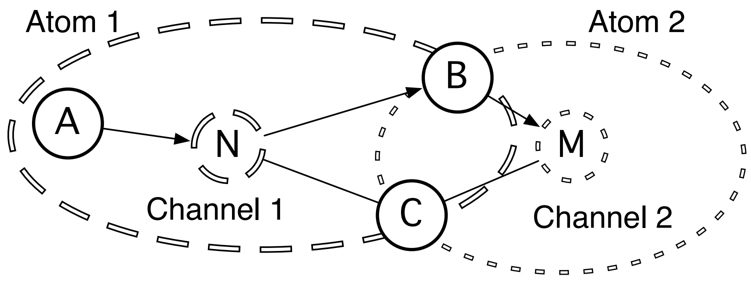



Figure 5.1: Two atoms sharing two electrons

wireless mesh network (cf. Figure 5.1). An *atom* is a basic entity for constructing a mesh network. It is composed of a *nucleus*, a mesh router that uses a fixed channel for communicating with *electrons*, its immediate neighbors. Atoms bond together to form a *molecule*, which corresponds to a connected wireless mesh network. Electrons belong to adjacent atoms and communicate with their nuclei. There is no direct link between two electrons nor between two nuclei. Figure 5.1 illustrates this view. Mesh routers N and M are nuclei of two atoms bonded by two mesh routers corresponding to electrons B and C.

To operate in parallel without interference, two neighboring atoms should use orthogonal channels. In this way, parallel frame transmissions will not suffer from contention nor from any undesirable spatial effect such as hidden and exposed nodes. At the same time, we need a means for forwarding packets between atoms that use different channels. Molecular MAC uses the nucleus as a kind of a *virtual access point*: it is either a receiver or a transmitter of any frame in the atom. Packet forwarding relies on electrons that switch between channels of all neighboring atoms. This scheme assigns one channel to each nucleus and all electrons use it for communication with a nucleus (either for reception or for transmission). Upon construction of a mesh, a nucleus dynamically chooses its channel depending on the channels used in its neighborhood by other atoms. By choosing different channels, two neighboring atoms that might interfere if they would use the same channel, can limit interference problems, which contributes to achieving high capacity.

There is a possible contention in an atom between entities that use the same channel (e.g. N, A, B, and C in Atom 1). Nodes in an atom can manage contention with the standard IEEE 802.11 DCF, however it may need to enable the RTS-CTS option to avoid hidden node problem between electrons. Any other optimized access method such as *Idle Sense* [40] can also be used within an atom.

The architecture presented in Figure 5.2 shows a more general example of molecular wireless mesh networks. Forwarding packets on a path between a source and a destination involves a series of alternating electron and nucleus nodes. For instance, node S can send packets to node D over a path spanning four atoms that use different channels.

In the next sections, we present the details of the Molecular MAC mechanism for packet forwarding.

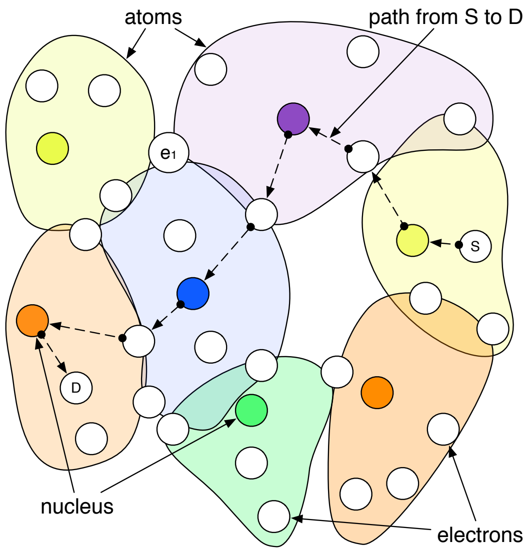



Figure 5.2: Molecular architecture of a wireless mesh network

## 5.3 Molecular MAC

To interconnect atoms and achieve efficient packet forwarding electrons need to switch between channels used by neighboring nuclei so that packets going through adjacent atoms benefit from parallel communications. Current 802.11a/b/g wireless cards can switch channels in less than $30\mu s$ [104] so electrons can alternate transmissions on different channels on a per-packet basis. Besides, no hardware constraint justifies that this delay cannot be further reduced.

However, when neighboring nodes dynamically choose different channels for transmissions, they face the problem of *deafness* when a node tries to send a frame on one channel while the intended receiver is listening to or is sending on another one.

### 5.3.1 Deafness Avoidance

Deafness is a challenging problem for multichannel nodes. There are two main approaches to solve it: either nodes reserve a common channel for signaling and use it to agree on another channel for a given transmission [101, 62], or use periodical rendezvous points to negotiate which channel to use for further transmissions [88, 7, 23]. Both solutions have drawbacks: a fixed signaling channel means less available resources for data transmission, while periodical rendezvous increases the overhead and requires some form of temporal synchronization between mesh routers.

Molecular MAC solves the deafness problem without these drawbacks: a nucleus chooses its channel, announces it to all electrons, and stays tuned to the channel. An electron belonging to several atoms knows the channels of its nuclei and can switch to a given channel when it wants to communicate. Packet forwarding in this case further

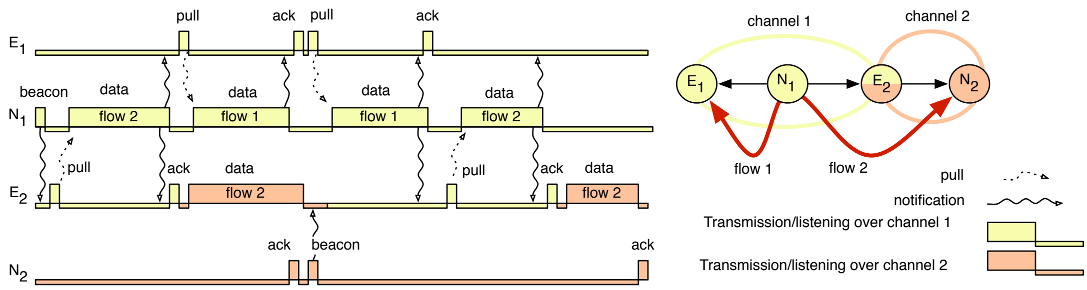



Figure 5.3: Pull mechanism and notifications for packet forwarding between nodes $E_1$ and $N_2$

requires two functionalities: a nucleus needs to *notify* an electron that there is a packet to forward and the electron needs to *pull* the frame containing the packet from the nucleus. We propose to implement frame notification through two mechanisms: piggybacking on data frames or including the information in periodically sent beacons. We redefine the semantics of the CTS control frame to use it when an electron needs to pull a frame from a nucleus. The role of a nucleus in forwarding reduces to buffering packets for electrons, notifying them that there is something to receive, and transmitting a packet on-demand when an electron asks for a frame.

Figure 5.3 presents an example of packet forwarding along two flows: Flow 1 between nodes $N_1$ and $E_1$, and Flow 2 between nodes $N_1$ and $N_2$. To simplify, the figure neglects all backoffs generated by the underlying channel access method. The example starts when node $N_1$ has a packet to send to electron $E_2$. It notifies the electron through a beacon on channel 1 that there is a pending frame to pull. Electron $E_2$ sends the CTS control frame (pull) to request the data frame. $N_1$ responds to the pull request of $E_2$ by sending its data frame. With this data frame, $N_1$ includes also a piggybacked notification about a pending data frame of Flow 1 ready for electron $E_1$. Electron $E_1$, listening on the channel 1, receives this notification. It, then, sends the CTS control frame to request the data frame.

When $N_1$ sends this frame containing the packet of Flow 1 on channel 1, electron $E_2$ can simultaneously forward the packet of Flow 2 to $N_2$ on channel 2. Note that transmissions do not suffer from deafness, because nuclei always use their channels and electrons pull frames on a given channel before receiving them on the same one.

### 5.3.2 Frame Notification

An electron must learn about data frames buffered at a nucleus. As several electrons are potential destinations of the frames, a nucleus needs to maintain a list of frame destinations and advertise it to all electrons. One way of notifying is to piggyback the list of pending destinations onto data frames. We can easily implement piggybacking in 802.11, because the maximal frame size is much larger than the common limitation of 1500 B due to Ethernet compatibility.

When an atom is idle, i.e. there is no traffic to forward, there is also no opportunity to piggyback notifications onto data frames. We thus propose to enhance 802.11 `beacon` frames to include the list of pending destinations in periodic broadcasts. A nucleus must send either a beacon or a notification piggybacked onto a data frame at most every $T_b$ interval. However, to speed up notification, the nucleus can send beacons with a higher frequency (e.g. an interval of several DIFS in the case of DCF), if it is idle, because anyway there is no other traffic to forward (we have used $T_b = 5ms$ in our simulations).

Figure 5.4(a) shows a beacon frame issued by a nucleus to notify its electrons of their pending frames. Its fields are as followings: Sender address, its role (nucleus), its channel, and the list of electrons ($E_1$ and $E_2$) for which it has a data frame to send. In Figure 5.4(b), electrons list is piggybacked to a data frame.


(a) Notification via a beacon


(b) Notification piggybacked to data frame

Figure 5.4: A nucleus notifies its electrons of their pending data frames using beacon or data frame

Since an electron may belong to several atoms, it needs to periodically switch its channel to listen to beacons from all neighboring nucleus. Thus, an electron must listen during at most $T_N$ interval to each of its neighboring nuclei. As soon as the electron receives either a `beacon` or a piggybacked notification, the electron can send a `pull` control frame (CTS) to the nucleus to receive the data frame. On the contrary, if the electron does not need to receive a data frame, it can switch to another atom.

When an electron knows that at least one nucleus has a data frame to transmit and at the same time it also has a data frame to send, it randomly chooses between sending or pulling the frame. Such a strategy maximizes fairness and leads to shorter forwarding delays since it does not privilege neither reception nor emission.

When an electron knows that one of its atoms is busy, it can perform other tasks, i.e. listen to other nuclei for notifications or perform neighbor discovery to establish links with nodes entering the mesh (cf. Section 5.4.3).

### 5.3.3 Optimization: Fast Reply

To improve throughput and lower the overhead due to piggybacking and pulling, we propose a fast reply mechanism: upon receiving a data frame from a electron, a nucleus can reply with another data frame, if present in its buffer, piggybacking the required acknowledgement. Figure 5.5 illustrates this sequence of three frames `data/data/ack`, two data frames being transmitted in the opposite direction. The scheme reduces the overhead introduced by piggybacking and pulling if flows are bidirectional.

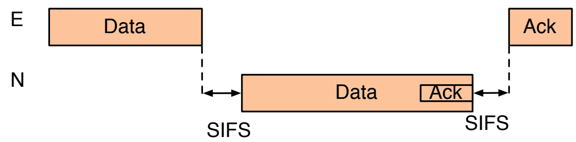



Figure 5.5: Fast reply optimization for frame transmission Electron→Nucleus

### 5.3.4 Handling Multi-Channel Hidden Terminal

As all nodes belonging to an atom are in the transmission range of the nucleus, RTS/CTS handshaking can alleviate the single-channel hidden station problem in an atom. However, due to channel switching a multi-channel hidden station problem may arise. When electron $E$ switches to a new channel to send a data packet to one of its neighboring nuclei, its NAV value is not up to date with respect to the new atom, because it has no prior information about the state of the new channel. Now, if it senses the new channel busy, it defers its transmission. Otherwise, it transmits its data frame after the contention period. In the latter case, there may be an on-going transmission from a hidden station that cannot be perceived by electron $E$ and leads to a collision.

One way for $E$ to avoid this problem is to defer its transmission during the time needed to transmit a MTU, but the problem of such a method is that it wastes the channel time. Another solution is to use a busy tone in each nucleus and to tune it as long as there is a on-going transmission within an atom. A newly arrived electron defers its transmission, if the busy tone is on. In our simulations, there are no obstacles and all stations of the atom are within the carrier sense of each other. As a result, the newly arrived electron perceives any transmission within the atom. Therefore, it does not transmit during the frame transmission as long as there is a frame in transmission. In this way, the multi-channel or single channel hidden terminal problem disappears. However, our main concern in Molecular MAC is to avoid deafness because its impact on performance is much more than that of the multi-channel hidden problem.

### 5.3.5 Multi-Interface Mesh Networks

Although we have originally designed Molecular MAC for mesh routers with single network interfaces, it can also operate without any modification when nodes have multiple interfaces. A node has just to assign the role of an electron or a nucleus to each of its interfaces. Moreover, if we want to achieve the best network throughput, we need to carefully assign roles, for instance, two neighbors that are electrons for their first interface should maintain a radio link with each other for their second interface: one of them should become a nucleus. In this way, we would reduce the route stretch factor[1]: two

[1] The efficiency of a routing scheme is often measured in terms of its stretch factor, namely, the maximum ratio between the length of the path traversed by a packet and the length of the shortest path between its source and destination.

neighbors will probably be nucleus for one of their interfaces and all the radio links will be able to forward traffic.

Figure 5.6 illustrates a chain topology configured to use molecular mechanism . Each node from B and C is equipped with two interfaces: one electron and another nucleus. By assigning orthogonal channels to nuclei (interfaces), communications from A to B, B to C and C to D can be done simultaneously. Note that in this scenario there is no need to channel switching because e.g. node B has two interfaces, the nucleus one is used to communicate with A and its electron interface manages the communication with C.

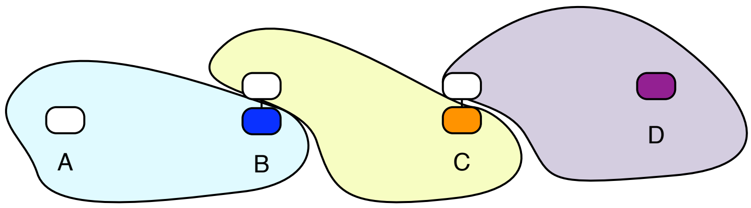



Figure 5.6: A multi-interface molecular topology: stations B and C have two interfaces, one nucleus and one electron.

## 5.4 Molecular Topology

In this section, we describe the mesh topology needed by Molecular MAC to operate and we present the mechanisms for distributed multichannel neighborhood discovery.

### 5.4.1 Scatternet and WCDS

As our structure is somehow related to the Bluetooth scatternet, which also builds on the weakly-connected dominating set (WCDS) structure, we briefly describe both of them in the following subsections.

#### WCDS

A set $S$ is dominating if each node in the graph $G = (V, E)$ is either in $S$ or adjacent to at least one of the nodes in $S$. The subgraph weakly induced by $S$ is the graph $G' = (V, E')$ such that each edge in $E'$ has at least one end point in $S$. The set $S$ is a WCDS of $G$ if $S$ is dominating and $G'$ is connected. Any node belonging to $S$ is called *dominator* and other nodes called *dominated*. Figure 5.7 shows an example of WCDS. In this graph nodes $b$, $c$, and $g$ form a WCDS and $E' = E - \{(d, e),\ (e, f)\}$. Dunbar *et al.* studied weakly-connected domination in graphs [34]. They also showed that the problem of finding a minimum size WCDS in a given graph is NP-Complete [34]. In our work the size of WCDS depends on number of nodes as well as network density.

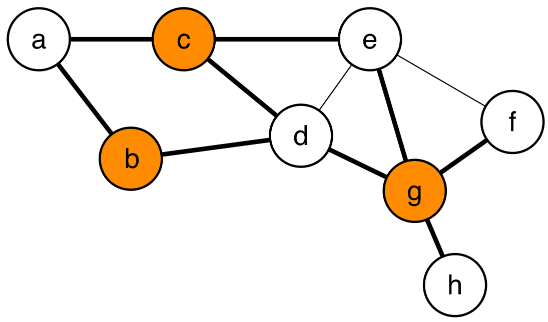



Figure 5.7: An example of Weakly Connected Dominating Set (WCDS)

#### Scatternet

A *piconet* is the type of connection that is formed between two or more Bluetooth-enabled devices such as modern cell phones or PDA's. When a piconet is formed between two or more devices, one device is dynamically elected to take the role of *master*, and all other devices assume a *slave* role for synchronization reasons. Piconets have a 3-bit address space, which limits the maximum size of a piconet to 8 devices, i.e. 1 master and 7 slaves. Inside each piconet, the medium access control is contention free and centrally regulated by a master device, which periodically polls the other devices (slaves).

A *scatternet* is a number of interconnected piconets that supports communication between more than 8 devices. Scatternets can be formed when a member of one piconet (either the master or one of the slaves) elects to participate as a slave in a second, separate piconet. The device participating in both piconets can relay data between members of both networks. Each piconet uses a different radio channel constituted by a frequency hopping code. Using this approach, it is possible to join together numerous piconets into a large scatternet, and to expand the physical size of the network beyond Bluetooth's limited range.

The way the devices are grouped in different piconets and the way the piconets are interconnected greatly affect the performance of the scatternet in terms of capacity, data transfer delay, and energy consumption. Bluetooth scatternet formation has been extensively discussed in the last few years [105, 98, 11, 30]. Basagni *et al.* [10] proposed a scatternet formation mechanism which builds on the WCDS structure.

### 5.4.2 Molecule Construction

Molecular MAC supports efficient packet forwarding over a path of alternating electron and nucleus nodes. In terms of graphs, this means that a molecule mesh network needs to form a bipartite graph. In addition to that, we can add more constraints that aim at minimizing interference and guaranteeing connectivity. Thus, the structure of a molecule mesh network must satisfy the following constraints:

1. Any electron must be a neighbor of at least one nucleus.

2. We should maximize the number of nuclei since in this way we can multiplex communications across different non interfering atoms. At the same time, we need to minimize the number of interfering nuclei because of the limited number of available channels. Otherwise, hidden and exposed terminal problems would arise during transmissions of different atoms.

3. The set of radio links between an electron and a nucleus must form a connected subset of the graph (all other links will not participate in packet forwarding).

If we construct the familiar WCDS structure [34] and remove the edges between dominator nodes, we obtain a graph that satisfies the constraints. *Dominator* nodes in a WCDS correspond to our nuclei and *dominated* nodes to electrons. The required structure is also related to the Bluetooth scatternet, which also builds on the WCDS structure [10]. Consequently, to construct a molecule mesh network, we can use an efficient algorithm that constructs a WCDS [33, 26], or adapt a Bluetooth scatternet construction algorithm. However, an issue is to choose the number of nuclei in the molecule which is the size of WCDS. A more number of nuclei enable the network to multiplex communications across different non interfering atoms, which results in increasing of overall throughput in the network. A less number of nuclei minimize the interference between neighboring atoms. Therefore, any molecule construction mechanism has to find a trade-off between number of nuclei and interference between interconnected atoms. Furthermore, it would be convenient to find a WCDS with a good route stretch factor in order to use short routes.

Molecular MAC shares several similar concepts with Bluetooth. First of all, Bluetooth uses the same kind of the WCDS structure for scatternet construction [10]. Thus, the distributed algorithms to form a Bluetooth scatternet [30, 10] can be more investigated to adapt to molecular mesh characteristics. The difference lies in the fact that in Molecular MAC a node cannot be both a nucleus and an electron while a Bluetooth node can play the role of a master and a slave at the same time. Moreover, Molecular MAC and Bluetooth both address the problem of multichannel MAC layers. However, they follow different approaches: while Bluetooth adopts a TDMA-like scheme, Molecular MAC adapts the 802.11 DCF. We consider that random access methods are more suitable for spontaneous mesh networks carrying bursty traffic. Distributed TDMA scheduling for inter-piconets communications may become a difficult task in large wireless networks even knowing the capacity of each link and with a stable topology [97]. Moreover, it requires temporal synchronization so that slots allocated by different masters do not overlap. On the contrary, Molecular MAC does not require any synchronization, is robust to traffic variations, does not limit the number of active nodes in an atom, and is entirely distributed.

### 5.4.3 Multichannel Neighborhood Discovery

Molecule construction and deciding on the roles of nodes requires neighbor discovery to learn which nodes are around, what are their roles (nucleus or electron), and which channels the neighbor nuclei use. As neighbor nodes may belong to different atoms and use different channels to transmit data, we need a multi-channel neighbor discovery protocol.

Our neighbor discovery scheme builds upon periodic broadcasting of `hello` packets: nuclei send them on their fixed channels and electrons on all channels they use. When a node wants to join a mesh molecule, it waits for a `hello` packet on a random channel. If it does not receive any `hello` packet during a predetermined time interval, it will scan all the channels by sending a `hello` packet. If a neighbor receives a `hello` packet, it replies with the required information (its channel, role, and the identity of its neighbors).

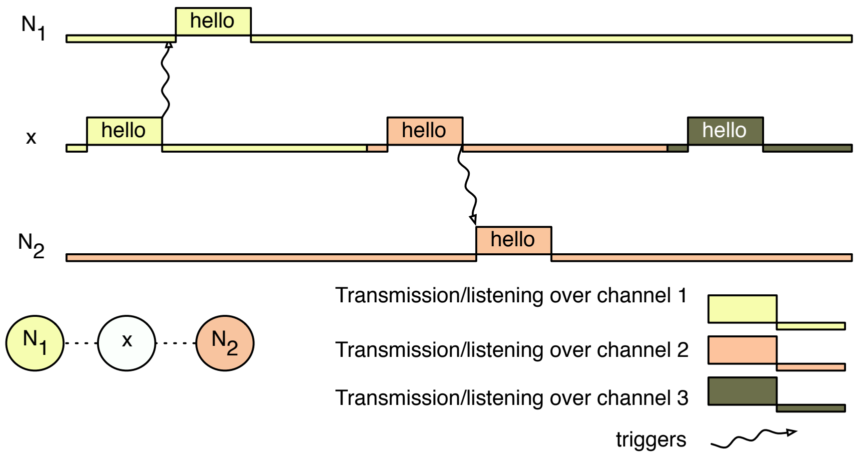



Figure 5.8: An example of new node $x$ joining a molecular mesh network

Thus, a node can achieve neighbor discovery in a single scan. Figure 5.8 shows how new node $x$ joins the mesh. It starts scanning the channels and sends the first `hello` on channel 1. Nucleus $N_1$ that listens to this channel receives this `hello` packet and replies with its own `hello` packet. Note that each `hello` packet is transmitted after a contention period according to 802.11 DCF. Then, node $x$ iteratively scans other channels. By receiving a `hello` packet, a node updates the related entry in its neighborhood table. Figure 5.9 gives an overview of the content of different neighborhood tables: a node must store the address, role (either nucleus or electron), and channel used by its neighboring nuclei if it is an electron.

A node considers a link with a given neighbor broken, if it does not receive neither data traffic nor `hello` packets.

An electron must periodically either transmit data frames to its nucleus or send one `hello` to announce its presence. Besides, an electron must discover eventual new neighbors. Thus, it will periodically scan channels for which no neighbor is registered. It can use *dead times* in its atom. For instance, if RTS/CTS is activated, a node can scan another channel when another node reserved the medium for a sufficiently long time in its atom.

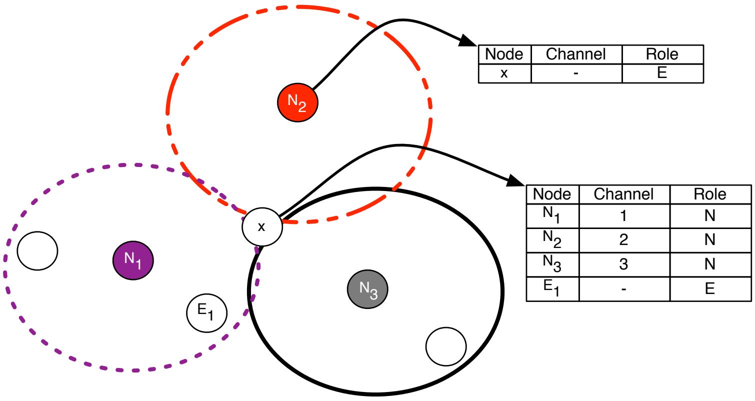



Figure 5.9: Content of the neighborhood table

### 5.4.4 Channel Assignment

When a node joins a mesh molecule, it will either become an electron of an existing atom or a nucleus of a new atom depending on the role in the WCDS structure. A nucleus needs to decide which channel to use in a given location to minimize interference with adjacent atoms. To do this, it needs to know the channels used in its neighborhood: Each electron maintains some information about channels usage in its neighborhood. It places this information in each `hello` packet. After obtaining information from all its electrons, the nucleus selects the least used channel. Finally, it informs the electrons about the chosen channel so that they can use it for communicating with the nucleus.

Obviously, we can implement in a real testbed a more sophisticated algorithm, as described in [60], or infer transmissions from the SINR measurments [85]. Channel assignment constitutes one of the future works of this thesis.

## 5.5 Performance Evaluation

We have simulated Molecular MAC in OPNET with the parameters presented in Table 5.1. OPNET is an event based network level simulation tool that operates at packet level. OPNET provides also the possibility of simulating wireless networks and specially models the physical layer (of a radio interface) using an accurate radio transmission pipeline stage.

We have compared the performance of our proposal with the standard IEEE 802.11 and MMAC (Multi-Channel MAC, which was described in Section 2.7 ) [88]. We assume that the standard IEEE 802.11 uses one channel in the whole mesh. We have chosen MMAC as one of the representative mechanisms that use channel switching. We implemented MMAC in OPNET with the parameters also presented in Table 5.1.

We have evaluated all three MAC protocols in various configurations: a line, a shared link (cross), 3 pairs (cf. Figure 5.10), and a random topology of nodes placed in a circular

| Parameter | Value |
|---|---|
| simulator | Opnet 12.0.A |
| Simulation duration | 240s |
| Physical layer | IEEE 802.11 a (OFDM) |
| Bit rate | 54Mb/s |
| Packet reception threshold | -86dBm |
| Transmit power | 5mW |
| RTS/CTS | desactivated |
| Buffer size | 256 kbits |
| Packet size | 1,500 bytes |
| $T_b$: Maximum time between two notifications | 5ms |
| $T_N$: Maximum listen time for notification | 10ms |
| Hello interval (Molecular) | 1s |
| Beacon Interval (MMAC) | 25ms |
| ATIM Window (MMAC) | 5ms |

Table 5.1: Simulations parameters for Molecular MAC and MMAC mechanisms

simulation area. The three basic topologies are well known challenging cases that we consider to gain insight into the behavior of the studied protocols. Data traffic consists of several constant-bitrate (CBR) flows, their rate being represented in the figures below as the offered load in Mb/s. For the basic configurations, Figure 5.10 presents the source and the destination of each flow. Each simulation runs for a period of 240 seconds. The results presented below are averaged over 5 different simulation runs and the 95% confidence intervals are within 1% of a given value so we have chosen not to represent them in the figures for better readability.

We have evaluated the performance of three MAC layers according to three following metrics:

1. *Packet delivery ratio*: the ratio of packets received by the destination and the total number of generated packets.

2. *End-to-end delay*: the delay between packet generation and its reception by the final destination

3. *Aggregated throughput*: the volume of all received data in the network per unit time, in Mb/s. We use this metric, because we expect our protocol to increase the total throughput of the network. Thus, this metric will directly show how molecular MAC achieves this goal.

4. *Jain index*: already defined in Chapter 2. We recall that a low Jain index means poor fairness.

To only measure the performance of variant MAC mechanisms, we use a static shortest path routing in our simulations. The Molecular MAC is independent of routing protocol.

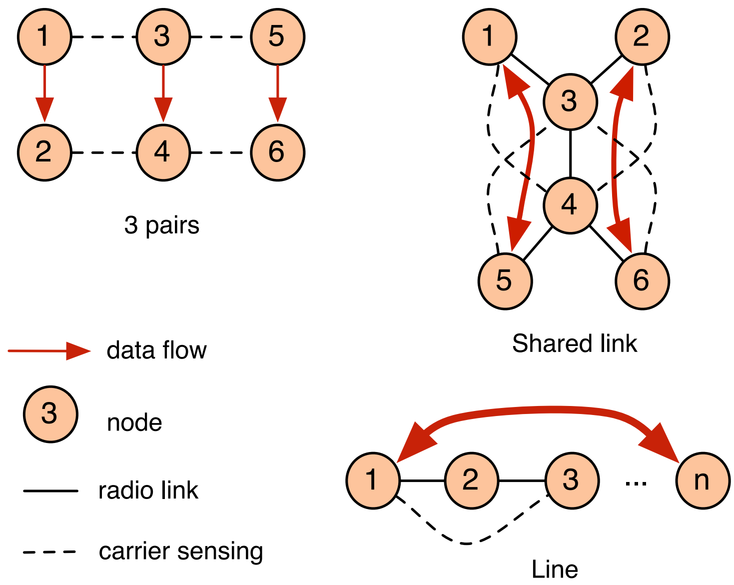


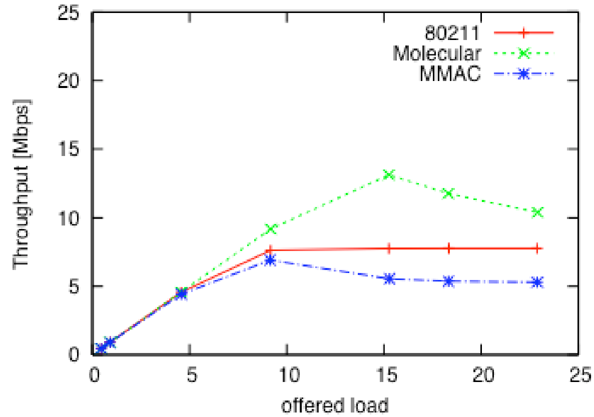

Figure 5.10: 3 pairs, shared link, and line configurations

However, its neighborhood discovery mechanism can be used by any routing protocol that needs information about neighboring nodes.

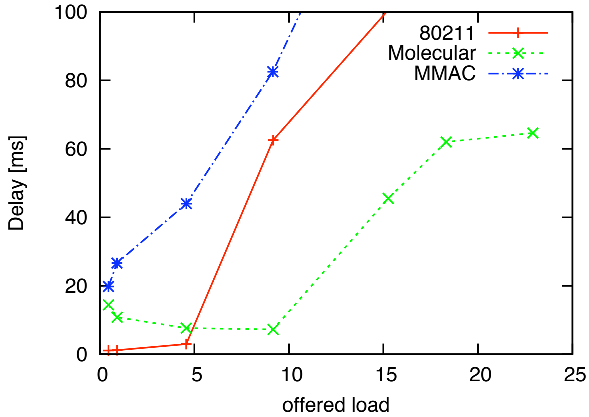
### 5.5.1 Shared Link Topology

We consider first the shared link topology (cf. Figure 5.10). Under Molecular MAC, nodes 1, 2, and 4 become nuclei and choose different orthogonal channels while the other nodes become electrons. Figure 5.11 shows the simulations results in this topology. Since all traffic passes through nodes 3 and 4, the link between them becomes a bottleneck.

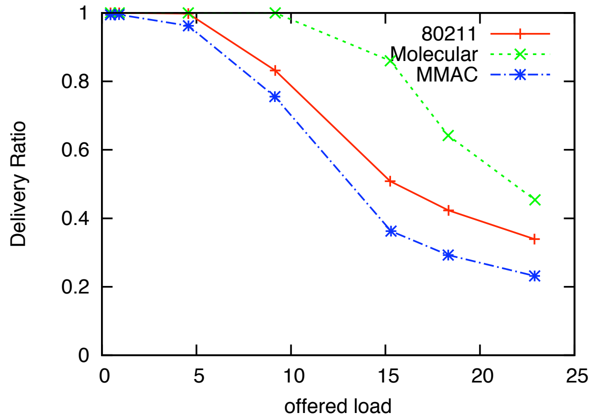
Figure 5.11(a) shows that even in saturation, MMAC obtains less throughput than 802.11 DCF. Indeed, MMAC assumes that nodes have to forward the same amount of traffic during one beacon interval although it is not the case in multihop networks (e.g. link (4,3) must forward more traffic than link (5,6)). Currently, MMAC does not consider the load on each radio link for channel reservation. Moreover, channel reservation of MMAC can become inefficient in some cases due to cascading effects: the choice of a channel implies the choices of other channels. For instance, if nodes 3 and 4 reserve a channel, all other nodes have to choose the same channel, because they have to transmit frames either to node 3 or to node 4. During such a beacon interval, MMAC would perform exactly like a single-channel IEEE 802.11. Conversely, MMAC creates islands isolated from each other during the beacon interval. For instance, it creates two different groups if radio links (3,1) and (4,5) reserve a different channel. Since the number of packets to buffer during a beacon interval can be large, buffer overflows are




(a) Aggregated throughput vs. offered load




(b) Delay vs. offered load




(c) Delivery ratio vs. offered load

Figure 5.11: Simulation results for the shared link topology with two active CBR bidirectional flows between nodes 1-5 and 2-6.

frequent, which explains the lower throughput compared to IEEE 802.11 in this topology. Accordingly, the delay with MMAC increases sharply with the load (cf. Figure 5.11(b)).

We can see from Figure 5.11 that in the shared link topology Molecular MAC outperforms both IEEE 802.11 and MMAC in terms of all indices: throughput, delay, and packet delivery ratio. This last measure decreases fairly fast under IEEE 802.11 and MMAC for increasing load while Molecular MAC still obtains good results for the load of 15 Mb/s. Note that the average end-to-end delay remains acceptable for Molecular MAC even in saturation while for two other access methods this metric increases sharply with the load (cf. Figure 5.11(b)).

### 5.5.2 Line Topology

The line topology often appears in multihop networks when nodes forward a packet along a route [22]. We have simulated a line with bidirectional traffic (two CBR flows in opposite directions). Under Molecular MAC, the line becomes a series of alternating electron and nucleus nodes.

Figure 5.12 shows the aggregate throughput with respect to the offered load in a line of seven nodes. We can observe that the Molecular MAC achieves much higher throughput than two other methods especially under high load.

When a node has less opportunity to transmit its packets, its buffer becomes full rapidly which leads to rejection of new packets. Figures 5.12(b) represents the packet delivery ratio with respect to the offered load. We can observe that the Molecular MAC is able to deliver all packets when the incoming load is not greater than 12 Mb/s while for two other methods this limit is even less than 2.5 Mb/s.

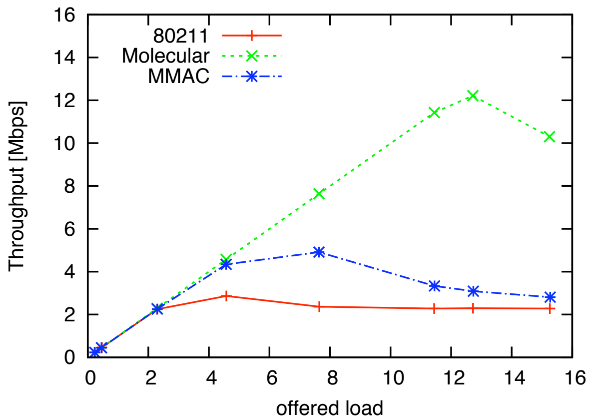
We have also studied the impact of the route length on the performance. Figure 5.12(c) presents the maximal throughput obtained by a given method for a given length of the line. We can observe that Molecular MAC is almost insensitive to the route length, from that we can conclude that our mechanism is scalable.

Conversely, IEEE 802.11 suffers from poor throughput that decreases with the number of hops, which confirms already published results [22]. MMAC is more scalable than IEEE 802.11, but also obtains lower throughput, because it benefits less from the channel diversity than Molecular MAC: channel reservations are dynamical and can create cascading effects as explained in detail above.

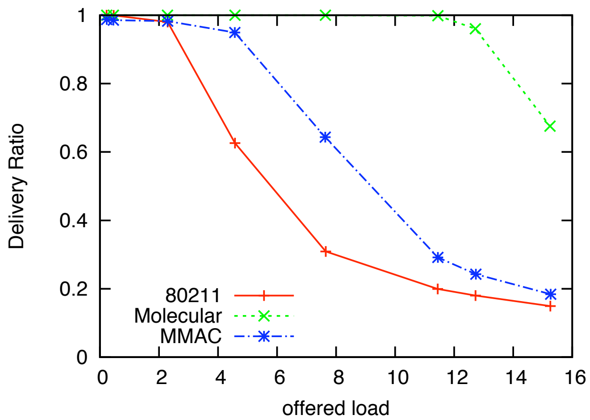
### 5.5.3 3 Pairs Topology

In this scenario, three pairs of stations communicate in parallel: $1 \rightarrow 2$, $3 \rightarrow 4$, $5 \rightarrow 6$ (cf. Figure 5.10). The middle pair senses the carrier of two other pairs, but it is too far to decode their frames. IEEE 802.11 performs very poorly in these conditions: the middle pair is starved while other pairs operate at the maximal throughput [22, 19, 71]. We have already studied the 3 pair topology in Chapter 4.

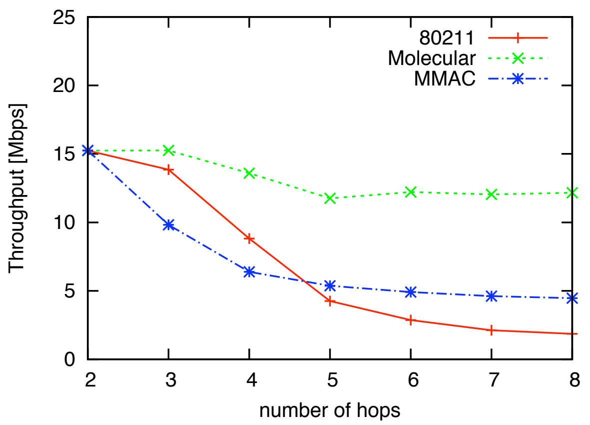
Molecular MAC solves the problem of unfairness in the 3 pairs topology and decreases the collision probability to 0. Indeed, the three pairs naturally create three different and non interfering atoms. In particular, the atom in the middle chooses a different channel




(a) Aggregate throughput vs. offered load, line of seven hops



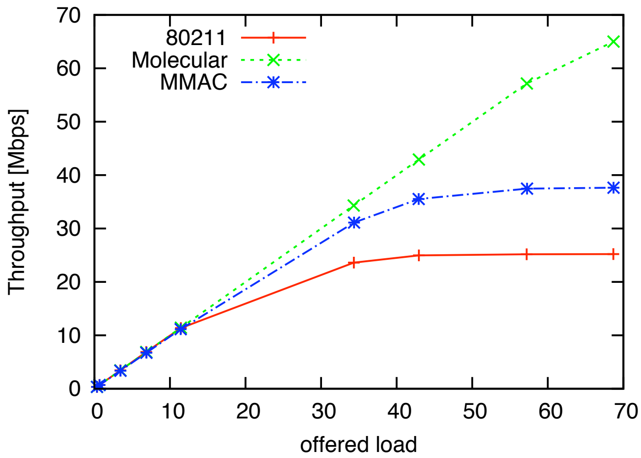

(b) Delivery ratio vs. offered load



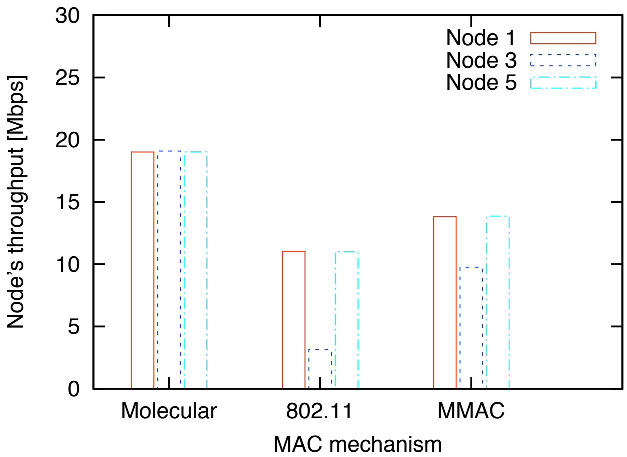

(c) Aggregate throughput vs. length of the line

Figure 5.12: Simulation results for the line topology with one active CBR bidirectional flow between nodes 1-7.

than the other pairs and the extreme pairs uses a different channel than the pair in the middle. With any channel usage measurement mechanism, channel allocation will converge to form such non interfering atoms. We used in this topology two orthogonal channels for Molecular MAC and MMAC.




(a) Aggregate throughput vs. offered load




(b) Node's throughput vs. MAC mechanism

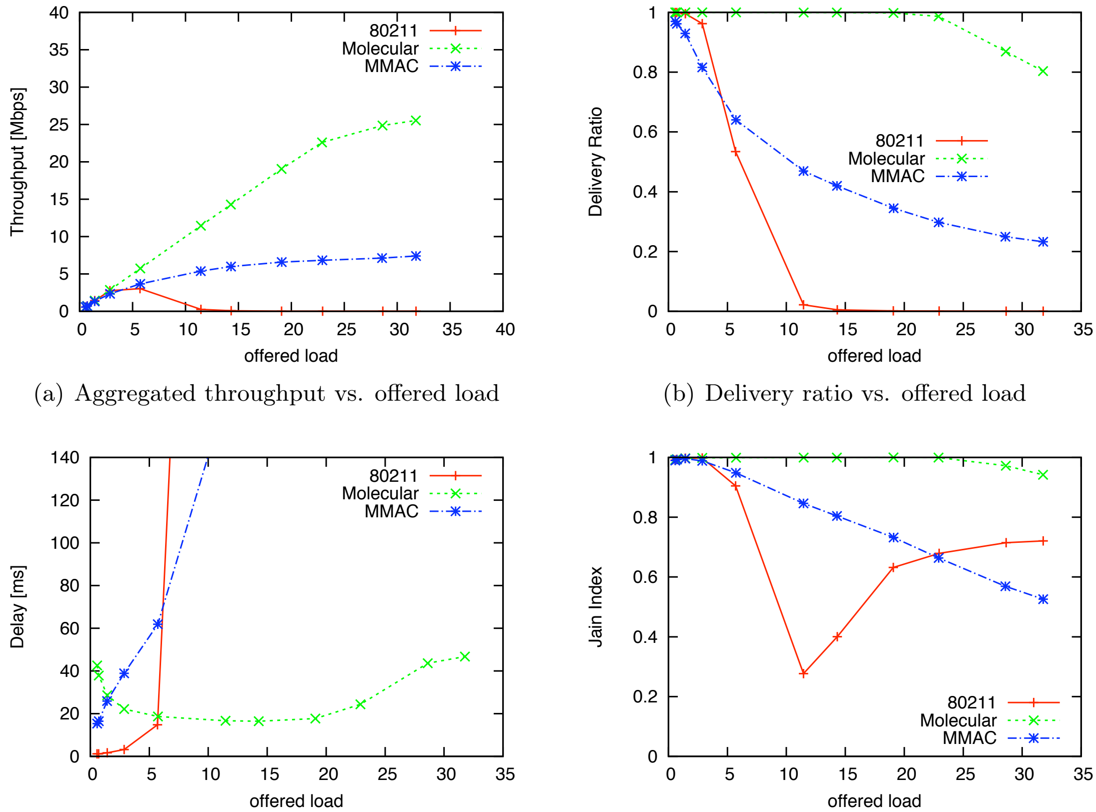
Figure 5.13: Throughput in the topology of 3 pairs.

Figure 5.13(a) shows the aggregate throughput with respect to the offered load. We observe that molecular MAC outperform MMAC in this configuration.

On the other hand, Molecular MAC provides totally fair allocation to three pairs (cf. Figure 5.13(b)) while the middle pair suffers from starvation under 802.11 and even under MMAC. The problem of MMAC for this topology is that during channel negotiation, ATIM packets cannot be decoded by interfering pairs. In other words, interfering pairs may end up choosing a common channel, which lowers throughput.

### 5.5.4 Random Topology

We have also evaluated the performance of the three access methods in more complex topologies corresponding to realistic wireless mesh networks. We have simulated a dense mesh network of 50 nodes randomly deployed in a circular simulation area of radius 270m. Traffic consists of 25 bidirectionnal CBR flows between randomly chosen pairs of source and destination.




(a) Aggregated throughput vs. offered load

(b) Delivery ratio vs. offered load

(c) Delay vs. offered load

(d) Jain index vs. offered load

Figure 5.14: Random mesh with 50 nodes uniformly distributed within a circular area with the radius of 270 m.

To organize the random topology in the molecule mesh network, we have used a geographic-oriented approach to construct a WCDS: we place nodes on a grid composed of sufficiently small squares and we elect one node as a nucleus in each square similarly to the GAF (Geographical Adaptive Fidelity) algorithm [102]. Other nodes automatically become electrons. Since our random network is sufficiently dense, there is at least one electron in each square to interconnect nuclei. Moreover, the small size of a square guarantees that an electron correctly receives packets from neighboring nuclei so that the resulting graph is connected. The molecule construction algorithm elects 12 nuclei on the average in our random topologies.

Figure 5.14 presents the performance indices in the randomly generated topology. We can observe that when load increases, IEEE 802.11 DCF performs poorly because of spatial problems (hidden and exposed nodes) and increased contention. The aggregated throughput becomes almost null when offered load is larger than 12 Mb/s. On the contrary, MMAC and Molecular MAC are much more scalable, but Molecular MAC outperforms MMAC: while MMAC begins to saturate when the load reaches 20 Mb/s with a throughput of 7 Mb/s, Molecular MAC attains 26 Mb/s in saturation. The throughput of Molecular MAC is roughly 4 times larger than under MMAC. The measures of the packet delivery ratio further show the performance advantage of Molecular MAC over MMAC (cf. Figure 5.14(b)).

Figure 5.14(c) presents the end-to-end delay, which under Molecular MAC is slightly longer than under IEEE 802.11 or MMAC for low load. This is expected because of the constant overhead experienced by electrons when they pull data frames from nuclei. However when load increases, nuclei can piggyback notifications in data frames so that the delay decreases. For increasing load, the delay under IEEE 802.11 or MMAC quickly becomes fairly long while under Molecular MAC, it is short and much more scalable with only a slight increase in function of load—it achieves packet delivery in less than 50ms independently of the load. Even in saturation, Molecular MAC offers an acceptable delay.

Finally, we have evaluated fairness measured by the Jain index (cf. Figure 5.14(d)). IEEE 802.11 presents the worst fairness for intermediate levels of load: nodes suffer from the inherent unfairness of the exponential backoff further amplified by spatial problems. The Jain index of IEEE 802.11 does increase in saturation, but it is an artifact, because only a small fraction of packets reach their destination (cf. Figure 5.14(b)). The fairness of MMAC decreases linearly with increasing load, because MMAC does not fairly distribute the bandwidth among different flows when a bottleneck appears. Thus, higher load amplifies the unfairness. Finally, we can observe that molecular MAC achieves almost a perfect fair allocation of capacity among different flows even in saturation. This nice property comes from the fact that electrons equally share their activity among neighboring nuclei.

It could appear that the route length increases when transmissions use the molecular architecture compared to the shortest path. Thus, we simulated the molecule construction and measured the route stretch factor (the ratio of the route used by molecular and the shortest route). It appears that this stretch factor is equal to 1.1 in a Unit-Disk Graph, even with a small degree: we can neglect safely this route lengthening.

## 5.6 Conclusion

In this chapter, we have presented a novel view on packet forwarding in wireless mesh networks. By adopting a molecular analogy, we can assign roles to mesh routers so that the resulting mesh is composed of independent atoms using different channels to limit interference. We define an efficient frame forwarding mechanism between adjacent atoms by extending the operation of the standard 802.11 DCF. The modifications to the standard IEEE 802.11 frame format and operation required by our mechanisms are minor: we reuse the CTS control frame for pulling frames as well as beacon frames and piggybacking onto data frames for notifying pending frames.

Our performance comparison with IEEE 802.11 and MMAC shows that Molecular MAC obtains much better throughput and packet delivery ratio, offers short end-to-end delays even in saturation, and exhibits very good fairness. Scalability is also fairly good: the overall performance remains at a high level even for an increasing number of hops or for higher loads.

# Part III

# Conclusions

Chapter 6

# Conclusion and Perspective

In this thesis, we studied the important performance problems that arise at the MAC layer in 802.11 wireless networks when they are used in the context of mesh or ad hoc networks or even in a dense environment. We proposed several mechanisms to overcome these performance issues.

In Chapter 3, we proposed a prioritized access method that supports both proportional throughput allocation and absolute priorities in 802.11 wireless LANs. It builds on the *Idle Sense* method, which provides near optimal throughput and fairness between contending stations. Idle Sense is a localized algorithm in which the stations adjust their contention window based on mean number of idle slots that they observe before each transmission. They attempt collectively to maintain this number of idle slot near a given target value that correspond to near optimal channel utilzation. We achieve absolute priority differentiation by setting different target values for the number of idle slots between transmissions. Basically, the contention window of a traffic class with a smaller target value converges rapidly to a smaller value too. At the same time, the contention window of a traffic class with a bigger target value diverges. Consequently the class with a smaller target value captures the medium. However, a very small target value leads to collisions between contending absolute priority classes. To choose a proper target value, we used analysis and simulation and we characterized what was the tradeoff between differentiation and efficiency. For example, a target value of 3 for the higher priority class, compared to 5.68 for best effort traffic gives satisfactory results for 802.11b.

In relative proportional allocation, one class benefits from a greater throughput than another one, the proportion of the throughputs being defined by a given ratio. The idea to obtain such a differentiation comes from the proportionality between the channel access probability and the contention window. Thus, in our mechanism, we periodically adjust $CW$ of relative priority classes in such a way that the desired throughput ratios are satisfied. Nevertheless, all low priority classes use the same target value of the number of idle slots as defined in *Idle Sense*. Our simulations show that the proposed method achieves its objectives of relative and absolute service differentiation both with respect to the aggregated throughput and the speed of convergence. Unlike 802.11e EDCA, it presents very good scalability—the throughput is not degraded and remains almost constant when the number of active stations increase. Also, as our mechanism is based on *Idle Sense* access method, it provides good fairness for traffic of the same priority.

The second problem that we considered is also common in the context of wireless ad hoc or mesh networks. We started from the basic ad hoc topology known as three parallel pair topology.In this scenario the central emitter, under DCF is almost always blocked while the external emitters capture the channel, which leads to dramatically unfair throughput allocation. We enhanced DCF by proposing *Forced Transmissions*, a simple and efficient solution to the problem of blocked stations. It consists of detecting that a station is blocked by others stations and forcing a transmission. The forced transmission from the central emitter results in a collision, which in turn synchronize all three emitters and then they will contend for the channel. Also, a blocked station uses $CW_{\min}$ just after a forced transmission while blocking stations double their contention windows after the collision. As a result, the blocked station is more likely to gain access to the channel. The blocked station forces a transmission with some low probability adjusted in function of the time spent waiting for the channel to become idle. We recall that Forced Transmissions can also be used in any enhancement to DCF access method such as Idle Sense and AOB. Our simulations show that the proposed method also increases the minimal throughput of the blocked stations in the configuration of $n$ parallel pairs. The price for fixing the fairness problem is an increased number of collisions and a slightly lower overall throughput.

One nice feature of our method is that the decision on reacting to bad spatial conditions is left to the blocked station. In contrast, in other propositions a station needs to periodically defer access even if there are no blocked stations nearby, which impacts channel utilization. When using *Forced Transmissions*, a station decides by itself that it is blocked and reacts independently of what other stations do. If there are no blocked stations, our proposal has no impact on the performance of the network because all stations behave as under the standard DCF access method. The proposed method is fairly general and can be deployed in any multi-hop environment (e.g. wireless sensor networks, wireless mesh networks, or wireless ad hoc networks) in which the problem of blocked stations and starvation may appear.

Due to different spatial problems such as *hidden*, *exposed*, *blocked*, and *masked* terminal problems, the overall throughput of a wireless multi-hop network can significantly degrade in certain setups. More specifically, the performance of DCF access method is almost unacceptable in wireless mesh networks because the traffic to transmit is more than in ad-hoc networks. Indeed, we observe in this case that most of the time, stations retransmit a large proportion of frames. Thus, our last objective in this thesis is to improve MAC performance in wireless mesh networks. To achieve this goal, we used the multi-channel and channel switching capabilities of 802.11 devices and proposed *Molecular MAC* mechanism in Chapter 5. In this mechanism, we adopt a molecular analogy. We consider a wireless mesh network as a *molecule* composed of interconnected atoms. In each atom, there is a nucleus able to communicate directly with all electrons belonging to the atom. To limit interference, neighboring atoms are assigned different non-overlapping channels. We also defined an efficient frame forwarding mechanism between adjacent atoms by extending the operation of the standard 802.11 DCF. The modifications to the standard IEEE 802.11 frame format and operation required by our mechanisms are minor: electrons reuse the CTS control frame for pulling their pending data frames from a nucleus. Moreover, nuclei reuse beacon frames and piggybacking onto data frames to notify electrons of their pending frames.

We also used simulation to evaluate our mechanism and to compare it with IEEE 802.11 and MMAC [88]. We evaluated the performance of each MAC in several well-known basic topologies as well as in a wireless mesh network with 50 nodes randomly placed throughout the network. Our performance comparison shows that Molecular MAC obtains much better throughput and packet delivery ratio, offers short end-to-end delays even in saturation, and exhibits very good fairness among contending flows. Scalability is also fairly good: the overall performance remains at a high level even for an increasing number of hops or for higher loads.

## 6.1 Future Work

In Forced Transmissions, there is still an open issue of a good method for deciding whether a given station is blocked. Our method based on observing the channel for a sufficient time and detecting an idle interval greater than DIFS works correctly in the studied configurations of parallel pairs of senders and receivers, however it may fail in a general scenario of a multi-hop wireless network. One direction to explore is to consider the waiting time before sending a frame—if it is too long, then the station is probably blocked.

However, for the perspectives of this thesis we mainly focus on our last contribution, Molecular MAC. We used in our simulations a centralized algorithm to assign a role (nucleus or electron) to each station. We need the definition of an efficient distributed algorithm for molecule construction. The set of nuclei in a molecular mesh forms a weakly connected dominating set. Therefore, an efficient algorithm to form a WCDS can also be adopted in a molecular mesh. An issue is the size of WCDS, i.e the number of nuclei in our molecular mesh. By maximizing the number of nuclei, we can multiplex communications across different non interfering atoms, which results in increasing the overall throughput in the network. On the other hand, the number of non-overlapping channels is limited in variants of 802.11 technologies: 3 for 802.11b/g and 12 for 802.11a. Therefore, a trade-off should be achieved between maximizing the number of nuclei and minimizing the interference between interconnected atoms. The distributed algorithms to form a Bluetooth scatternet [30, 10] can also be considered to adopt to the characteristics of the molecular mesh.

It is also possible to construct a molecule using a localized algorithm. In this way, any node elects its role itself according to the information gathered from its neighbors such as the number of neighbors and neighbor addresses. The main advantage of this mechanism is its speed of convergence, but the issue is the connectivity of the molecule, because an electron that has no nucleus in its neighborhood is disconnected from the network. We think that such a localized mechanism can be more efficient especially in a dense wireless mesh network, because the corresponding molecule is almost always connected.

Channel allocation mechanism is another research direction to follow. However, it should be integrated with the molecule construction algorithm. In fact, a freshly elected nucleus has to immediately initiate a channel assignment procedure. In addition to the neighborhood of the nucleus, the neighborhood of all electrons in the atom must also be taken into account when assigning a channel. Otherwise, any transmission from an electron to the nucleus may interfere with other transmissions in a neighboring atom. We recall also that unlike electrons, a nucleus continuously operates on the same channel and does not change its channel unless upon a major change in topology or in channel quality. Thus, electrons have to dynamically measure the utilization for all channels. An idea could be as follows: an electron measures Signal Interference Noise Ratio (SINR) for channels used in its vicinity. The electron periodically sends this information to each of the nuclei in its neighborhood. An interference estimation algorithm based on SINR measurement is proposed [60], infer the degree of interference due to different nodes within the carrier sensing range of a receiving node. A nucleus may decide to change its main channel according to the received information from its electrons. However, the channel allocation process should be stable enough to avoid cascading effects.

In order to further improve the capacity of a wireless mesh network, a joint (or cross-layer) design of medium access and routing can be investigated: a new metric based on Molecular MAC properties could be defined. This metric would be then used by the routing protocol to select the most efficient route for each pair of communicating nodes. Although the Molecular MAC is independent of a routing protocol, its neighborhood discovery mechanism can be used by any routing protocol that needs the information about neighboring nodes. Molecular MAC and the routing protocol should also interact in order to make sure that the link between the current node and any next hop in the routing table is authorized by Molecular architecture.

We think that the Optimized Link State Routing Protocol (OLSR) can be easily adapted to Molecular MAC. At each node, the OLSR protocol uses `hello` messages to discover 2-hop neighbor information and performs a distributed election of a set of multipoint relays (MPRs). Nodes select MPRs such that there exists a path to each of its 2-hop neighbors via a node selected as an MPR. It may be efficient to integrate MPR election of the routing protocol with molecule construction of the MAC layer. In this case, nuclei could also play the role of MPRs.

As mentioned before, we use simulation to evaluate our work. The main reason was the lack of an experimental testbed in our laboratory especially for wireless mesh networks. However, to better evaluate the performance of Molecular MAC and adapt it to the real world of wireless networks, it should be implemented in a real testbed.

# Part IV

# Bibliography

# Publications


[NTHD09] Mohammad Nassiri, Fabrice Theoleyre, Martin Heusse, and Andrzej Duda. Molecular MAC for Multichannel Wireless Mesh Networks. In *submitted to INFOCOM'09: The 28th IEEE Conference on Computer Communications.*, Rio de Janeiro, Brazil, April 2009.

[NTHD08] Mohammad Nassiri, Fabrice Theoleyre, Martin Heusse, and Andrzej Duda. Molecular Architecture for Spontaneous Wireless Mesh Networks. In *AD-HOC'08: 8th Scandinavian Workshop on Wireless Ad-hoc Networks*, Stockholm, Sweden, May 2008.

[NHD08] Mohammad Nassiri, Martin Heusse, and Andrzej Duda. A Novel Access Method for Supporting Absolute and Proportional Priorities in 802.11 WLANs. In *INFOCOM'08: The 27th IEEE Conference on Computer Communications.*, pages 709–717, Phoenix, AZ|USA, April 2008.

[NTHD07] Mohammad Nassiri, Fabrice Theoleyre, Martin Heusse, and Andrzej Duda. Molecular Architecture for Autonomic Wireless Mesh Networks. In *CoNEXT'07: Proceedings of the ACM CoNEXT student conference*, pages 1–2, New York, NY|USA, December 2007.

[NHD07] Mohammad Nassiri, Martin Heusse, and Andrzej Duda. Forced Transmissions for Coping with the Effect of Blocked Stations in 802.11 Wireless Networks. In *MASS'07: IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems*, pages 1–9, Pisa, Italy, October 2007.

# References

[1] I. Aad and C. Castelluccia, “Differentiation Mechanisms for IEEE 802.11.” in *Proc. IEEE INFOCOM'01*, 2001, pp. 209–218.

[2] N. Abramson, “The ALOHA System—Another Alternative for Computer Communications,” in *Proc. Fall Joint Computer Conference, AFIPS Conference*, 1970, pp. 37+.

[3] ——, “Development of the ALOHANET,” *IEEE Transactions on Information Theory*, vol. IT-31, no. 2, March 1985.

[4] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou, “A Multi-Radio Unification Protocol for IEEE 802.11 Wireless Networks,” *in Proc. BroadNets'04: First International Conference on Broadband Networks*, pp. 344–354, Oct. 2004.

[5] I. Akyildiz and X. Wang, “A Survey on Wireless Mesh Networks,” *IEEE Communications Magazine*, vol. 43, no. 9, pp. S23–S30, Sept. 2005.

[6] M. Alicherry, R. Bhatia, and L. E. Li, “Joint Channel Assignment and Routing for Throughput Optimization in Multi-Radio Wireless Mesh Networks,” in *Proc. ACM MobiCom'05*. New York, NY, USA: ACM, 2005, pp. 58–72.

[7] V. Bahl, R. Chandra, and J. Dunagan, “SSCH: Slotted Seeded Channel Hopping For Capacity Improvement in IEEE 802.11 Ad Hoc Wireless Networks,” in *Proc. ACM MOBICOM'04*. Philadelphia, USA: ACM, October 2004, pp. 216–230.

[8] A. Banchs and X. Perez, “Distributed Weighted Fair Queuing in 802.11 Wireless LAN.” in *Proc. IEEE ICC'02*, vol. 5, 2002, pp. 3121–3127.

[9] V. Barghavan, A. Demers, S. Shenker, and L. Zhang, “MACAW: a Media Access Protocol for Wireless LAN's,” in *Proc. ACM SIGCOMM*, August 1994.

[10] S. Basagni, R. Bruno, G. Mambrini, and C. Petrioli, "Comparative Performance Evaluation of Scatternet Formation Protocols for Networks of Bluetooth Devices," *Wireless Networks*, vol. 10, no. 2, pp. 197–213, March 2004.

[11] S. Basagni and C. Petrieli, "Degree-Constrained Multihop Scatternet Formation for Bluetooth Networks," in *Proc. IEEE Globecom'02*, 2002, pp. 33–47.

[12] M. Benveniste and Z. Tao, "Performance Evaluation of a Medium Access Control Protocol for IEEE 802.11s Mesh Networks," in *Proc. IEEE Sarnoff Symposium*, 2006.

[13] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535–547, 2000.

[14] G. Bianchi and I. Tinnirello, "Kalman Filter Estimation of the Number of Competing Terminals in an IEEE 802.11 network," in *Proc. IEEE INFOCOM'03*, vol. 2, March-April 2003, pp. 844–852.

[15] L. Bononi, M. Conti, and E. Gregori, "Runtime Optimization of IEEE 802.11 Wireless LANs Performance," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, no. 1, pp. 66–80, 2004.

[16] F. Cali, M. Conti, and E. Gregori, "IEEE 802.11 Wireless LAN: Capacity Analysis and Protocol Enhancement," in *Proc. IEEE INFOCOM'98*, 1998.

[17] ——, "Dynamic Tuning of the IEEE 802.11 Protocol," *ACM/IEEE Transactions on Networking*, vol. 8, no. 6, pp. 785–799, December 2000.

[18] R. Chandra and P. Bahl, "MultiNet: Connecting to Multiple IEEE 802.11 Networks using a Single Wireless Card," *in Proc. IEEE INFOCOM'04*, vol. 2, pp. 882–893, March 2004.

[19] C. Chaudet, G. Chelius, H. Meunier, and D. Simplot-Ryl, "Adaptive Probabilistic NAV to Increase Fairness in Ad Hoc 802.11 MAC," *Ad Hoc and Sensor Wireless Networks: an International Journal (AHSWN)*, vol. 2, no. 2, 2006.

[20] C. Chaudet, D. Dhoutaut, and I. Guérin-Lassous, "Experiments of Some Performance Issues with IEEE 802.11b in Ad Hoc Networks," in *Proc. WONS 2005*, St Moritz, Switzerland, 2005.

[21] C. Chaudet, I. Guérin-Lassous, E. Thierry, and B. Gaujal, "Study of the Impact of Asymmetry and Carrier Sense Mechanism in IEEE 802.11 Multi-Hops Networks Through a Basic Case," in *ACM Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN'04)*, Venice, Italy, October 2004.

[22] C. Chaudet, D. Dhoutaut, and I. Guérin-Lassous, "Performance Issues with IEEE 802.11 in Ad Hoc Networking," *IEEE Communications Magazine*, vol. 43, no. 7, pp. 110–116, July 2005.

[23] J. Chen and S.-T. Sheu, "Distributed Multichannel MAC Protocol for IEEE 802.11 Ad Hoc Wireless LANs," *Computer Communications*, vol. 28, no. 9, pp. 1000–1013, June 2005.

[24] J. Chen, S.-T. Sheu, and C.-A. Yang, "A New Multichannel Access Protocol for IEEE 802.11 Ad Hoc Wireless LANs," *in Proc. IEEE PIMRC'03*, vol. 3, pp. 2291–2296, 7-10 Sept. 2003.

[25] W.-T. Chen, T.-K. Huang, Y.-C. Chang, and J.-C. Liu, "An Adaptive Multi-channel MAC protocol for Wireless Ad Hoc Networks," in *Proc. IEEE ICC'06*. Istanbul, Turkey: IEEE, June 2006.

[26] Y. P. Chen and A. L. Liestman, "Approximating Minimum Size Weakly-Connected Dominating Sets for Clustering Mobile Ad Hoc Networks," in *Proc. ACM MOBICOM'02*. Lausanne, Switzerland: ACM, June 2002, pp. 165–172.

[27] D. Chiu and R. Jain, "Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks," *Journal of Computer Networks and ISDN*, vol. 17, no. 1, June 1989.

[28] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)," Published Online, Internet Engineering Task Force, RFC 3626, October 2003. [Online]. Available: http://rfc.net/rfc3626.txt

[29] S. Corson and J. Macker, "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations," Published Online, Internet Engineering Task Force, United States, RFC, January 1999. [Online]. Available: http://rfc.net/rfc2501.txt

[30] F. Cuomo, T. Melodia, and I. Akyildiz, "Distributed Self-Healing and Variable Topology Optimization Algorithms for QoS Provisioning in Scatternets," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 7, pp. 1220–1236, Sept. 2004.

[31] J. Deng and R. S. Chang, "A Priority Scheme for IEEE 802.11 DCF Access Method." *IEICE Transactions on Communications*, vol. E82-B, no. 1, pp. 96–102, 1999.

[32] D. Dhoutaut, "Study of the 802.11 Standard in Ad Hoc Networks: from Simulation to Experiments," Ph.D. dissertation, in French, INSA de Lyon, France, December 2003.

[33] D. Dubhashi, A. Mei, A. Panconesi, J. Radhakrishnan, and A. Srinivasan, "Fast Distributed Algorithms for (Weakly) Connected Dominating Sets and Linear-Size Skeletons," in *Symposium on Discrete Algorithms (SODA)*. Baltimore, USA: ACM–SIAM, January 2003, pp. 717–724.

[34] J. E. Dunbar, J. W. Grossman, J. H. Hattingh, S. T. Hedetniemi, and A. A. McRae, "On Weakly Connected Domination in Graphs," *Discrete Mathematics*, vol. 167-168, pp. 261–269, April 1997.

[35] J. J. Garcia-Luna-Aceves and A. Tzamaloukas, “Reversing the Collision-Avoidance Handshake in Wireless Networks,” in *Proc. ACM MOBICOM ’99*, New York, USA, 1999, pp. 120–131.

[36] A. Goldsmith, *Wireless Communications.* New York, NY, USA: Cambridge University Press, 2005.

[37] Y. Grunenberger, M. Heusse, F. Rousseau, and A. Duda, “Experience with an Implementation of the Idle Sense Wireless Access Method,” in *Proc. ACM CoNEXT 2007*, New York, 2007.

[38] P. Gupta and P. Kumar, “The Capacity of Wireless Networks,” *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388–404, Mar 2000.

[39] Z. J. Haas and J. Deng, “Dual Busy Tone Multiple Access (DBTMA)—a Multiple Access Control Scheme for Ad Hoc Networks,” *IEEE Trans. Commun.*, vol. 50, no. 6, pp. 975–985, 2002.

[40] M. Heusse, F. Rousseau, R. Guillier, and A. Duda, “Idle Sense: An Optimal Access Method for High Throughput and Fairness in Rate Diverse Wireless LANs,” in *Proc. ACM SIGCOMM’05*, vol. 35, no. 4, August 2005, pp. 121–132.

[41] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, “Performance Anomaly of 802.11b,” in *Proc. INFOCOM’03*, 30– 3 2003.

[42] G. Holland, N. Vaidya, and P. Bahl, “A Rate-Adaptive MAC Protocol for Multi-Hop Wireless Networks,” in *Proc. MobiCom ’01.* New York, NY, USA: ACM Press, 2001, pp. 236–251. [Online]. Available: http://dx.doi.org/10.1145/381677.381700

[43] C. Hu and J. C. Hou, “A Novel Approach to Contention Control in IEEE 802.11e-Operated WLANs,” in *Proc. IEEE INFOCOM’07*, May 2007.

[44] IEEE, “IEEE 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,” 1999.

[45] ——, “IEEE 802.11a: Wireless LAN MAC PHY: High-Speed Physical Layer in the 5 GHz Band,” 1999.

[46] ——, “IEEE 802.11b: Wireless LAN MAC PHY: Higher-speed Physical Layer Extension in the 2.4 GHz Band,” 1999.

[47] ——, “IEEE 802.11g: Wireless LAN MAC PHY: Further Higher Data Rate Extension in the 2.4 GHz Band,” 1999.

[48] ——, “IEEE 802.11e: Wireless LAN MAC PHY: Medium Access Control (MAC) Quality of Service Enhancements,” 2005.

[49] ——, “IEEE 802.11n Working Group: Draft 5.0,” July 2008.

[50] IEEE, "Status of Project IEEE 802.11n," July 2008. [Online]. Available: http://www.ieee802.org/11/Reports/tgn_update.htm

[51] A. Iyer and C. Rosenberg, "Understanding the Key Performance Issues with MAC Protocols for Multi-hop Wireless Networks," *Wireless Communications And Mobile Computing*, vol. 6, no. 6, Sep. 2006.

[52] R. Jain, D. Chiu, and W. Hawe, "A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems," *DEC Research Report TR-301*, September 1984.

[53] D. Johnson, Y. Hu, and D. Maltz, "The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4," Published Online, Internet Engineering Task Force, RFC 3626, October 2007. [Online]. Available: http://rfc.net/rfc4728.txt

[54] A. Kamerman and L. Monteban, "WaveLAN-II: a High-Performance Wireless LAN for the Unlicensed Band," *Bell Labs Technical Journal*, vol. 2, no. 3, pp. 118–133, 1997. [Online]. Available: http://dx.doi.org/10.1002/bltj.2069

[55] P. Karn, "MACA: A New Channel Access Method for Packet Radio," in *Proc. ARRL/CRRL Amateur Radio 9th Computer Networking Conf.*, 1990.

[56] L. Kleinrock and F. Tobagi, "Packet Switching in Radio Channels: Part I–Carrier Sense Multiple-Access Modes and Their Throughput-Delay Characteristics," *IEEE Transactions on Communications*, vol. 23, no. 12, pp. 1400–1416, Dec 1975.

[57] A. Kochut, A. Vasan, U. Shankar, and A. Agrawala, "Sniffing Out the Correct Physical Layer Capture Model in 802.11b," in *Proc. IEEE ICNP'04*, Berlin, Germany, October 2004, pp. 252–261.

[58] M. Kodialam and T. Nandagopal, "Characterizing the Capacity Region in Multi-Radio Multi-Channel Wireless Mesh Networks," in *Proc. ACM MobiCom'05*. New York, NY, USA: ACM, 2005, pp. 73–87.

[59] C. Koksal, H. Kassab, and H. Balakrishnan, "An Analysis of Short-Term Fairness in Wireless Media Access Protocols," in *Proc. ACM SIGMETRICS*, 2000.

[60] R. M. Kortebi, Y. Gourhant, and N. Agoulmine, "On the Use of SINR for Interference-aware Routing in Wireless Multi-hop Networks," in *Proc. ACM MSWIM'07*. Chania, Greece: ACM, October 2007, pp. 395–399.

[61] Y. Kwon, Y. Fang, and H. Latchman, "A Novel MAC Protocol with Fast Collision Resolution for Wireless LANs," in *Proc. INFOCOM 2003*, San Francisco, USA, March-April 2003.

[62] P. Kyasanur and N. H. Vaidya, "Routing and Interface Assignment in Multi-Channel Multi-Interface Wireless Networks." in *Proc. IEEE WCNC'05*. IEEE, March 2005.

[63] J. Li, C. Blake, D. S. J. De Couto, H. I. Lee, and R. Morris, "Capacity of Ad Hoc Wireless Networks," in *Proc. ACM MOBICOM'01*, Rome, Italy, July 2001, pp. 61–69.

[64] C.-Y. Lin, "A Multi-Channel MAC Protocol with Power Control for Multi-Hop Mobile Ad Hoc Networks," in *Proc. ICDCSW '01: the 21st International Conference on Distributed Computing Systems*. Washington, DC, USA: IEEE Computer Society, 2001, p. 419.

[65] E. Lopez-Aguilera, M. Heusse, Y. Grunenberger, F. Rousseau, A. Duda, and J. Casademont, "An Asymmetric Access Point for Solving the Unfairness Problem in WLANs ," *IEEE Transactions on Mobile Computing*, 2008.

[66] H. Ma, H. Li, P. Zhang, S. Luo, C. Yuan, and X. Li, "Dynamic Optimization of IEEE 802.11 CSMA/CA Based on the Number of Competing Stations," in *Proc. IEEE ICC'04*, Paris, France, June 2004.

[67] R. Maheshwari, H. Gupta, and S. Das, "Multichannel MAC Protocols for Wireless Networks," *in Proc. SECON '06: 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks*, vol. 2, pp. 393–401, Sept. 2006.

[68] R. M. Metcalfe and D. R. Boggs, "Ethernet: Distributed Packet Switching for Local Computer Networks," *Commun. ACM*, vol. 19, no. 7, pp. 395–404, 1976.

[69] K. Mittal and E. M. Belding, "RTSS/CTSS: Mitigation of Exposed Terminals in Static 802.11-based Mesh Networks," in *Proc. IEEE WiMesh*, September 2006.

[70] J. Mo, H.-S. W. So, and J. Walrand, "Comparison of Multichannel MAC Protocols," *IEEE Transactions on Mobile Computing*, vol. 7, no. 1, pp. 50–65, 2008.

[71] M. Nassiri, M. Heusse, and A. Duda, "Forced Transmissions for Coping with the Effect of Blocked Stations in 802.11 Wireless Networks," *in Proc. MASS'07: IEEE International Conference on Mobile Adhoc and Sensor Systems*, pp. 1–9, Oct. 2007.

[72] Q. Ni, I. Aad, C. Barakat, and T. Turletti, "Modeling and Analysis of Slow CW Decrease for IEEE 802.11 WLAN," in *Proc. IEEE PIMRC*, 2003.

[73] The Network Simulator NS-2 (v2.29). [Online]. Available: http://www.isi.edu/nsnam/ns/

[74] W. Pattara-Atikom, S. Banerjee, and P. Krishnamurthy, "Starvation Prevention and Quality of service in Wireless LANS," in *Proc. IEEE WPMC'02*, 2002.

[75] ——, "A-DRAFT: an adaptive QoS mechanism to support absolute and relative throughput in 802.11 wireless LANs." in *Proc. ACM MSWiM'04*. ACM, 2004, pp. 117–126.

[76] W. Pattara-Atikom, P. Krishnamurthy, and S. Banerjee, "Distributed Mechanisms for Quality of Service in Wireless LANs." *IEEE Wireless Communications Magazine*, June 2003.

[77] C. E. Perkins, E. M. Belding-Royer, and S. R. Das, “Ad hoc On-Demand Distance Vector (AODV) Routing,” Published Online, Internet Engineering Task Force, RFC 3561, July 2003. [Online]. Available: http://rfc.net/rfc3561.txt

[78] D. Qiao and K. Shin, “Achieving Efficient Channel Utilization and Weighted Fairness for Data Communications in IEEE 802.11 WLAN under the DCF,” in *IEEE International Workshop on QoS*, 2002, pp. 227–236.

[79] ——, “UMAV: a Simple Enhancement to the IEEE 802.11 DCF,” *in Proc. HICSS'03: the 36th Annual Hawaii International Conference on System Sciences*, pp. 9 pp.–, Jan. 2003.

[80] A. Raniwala and T. Chiueh, “Architecture and Algorithms for an IEEE 802.11-based Multi-Channel Wireless Mesh Network,” *in Proc. IEEE INFOCOM'05*, vol. 3, pp. 2223–2234, March 2005.

[81] A. Raniwala, K. Gopalan, and T. cker Chiueh, “Centralized Channel Assignment and Routing Algorithms for Multi-Channel Wireless Mesh Networks,” *ACM Mobile Computing and Communications Review*, vol. 8, pp. 50–65, 2004.

[82] S. Ray, J. B. Carruthers, and D. Starobinski, “Evaluation of the Masked Node Problem in Ad Hoc Wireless LANs,” *IEEE Trans. on Mob. Computing*, vol. 4, no. 5, pp. 430–442, 2005.

[83] S. Ray and D. Starobinski, “Adapting Physical Carrier Sensing to Maximize Spatial Reuse in 802.11 Mesh Networks,” *Wireless Communications and Mobile Computing*, vol. 8, no. 4, pp. 933–946, 2004.

[84] T. Razafindralambo and I. Guérin-Lassous, “Increasing Fairness and Efficiency using the MadMac Protocol in Ad Hoc Networks,” in *IFIP Networking*, May 2006, pp. 403–414. [Online]. Available: http://www.springerlink.com/content/w1t618qnp4388533/

[85] K. Sanzgiri, I. Chakeres, and E. Belding-Royer, “The Utility of Perceptive Communication between Distant Wireless Nodes,” in *IEEE Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TRIDENTCOM)*. Barcelona, Spain: IEEE, March 2006.

[86] N. Shacham and P. J. B. King, “Architectures and Performance of Multichannel Multihop Packet Radio Networks,” *Journal on Selected Areas in Communications*, vol. 5, no. 6, pp. 1013–1025, July 1987.

[87] M. Shreedhar and G. Varghese, “Efficient Fair Queueing Using Deficit Round Robin,” in *Proc. ACM SIGCOMM'95*, 1995, pp. 231–242.

[88] J. So and N. H. Vaidya, “Multi-Channel Mac for Ad Hoc Networks: Handling Multi-Channel Hidden Terminals Using a Single Transceiver,” in *Proc. of ACM MOBIHOC'04*. Tokyo, Japan: ACM, May 2004, pp. 222–233.

[89] N. O. Song, B. J. Kwak, J. Song, and L. E. Miller, "Enhancement of IEEE 802.11 Distributed Coordination Function with Exponential Increase Exponential Decrease Backoff Algorithm," in *Proc. VTC 2003 (Spring)*, New Orleans, 2003.

[90] F. Talucci and M. Gerla, "MACA-BI (MACA by Invitation). A Wireless MAC Protocol for High Speed Ad Hoc Networking," in *Proc. IEEE ICUPC*. San Diego, USA: IEEE, October 1997.

[91] G. Tan and J. Guttag, "Time-based Fairness Improves Performance in Multi-rate Wireless LANs," in *Proc. USENIX'04*, Boston, June 2004.

[92] Z. Tang and J. J. Garcia-Luna-Aceves, "Hop-Reservation Multiple Access (HRMA) for Ad-Hoc Networks," in *Proc. IEEE INFOCOM*. New-York, USA: IEEE, March 1999, pp. 194–201.

[93] F. Tobagi and L. Kleinrock, "Packet Switching in Radio Channels: Part II—The Hidden Terminal Problem in Carrier Sense Multiple-Access and the Busy-Tone Solution," *IEEE Transactions on Communications*, vol. 23, 1975.

[94] A. Tzamaloukas and J. J. Garcia-Luna-Aceves, "Channel-Hopping Multiple Access," in *Proc. IEEE ICC'00*, 2000, pp. 415–419.

[95] ——, "Channel Hopping Multiple Access with Packet Trains for Ad Hoc Networks," in *Proc. IEEE MoMuC '00*, Tokyo, Japan, October 2000.

[96] N. H. Vaidya, A. Dugar, S. Gupta, and P. Bahl, "Distributed Fair Scheduling in a Wireless LAN." *IEEE Trans. Mob. Comput.*, vol. 4, no. 6, pp. 616–629, 2005.

[97] W. Wang, Y. Wang, X.-Y. Li, W.-Z. Song, and O. Frieder, "Efficient Interference-Aware TDMA Link Scheduling for Static Wireless Networks," in *Proc. ACM MOBICOM'06*. Los Angeles, USA: ACM, September 2006, pp. 262–273.

[98] Z. Wang, R. Thomas, and Z. Haas, "BlueNet - a New Scatternet Formation Scheme," *in Proc. HICSS'02*, vol. 2, p. 61, Jan. 2002.

[99] S. Wiethoelter, M. Emmelmann, C. Hoene, and A. Wolisz, "TKN EDCA Model for NS2," Technische Universität Berlin, Tech. Rep. TKN-06-003, Jun. 2006.

[100] C. Wu and V. Li, "Receiver-Initiated Busy-Tone Multiple Access in Packet Radio Networks," *SIGCOMM Comput. Commun. Rev.*, vol. 17, no. 5, pp. 336–342, 1987.

[101] S.-L. Wu, C.-Y. Lin, Y.-C. Tseng, and J.-P. Sheu, "A New Multi-Channel MAC Protocol with On-Demand Channel Assignment for Multi-Hop Mobile Ad Hoc Networks," in *Proc. IEEE ISPAN*. Dallas, USA: IEEE, December 2000, pp. 232–237.

[102] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed Energy Conservation for Ad Hoc Routing," in *Proc. ACM MOBICOM'01*. Roma, Italy: ACM, July 2001, pp. 70–84.

[103] T. You, C.-H. Yeh, and H. Hassanein, "A New Class of Collision-Prevention MAC Protocols for Ad Hoc Wireless Networks," in *Proc. IEEE ICC'03*, San Francisco, USA, May 2003.

[104] M. Zannoth, T. Ruhlicke, and B.-U. Klepser, "A Highly Integrated Dual-band Multimode Wireless LAN Transceiver," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 7, pp. 1191–1195, July 2004.

[105] G. Zaruba, S. Basagni, and I. Chlamtac, "BlueTrees — Scatternet Formation to Enable Bluetooth-based Ad Hoc Networks," *in Proc. IEEE ICC'01*, vol. 1, pp. 273–277, Jun 2001.

[106] J. Zhu *et al.*, "Adapting Physical Carrier Sensing to Maximize Spatial Reuse in 802.11 Mesh Networks," *Wireless Communications and Mobile Computing*, vol. 8, no. 4, pp. 933–946, 2004.

# Part V

# Long Abstract in French