Supporting Internet Access and Quality of Service in Distributed Wireless Ad Hoc Networks

Hamidian, Ali

2009

Supporting Internet Access and Quality of Service in Distributed Wireless Ad Hoc Networks

Doctoral dissertation
Ali Hamidian

LUND UNIVERSITY

Department of Electrical and Information Technology
Faculty of Engineering

Department of Electrical and Information Technology
Lund University
P.O. Box 118
SE-221 00 Lund
Sweden
e-mail: ali.hamidian@eit.lth.se

ISSN 1654-790X
No. 17

Printed by *Tryckeriet i E-huset*, Lund, Sweden 2009

To Davoud, Susanne, Reza, Amir, Arash, and Sara

Abstract

In this era of wireless hysteria, with continuous technological advances in wireless communication and new wireless technologies becoming standardized at a fast rate, we can expect an increased interest for wireless networks, such as ad hoc and mesh networks. These networks operate in a distributed manner, independent of any centralized device. In order to realize the practical benefits of ad hoc networks, two challenges (among others) need to be considered: distributed QoS guarantees and multi-hop Internet access. In this thesis we present conceivable solutions to both of these problems.

An autonomous, stand-alone ad hoc network is useful in many cases, such as search and rescue operations and meetings where participants wish to quickly share information. However, an ad hoc network connected to the Internet is even more desirable. This is because Internet plays an important role in the daily life of many people by offering a broad range of services. In this thesis we present *AODV+*, which is our solution to achieve this network interconnection between a wireless ad hoc network and the wired Internet.

Providing QoS in distributed wireless networks is another challenging, but yet important, task mainly because there is no central device controlling the medium access. In this thesis we propose EDCA with Resource Reservation (EDCA/RR), which is a fully distributed MAC scheme that provides QoS guarantees by allowing applications with strict QoS requirements to reserve transmission time for contention-free medium access. Our scheme is compatible with existing standards and provides both parameterized and prioritized QoS. In addition, we present the Distributed Deterministic Channel Access (DDCA) scheme, which is a multi-hop extension of EDCA/RR and can be used in wireless mesh networks.

Finally, we have complemented our simulation studies with real-world ad hoc and mesh network experiments. With the experience from these experiments, we obtained a clear insight into the limitations of wireless channels. We could conclude that a wise design of the network architecture that limits the number of consecutive wireless hops may result in a wireless mesh network that is able to satisfy users' needs. Moreover, by using QoS mechanisms like EDCA/RR or DDCA we are able to provide different priorities to traffic flows and reserve resources for the most time-critical applications.

Acknowledgments

I would like to begin thanking my supervisor Professor Ulf Körner for supporting and guiding me through my research. Thank you Ulf for introducing me to the fascinating topic of wireless ad hoc and mesh networks, and giving me the opportunity to pursue my Ph.D. studies here at Lund University. I would also like to express my gratitude to Associate Professor Christian Nyberg for his pedagogical explanations to the most difficult and abstract problems. Furthermore, I owe my colleague Anders Nilsson Plymoth special thanks for sharing his experiences and for all interesting discussions we have had. I am also grateful to Jianhua Cao and Eligijus Kubilinskas for always being there answering my questions related to Linux. Of course I will not forget everyone else at the department for making the place of work such a pleasant environment - it really makes a difference! Finally, I would like to thank the most important persons in my life, namely my parents Susanne and Davoud, my brothers Reza, Amir and Arash, and my girlfriend Sara. Thank you all for your support and constant encouragement.

Ali Hamidian
Lund, Sweden
May 2009

Abbreviations and Acronyms

AC	Access Category
ACK	Acknowledgment
ACU	Admission Control Unit
ADDBA	Add Block Acknowledgment
ADDTS	Add Traffic Stream
AIFS	Arbitration Interframe Space
AIFSN	Arbitration Interframe Space Number
AODV	Ad hoc On-Demand Distance Vector
AP	Access Point
APSD	Automatic Power-Save Delivery
AR	Augmented Reality
ARP	Address Resolution Protocol
BA	Block Acknowledgment
BSD	Berkeley Software Distribution
BSS	Basic Service Set
CAP	Controlled Access Period
CBR	Constant Bit Rate
CCDF	Complementary Cumulative Distribution Function
CFP	Contention-Free Period
CP	Contention Period
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CTS	Clear To Send
CW	Contention Window
DCF	Distributed Coordination Function
DDCA	Distributed Deterministic Channel Access
DELBA	Delete Block Ack
DELTS	Delete Traffic Stream
DIFS	DCF Interframe Space
DLS	Direct-Link Setup
DS	Differentiated Services
DSCP	Differentiated Services Code Point
DSDV	Destination-Sequenced Distance Vector
DSR	Dynamic Source Routing

DSSS	Direct Sequence Spread Spectrum
DYMO	Dynamic MANET On-demand
EDCA	Enhanced Distributed Channel Access
EDCAF	Enhanced Distributed Channel Access Function
EDCA/RR	Enhanced Distributed Channel Access with Resource Reservation
EIFS	Extended Interframe Space
ESS	Extended Service Set
ETSI	European Telecommunications Standards Institute
FHSS	Frequency-Hopping Spread Spectrum
FTP	File Transfer Protocol
GWADV	Gateway Advertisement
HC	Hybrid Coordinator
HCCA	HCF Controlled Channel Access
HCF	Hybrid Coordination Function
HiperLAN/2	High Performance Radio Local Area Network type 2
HR/DSSS	High Rate Direct Sequence Spread Spectrum
HTTP	Hypertext Transfer Protocol
HUD	Head-Up Display
HWMP	Hybrid Wireless Mesh Protocol
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
IR	Infrared
ISM	Industrial, Scientific and Medical
ITS	Intelligent Transportation System
LLC	Logical Link Control
LQSR	Link-Quality Source Routing
MAC	Medium Access Control
MANET	Mobile Ad Hoc Network
MAP	Mesh Access Point
MCL	Mesh Connectivity Layer
MIMO	Multiple-Input Multiple-Output
MP	Mesh Point
MPP	Mesh Point collocated with a mesh Portal
MPR	Multipoint Relay
MSDU	MAC Service Data Unit
NAM	Network Animator
NAV	Network Allocation Vector
NHDP	Neighborhood Discovery Protocol

OFDM	Orthogonal Frequency Division Multiplexing
OLSR	Optimized Link State Routing
OLSRv2	Optimized Link State Routing version 2
OTcl	Object Tool Command Language
PC	Point Coordinator
PCF	Point Coordination Function
PHY	Physical Layer
PIFS	PCF Interframe Space
QAP	QoS Access Point
QoS	Quality of Service
RERR	Route Error
RFC	Request for Comments
RM-AODV	Radio Metric AODV
RREP	Route Reply
RREQ	Route Request
RRP	Route and Reservation Reply
RRQ	Route and Reservation Request
RTP	Real-time Transport Protocol
RTS	Request To Send
SI	Service Interval
SIFS	Short Interframe Space
SSID	Service Set Identifier
STA	Station
TBRPF	Topology Dissemination Based on Reverse-Path Forwarding
TCP	Transmission Control Protocol
TG	Task Group
TOS	Type of Service
TSPEC	Traffic Specification
TTL	Time To Live
TXOP	Transmission Opportunity
UDP	User Datagram Protocol
UP	User Priority
VANET	Vehicular Ad Hoc Network
VBR	Variable Bit Rate
VR	Virtual Reality
WAVE	Wireless Access for the Vehicular Environment
WCETT	Weighted Cumulative Expected Transmission Time
WDS	Wireless Distribution System
WEP	Wired Equivalent Privacy
WLAN	Wireless Local Area Network
WMAN	Wireless Metropolitan Area Network
WMM	Wi-Fi MultiMedia

WMM-SA	WMM Scheduled Access
WMN	Wireless Mesh Network
WNIC	Wireless Network Interface Card
WPAN	Wireless Personal Area Network
WSN	Wireless Sensor Network

Contents

Chapter 1

Introduction

The interest for Wireless Local Area Networks (WLANs) based on IEEE 802.11 [1] has been growing quickly during recent years. Today, IEEE 802.11 has become a de facto standard for WLANs. As a consequence of the increased popularity of WLANs, the interest for ad hoc networks has also increased. An ad hoc network is a wireless network composed of stations that communicate with each other directly in a peer-to-peer fashion. Thus, an ad hoc network is independent of any existing network infrastructure, such as base stations and access points. Examples of simple ad hoc networks are two mobile phones connected through Bluetooth or two laptops connected through IEEE 802.11 (operating in ad hoc mode).

Two challenges, among many others, that need to be paid attention to in order to realize the practical benefits of ad hoc networks, are providing distributed Quality of Service (QoS) guarantees and multi-hop wireless Internet access. In this thesis, both of these topics are investigated.

1.1 Thesis Outline

After a short introduction in the first chapter, Chapter 2 gives a brief overview of ad hoc networks, the IEEE 802.11 technology that supports ad hoc networking, and routing protocols for ad hoc networks. In Chapter 3, we study the interconnection between wireless and wired networks, whereas Chapter 4 investigates the provisioning of QoS guarantees in distributed wireless networks. More specifically, it proposes an enhancement to the contention-free medium access mechanism of IEEE 802.11e to provide QoS guarantees. Chapter 5 presents our experience from deploying a real mesh network testbed and, finally, in Chapter 6 we summarize the contributions of the thesis and provide general conclusions.

1.2 List of Publications

This thesis is mainly, but not exclusively, based on the following papers:

Paper I

Ali Hamidian, Ulf Körner, and Anders Nilsson, **Evaluation of Solutions for Internet Access in Mobile Ad Hoc Networks**, 17th Nordic Teletraffic Seminar (NTS 17), Oslo, Norway, Aug 2004.

Paper II

Ali Hamidian, Ulf Körner, and Anders Nilsson, **Performance of Internet Access Solutions in Mobile Ad Hoc Networks**, Lecture Notes in Computer Science (LNCS), vol. 3427, pp. 189-201, 2004.

Paper III

Ali Hamidian and Ulf Körner, **Towards a Solution Providing QoS in Ad Hoc Networks**, 19th International Teletraffic Congress (ITC 19), Beijing, China, Aug 2005.

Paper IV

Ali Hamidian, **Providing QoS Guarantees in Ad Hoc Networks through EDCA with Resource Reservation**, Technical report, CODEN: LUTEDX (TETS-7217)/1-8/(2006) & local 6, Faculty of Engineering, LTH at Lund University, 2006.

Paper V

Ali Hamidian and Ulf Körner, **An Enhancement to the IEEE 802.11e EDCA Providing QoS Guarantees**, Telecommunication Systems, vol. 31, no. 2-3, pp. 195-212, 2006.

Paper VI

Ali Hamidian and Ulf Körner, **Providing QoS in Ad Hoc Networks with Distributed Resource Reservation**, 20th International Teletraffic Congress (ITC 20), Ottawa, Canada, Jun 2007.

Paper VII

Ali Hamidian and Ulf Körner, **Extending EDCA with Distributed Resource Reservation for QoS Guarantees**, Telecommunication Systems, vol. 39, no. 3, pp. 187-194, 2008.

Paper VIII

Ali Hamidian, Claudio E. Palazzi, Tin Y. Chong, Mario Gerla, and Ulf Körner, **Deployment and Evaluation of a Wireless Mesh Network**, 2nd International Conference on Advances in Mesh Networks (MESH 2009), Athens, Greece, Jun 2009.

Paper IX

Ali Hamidian and Ulf Körner, **Distributed Reservation-based QoS in Ad Hoc Networks with Internet Access Connectivity**, submitted to the 21st International Teletraffic Congress (ITC 21), Paris, France, Sep 2009.

Paper X

Ali Hamidian, Claudio E. Palazzi, Tin Y. Chong, Mario Gerla, and Ulf Körner, **Exploring Wireless Mesh Networks for Collaborative Augmented Reality Environments**, to be submitted to International Journal of Virtual Reality (IJVR), 2009.

Paper XI

Ali Hamidian, Claudio E. Palazzi, Tin Y. Chong, Mario Gerla, and Ulf Körner, **Augmented Reality Multiplayer Games over Wireless Mesh Networks**, to be submitted to IEEE Transactions on Consumer Electronics, 2009.

Chapter 2

Ad Hoc Networks: QoS and Routing

The meaning of the Latin word “ad hoc” is “to this”, referring to something that is intended for a specific and non-continuing purpose. In other words, the term refers to dealing with special needs as they occur rather than situations that are repeated on a regular basis. When it comes to computer networks, the term refers to wireless and spontaneous computer networks without any centralized administration or established infrastructure. A simple form of an ad hoc network is composed of two wireless devices that are connected to each other directly and communicate in a peer-to-peer fashion. For example, when two mobile phones send music or pictures to each other via Bluetooth or infrared (IR) technology, they form an ad hoc network because the connection is not permanent but rather temporarily formed to meet a particular need.

In this chapter, we start by giving an overview of ad hoc networks. Next we present the IEEE 802.11 technology that supports ad hoc networking and identify its limitations when it comes to providing QoS guarantees. Then we describe a few routing protocols designed for ad hoc networks. Finally, we present the simulation tool that has been used in large parts of this thesis.

2.1 Ad Hoc Networks

Ad hoc networks require little configuration and allow for quick deployment. These features make them suitable for use in situations where an infrastructure is unavailable or to deploy one is not cost- or time-effective, such as search and rescue operations, meetings where participants wish to quickly share information, and disaster recovery where the entire communication infrastructure is destroyed and restoring communication quickly is crucial.

There is no specific requirement for ad hoc networks to support multi-hopping. Bluetooth and IR, for example, support single-hop ad hoc networks only. However, many ad hoc networks have routing capability and thus support multi-hopping. There are different types of ad hoc networks and the research on the area is mainly focused on:

- Mobile Ad Hoc Networks (MANETs)
- Vehicular Ad Hoc Networks (VANETs)
- Wireless Sensor Networks (WSNs)
- Wireless Mesh Networks (WMNs)

These can be classified according to their respective typical features: MANETs are both mobile and power-constrained, VANETs are mobile but not power-constrained, WSNs are not mobile but very power-constrained, and WMNs are neither mobile nor power-constrained. In this thesis, we focus on MANETs and WMNs.

2.1.1 Mobile Ad Hoc Networks (MANETs)

A mobile ad hoc network is a multi-hop wireless network consisting of stations that are free to move randomly and organize themselves arbitrarily; thus, the stations are usually power-constrained. These mobile stations serve as both hosts and routers so they can forward packets on behalf of each other. Hence, dynamic and adaptive routing protocols enable mobile stations to communicate beyond their transmission range by supporting multi-hop communication.

Since the stations are mobile and free to move randomly, the issue of routing in a MANET has been a challenging task for a long time. Consequently, there has been a lot of research focusing on routing protocols for MANETs. Within the Internet Engineering Task Force (IETF), there is a MANET working group with the primary goal to "standardize IP [Internet Protocol] routing protocol functionality suitable for wireless routing application within both static and dynamic topologies with increased dynamics due to node motion or other factors" [2]. Among the many proposed routing protocols, the working group chose four to go on with and published them as experimental Request for Comments (RFC). These protocols are Ad hoc On-Demand Distance Vector (AODV) [3], Dynamic Source Routing (DSR) [4], Optimized Link State Routing (OLSR) [5], and Topology Dissemination Based on Reverse-Path Forwarding (TBRPF) [6]. Two of these protocols, AODV and DSR, are reactive, whereas the other two, OLSR and TBRPF, are proactive. Based on the work and experience on these protocols, the working group is developing two standard routing protocol specifications: one reactive MANET protocol and one proactive MANET protocol. If the reactive and the proactive protocol turn out to have many parts in common, the working group may decide to continue with a converged approach. The goal is that the final protocols support both IPv4 and IPv6 and that they address routing security requirements. The work with the reactive MANET protocol has resulted in the Dynamic MANET On-demand (DYMO) [7] routing protocol, whereas Optimized Link State Routing version 2 (OLSRv2) [8] is considered as the proactive MANET protocol.

2.1.2 Wireless Mesh Networks (WMNs)

A wireless mesh network is a multi-hop wireless network consisting of static wireless devices that are usually not power-constrained. WMNs are the next step in the evolution of a wireless architecture, delivering services for a large variety of applications in Wireless Personal Area Networks (WPANs), Wireless Local Area Networks (WLANs), and Wireless Metropolitan Area Networks (WMANs). Currently, there are three working groups within the IEEE 802 project that are working on WMNs. The IEEE 802.15 working group is working with a WPAN mesh standard (IEEE 802.15.5), the IEEE 802.11 working group is working with a WLAN mesh standard (IEEE 802.11s), and IEEE 802.16 is working with WMAN mesh standard (IEEE 802.16j). In this thesis, we focus on IEEE 802.11-based mesh networks.

Unlike WLANs, mesh networks are self-configuring systems where each Access Point (AP) may relay messages on behalf of others and thus increase the communication range. Moreover, in regular WLANs, the (wireless) AP has to be wired to the infrastructure; this is a paradox overcome by WMNs, where APs can be connected to the rest of the network by wireless radio links. Other key advantages of WMNs include ease of installation, no cable cost, automatic connection among nodes, network flexibility, discovery of newly added nodes, redundancy, self-healing, and reliability. Due to these attractive characteristics, WMNs have gained the interest of researchers all around the world.

Although ad hoc networks are said to be infrastructure-less, a WMN can be viewed as a *wireless and distributed infrastructure*. Thus, the architecture of a WMN makes it a hybrid wireless network between an infrastructure WLAN and an ad hoc network. In essence, a WMN is able to extend the coverage of the infrastructure network by multi-hop wireless connections between APs. The APs in a WMN can hence be detached from any wired infrastructure while being connected to each other through wireless links. Essentially, we can view a mesh network as a packet-switched, multi-hop ad hoc network composed of mesh-enabled wireless devices that form a wireless communication infrastructure.

2.2 QoS in IEEE 802.11 Ad Hoc Networks

From a layered point of view, the QoS issue can be treated in different layers of the protocol stack. However, QoS provisioning at the Medium Access Control (MAC) sublayer[1] is a necessary (but sometimes not sufficient) condition. In other words, QoS provisioning in IEEE 802.11-based ad hoc networks is not possible unless supported by the MAC protocol. For example, no matter what QoS approach is used at higher layers, one cannot guarantee, e.g., delay, jitter, throughput, and packet loss because the IEEE 802.11 MAC protocol gives an unpredictable random waiting time before accessing the medium.

When talking about QoS guarantees, we must keep in mind that since a wireless medium is much more unpredictable and error-prone than a wired medium, QoS cannot be guaranteed as in a wired system, especially in unlicensed spectra. However, it is possible to provide techniques that increase the probability that certain traffic classes get adequate QoS and that can provide QoS guarantees in controlled environments. This is also formulated in the IEEE 802.11e standard amendment (Section 5.1.1.2 - Media impact on design and performance) [9]:

> *When providing QoS services it should be understood that the MAC endeavors to provide QoS "service guarantees" within the limitations of the medium properties identified above. That is, particularly in unlicensed spectrum, true guarantees are often not possible. However gradations of service are always possible, and in sufficiently controlled environments, QoS guarantees can truly be made.*

This section gives an overview of the IEEE 802.11 standard. In particular, it describes the extensions and enhancements to the IEEE 802.11 standard with focus on IEEE 802.11e, which aims at providing QoS.

2.2.1 IEEE 802.11 and its QoS Limitations

The Wi-Fi Alliance is an industry group composed of leading WLAN manufacturers. The goal of the group is to drive the adoption of a single worldwide-accepted standard for WLANs. The group has certification programs to ensure compatibility between different Wi-Fi devices. The certification programs are based on subsets of the IEEE 802.11 standard.

The IEEE 802.11 standard, which covers both the Physical Layer (PHY) and the MAC sublayer, specifies two network configuration modes: *infrastructure* and *ad hoc*. Stations can be configured to operate in either of

[1] The data link layer is composed of two sublayers: Logical Link Control (LLC) and MAC.

these modes, but not in both at the same time. Using the infrastructure network configuration, all unicast transmissions between two stations must pass through an AP that relays them to the destination. The AP can also be used by the stations to access the Internet. Using the ad hoc network configuration, any station can communicate to another directly without the need of any AP.

2.2.1.1 IEEE 802.11 PHY

The first version of the IEEE 802.11 standard was released in 1997. It specified three physical layer options: IR, Frequency-Hopping Spread Spectrum (FHSS), and Direct Sequence Spread Spectrum (DSSS). Whereas FHSS and DSSS operate at the Industrial, Scientific and Medical (ISM) band at 2.4 GHz, IR uses near-visible light in the 850-950 nm range for signaling. Since all three PHY options offered low data rates of up to 2 Mbps, none of them became widely used. The breakthrough came in 1999 with IEEE 802.11b specifying the High Rate DSSS (HR/DSSS), with a maximum data rate of 11 Mbps. The same year, IEEE ratified IEEE 802.11a, which is based on Orthogonal Frequency Division Multiplexing (OFDM) and allows for data rates theoretically up to 54 Mbps. However, despite the higher throughput, IEEE 802.11a did not become widely accepted as opposed to IEEE 802.11b. One important reason for this is that IEEE 802.11a operates in the 5 GHz band, implying that it is not backward compatible with the original standard. Another reason is that IEEE 802.11a has a shorter transmission range; also a consequence of the operation in the 5 GHz band. On the contrary, IEEE 802.11b operates in the ISM band at 2.4 GHz implying longer transmission range compared to IEEE 802.11a and backward compatibility with the original standard. Later in 2003, IEEE 802.11g was ratified. This version, which rapidly became the most popular standard, uses both DSSS and OFDM, operates at the 2.4 GHz ISM band, is backward compatible with IEEE 802.11b and allows for data rates theoretically up to 54 Mbps.

Currently, Task Group n (TGn) is working on IEEE 802.11n that is expected to become the next-generation standard for WLANs. The task group is studying various enhancements to the PHY and the MAC sublayer; the goal is to support data rates of at least 100 Mbps, measured at the interface between the MAC sublayer and higher layers. The motivation to measure at a higher layer than at the physical interface to the wireless medium (where IEEE 802.11/a/b/g measure the data rate), is to better match the data rates that a user experiences. In addition to higher data rates, IEEE 802.11n addresses longer transmission range at existing data rates and increased resistance to interference. One way to achieve these improvements is to use the Multiple-Input Multiple-Output (MIMO) technology, which uses multiple transmitter and receiver antennas to allow for higher data rates through spatial multiplexing and longer range by exploiting the spatial diversity. In order to improve the transfer efficiency, the MAC sublayer can be enhanced by aggregating multiple frames into a single PHY unit instead of initiating a new transfer for every frame. The IEEE 802.11n amendment is expected to be ready January 2010 as predicted by the official IEEE 802.11 working group project timelines.

2.2.1.2 IEEE 802.11 MAC

Although there has been several enhancements to the physical layer since the first version of the standard was released (more specifically, these enhancements are IEEE 802.11a/b/g focusing on higher data rates), the medium access mechanism in the MAC sublayer was not changed until 2005 (with IEEE 802.11e that specifies a new coordination function). In other words, IEEE 802.11/a/b/g used the same protocol in the MAC sublayer. The original IEEE 802.11 standard has defined two medium access mechanisms: the mandatory Distributed Coordination Function (DCF) and the optional Point Coordination Function (PCF). DCF is the basis for PCF and is used for best effort contention services. Since DCF is a distributed access method, it can be used not only in infrastructure network configurations, but also in ad hoc network configurations. PCF, on the other hand, is required for contention-free services and only usable in infrastructure network configurations.

Distributed Coordination Function (DCF) To control the waiting time before medium access, DCF and PCF use four parameters called interframe spaces (illustrated in Figure 2.1): Short Interframe Space (SIFS), PCF Interframe Space (PIFS), DCF Interframe Space (DIFS), and Extended Interframe Space (EIFS). SIFS is the shortest waiting time and thus used to give the highest priority for medium access. It is used before sending short control frames, such as Clear To Send (CTS) and Acknowledgment (ACK) frames. PIFS is a waiting time longer than SIFS but shorter than DIFS, resulting in medium priority. It is used only by stations operating under PCF, e.g., by the AP polling other stations. DIFS is a waiting time longer than both SIFS and PIFS and gives therefore the lowest priority for medium access. It is used only by stations operating under DCF, transmitting data or management frames. EIFS is the longest waiting time used by stations operating under DCF, but only when a transmission failure occurs. A station that receives an incorrect frame must wait for EIFS before starting its transmission in order to give other stations enough time to acknowledge the frame that the station received incorrectly.

The Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)

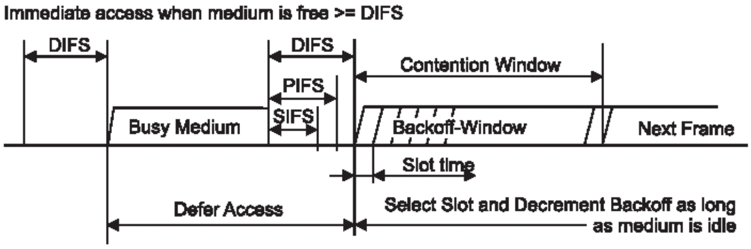

Immediate access when medium is free >= DIFS
DIFS
DIFS
PIFS
SIFS
Contention Window
Busy Medium
Backoff-Window
Next Frame
Slot time
Defer Access
Select Slot and Decrement Backoff as long as medium is idle

Figure 2.1: The interframe space relationships (copied from [10]).

mechanism is used by DCF to regulate access to the shared medium. Whenever a station desires to transmit a frame, it must invoke the carrier sense mechanism to determine whether the medium is busy or idle. There are two carrier sense mechanisms: one physical provided by the PHY and one virtual, referred to as the Network Allocation Vector (NAV), provided by the MAC sublayer. If both of these functions indicate an idle medium, the medium is considered idle; otherwise, the medium is considered busy. If the medium has been sensed to be idle for the duration of at least a DIFS period, the station can initiate a transmission immediately. Otherwise, if the medium is sensed to be busy or if it is sensed to be idle but becomes busy before the duration of a DIFS period, the station must defer until the end of the ongoing transmission. Then, the station must wait until the medium is determined to be idle without interruption for the duration of a DIFS period. Finally, it must invoke the random backoff process, which is an additional random waiting time necessary to reduce the probability of collisions. The random backoff is necessary because once the medium becomes idle following a busy medium, there is a high probability of collision since several stations may be waiting for the medium to become idle. The random backoff time is calculated as follows:

$$\text{backoff time} = \text{random}() \times \text{aSlotTime},$$

where random() is a uniformly distributed pseudo-random integer between zero and Contention Window (CW) and aSlotTime is a PHY-dependent value. The value of CW varies between CWmin and CWmax, which are PHY-dependent, but initially the CW is set equal to CWmin. The backoff time can be seen as a waiting time before accessing the medium and equals to a random number of time slots. If the medium is sensed to be idle for the duration of one complete time slot, the backoff time is decremented with one aSlotTime. If the medium becomes busy in the middle of a time slot, the backoff time becomes suspended until the medium is sensed to be idle for the duration of a DIFS period. Then the backoff procedure resumes and starts decrementing the backoff time again. Once the backoff timer reaches zero, the station can start transmitting. When the destination receives the data frame, it waits for the duration of a SIFS period and responds with an ACK frame to notify the sender of a successful reception. If two or more stations choose the same random number, their backoff timers will reach zero simultaneously resulting in a collision since both begin transmitting. To reduce the probability of choosing the same random number, the CW is doubled every time a collision occurs, from its initial value of CWmin until CWmax is reached. Thus, the CW increases exponentially.

DCF suffers from the well-known hidden station problem that exists among CSMA/CA-based protocols. Two stations are hidden from each other if they are out of signal range and thus cannot hear each other. In such a case, the carrier sense mechanisms will not work properly and the stations may both sense the medium idle and start transmitting at the same time to a common receiver causing a collision. To deal with the hidden station problem, an optional handshake mechanism with Request To Send (RTS) and CTS control frames is used to announce the neighborhood of the impending use of the medium. Before sending a data or a management frame, a station can transmit an RTS frame and await a CTS frame. The control frames contain a duration field that is used to tell the neighbors about the duration of the impending data transmission. When neighbors receive the RTS or CTS frames, they update their NAVs so that they consider the medium busy until the end of the transmission. Thus, collisions occur only on RTS frames and are detected by the absence of a CTS frame. Because of the additional overhead imposed by the handshake mechanism, it is not recommended to be used for short data frames. In other words, the RTS/CTS mechanism is used only if the length of the data or management frames is greater than a threshold and only for unicast frames. A typical frame exchange sequence (if the RTS/CTS mechanism is used) in DCF is RTS-{SIFS}-CTS-{SIFS}-data-{SIFS}-ACK.

Point Coordination Function (PCF) PCF uses polling to regulate access to the shared medium. Instead of contending for access to the medium, the stations are polled by a Point Coordinator (PC). The PC performs the role of the polling master and operates at an AP, which explains why the operation of PCF is restricted to infrastructure networks. When PCF is used, the time is divided into a Contention Period (CP), when DCF manages access to the medium, and a Contention-Free Period (CFP), when PCF controls the medium access. During a CFP, the PC maintains a polling list of registered stations and polls them according to the list. A station is allowed to start a transmission only after it has been polled. At the nominal start of a CFP, the PC starts sensing the medium. Once the medium is determined to be idle for the duration of a PIFS period, the PC transmits a Beacon frame. Thus, a CFP begins with a Beacon frame and is generated by the PC at a defined rate, called the CFP Repetition Interval (CFPPeriod). At the nominal start time of each CFP, stations set their NAV to CFPMaxDuration, which is a parameter that indicates the maximum duration of the CFP. This action prevents stations to contend for access to the medium and thus initiate a transmission, unless they are polled by the PC. The actual duration of a CFP is controlled by the PC, which can terminate a CFP at or before the CFPMaxDuration, based on available traffic and size of the polling list. During the CFP, the stations update their NAV using the CFPDurRemaining parameter.

To give the PC a higher priority to access the medium, it waits for the duration of a PIFS period before accessing the medium, whereas the stations must wait for the duration of a DIFS period, which is longer. At the end of a CFP, the PC transmits a CF-End or CF-End+ACK frame, which will cause stations receiving the frame resetting their NAV and start contending for access to the medium.

Regarding QoS provisioning, unfortunately both PCF and DCF have their limitations. DCF can only provide a best-effort service and all stations contend for access to the medium with the same priority. Thus, DCF does not provide any differentiation mechanism to give application with QoS requirements better service than other applications. Although PCF was designed to support time-sensitive applications, it has a few problems that lead to poor QoS performance. Due to this fact, together with the fact that it is an optional and centralized access mechanism, PCF never became commonly implemented. The main problems with PCF are the following:

- A Beacon frame transmission, which indicates the start of a CFP, may be delayed because of a possible transmission in progress from the CP. This will result in a shortened CFP and less time for stations with QoS applications requiring contention-free access to the medium.

- PCF cannot handle the various QoS requirements of different types of applications because there is no way for stations to send their requirements to the PC. Furthermore, the scheduling algorithm used by the PC is rather simple, polling stations one after another.

- When a station is polled, it may send a frame between 0-2304 bytes. Therefore, the PC is not able to predict the transmission time of the polled stations; thus, it cannot provide any delay guarantees.

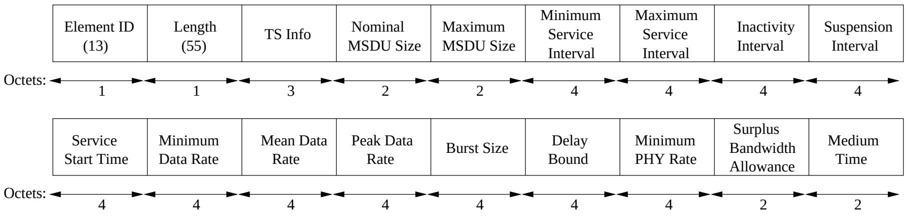
Another QoS problem, which is common for both DCF and PCF, is that there is no admission control mechanism to regulate the usage of the medium. An admission control mechanism is necessary to prevent performance degradation of existing traffic streams when the network becomes overloaded.

2.2.2 IEEE 802.11e QoS Provisioning

Due to the QoS limitations of DCF and PCF, there has been a lot of research focusing on the enhancement of the MAC sublayer of IEEE 802.11 to provide QoS. To support multimedia applications with QoS requirements, the IEEE 802.11 working group started TGe to address the QoS issues in the MAC sublayer. The work of the task group resulted in the IEEE 802.11e standard amendment, which was completed in 2005. The new standard specifies a new coordination function, the Hybrid Coordination Function (HCF), which has both a contention-based and a contention-free medium access method. The Enhanced Distributed Channel Access (EDCA) mechanism is contention-based and provides prioritized QoS support, whereas the HCF Controlled Channel Access (HCCA) mechanism is contention-free and provides support for parameterized QoS. Whereas EDCA is distributed (like DCF) and can be used in ad hoc networks, HCCA is centralized (like PCF) and thus useful only in infrastructure networks. HCF has introduced two new important features to solve the QoS limitations of PCF:

- **Transmission Opportunity (TXOP)**: A TXOP is a time interval defined by a starting time and a maximum duration. During a TXOP, a station may send several frames as long as the duration of the transmissions does not extend beyond the maximum duration. Since no transmission can violate the TXOP limit, frames that are too large to be transmitted in a single TXOP, must be fragmented into smaller frames. Using ad hoc network configuration, broadcast and multicast frames are not allowed to be sent more than one at a time. These bounded time intervals were introduced to solve the problem with unknown transmission times of polled stations in PCF. Furthermore, the problem of shortened CFPs is solved by requiring TXOPs not to extend across the time for a Beacon frame transmission.

- **Traffic Specification (TSPEC)**: A TSPEC describes the QoS requirements of a traffic stream by specifying a set of parameters, such as nominal/maximum MAC Service Data Unit (MSDU) size, minimum/maximum Service Interval (SI), minimum/mean/peak data rate, service start time, burst size, delay bound, minimum PHY rate, and medium time (see Figure 2.2). Most of the above-mentioned parameters are typically set according to the requirements from the application, whereas some are generated locally within the MAC. The parameter minimum/maximum SI specifies the minimum/maximum time interval between the start of two consecutive TXOPs and service start time specifies the time when the service period starts, i.e., when the station expects to be ready to send frames. The burst size parameter specifies the maximum size of the data burst that can be transmitted



	Element ID (13)	Length (55)	TS Info	Nominal MSDU Size	Maximum MSDU Size	Minimum Service Interval	Maximum Service Interval	Inactivity Interval	Suspension Interval
Octets:	1	1	3	2	2	4	4	4	4

	Service Start Time	Minimum Data Rate	Mean Data Rate	Peak Data Rate	Burst Size	Delay Bound	Minimum PHY Rate	Surplus Bandwidth Allowance	Medium Time
Octets:	4	4	4	4	4	4	4	2	2

Figure 2.2: The TSPEC element format.

at the peak data rate. The TSPEC was introduced to solve the problem with the inability for a station to send its QoS requirements to an AP.

In addition to the QoS enhancements, the IEEE 802.11e specification has defined the following optional features to improve the MAC performance: Automatic Power-Save Delivery (APSD), Block Acknowledgment (BA), and Direct-Link Setup (DLS). The APSD is an enhancement to the existing power save mechanism in IEEE 802.11 and used for delivery of downlink unicast frames to power-saving stations.

The BA mechanism allows a station to aggregate several (up to 64) ACK frames into one, instead of sending one ACK after each successfully received data frame. Once the BA mechanism is initialized by an exchange of Add Block Acknowledgment (ADDBA) Request/Response frames, blocks of data frames can be transmitted. When the sender needs an ACK, it sends a Block Ack Request (BlockAckReq) control frame to the receiver, which replies with a Block Ack (BlockAck) control frame acknowledging the successfully received data frames. There are two types of BA mechanisms: immediate BA and delayed BA. If the immediate BA mechanism is used, the BlockAck frame must be sent immediately after a BlockAckReq frame is received. However, if the delayed BA mechanism is used, an ACK frame is used to respond to a BlockAckReq frame and the BlockAck frame can be delayed and sent somewhat later. The delayed BA option is intended to be used by stations with low processing power, i.e., to give these stations enough time to calculate and prepare the content of the BlockAck frame. A BA setup can be torn down, e.g., when there are no more data frames to be sent, by sending a Delete Block Ack (DELBA) frame.

The DLS mechanism allows stations operating in infrastructure mode to transmit frames directly to each other (the same way that stations operating in ad hoc mode communicate) without relying on the AP to forward the frames. DLS requires a handshake process where the station intending to initiate a direct link to another station sends a DLS Request Action frame to the QoS Access Point (QAP). The QAP relays the request to the other station, which responds with a DLS Response. The QAP relays the response back to the station that requested the DLS and finally, in case of successful negotiation, the two stations can communicate with each other directly.

2.2.2.1 Enhanced Distributed Channel Access (EDCA)

EDCA is a distributed, contention-based medium access mechanism and an enhanced variant of DCF. The main problem with DCF, regarding QoS provisioning, is that it cannot provide any service differentiation since all stations have the same priority, i.e., the same CWmin, CWmax and waiting time before backoff or transmission (equal to DIFS). In addition, DCF uses one single transmit queue and one channel access function. To overcome these problems, each station using the EDCA mechanism has four Access Categories (ACs); for each of these there is one transmit queue with an Enhanced Distributed Channel Access Function (EDCAF) that contends for TXOPs independently of the EDCAFs of the other ACs. Thus, each AC behaves like an enhanced and independent DCF contending for medium access.

In order to prioritize a traffic stream, the Differentiated Services Code Point (DSCP) value must be modified by the application. This value is contained in the six least significant bits of the Differentiated Services (DS) field, which supersedes the Type of Service (TOS) field in the IPv4 header and the Traffic Class field in the IPv6 header. Once the DSCP value is set in the IP header, the packet is assigned a User Priority (UP) and based on these UPs each frame is mapped to an AC according to Table 2.1. Besides using the UPs, frames can be mapped to ACs based on frame types. The management type frames, for example, shall be sent from AC_VO (without being restricted by any admission control though) and RTS frames shall use the same AC as the corresponding data or management frame(s). The four ACs can be used for different types of traffic: AC_BK for background traffic, AC_BE for best effort traffic, AC_VI for video traffic and AC_VO for voice traffic. Differentiated medium access is realized by varying the contention parameters for each AC:

- CWmin[AC] and CWmax[AC] - the minimum and maximum value of the CW used for calculation of the backoff time. These values are variable and no longer fixed per PHY as with DCF. By assigning low values to CWmin[AC] and CWmax[AC], an AC is given a higher priority.

- Arbitration Interframe Space Number (AIFSN[AC]) - the number of time slots after a SIFS duration that a station has to defer before either invoking a backoff or starting a transmission. AIFSN[AC] affects the Arbitration Interframe Space (AIFS[AC]), which specifies the duration

Table 2.1: Mapping from UPs to ACs.

Priority	**User Priority (same as in 802.1D)**	**Access Category**	**Designation**	**TOS**	**DSCP**
lowest	1	AC_BK	Background	32	8
	2	AC_BK	Background	64	16
	0	AC_BE	Best Effort	0	0
	3	AC_BE	Best Effort	96	24
	4	AC_VI	Video	128	32
	5	AC_VI	Video	160	40
	6	AC_VO	Voice	192	48
highest	7	AC_VO	Voice	224	56

(in time instead of number of time slots) a station must defer before backoff or transmission:

$$AIFS[AC] \quad = \quad SIFS + AIFSN[AC] \times aSlotTime$$

Thus, by assigning a low value to AIFSN[AC], an AC is given a high priority.

- TXOP_limit[AC] - the maximum duration of a TXOP. A value higher than zero means that an AC may transmit multiple frames (if all belong to the same AC since a TXOP is given to an EDCAF in a specific AC and not to a station) as long as the duration of the transmissions does not extend beyond the TXOP_limit[AC]. A TXOP_limit[AC] value equal to zero indicates that only one data or management frame (plus any corresponding RTS/CTS frames) may be sent. Thus, by assigning a high value to the TXOP_limit[AC], an AC is given a high priority.

Table 2.2a and Table 2.2b show the default values for the contention parameters of each AC. The values of these parameters can be changed by the QAP announcing the new values in selected Beacon frames, any Probe Response or (Re)Association Response frames. In order to prevent stations to interfere with the operation of APs, it is important to have AIFSN[AC] $\geq$ 2 for stations, resulting in AIFS[AC] $\geq$ DIFS. If the backoff timer of two or more ACs in a single station counts down to zero at the same time, an internal collision occurs. These virtual collisions are resolved such that the high-priority AC is given access to the medium, whereas the other AC(s) act as if there was an external collision on the wireless medium.

Table 2.2: The default EDCA parameter set.

(a) IEEE 802.11b PHY

AC	CWmin	CWmax	AIFSN	TXOP limit
AC_BK	31	1023	7	0
AC_BE	31	1023	3	0
AC_VI	15	31	2	6.016 ms
AC_VO	7	15	2	3.264 ms

(b) IEEE 802.11a/802.11g PHY

AC	CWmin	CWmax	AIFSN	TXOP limit
AC_BK	15	1023	7	0
AC_BE	15	1023	3	0
AC_VI	7	15	2	3.008 ms
AC_VO	3	7	2	1.504 ms

2.2.2.2 Hybrid Controlled Channel Access (HCCA)

HCCA is the centralized, contention-free medium access mechanism of HCF and uses a Hybrid Coordinator (HC), collocated with the QAP, to manage access to the medium. Whenever the medium is sensed to be idle for at least PIFS, the QAP may take control over the medium and start a Controlled Access Period (CAP). A CAP is a time period during which a QAP maintains control of the medium to allocate TXOPs to itself or other stations for contention-free medium access. It may span multiple consecutive TXOPs and its maximum duration is limited by a parameter specified in the standard. The QAP has a higher medium access priority than other stations since it needs to wait for only PIFS, which is shorter than DIFS and AIFS that other stations must wait, before accessing the medium. In HCCA, stations are allowed to reserve TXOPs for their traffic streams by sending an Add Traffic Stream (ADDTS) Request frame to a QAP. The ADDTS Request is a management Action frame and contains a TSPEC. Since applications with QoS requirements have the possibility to specify required QoS parameters in the TSPEC, HCCA is said to provide parameterized QoS.

The IEEE 802.11e standard amendment specifies a reference design for a sample scheduler and an Admission Control Unit (ACU). These use the mandatory set of TSPEC parameters to decide on admission and generate a schedule:

- ρ: Mean Data Rate (from the negotiated TSPEC)
- L: Nominal MSDU Size (from the negotiated TSPEC)

- SI_{max}: Maximum Service Interval *or* D: Delay Bound
- R: Physical Transmission Rate (equal to the Minimum PHY Rate negotiated in the TSPEC or the observed PHY Rate)
- M: Maximum Allowable Size of MSDU
- O: Overheads in time units (including interframe spaces, ACK frames and CF-Poll frames)

The admission control and scheduling procedure can be described according to the following four steps:

1. **Calculate SI**.
   First the scheduler calculates the minimum m of all SI_{max} for all admitted streams. Then SI equals a value lower than m that is a submultiple of the beacon interval. For example, if the beacon interval is equal to 100 ms, SI can have one of the following values: {2, 4, 5, 10, 20, 25, 50} ms.

2. **Calculate the number of MSDUs that arrive at the mean data rate during SI**.
   For traffic stream i, where L and ρ are given by the application, the number of arrived MSDUs during SI equals

$$N_i = \left\lceil \frac{SI \times \rho_i}{L_i} \right\rceil$$

3. **Calculate the TXOP duration**.
   The scheduler calculates the TXOP duration as the maximum of (a) the time to transmit N_i frames at R_i plus overhead and (b) the time to transmit one maximum size MSDU at R_i plus overhead. This way, the scheduler ensures that the station can transmit at least one maximum-sized MSDU during a TXOP. Thus, the TXOP duration for traffic stream i equals

$$TXOP_i = \max\left(\frac{N_i \times L_i}{R_i} + O, \frac{M}{R_i} + O\right)$$

4. **Admit or reject the traffic stream**.
   Assuming that there are k admitted streams, a new stream (k+*1*) can be admitted if it satisfies the following inequality:

$$\frac{TXOP_{k+1}}{SI} + \sum_{i=1}^{k} \frac{TXOP_i}{SI} \leq \frac{T - T_{CP}}{T},$$

where T is the beacon interval and T_{CP} is the time used for EDCA traffic. The last term ensures that some amount of time is saved for contending low-priority streams.

It should be noticed that SI must be recalculated when a new traffic stream with SI_{max} smaller than the current SI is admitted. This will in turn lead to the recalculation of the TXOP duration based on the new value of SI.

To improve the performance of the scheduler, it can for example be modified to generate different SIs for different traffic streams or consider retransmissions while calculating TXOP durations. To improve the performance of the admission control algorithm, it might include UPs in the decision of admitting, retaining, or dropping a traffic stream. Thus, the scheduler and the admission control algorithm presented in the IEEE 802.11e specification are just examples and any modification can be made in order to improve their performance (see Section 4.2.1).

2.2.3 Real-World EDCA Experiments

As discussed in Section 2.2.2.1, we can use the EDCA medium access protocol to provide service differentiation. We have hence run a preliminary QoS experiment with integrated Intel PRO/Wireless 3945ABG Network Connection wireless cards on three laptops (say A, B, and C) operating in ad hoc mode. In the experiments, we used the default values of the wireless cards, i.e., channel 11, IEEE 802.11b/g, ad hoc power management disabled, EDCA enabled, HD mode disabled (a feature reducing interference in environments with several nearby APs), CTS-to-self enabled, power management optimized for maximum battery life, balanced setting between roaming and performance, throughput enhancement disabled, and highest transmit power optimized for maximum coverage.

In the first experiment, three stations are positioned in a straight line, equally distanced, with B in the middle. Both nodes A and C send data to B. The outcome has been captured using the network protocol analyzer WireShark and is shown in Figure 2.3. Basically, two streams are simultaneously present on the channel: a best effort stream (the red curve) and a prioritized voice stream (the green curve). The black curve denotes the sum of the throughput of both streams.

First, we start a best effort stream from node A to B starting at 4 s and going on until 6 s. Since this is the only active stream during this period, its throughput curve (in red) overlaps with the curve representing the total throughput (in black). We can see that the throughput of the best effort stream is around 1 Mbps. Then, at 6 s, a voice stream is started from node C to B. This is depicted by the green curve going on approximately from

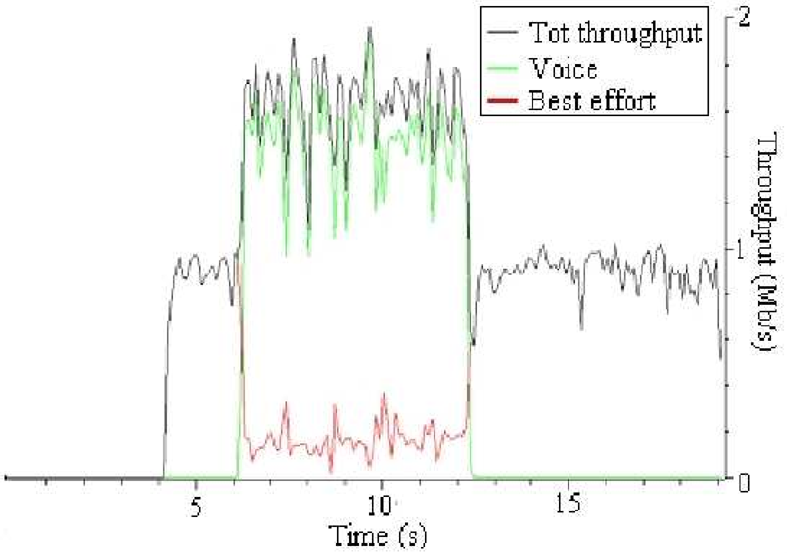
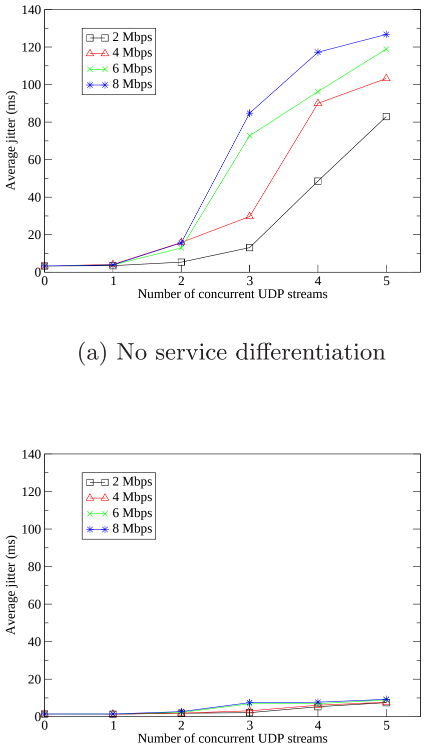


Figure 2.3: EDCA experiments with voice (green) and best effort (red) streams. The black curve shows the sum of the throughput of both streams.

6 s to 12 s. The figure shows that the throughput of the best effort stream falls to around 200 kbps, whereas the voice stream gets around 1.5 Mbps. This is obviously because EDCA succeeds in prioritizing the high-priority voice stream and transmit its data frames using AC_VO. The low-priority best effort stream, on the other hand, is transmitted using AC_BE. This results in longer waiting times before trying to access the wireless medium and no possibility to transmit more than a single data frame at a time; or in other words, lower throughput. At 12 s, the voice stream is stopped and the best effort stream recaptures the available bandwidth of the channel, again climbing up to 1 Mbps.

In the second experiment, we continued to analyze the effectiveness of EDCA in prioritizing data flows. The outcomes are reported in Figure 2.4a and Figure 2.4b, showing the average jitter of a Voice over IP (VoIP) stream among a node pair, when a variable number (from zero to five) of concurrent User Datagram Protocol (UDP) streams share the same link. Moreover, we also vary the data rate of each of the considered UDP streams for a certain testbed configuration from 2 Mbps to 8 Mbps (*2 Mbps*, *4 Mbps* ... *8 Mbps* in the charts). Figure 2.4a presents the average jitter of the VoIP stream when no EDCA priority is exploited, whereas Figure 2.4b shows the same outcome when a high priority is assigned to the VoIP packets. As it is evident, without any priority (Figure 2.4a) the average jitter experienced by the



(a) No service differentiation

(b) With EDCA service differentiation

Figure 2.4: Average jitter experienced by a VoIP stream when competing with concurrent UDP streams: (a) all streams have the same priority equal to zero (see Table 2.1) and (b) the VoIP stream is given high priority equal to seven.

VoIP stream grows very quickly as soon as we introduce some background traffic. Instead, Figure 2.4b demonstrates that when EDCA is employed, even with five concurrent UDP streams of 8 Mbps each, the average jitter experienced by the high-priority VoIP stream remains very little.

2.2.4 IEEE 802.11 Standards and Recommendations

The standards, recommendations, and TGs in the IEEE 802.11 working group have been expanding for a long time. To bring some order into the alphabet soup, the work of all TGs is summarized here [11, 12]. Note that there is no standard or task group called ”802.11x”, which sometimes is used to denote any current or future IEEE 802.11 standard. Furthermore, IEEE 802.11l, IEEE 802.11o, and IEEE 802.11q are not used.

- IEEE 802.11 - a WLAN standard specifying the PHY and the MAC sublayer. It specifies three PHY options with data rates of 1-2 Mbps: IR at 850-950 nm, FHSS and DSSS at 2.4 GHz. (1997)

- IEEE 802.11a - a PHY amendment at the 5 GHz band providing data rates up to 54 Mbps. (1999)

- IEEE 802.11b - a PHY amendment extending IEEE 802.11 PHY (DSSS) at the 2.4 GHz band to support 5.5 and 11 Mbps. (1999)

- IEEE 802.11c - a network interoperability amendment that deals with bridge operation procedures. A bridge is a device that connects local area networks with a similar or identical MAC protocol. This amendment is included in the IEEE 802.1D standard. (2003)

- IEEE 802.11d - a "global harmonization" amendment that defines PHY requirements to satisfy regulatory domains since the allowed frequencies, power levels and signal bandwidth may differ between different countries. Thus, the specification eliminates the need for manufacturing country-specific products. (2001)

- IEEE 802.11e - a QoS amendment that defines enhancements to the IEEE 802.11 MAC sublayer. In addition to providing QoS, the amendment improves the MAC performance by specifying functions such as Block Acknowledgment (BA), Direct-Link Setup (DLS), and Automatic Power-Save Delivery (APSD). (2005)

- IEEE 802.11F - an AP interoperability *recommendation* that defines an extension (Inter-Access Point Protocol) to IEEE 802.11 to simplify wireless communications among APs from different vendors. This document was published in 2003 but withdrawn in 2006. (2003)

- IEEE 802.11g - a PHY amendment extending IEEE 802.11 PHY to support data rates up to 54 Mbps at the 2.4 GHz band. This amendment is backward compatible with IEEE 802.11b. (2003)

- IEEE 802.11h - a spectrum and transmit power management amendment that allows IEEE 802.11a devices to co-exist with devices using other amendments operating at the same 5 GHz frequency band. In the European Union, the 5 GHz band is used for, e.g., satellite communication, so the amendment uses a dynamic frequency selection mechanism to prevent selection of congested channels. The transmit power control function of the amendment adjusts the power to the EU requirements. (2004)

- IEEE 802.11i - a security amendment that defines enhancements to the IEEE 802.11 MAC sublayer. The amendment supersedes the Wired Equivalent Privacy (WEP) algorithm, which was specified in the original standard and had severe security weaknesses. (2004)
- IEEE 802.11j - an amendment that specifies the 4.9-5 GHz operation in Japan. It defines methods that let APs move to new frequencies or change the channel width for better performance and capacity and not to interfere with other wireless devices using the same frequency band in Japan. (2004)
- IEEE 802.11k - an amendment for radio resource measurement that is intended to improve the way the traffic is distributed within a WLAN. Instead of connecting to the AP with the strongest signal, a station will also consider the load of the existing APs. In other words, IEEE 802.11k provides information to discover the best available AP. Thus, a station may connect to an AP with a weaker signal but that is underutilized, thereby increasing the overall performance in the WLAN. (2008)
- IEEE 802.11l - not used
- IEEE 802.11m - an initiative to perform editorial corrections, clarifications, and interpretations in the IEEE 802.11 family specifications. After merging eight published amendments (IEEE 802.11a/b/d/e/g/h/i/j) with the IEEE 802.11-1999 standard, the IEEE 802.11 standard was updated in 2007 and called IEEE 802.11-2007. The work of TGm is still continuing.
- IEEE 802.11n - an upcoming amendment that is expected to be the successor of IEEE 802.11g. The goal is to improve the PHY and the MAC sublayer to enable higher throughput (several hundreds of Mbps), partly by adding MIMO technology, i.e., by using multiple transmitter and receiver antennas.
- IEEE 802.11o - not used
- IEEE 802.11p - an upcoming amendment that defines enhancements required to support Intelligent Transportation Systems (ITS) applications, such as toll collection, vehicle safety services, and commerce transactions via cars. The goal is to enable communication between vehicles and roadside APs or other vehicles. This includes data exchange between high-speed vehicles and between these vehicles and the roadside infrastructure in the licensed ITS band of 5.9 GHz. The amendment is sometimes referred to as Wireless Access for the Vehicular Environment (WAVE).

- IEEE 802.11q - not used

- IEEE 802.11r - an amendment for fast roaming of stations. This amendment will enable connectivity aboard vehicles in motion, with fast roaming from one AP to another. Furthermore, it will facilitate the deployment of IP-based telephony over IEEE 802.11-enabled phones. (2008)

- IEEE 802.11s - an upcoming amendment for mesh networking that defines extensions to the IEEE 802.11 MAC sublayer. The purpose of the project is to provide a protocol for auto-configuring paths between APs over multi-hop topologies.

- IEEE 802.11T - a *recommendation* that specifies test methods and metrics to measure and evaluate the performance of IEEE 802.11-based devices and networks. (2008)

- IEEE 802.11u - an upcoming amendment extending both the PHY and the MAC sublayer to enable interworking with external networks (e.g., the Internet or cellular networks). Since IEEE 802.11-based WLANs has become more widespread, this work started to solve the problems related to the connection of a WLAN to an external network in a standardized manner.

- IEEE 802.11v - an upcoming amendment extending both the PHY and the MAC sublayer to provide wireless network management for stations.

- IEEE 802.11w - an upcoming amendment with focus on security of management frames by enhancing the MAC sublayer. The IEEE 802.11i amendment addresses the security of only data frames so the WLANs are still vulnerable to malicious attacks because of the unprotected management frames.

- IEEE 802.11x - not used

- IEEE 802.11y - an amendment extending the PHY and using the 3.65-3.7 GHz band, which previously was reserved for fixed satellite service networks. The amendment will provide a standardized interference avoidance mechanism and streamline the adoption of new frequencies in the future. (2008)

- IEEE 802.11z - an upcoming amendment specifying DLS extensions. The goal is to allow operation with non-DLS capable APs and allow stations with an active DLS session to enter power save mode.

- IEEE 802.11aa - an upcoming amendment for robust streaming of audio video transport streams.
- IEEE 802.11ac - an upcoming amendment for very high throughput below the 6 GHz frequency band.
- IEEE 802.11ad - an upcoming amendment for very high throughput in the 60 GHz band. The goal is to specify modifications to both PHY and MAC to enable very high throughput in the 60 GHz band (typically 57-66 GHz).

2.3 Routing in Ad Hoc Networks

In order to allow for multi-hop communication, a source needs a routing protocol to find a route, through possible intermediate stations, to the destination. These routing protocols can be classified into two main classes[2]: *proactive* and *reactive* routing protocols. In proactive routing, the routing table of every station is updated periodically. Thus, the delay before sending a packet is minimal but at the cost of increased routing overhead. On the contrary, reactive routing is performed on-demand, i.e., the sending station searches for a route to the destination station only when it needs to communicate with it. Hence, the routing overhead is minimized but the route discovery process may result in considerable delay. The following sections give an overview of some common ad hoc routing protocols.

2.3.1 Ad hoc On-Demand Distance Vector (AODV)

AODV is a popular, reactive routing protocol, which guarantees loop-free routes by using sequence numbers that indicate how new a route is. Its reactive property implies that a station requests a route only when it needs one. AODV requires each station to maintain a routing table containing one route entry for each destination that the station is communicating with. Each route entry keeps track of certain fields, including *Destination IP Address*, *Destination Sequence Number*, *Next Hop* (a neighbor station chosen to forward packets to the destination), *Hop Count* (the number of hops needed to reach the destination), and *Lifetime* (the expiration or deletion time of the route).

Whenever a station determines that it needs a route to a station for which it does not have a route, it starts the route discovery process by broadcasting a Route Request (RREQ) and starting a timer to wait for the reception of a Route Reply (RREP). A neighbor receiving a RREQ sends a RREP back to the source if it is either the destination or if it has an unexpired route to the

[2] There are other ways to categorize ad hoc routing protocols [13, 14, 15, 16].

destination. If none of these two cases is satisfied, the neighbor rebroadcasts the RREQ. To prevent dissemination of duplicated RREQs, stations keep a cache where they store the source IP address and ID of the received RREQs during a short period of time. If the stations receive another RREQ with the same source IP address and RREQ ID during this period, it is discarded.

When searching for a route to the destination, the source may use the *expanding ring search* technique to prevent unnecessary network-wide dissemination of RREQs. This is done by controlling the value of the Time To Live (TTL) field in the IP header, which defines the maximal number of hops a RREQ can move through the network.

When a link break occurs, the station upstream of the break invalidates all its routes that use the broken link and broadcasts a Route Error (RERR). The RERR contains a list of each destination that has become unreachable due to the link break. Upon reception of a RERR, a station invalidates possible routes to the unreachable destinations and broadcasts a new RERR. This process continues until the source receives a RERR. The source invalidates the listed routes and re-initiates a route discovery process if needed.

2.3.2 Dynamic Source Routing (DSR)

DSR is a purely reactive routing protocol that uses *source routing* to send packets. Source routing means that the header of each data packet carries the complete list of intermediate stations to the destination. Consequently, the overhead caused by DSR increases but, on the other hand, its entirely reactive behavior means that DSR requires no periodic packets of any kind; thereby decreasing the overhead. An advantage of source routing is that stations forwarding or overhearing data packets, can cache the routing information included in the header of the data packets for future use. Other advantages of using source routing are that it guarantees loop-free routes and supports the use of multiple routes to any destination. The support for multiple routes results in fast reaction to routing failures since a station can try another cached route.

The route discovery process is initiated only if a station needs a route that cannot be found in the route cache. The station broadcasts a Route Request, which contains the address of the source and the destination, and a unique identification number. An intermediate station that receives a Route Request searches its route cache for a route to the destination. If no route is found, the intermediate station appends its address to the route record of the Route Request and rebroadcasts the message. The message propagates through the network until it reaches either the destination or an intermediate station with a route to the destination. Then, a Route Reply, containing the proper hop sequence for reaching the destination, is generated and sent back to the source station. To limit the number of propagated Route Requests, a station discards Route Requests that it has received recently with the same identification number and destination address or if its address is already present in the route record of the Route Request.

Route maintenance is used to handle route breaks. When a station encounters a transmission problem at its data link layer, DSR removes the route with the broken link from its route cache and generates a *Route Error*. The Route Error is sent to each station that has sent a packet routed over the broken link. When a station receives a Route Error, it removes the hop in error from its route cache.

2.3.3 Optimized Link State Routing (OLSR)

OLSR is a proactive routing protocol developed for ad hoc networks. The routing tables are kept updated by regularly exchanging topology information; thus, routes are maintained for all known destinations at all times. OLSR substantially reduces the large message overhead, which usually is associated with classical flooding mechanisms, by reducing redundant retransmissions. This is done by allowing only some selected stations, called Multipoint Relays (MPRs), to forward the broadcast messages during the flooding process. Each station in the network selects a subset of its neighbors as MPRs. To avoid problems associated with uni-directional links, the candidates for MPRs must have a bi-directional link to the selecting station. Another optimization is achieved by minimizing the set of links flooded in the network. As opposed to the classic link state algorithm, a station declares only the MPR links to its neighbors, rather than all links to all neighbors.

The concept of relaying in OLSR has been inherited from the MAC protocol High Performance Radio Local Area Network type 2 (HiperLAN/2)[3], which is standardized by the European Telecommunications Standards Institute (ETSI).

2.3.4 Topology Dissemination Based on Reverse-Path Forwarding (TBRPF)

TBRPF is a proactive, link state routing protocol designed for ad hoc networks. It provides hop-by-hop routing along shortest paths to each destination. Each station running TBRPF computes a source tree (providing paths to all reachable stations) based on partial topology information stored

[3] Before IEEE 802.11 became a de facto standard for WLANs, HiperLAN/2 was considered as a competitor. However, today we know that HiperLAN/2 did not become widely used, despite advertised advantages compared to IEEE 802.11 (e.g., regarding QoS provisioning), which had many unsolved problems that were dealt with in later amendments.

in its topology table, using a modification of Dijkstra's algorithm. To minimize overhead, each station reports only part of its source tree to neighbors. TBRPF uses a combination of periodic and differential updates to keep all neighbors informed of the reported part of its source tree. Each station also has the option to report additional topology information (up to the full topology), to provide improved robustness in highly mobile networks. TBRPF performs neighbor discovery using "differential" HELLO messages that report only changes in the status of neighbors. This results in HELLO messages that are much smaller than those of other link state routing protocols.

2.3.5 Dynamic MANET On-demand (DYMO)

DYMO is a reactive routing protocol and the main candidate for the upcoming reactive MANET routing protocol. It is based on the work and experience from previous reactive routing protocols, especially AODV and DSR. To ensure loop-free routes, DYMO uses the same technique as in AODV, namely sequence numbers. The route discovery process is done using RREQs and RREPs, whereas RERRs are used to maintain routes. The stations monitor links on active routes through, e.g., link layer feedback, neighbor discovery, and route timeouts. The DYMO draft specifies the base specification but by using the generalized MANET packet and message format [17], it is prepared for extensions.

2.3.6 Optimized Link State Routing version 2 (OLSRv2)

As the name implies, the OLSRv2 is very similar to the OLSR protocol described earlier. The protocol has the same key optimization techniques as in the OLSR protocol, i.e., using MPRs responsible for forwarding control traffic that must be flooded in the entire network, and maintaining partial link state information to reduce the number and size of the network-wide broadcasts.

The main differences compared to the first version are more flexible signaling framework and some simplifications on the messages. As opposed to OLSR, but just like DYMO, OLSRv2 is prepared to use the MANET Neighborhood Discovery Protocol (NHDP) [18] and the generalized MANET packet and message format specified by the IETF MANET working group. Consequently, as opposed to OLSR, OLSRv2 allows for modifications of existing control messages and extensions with new message types.

2.4 Simulation of Ad Hoc Networks

There are many network simulation tools used by the research community; the most popular ones are ns-2, ns-3, OPNET, OMNET++, QualNet, NCTUns, etc. In this thesis we used the network simulator ns-2 [19], which is an object-oriented, discrete event simulator for networking research. The reasons for this choice are that ns-2 is free to download and use, provides a wide range of features, has an open source code that can be modified and extended, and has an active community continuously helping to improve the simulator. ns-2 provides substantial support for simulation of Transmission Control Protocol (TCP), routing and multicast protocols over wired and wireless networks. The simulator is a result of an on-going effort of research and development. Even though there is a considerable confidence in ns-2, it is not a polished and finished product yet and bugs are being discovered and corrected continuously.

The ns-2 simulator is written in C++, with an Object Tool Command Language (OTcl) interpreter as a command and configuration interface. The C++ part, which is fast to run but slower to change, is used for detailed protocol implementation. The OTcl part, on the other hand, which runs much slower but can be changed very quickly, is used for simulation configuration. One of the advantages of this split-language programming approach is that it allows for fast generation of large scenarios. On the other hand, one disadvantage is that modifying and extending the simulator requires programming and debugging in both languages simultaneously. To simply use the simulator, it is sufficient to know OTcl. However, modification to existing protocols or implementation of new ones requires C++ knowledge.

Network Animator (NAM) is an animation tool for viewing network simulation traces and real world packet traces. It supports topology layout, packet level animation and various data inspection tools. Figure 2.5 shows a NAM window explaining the most important functions.

Before starting to use NAM, a trace file needs to be created. This trace file is usually generated by ns-2. It contains topology information, e.g., stations and links, as well as packet traces. During a simulation, the user can produce topology configurations, layout information and packet traces using tracing events in ns-2.

Once the trace file is generated, NAM can be used to animate it. Upon startup, NAM will read the trace file, create topology, pop up a window, do layout if necessary and then pause at time 0. Through its user interface, NAM provides control over many aspects of the animation.

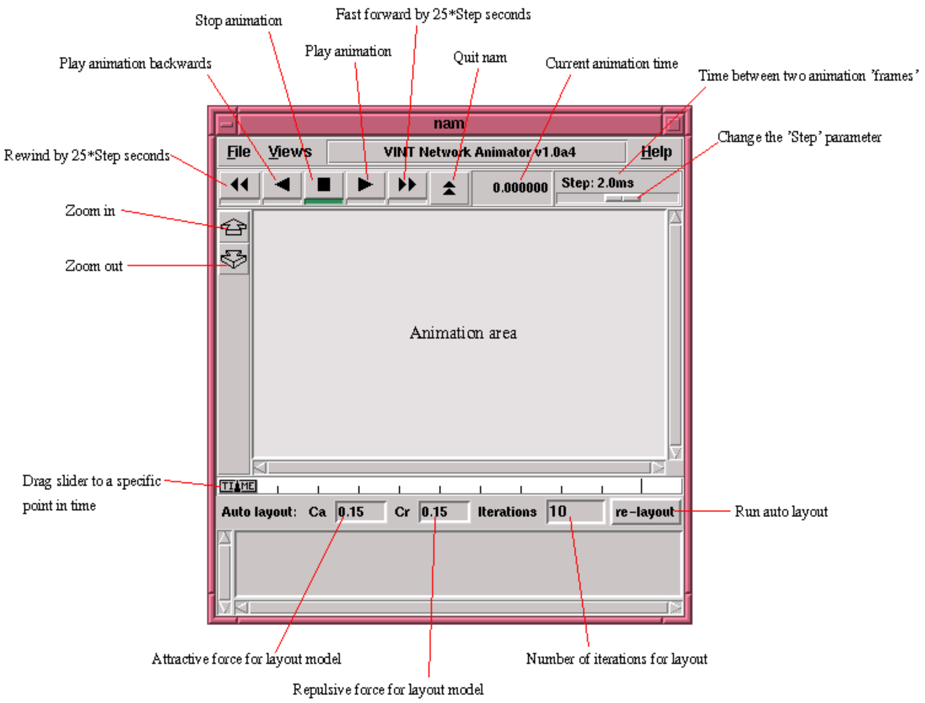

Stop animation
Fast forward by 25*Step seconds
Play animation
Play animation backwards
Quit nam
Current animation time
Time between two animation 'frames'
Change the 'Step' parameter
Rewind by 25*Step seconds
Zoom in
Zoom out
nam
File
Views
VINT Network Animator v1.0a4
Help
0.000000
Step: 2.0ms
Animation area
Drag slider to a specific point in time
TIME
Auto layout:
Ca
0.15
Cr
0.15
Iterations
10
re-layout
Run auto layout
Attractive force for layout model
Repulsive force for layout model
Number of iterations for layout

Figure 2.5: Screenshot of a NAM window.

Appendix A - Configuration of IEEE 802.11e-capable Stations

In order to ease the performance evaluation of EDCA technology, we have developed the tools described below. It is our hope that these tools may be valuable to continue ad hoc research. Therefore, we have decided to make them available online at http://www.eit.lth.se/staff/ali.hamidian.

- qostest.exe - This is a program written in C# that can behave as either a sender or receiver. It sets various socket options to either enable or disable QoS settings, such as the DSCP value in the IP header; it was hence utilized to perform the preliminary QoS evaluation in Section 2.2.3. When configured as sender, the sending rate can be specified by calculating the ratio of the packet size to the sending rate (taken as arguments). When configured as receiver, it can receive an indefinite number of packets: a timeout can be specified to eventually block the receive call. Upon timing out, the receiver will print statistics that are specific to a certain run.
- parsedotnet.pl - This is a Perl script that parses the output of the qostest.exe program for several runs, and prints summarized statistics.

It takes a filename as an argument: it assumes the file contains data for several runs of qostest.exe, and then proceeds to calculate averages based on the run information. The output of this program can then be easily loaded into tools such as MATLAB to ease the plotting and analysis process.

However, enabling QoS settings via software is not enough: it is necessary to activate the IEEE 802.11e EDCA also via hardware. To this aim, we have to point out that finding a wireless card that could support EDCA in ad hoc mode has not been an easy task; in fact, most commercial cards have support for EDCA in infrastructure mode only. Even worse, some vendors promise support for EDCA also in ad hoc mode without fulfilling this promise. It should also be mentioned that the Wi-Fi Alliance has an optional certification testing support for multimedia content over Wi-Fi networks, Wi-Fi MultiMedia (WMM), but this program checks EDCA support in infrastructure mode only and not in ad hoc mode. In other words, EDCA support in ad hoc mode is totally up to the vendor to implement. The only wireless cards we have found that could use EDCA in an ad hoc mode are:

- Intel PRO/Wireless 3945ABG Network Connection
- Intel Wireless WiFi Link 4965AGN

In our experiments, we used the Intel PRO/Wireless 3945ABG Network Connection wireless cards that were integrated in our laptops. Once having a wireless card that supports EDCA in ad hoc mode, this feature has to be enabled in the hardware. To this aim, in the following we report sketch instructions on how to do it.

1. **Set up an ad hoc network**:
   Start ⇒ Control Panel ⇒ Network Connection ⇒ Wireless Network Connection ⇒ Properties ⇒ Wireless Networks ⇒ Add...
   - Network name (SSID): MyAdHocNetwork
   - Network Authentication: Open
   - Data Encryption: Disabled
   - Click the box This is computer-to-computer (ad-hoc) network; wireless access points are not used
2. **Set a static IP address**:
   Start ⇒ Control Panel ⇒ Network Connection ⇒ Wireless Network Connection ⇒ Properties ⇒ Internet Protocol (TCP/IP) ⇒ Properties ⇒ Use the following IP address

   - IP address: 192.168.10.1
   - Subnet mask: 255.255.255.0

3. **Enable the use of the IP_TOS socket option**:
   Start ⇒ Run ⇒ regedit ⇒ HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters ⇒ Edit ⇒ New ⇒ DWORD Value

   - Create and set a new DWORD registry value:
     DisableUserTOSSetting=0

4. **Enable WMM in ad hoc mode**:
   Start ⇒ Control Panel ⇒ Network Connection ⇒ Wireless Network Connection ⇒ Properties ⇒ Configure ⇒ Advanced ⇒ Ad Hoc QoS Mode

Chapter 3

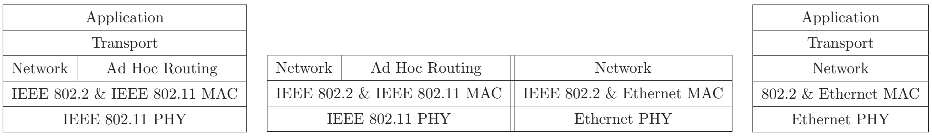
Internet Connectivity for MANETs

Although an autonomous, stand-alone MANET is useful in many cases, a MANET connected to the Internet is much more desirable. This is because Internet plays an important role in the daily life of many people by offering a broad range of services. This network interconnection between a wireless MANET and the wired Internet is achieved by using an ad hoc routing protocol that is able to route packets not only within a MANET, but also from a MANET to the Internet. However, most ad hoc routing protocols have been designed to route packets within an autonomous MANET and not between a MANET and a wired network. Therefore, we have extended the AODV routing protocol to provide Internet access to mobile stations in a MANET. Our solution is called *AODV+* and the reason we chose AODV as the basis for our solution is that AODV is a popular and widely used ad hoc routing protocol.

In addition to an ad hoc routing protocol with support for Internet access, we also need *Internet gateways*[1] that act as bridges between the two networks. In other words, we need gateways that can translate between both "languages" since all communication between the two networks pass through the gateways. Figure 3.1 illustrates the role of the gateway being able to communicate with both wired and wireless stations. Before a wireless device in a MANET can communicate with an Internet host, it needs to find a route to a gateway. Thus, a gateway discovery mechanism needs to be considered in Internet access solutions. Consequently, *AODV+* implements three different methods for gateway discovery: *reactive*, *proactive*, and *hybrid* gateway discovery.

As Figure 3.1 shows, the gateway implements the three lower layers in

[1] From now on and throughout this text we use the shorter term *gateway* instead of *Internet gateway* since no other kind of gateways are of importance in this thesis.



(a) MANET station (b) Gateway (c) Internet host

Figure 3.1: The gateway acts as a bridge between the wireless MANET and the wired Internet. Therefore, it must be able to understand the protocols used in both networks.

the protocol stack, that is, the physical layer, the data link layer, and the network layer. The MANET station and the Internet host, on the other hand, also implement protocols operating on higher layers, e.g., Real-time Transport Protocol (RTP), TCP, and UDP at the transport layer and Hypertext Transfer Protocol (HTTP), File Transfer Protocol (FTP), BitTorrent, and VoIP at the application layer.

In this chapter, we present three different methods for gateway discovery. Next we present an Internet access solution for MANETs, *AODV+*, and show by simulations that it is capable of providing Internet access to mobile stations in a MANET. Finally, after a short conclusion, we give a detailed description of the implementation of *AODV+* as well as the three gateway discovery methods.

3.1 Related Work

There has been much research on Internet access for ad hoc networks during the last decade. In the beginning of the decade some works focused on Internet access solution based on Mobile IPv4 [20] and Mobile IPv6 [21]. For example, in [22] the authors present a solution that provides Internet access by using tunneling and Mobile IP with foreign agent care-of addresses. In order to access the Internet, mobile stations need to register with a foreign agent and tunnel all packets destined for the Internet to the registered foreign agent. The foreign agent decapsulates the packets and forwards them to the destination. The ad hoc routing protocol AODV is used within ad hoc networks to route packets between mobile stations and foreign agents.

Another example of work based on Mobile IP is presented in [23], where AODV cooperates with the Mobile IP protocol. Mobile IP is used for mobile station registrations with a foreign agent, whereas AODV is used for routing within the ad hoc network and for obtaining routes to the foreign agent. In this solution, the foreign agent discovery mechanism is incorporated into the ad hoc routing protocol.

As the interest for Mobile IP decreased, the focus on Internet access research changed to solutions that were not based on Mobile IP. For example, in [24] the authors discuss issues like gateway discovery and address autoconfiguration. One of the leading and most promising works in the field was the now outdated Internet draft [25], which describes the operation of gateways and how to obtain globally routable addresses.

However, most of the related works propose solutions that have not been evaluated. This is due to lack of implementations of the proposed designs. An implementation and evaluation may reveal problems that are not obvious at first. As opposed to other works, we do not only propose a solution for Internet access and/or gateway discovery but also provide an implementation of the proposed mechanisms. Moreover, our open source solution is contributed to the ns-2 community and thus used research done in the field and allowing others to do a thorough investigation of our solution. Accordingly, we have found that our work has been used by researchers for various kinds of studies (e.g., [26, 27, 28, 29]), proposing improvements and enhancements to our solution, especially the gateway discovery mechanisms [30, 31, 32, 33, 34]. Thus, the main contribution of our work is a theoretic discussion of problems related to interconnecting wireless ad hoc networks with the wired Internet in combination with a public implementation and evaluation of this proposed solution.

3.2 Gateway Discovery

The question of whether the registration process with the gateway should be initiated by the gateway (proactive method), by the mobile station (reactive method) or by either of the two (hybrid proactive/reactive method) has been studied in this thesis.

3.2.1 Reactive Gateway Discovery

The reactive gateway discovery is initiated by a mobile station when it determines that it needs to access the Internet. The mobile station broadcasts a RREQ with an 'I'-flag set, i.e., a RREQ_I, to the ALL_MANET_GW_MULTICAST address, which is the IP address for the group of all gateways in a MANET. Thus, the RREQ_I is processed only by the gateways in the MANET. Intermediate mobile stations that receive a RREQ_I just rebroadcast it. Since the message format is RREQ, which has a RREQ ID field, duplicated RREQ_Is can be detected and discarded. When a gateway receives a RREQ_I it sends back a RREP with an 'I'-flag set, i.e., a RREP_I, which among other things, contains the IP address of the gateway.

The advantage of this approach is that RREQ_Is are generated only when a mobile station needs to find a route to reachable gateways. Hence, periodic flooding of the whole MANET, which has obvious disadvantages, is avoided. On the other hand, the disadvantage of reactive gateway discovery is that a handover cannot be initiated before a mobile station loses its Internet connection. As a consequence, a situation can occur where a mobile station uses a gateway for its Internet connection although there are other gateways that are more suitable. Moreover, there is a delay caused by the route discovery process.

3.2.2 Proactive Gateway Discovery

The proactive gateway discovery is initiated by the gateway itself. The gateway periodically broadcasts a Gateway Advertisement (GWADV) message to inform mobile stations in the network of its presence. The time between two consecutive advertisements is determined by the ADVERTISEMENT_INTERVAL parameter and must be chosen with care so that the network is not flooded too frequently.

Upon receipt of the GWADV, the mobile stations update their routing tables and rebroadcast the message in order to spread it further in the network. To assure that all mobile stations within the MANET receive the advertisement, the TTL is set to the maximum value, i.e., NET_DIAMETER defined by AODV. However, this will lead to lots of unnecessary duplicated GWADVs. A conceivable solution that prevents duplicated GWADVs, is to introduce a “GWADV ID” field in the GWADV message format similar to the “RREQ ID” field in the RREQ message format.

It is worth mentioning that mobile stations randomize their rebroadcasting of GWADV messages in order to avoid synchronization and subsequent collisions with other stations’ rebroadcasts.

The advantage of this approach is that there is a chance for mobile stations to initiate a handover before they lose their Internet connection. Moreover, the need for a time-consuming route discovery process is eliminated since the routes to the gateways are updated periodically. On the other hand, the disadvantage is that limited resources in a MANET will be used a lot since GWADVs are flooded through the whole MANET periodically.

3.2.2.1 Duplicated Broadcast Messages

The problem of duplicated broadcast messages in MANETs is well known. In AODV, RREQ messages are broadcasted. To avoid duplicated RREQs, a RREQ ID is used. When a RREQ is received by a mobile station, it first checks to determine whether it already has received a RREQ with the same originator IP address and RREQ ID. If such a RREQ has already been received, the station discards the newly received RREQ.

Here, we use the idea of comparing the RREQ ID with the originator

<table>
<tr><td>Type</td><td>R</td><td>A</td><td>I</td><td>Reserved</td><td>Prefix Sz</td><td>Hop Count</td></tr>
<tr><td colspan="7">GWADV ID</td></tr>
<tr><td colspan="7">Destination IP Address</td></tr>
<tr><td colspan="7">Destination Sequence Number</td></tr>
<tr><td colspan="7">Originator IP Address</td></tr>
<tr><td colspan="7">Lifetime</td></tr>
</table>

Figure 3.2: The format of the GWADV message.

IP address to solve the problem of duplicated GWADVs. A GWADV is an extended RREP_I message: since RREPs do not contain any field similar to the RREQ ID field in RREQ messages, we have extended the RREP message extended with the GWADV ID field. Figure 3.2 illustrates the GWADV message format.

When a mobile station receives a GWADV, it first checks to determine whether a GWADV with the same originator IP address and RREQ ID has already been received during the last BCAST_ID_SAVE seconds. If such a GWADV message has not been received, the message is rebroadcasted; otherwise, the newly received GWADV is discarded. Hence, duplicated GWADVs are not forwarded and the advertisement is flooded through the whole network without causing too much congestion.

3.2.3 Hybrid Gateway Discovery

To minimize the disadvantages of proactive and reactive gateway discovery strategies, they can be combined into a hybrid proactive/reactive method for gateway discovery. For mobile stations in a certain zone around a gateway, proactive gateway discovery is used, whereas mobile stations residing outside this zone use reactive gateway discovery to find a route to the gateway.

The gateway periodically broadcasts a GWADV message. Upon receipt of the message, the mobile stations update their routing table and then rebroadcast the message. The maximum number of hops a GWADV can disseminate through the MANET is determined by ADVERTISEMENT_ZONE. This value defines the zone within which proactive gateway discovery is used. When a mobile station residing outside this zone needs gateway information, it broadcasts a RREQ_I to the ALL_MANET_GW_MULTICAST address. Mobile stations receiving the RREQ_I just rebroadcast it. When a gateway receives a RREQ_I, it responds by sending a RREP_I towards the source.

Thus, the proactive gateway discovery method is used to handle the mobile stations less or equal than ADVERTISEMENT_ZONE hops away from the gateway and the reactive gateway discovery method is used to handle the mobile stations more than ADVERTISEMENT_ZONE hops away from the gateway.

3.3 Internet Connectivity for MANETs - *AODV+*

Whenever a mobile station is about to communicate with a fixed wired Internet host, the mobile station searches its routing table for a route towards the destination. If a route is found, the communication can be established; otherwise, the mobile station starts a route discovery process by broadcasting a RREQ message with its own IP address as the *Originator IP Address* field and the address of the Internet host specified in the *Destination IP Address* field.

When an intermediate mobile station receives a RREQ, it searches its routing table for a route towards the destination, i.e., the Internet host in our case. If a route is not found, the intermediate station simply updates its routing table and rebroadcasts the RREQ. However, if a route is found, according to the AODV operation rules the intermediate station would send a RREP back to the originator of the RREQ. However, in that case the source would think that the destination is a mobile station that can be reached via the intermediate station. It is important that the source knows that the destination is an Internet host and not a mobile station, because these are sometimes processed differently. In *AODV+*, this problem has been solved by preventing the intermediate station to send a RREP back to the originator of the RREQ if the destination is an Internet host. Instead, the intermediate station updates its routing table and rebroadcasts the received RREQ message. To determine whether the destination is an Internet host, an intermediate station consults its routing table. If the next hop address of the destination is a default route (see Figure 3.3), the destination is an Internet host; otherwise, it is a mobile station or a gateway.

Since the destination is an Internet host, no mobile station will ever send a RREP back to the originator of the RREQ. Thus, the RREQ is rebroadcasted until its TTL value reaches zero. When the timer of the RREQ expires, a new RREQ message is broadcasted with a larger TTL value. However, since the Internet host cannot receive the RREQ message (no matter how large the TTL value is) the source will never receive the RREP message it is waiting for. This problem has been solved by letting the source assume the destination is an Internet host if a *network-wide search* has been done without receiving any corresponding RREP. In that case, the source must find a route to a gateway (if it does not have one already) and send its data packets towards the gateway, which in turn forwards them towards the Internet host.

It should be mentioned that when using the expanding ring search, a considerable route discovery delay will occur if the destination is an Internet host. Modifying the parameters involved in the expanding ring search technique (such as TTL_START and TTL_THRESHOLD) can decrease the route discovery delay if the destination is an Internet host. However, the modification can also result in increased routing overhead if the destination is a mobile station. The modification could for example be to increase TTL_START. Assuming the destination is an Internet host, increasing TTL_START would result in less number of broadcasted RREQs (and, consequently, less delay) before the source assumes that the destination is an Internet host. As an alternative approach to waiting for a network-wide search, the gateway could respond to incoming RREQs on behalf of wired stations on the Internet.

3.3.1 Handover in Multiple Gateway Scenarios

Due to the multi-hop nature of a MANET, there might be several reachable gateways for a mobile station at some point of time. If a mobile station receives gateway advertisements from more than one gateway, it has to decide which gateway to use for its connection to the Internet. In *AODV+*, a mobile station initiates a handover when it receives an advertisement from a gateway that is closer (in terms of number of hops) than the one it is currently registered with. Apart from the hop count, there are other potential criteria that could be used to determine whether a handover is needed; e.g., geographical distance, radio signal level, signal delay and direction of station movement [35]. However, the question of a suitable metric for route selection is a general routing issue in MANET research.

3.3.2 Gateway Operation

When a gateway receives a RREQ, it consults its routing table for the destination IP address specified in the RREQ message. If the address is not found, the gateway sends a RREP with an 'I' flag (RREP_I) back to the originator of the RREQ. On the other hand, if the gateway finds the destination in its routing table, it sends a RREP according to ordinary AODV procedures, but may also optionally send a RREP_I back to the originator of the RREQ. This will provide the mobile station a default route although it has not requested it. If the mobile station is to communicate with the Internet later, the default route is already established, and a time-consuming gateway discovery process can be avoided.

3.3.3 Unreachable Gateway

An interesting issue to consider is when a mobile station becomes isolated from the rest of the network while communicating with an Internet host. This could happen, for example, if the mobile station moves outside the transmission range of not only the gateways, but also all other wireless stations in the MANET. Losing its Internet connection, the isolated station

Destination Address	**Next Hop Address**
Internet host	Default
Default	Gateway
Gateway	Intermediate mobile station

Figure 3.3: The routing table of a mobile station after creating a route entry for an Internet host.

will obviously initiate a gateway discovery process. However, since all gateways are unreachable, the gateway discovery will not succeed. After trying a gateway discovery RREQ_I_RETRIES times at the maximum TTL without receiving any RREP_I, all data packets destined for the Internet host will be dropped from the buffer.

3.3.4 Routing Table Management

Another issue that must be taken into consideration is how the routing table should be updated after a network-wide search without receiving any corresponding RREP. Once the source has determined that the destination is an Internet host located on the Internet, it has to create a route entry for the Internet host in its routing table. If the route entry for the wired destination would not be created in the routing table, the source would not find the address to the Internet host in its routing table when the next data packet would be generated and hence, the source would have to do another time consuming network-wide search.

Figure 3.3 shows how the routing table of a mobile station should look like after creating a route entry for an Internet host. The first entry indicates that the destination is an Internet host since the next hop is specified by the default route. The second entry specifies which gateway the station has chosen for its Internet connection. The last entry gives information about the next hop towards the gateway.

Another issue is how to setup the routing table of an intermediate mobile station chosen to forward data packets towards the gateway. Since the forward route entries are created for the gateway (the source of the RREP_I) and not for the Internet host, which is the final destination of the data packets, intermediate mobile stations will not find any valid route for the Internet host when it receives data packets from the source. Therefore, it would normally drop the data packets because it does not know how to forward them. In *AODV+*, if an intermediate mobile station does not find a valid route to the destination and if the destination is an Internet host, the intermediate mobile station creates or updates the route entry for the

Internet host in its routing table and forwards the data packets towards the gateway.

3.4 Evaluation

In order to evaluate the performance of the three gateway discovery methods, we used the network simulator ns-2 (ns-2.27). First, the source code of AODV in ns-2 was extended to provide Internet access to mobile stations. Then the three gateway discovery methods were implemented. This code, which is referred to as *AODV+*, has been contributed to the ns-2 community [36] and is free to be downloaded and used by everyone.

The presented results are averages over ten simulation runs, each with different randomly generated movement patterns. The simulation time is set to 1000 seconds. Since we are interested in studying the behavior of the network in steady state, i.e., after the transient state during which the connections are set up, the first 100 seconds of the simulation are ignored.

3.4.1 Simulation Setup

The studied scenario consists of 60 mobile stations, two gateways, two routers and two hosts. The topology is a rectangular area with 1300 m length and 800 m width. A rectangular area was chosen in order to force the use of longer routes between stations, compared to a square area with the same station density. The two gateways are placed on each side of the area; their x- and y-coordinates in meters are (200,500) and (1100,500).

Ten of the 60 mobile stations are Constant Bit Rate (CBR) traffic sources sending UDP data packets with a size of 512 bytes, to one of the two hosts, chosen randomly. The traffic sources are distributed randomly within the MANET. At the network layer *AODV+* is used as the ad hoc routing protocol, whereas DCF is used at the MAC sublayer with its default values for the contention parameters. Finally, at the physical stations use IEEE 802.11 DSSS.

A screenshot of the simulation scenario is shown in Figure 3.4. The 60 small circles represent the mobile stations. The two hexagonal stations at each side of the figure are the gateways and the four square stations are the two hosts and the two routers.

3.4.2 The Mobility Model

The mobile stations move according to an improved version of the commonly used random waypoint model. It has been shown that the original random waypoint model can generate misleading results [37]. With the improved

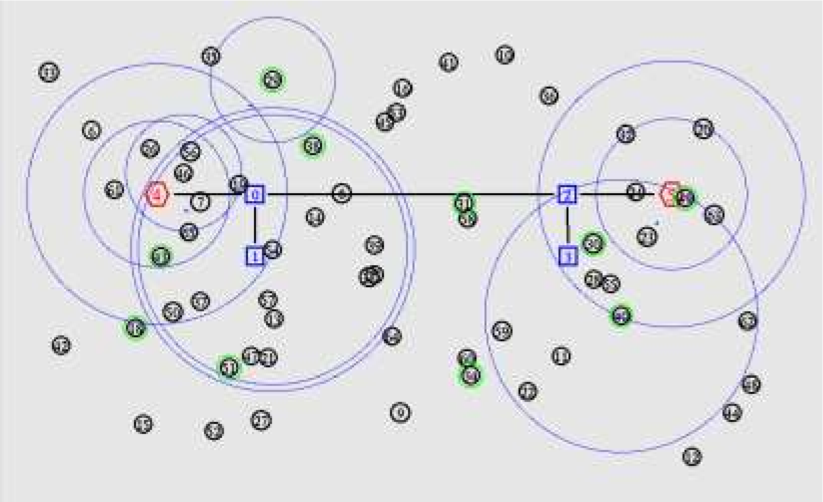


Figure 3.4: Screenshot of the simulation scenario.

random waypoint model, the mobile station speed reaches steady state after a quick warm-up period.

Each mobile station begins the simulation by selecting a random destination in the defined area and moves to that destination at a random speed. The random speed is distributed uniformly in the interval [1,19] m/s. Upon reaching the destination, the mobile station pauses for ten seconds, selects another destination, and proceeds as described. This movement pattern is repeated for the duration of the simulation. The movement patterns are generated using the movement generator tool *setdest* and the traffic connection pattern is generated by the traffic generator *cbrgen*.

If the gateways use proactive or hybrid gateway discovery, they broadcast GWADVs periodically every ADVERTISEMENT_INTERVAL seconds. ADVERTISEMENT_ZONE, which is set to three, is used for the hybrid gateway discovery method and defines the zone within which proactive gateway discovery is used. Outside this zone the reactive gateway discovery is used. We have summarized the simulation parameters in Table 3.1.

3.4.3 Performance Metrics

In comparing the gateway discovery approaches, the evaluation has been done according to the following three metrics:

- **The packet delivery ratio**: calculated as the number of data packets received at the destination's application layer divided by the number of data packets generated at the application layer of the source.

Table 3.1: The simulation parameters.

Parameter	Value
Topology area	1300 m × 800 m
Number of mobile stations	60
Number of traffic sources	10
Number of gateways	2
Packet size	512 bytes
Speed	[1,19] m/s
Pause time	5 s
Data rate	2 Mbps
Transmission range	250 m
Carrier sense range	550 m
Simulation time	1000 s
Warmup time	100 s
ADVERTISEMENT_INTERVAL	5 seconds
ADVERTISEMENT_ZONE	3 hops

- **The average end-to-end delay**: calculated as the time when a frame is received at the destination's application layer minus the time when the same frame was generated at the application layer of the source.

- **The AODV overhead**: calculated as the total amount of transmitted AODV messages in bytes divided by the sum of the transmitted AODV messages plus the data packets in bytes.

3.4.4 Simulation Results

In all figures discussed in this section it should be noted that the term "traffic load" denotes only the data traffic that each source generates, which is ten times less than the total data traffic in the whole network. To that come also control packets sent by the data link and network layers.

Figure 3.5 shows the impact of the advertisement interval on the average end-to-end delay when the traffic load changes for the proactive gateway discovery method. It can be observed that the curve representing the advertisement interval of one second differs greatly from other curves representing higher advertisement intervals. The reason is that a very short interval leads to a lot of advertisements and thus a lot of overhead, which in turn means many collisions, retransmissions and route discoveries that increase the end-to-end delay.

Figures 3.6, 3.7, and 3.8 show the packet delivery ratio, the average end-

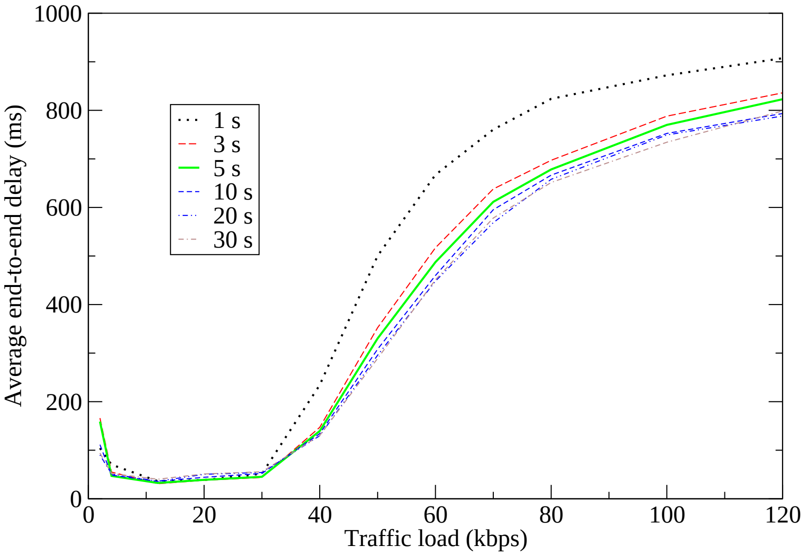


Figure 3.5: The impact of advertisement interval.

to-end delay, and the AODV overhead respectively for the three gateway discovery methods when the traffic load increases.

Packet losses occur frequently due to many reasons, e.g., when a source sends packets along a path with a recently broken link but the source has not been informed of that yet; or when a source has no other stations within its transmission range (i.e., the station is isolated) for some time and its outgoing buffer is full. Since the data packets use the connection-less UDP, which provides an unreliable delivery service, high packet losses may occur.

As Figure 3.6 shows, the packet delivery ratio is high when the traffic load is light but decreases when the traffic increases. This result is expected but it can also be seen that increasing the traffic affects all three approaches pretty much the same way. One can also see that the delivery ratio is somewhat lower for very light loads (5 kbps/source) compared to light loads (20 kbps/source). The reason for this is that once a connection has been established, it is not fully used when the traffic is very light. Therefore, only a few packets are sent before the connection breaks (e.g., due to mobility) and a new route must be discovered.

Figure 3.7 shows that the average end-to-end delay increases as expected when the traffic load increases, since increased traffic load means more collisions, retransmissions and route discoveries. We can also see that the

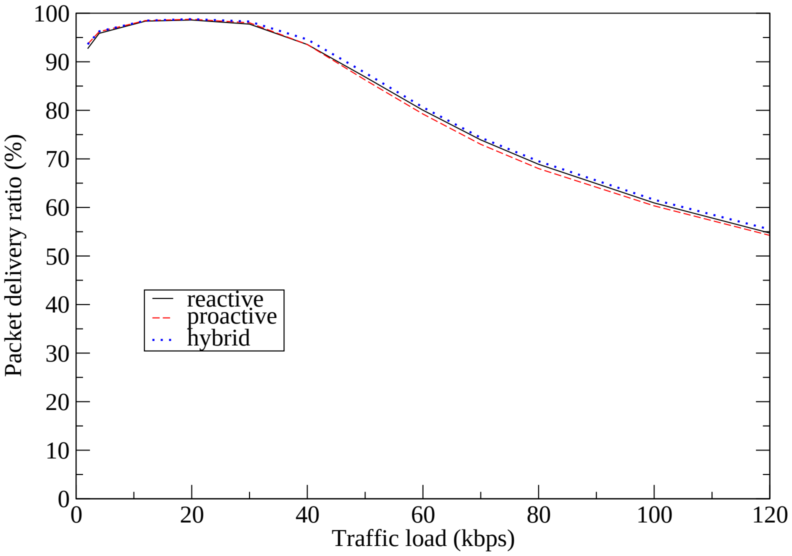


Figure 3.6: Packet delivery ratio vs. traffic load.

difference between the different strategies is negligible.

One might have expected that the delivery ratio and the average end-to-end delay would have been different for the reactive compared to, e.g., the proactive method. From one point of view, the reactive method should perform better since it generates less overhead, which should cause less number of collisions. On the other hand, the reactive method should perform worse because it does not send periodic advertisements, which would have given shorter routes (in terms of number of hops) in the long term. Since a number of other aspects need to be taken into account, it is our belief that the given scenario and the assumptions made for the simulation have a significant impact on the results.

There are some problems with the Address Resolution Protocol (ARP) implementation in ns-2, which is based on the Berkeley Software Distribution (BSD) implementation of ARP [38], that have negative impacts on our results. Each station has an ARP queue that can hold only one packet for each destination while requesting the MAC address of the next hop. If other packets arrive to the queue before the MAC address is resolved, all but the last one will be dropped [39]. This can lead to loss of important messages from upper layers, such as RREP or RREP_I messages from AODV. Consequently, if the source does not receive any RREP or RREP_I before its timer

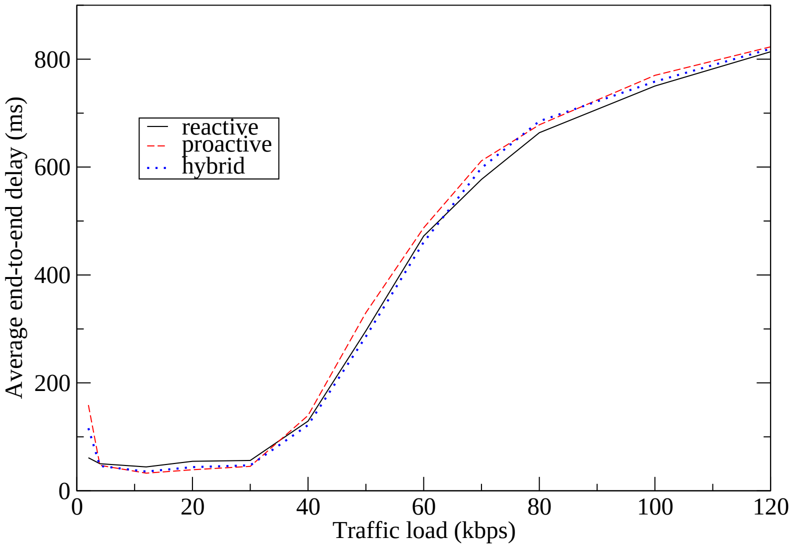


Figure 3.7: Average end-to-end delay vs. traffic load.

expires, it has to re-attempt its gateway discovery process where the reply could be lost again. It should be noted that these replies can be dropped by ARP on each hop between the gateway and the source where an address resolution process is started. In the worst case, the source will give up after some attempts and the session is aborted. Increasing the buffer size of ARP can prevent situations like this to occur.

There is another problem, where ARP is involved, which cannot be solved by increasing the buffer size. Since there is no timer involved in the address resolution process, a retransmission will not occur until it is triggered by a new incoming packet. This can have a significant impact on the end-to-end delay. Suppose that a data packet is sent to ARP from the routing protocol and that the address resolution fails because of some reason (e.g., collision). Before a new data packet is sent to ARP to trigger a new ARP Request transmission, the routing protocol changes its route towards the destination (with a new next hop) and, hence, no MAC address resolution is needed for the old next hop anymore. So far there is no problem except that the old data packet remains in the ARP queue. If much later, the station needs to resolve the MAC address of the old next hop and the ARP resolution succeeds, the data packet waiting in the queue will be sent to the next hop resulting in a very high end-to-end delay. Increasing the buffer size will in fact only make the problem even worse since then there are more than a single data packet that will be delivered to the next hop with a very long end-to-end delay. This problem could be solved by periodically purging the ARP queue.

Finally, the lack of retransmissions means that one single loss of an ARP Request or an ARP Reply means that the data packet (e.g., RREP_I) cannot be sent to the source, which will be forced to re-attempt its gateway discovery process.

The first problem caused by ARP has been investigated in [40], which shows that increasing the ARP buffer size makes the situation much better (although another solution is preferred). The second problem is discussed in [41], which suggests a cross-layer feedback mechanism from MAC to ARP.

Another thing that affects the simulation results in a negative way is when sources become isolated from the MANET such that they cannot reach any gateway. Isolated sources result in decreased packet delivery ratio and increased end-to-end delay.

In Figure 3.8 the AODV overhead is dominated by the periodically broadcasted GWADV messages. As the figure shows, the AODV overhead is significantly larger for the proactive approach than for the reactive approach, especially for light traffic loads. This result is expected since the proactive approach periodically broadcasts gateway information, no matter if the mobile stations need them or not, whereas the reactive approach broadcasts gateway information only when a mobile station needs it. Moreover, the figure shows that the overhead of the hybrid approach, which is a mixture of both the proactive and the reactive approach, is between the two other methods.

3.5 Conclusion

We have presented a solution for Internet access for mobile stations in a MANET. The MANET routing protocol AODV has been extended to route packets between a wireless MANET and the wired Internet. To achieve this, we need devices that are able to communicate with both the MANET and the wired Internet. As all communication between the wireless and the wired network must pass through these devices, they are referred to as gateways. In this thesis, three methods for detection of these gateways have been presented, implemented and compared. The three methods for gateway detection are referred to as reactive, proactive and hybrid gateway discovery. When it comes to end-to-end delay and packet delivery ratio, the three methods show surprisingly similar behavior. The fact that the proactive method shows much higher overhead in terms of control packets than the other methods is more obvious.

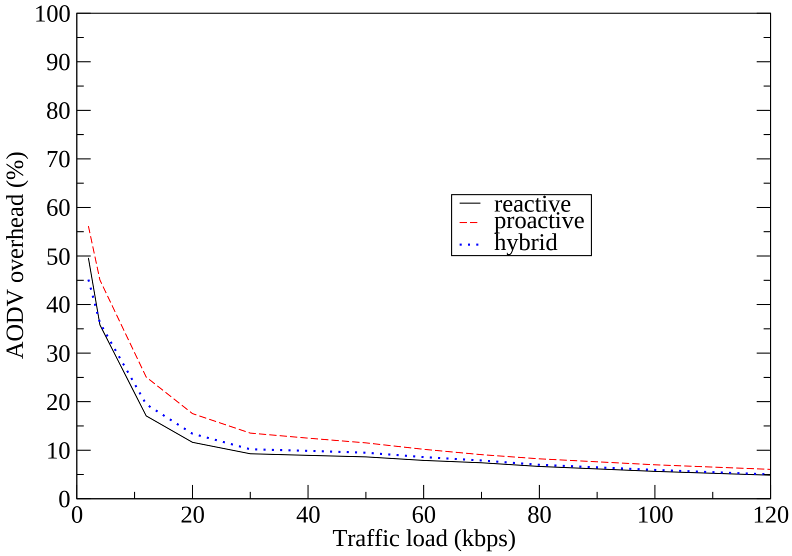


Figure 3.8: AODV overhead vs. traffic load.

Thus, we have demonstrated the ability of *AODV+* providing Internet access to mobile nodes in a MANET. By contributing the code to the ns-2 community, we made it possible to be used by many research groups to further study the interconnection between wireless MANETs and the wired Internet.

Appendix A - Implementation of *AODV+*

The following sections we describe some details about the implementation of *AODV+*. The AODV-related files in the ns-2 distribution are aodv.{h,cc}, aodv_packet.h, aodv_rqueue.{h,cc}, aodv_rtable.{h,cc}, and aodv_logs.{h,cc}. The main code is implemented in aodv.cc and the functions are declared in aodv.h. The AODV message formats (RREQ, RREP, RERR, RREP-ACK, and HELLO) are defined in aodv_packet.h. Moreover, the new GWADV message format has been added and defined in this file.

Here, we explain the main modifications, which have been done in aodv.cc. The functions are explained in a logical order: when a mobile station is to send a data packet to a destination, it tries to find a route to the destination (rt_resolve). If the mobile station does not have any valid route to the destination it broadcasts a RREQ message (sendRequest). The RREQ message is eventually received by the destination or another station that knows a route to the destination (recvRequest). The station sends a RREP/RREP_I message back to the originator of the RREQ (sendReply). The originator of the RREQ receives the RREP/RREP_I message (recvReply) and starts sending data packets to the destination (find_send_entry if the destination is an Internet host).

Modifications in aodv.cc

- **void AODV::rt_resolve**
  This function is invoked in two situations. First, when a mobile station is to send a data packet and second, when an intermediate station receives a data packet, which it must forward towards its destination.

  If the function is invoked by a (source) mobile station that wants to send data packets to some destination, there is a check to determine if the destination is an Internet host and the route to the Internet host is invalid. If this is the case, the source broadcasts a RREQ_I message to find a route to a gateway; otherwise, the mobile station acts as described in the AODV RFC without any modifications.

  If the function is invoked by an intermediate station, which has received a data packet that must be forwarded, the packet is processed differently depending on if the intermediate station is a mobile station or a gateway. If the intermediate station is a mobile station and it has a default route, the data packet is destined for an Internet host. Therefore, the intermediate mobile station updates its route entry for the Internet host and forwards the packet towards the gateway. On the other hand, if the intermediate station is a gateway, it has received a data packet from an Internet host destined for a mobile station. Consequently, the gateway broadcasts a RREQ message to discover a route to the destination.

- **void AODV::sendRequest**
  This function is invoked when a mobile station needs to find a route to another station by broadcasting a RREQ. The RREQ is broadcasted according to the expanding ring search algorithm described in Section 2.3.1. However, when a RREQ has been broadcasted through the whole network, i.e., when a mobile station has done a network-wide search, without receiving any corresponding RREP, it assumes that the destination station is an Internet host located on the Internet. First, the mobile station updates its route entry for the Internet host. Then, it checks for buffered packets destined for the Internet host. In case there are such packets, they are forwarded towards the gateway.

This function is also invoked when a mobile station needs to find a route to a gateway by broadcasting a RREQ_I. The RREQ_I is broadcasted in the same way as a RREQ message. A mobile station needs to find a route to a gateway when it detects a link break and the destination of the route is an Internet host.

- **void AODV::recvRequest**
  This function is invoked when a station receives a RREQ or a RREQ_I message. The message is processed differently depending on if the station is a mobile station or a gateway. If the station is a mobile station, the code runs without any modifications, i.e., the station tries to sends a RREP back to the originator of the RREQ message. However, in case the station is a gateway, a RREP_I is sent back to the originator of the RREQ message.

- **void AODV::sendReply**
  This function is invoked by a station that has received a RREQ or a RREQ_I message and either it is the destination or it has a valid route to the destination. The function just sends back a RREP or RREP_I message (depending on the value in the “flag” field) to the originator of the RREQ or RREQ_I. See also the comments on the recvRequest function.

- **void AODV::recvReply**
  This function is invoked when a mobile station receives a RREP or a RREP_I message. The message is processed differently depending on if it is a RREP or a RREP_I. If the message is a RREP, the code runs without any modifications. However, if the message is a RREP_I the mobile station saves the address of the gateway and creates or updates the route entry for the default route with the address of the gateway as the next hop. If the mobile station already has a route to another gateway than the originator of the newly received RREP_I message, it performs gateway selection with the number of hops to the gateways as metric.

  Then the mobile station checks if it has any packets queued in its buffer destined for an Internet host. If such packets exist and there exists a valid default route, all packets queued in the buffer are forwarded towards the gateway.

- **rt_entry* AODV::find_send_entry**
  This function is invoked when a station needs to find the correct next hop towards an Internet host. The function searches the routing table and returns the correct next hop. This is needed since default routes, which are not valid next hops, have been introduced.

Table 3.2: Default values for some important parameters associated with the operations of AODV.

Parameter	**Value**
MY_ROUTE_TIMEOUT	10 s
ACTIVE_ROUTE_TIMEOUT	10 s
REV_ROUTE_LIFE	6 s
BCAST_ID_SAVE	6 s
NETWORK_DIAMETER	30 hops
NODE_TRAVERSAL_TIME	30 ms
RREQ_RETRIES	2
MAX_RREQ_TIMEOUT	10 s
TTL_START	1
TTL_INCREMENT	2
TTL_THRESHOLD	7

As an example, take a look at Figure 3.3 in Section 3.3.4. If this function is not invoked, the packets destined for FS will be sent to next hop “DEFAULT”, which is defined as a constant in aodv.h. Since there is no station with this address, the packets will be dropped. Therefore, this function is invoked to find the correct next hop, i.e., the intermediate mobile station in this example. Hence, the packets are forwarded to the intermediate mobile station, which forwards them to the gateway, which in turn, forwards them towards the Internet host.

Modifications in aodv.h

The default values for some important parameters associated with the operation of AODV are specified in aodv.h. The parameters and their default values are presented in Table 3.2 and described below. These values do not always match the default values given in the AODV RFC.

- MY_ROUTE_TIMEOUT: the value copied into the Lifetime field of RREPs generated by destination stations. The Lifetime field specifies the expiration or deletion time of the route.
- ACTIVE_ROUTE_TIMEOUT: the lifetime of an active route.
- REV_ROUTE_LIFE: the lifetime of a reverse route created when an intermediate mobile station receives a RREQ originated by another mobile station. There is no such parameter in the AODV RFC; instead, it is calculated according to a simple formula.

Table 3.3: Some important parameters associated with the gateway operation.

Parameter	**Value**
DEFAULT	-10
ALL_MANET_GW_MULTICAST	-20
GWINFO_LIFETIME	10 seconds
ADVERTISEMENT_INTERVAL	5 seconds
ADVERTISEMENT_ZONE	3 hops

- BCAST_ID_SAVE: the time the RREQ ID should be saved. After BCAST_ID_SAVE seconds, the RREQ ID is deleted. The RREQ ID is stored in order to prevent duplicated RREQs and GWADVs being forwarded. This parameter is replaced by PATH_DISCOVERY_TIME in the AODV RFC.

- NETWORK_DIAMETER: the maximum value for the TTL field in the IP header of the RREQ message. A RREQ dissemination with TTL equal to NETWORK_DIAMETER is referred to as a network-wide search. This parameter is called NET_DIAMETER in the AODV RFC.

- NODE_TRAVERSAL_TIME: the time it takes for a station to process a packet.

- RREQ_RETRIES: the number of times to reattempt a network-wide search before timing out for MAX_RREQ_TIMEOUT seconds.

- MAX_RREQ_TIMEOUT: the time a mobile station has to wait after doing network-wide search RREQ_RETRIES times. This parameter is not defined in the AODV RFC.

- TTL_START, TTL_INCREMENT, and TTL_THRESHOLD: the parameters used by the expanding ring search technique.

- DEFAULT: the address of the default route. The value of this parameter is chosen to be negative so it cannot be mixed with the address of a mobile station.

- ALL_MANET_GW_MULTICAST: the multicast address of all the gateways in the MANET. The value of this parameter is chosen to be negative so it cannot be mixed with the address of a mobile station.

- GWINFO_LIFETIME: the lifetime of a RREP_I sent by a gateway.

- ADVERTISEMENT_INTERVAL: the interval between two consecutive GWADV messages.
- ADVERTISEMENT_ZONE: the zone within which mobile stations receive the gateway information message. Hence, this value limits the GWADV message propagation.

Modifications in aodv_packet.h

- **struct hdr_aodv_request**
  A field for flags has been added to RREQ messages. This field is used to set the I-flag when necessary.
- **struct hdr_aodv_reply**
  A field for flags has been added to RREP messages. This field is used to set the I-flag when necessary.
- **struct hdr_aodv_advertisement**
  The GWADV is a new AODV message that is basically a RREP message extended with the GWADV ID field. Figure 3.2 illustrates the GWADV message format. This message is periodically broadcasted by gateways to advertise their addresses to mobile stations in the MANET.

Appendix B - Implementation of Gateway Discovery Methods

The following sections present the implementation of the three discovery methods examined in this chapter. The main part of the implementation has been done in aodv.cc.

Implementation of Proactive Gateway Discovery

```
void AODV::sendAdvertisement() {
  /*
    Only gateways broadcast GWADV messages
  */
  if(index != thisnode->base_stn()) {
    //I’m not gateway; return
    return;
  }

  //Allocate a GWADV message
  Packet *p = Packet::alloc();
  struct hdr_cmn *ch = HDR_CMN(p);
  struct hdr_ip *ih = HDR_IP(p);
  struct hdr_aodv_advertisement *ad = HDR_AODV_ADVERTISEMENT(p);

  //Fill in the GWADV message
  ad->ad_type = AODVTYPE_ADVERTISEMENT;
  ad->ad_hop_count = 1;
  seqno++;
  if(seqno%2) seqno++;
  ad->ad_dst_seqno = seqno;
  ad->ad_src = index;
  ad->ad_lifetime = (1 + ALLOWED_HELLO_LOSS) * (u_int32_t)
    ADVERTISEMENT_INTERVAL;
  ad->ad_bcast_id = ad_bid++;

  ch->ptype() = PT_AODV;
  ch->size() = IP_HDR_LEN + ad->size();
  ch->iface() = -2;
  ch->error() = 0;
  ch->addr_type() = NS_AF_NONE;
  ch->prev_hop_ = index;

  ih->saddr() = index;
  ih->daddr() = IP_BROADCAST;
  ih->sport() = RT_PORT;
  ih->dport() = RT_PORT;
  //The GWADV is flooded through the whole MANET
  ih->ttl_ = NETWORK_DIAMETER;

  Scheduler::instance().schedule(target_, p, 0.0);
}
```

Implementation of Reactive Gateway Discovery

```
void AODV::sendRequest(nsaddr_t dst, u_int8_t flag) {

  //Allocate a RREQ message
  Packet *p = Packet::alloc();
  struct hdr_cmn *ch = HDR_CMN(p);
  struct hdr_ip *ih = HDR_IP(p);
  struct hdr_aodv_request *rq = HDR_AODV_REQUEST(p);
  aodv_rt_entry *rt = rtable.rt_lookup(dst);
  assert(rt);
```

```
/*
  Return if
  1. route is up
  2. RREQ_I has already been sent
  3. network-wide search has been done three times

  rt_req_cnt is the number of times we did network-wide
  search. RREQ_RETRIES is the maximum number we will
  allow broadcasting RREQs before going to a long timeout.
*/
if(rt->rt_flags == RTF_UP) {
  assert(rt->rt_hops != INFINITY2);
  Packet::free((Packet *)p);
  return;
}

if(rt->rt_req_timeout > CURRENT_TIME) {
  Packet::free((Packet *)p);
  return;
}

if((rt->rt_req_cnt > RREQ_RETRIES)) {
  rt->rt_req_timeout = CURRENT_TIME + MAX_RREQ_TIMEOUT;
  rt->rt_req_cnt = 0;
  Packet *buf_pkt;
  while ((buf_pkt = rqueue.deque(rt->rt_dst))) {
    drop(buf_pkt, DROP_RTR_NO_ROUTE);
  }
  Packet::free((Packet *)p);
  return;
}


//... OMITTED CODE NOT RELEVANT TO GATEWAY DISCOVERY ...//


//Determine the TTL to be used this time.
if(rt->rt_last_hop_count < INFINITY2) {
  rt->rt_req_last_ttl =
     max(rt->rt_req_last_ttl, rt->rt_last_hop_count);
}
```

```
if (0 == rt->rt_req_last_ttl) {
  //First time query broadcast
  ih->ttl_ = TTL_START;
}
else {
  //expanding ring search
  if (rt->rt_req_last_ttl < TTL_THRESHOLD)
    ih->ttl_ = rt->rt_req_last_ttl + TTL_INCREMENT;
  else {
    //network-wide broadcast
    ih->ttl_ = NETWORK_DIAMETER;
    rt->rt_req_cnt += 1;
  }
}

//remember the TTL used for the next time
rt->rt_req_last_ttl = ih->ttl_;

//PerHopTime is the roundtrip time per hop for route
//requests. Also note that we are making timeouts to
//be larger if we have done network wide broadcast before.
rt->rt_req_timeout = 2.0 * (double) ih->ttl_ * PerHopTime(rt);
if (rt->rt_req_cnt > 0)
  rt->rt_req_timeout *= rt->rt_req_cnt;
rt->rt_req_timeout += CURRENT_TIME;

//Don’t let the timeout to be too large, however ...
if (rt->rt_req_timeout > CURRENT_TIME + MAX_RREQ_TIMEOUT)
  rt->rt_req_timeout = CURRENT_TIME + MAX_RREQ_TIMEOUT;
rt->rt_expire = 0;

//Fill in the RREQ message
ch->ptype() = PT_AODV;
ch->size() = IP_HDR_LEN + rq->size();
ch->iface() = -2;
ch->error() = 0;
ch->addr_type() = NS_AF_NONE;
ch->prev_hop_ = index;

ih->saddr() = index;
ih->daddr() = IP_BROADCAST;
```

```
  ih->sport() = RT_PORT;
  ih->dport() = RT_PORT;

  rq->rq_type = AODVTYPE_RREQ;
  rq->rq_hop_count = 1;
  rq->rq_bcast_id = bid++;
  rq->rq_dst = dst;
  rq->rq_dst_seqno = (rt ? rt->rt_seqno : 0);
  rq->rq_src = index;
  seqno += 2;
  assert ((seqno%2) == 0);
  rq->rq_src_seqno = seqno;
  rq->rq_timestamp = CURRENT_TIME;
  //The I-flag is set for RREQ_I messages
  rq->rq_flags = flag;
  Scheduler::instance().schedule(target_, p, 0.);
}
```

Implementation of Hybrid Gateway Discovery

```
void AODV::sendReply_I() {
  /*
    Only gateways broadcast RREP_I messages
  */
  if(index != thisnode->base_stn()) {
    //I'm not gateway; return
    return;
  }

  //Allocate a RREP_I message
  Packet *p = Packet::alloc();
  struct hdr_cmn *ch = HDR_CMN(p);
  struct hdr_ip *ih = HDR_IP(p);
  struct hdr_aodv_reply *rp = HDR_AODV_REPLY(p);

  //Fill in the RREP_I message
  rp->rp_type = AODVTYPE_RREP;
  //The I-flag is set for RREP_I messages
  rp->rp_flags = RREP_IFLAG;
  rp->rp_hop_count = 1;
  rp->rp_dst = index;
  seqno++;
  if(seqno%2) seqno++;
  rp->rp_dst_seqno = seqno;
  rp->rp_lifetime = (1 + ALLOWED_HELLO_LOSS) * (u_int32_t)
    ADVERTISEMENT_INTERVAL;

  ch->ptype() = PT_AODV;
  ch->size() = IP_HDR_LEN + rp->size();
  ch->iface() = -2;
  ch->error() = 0;
  ch->addr_type() = NS_AF_NONE;
  ch->prev_hop_ = index;

  ih->saddr() = index;
  ih->daddr() = IP_BROADCAST;
  ih->sport() = RT_PORT;
  ih->dport() = RT_PORT;
  //TTL is limited in order to avoid too much advertisement
  //duplication
  ih->ttl_ = ADVERTISEMENT_ZONE;

  Scheduler::instance().schedule(target_, p, 0.0);
}
```

Chapter 4

QoS in Distributed Wireless Networks

In this era of wireless hysteria, with new wireless technologies becoming standardized at a fast rate, we can expect an increased interest for wireless networks, such as ad hoc and mesh networks. These networks all operate in a distributed manner, independent of any infrastructure or centralized device. Providing QoS in these networks is a challenging, but yet important, task mainly because there is no central device controlling the medium access. Despite this fact, distributed approaches have shown to be much more popular to implement in today's IEEE 802.11-based wireless networks. Centrally controlled medium access mechanisms are hardly implemented by the vendors. For example, the distributed medium access mechanism DCF has been implemented in all products supporting IEEE 802.11, whereas the centralized PCF has been totally ignored. Although this could be explained by PCF being optional and its poor QoS support, but still, its centralized operation is an important factor in this development. This claim is supported by the development of the centralized HCCA, which has also been ignored despite solving the QoS problems of PCF. Thus, it seems that the destiny of EDCA and HCCA will be similar to that of their predecessors, i.e., DCF and PCF. Sure enough, since the standardization of HCF, we have seen Wi-Fi MultiMedia (WMM), which is a subset of EDCA, replacing the older DCF as the dominant medium access scheme for wireless networks based on IEEE 802.11. At the same time, WMM Scheduled Access (WMM-SA), which is a subset of HCCA, has been ignored by the Wi-Fi Alliance. The most important reasons for this development are simple and fast installation for the distributed techniques and high complexity for the centralized ones. Thus, when it comes to providing QoS, we believe that any realistic proposed enhancement for IEEE 802.11 networks should be distributed and compatible with EDCA. Accordingly, we have designed a MAC scheme that has these characteristics.

In this chapter we will present our scheme, called *EDCA with Resource Reservation* (EDCA/RR), and evaluate its performance regarding QoS provisioning of real-time applications. In addition, we will present the *Distributed Deterministic Channel Access* (DDCA) scheme, which is a multihop extension of EDCA/RR and can be used in WMNs.

4.1 Related Work

There has been a lot of research on providing QoS in ad hoc networks. However, some of these suggest IEEE 802.11-incompatible solutions - complete architectures [42], cross-layer frameworks [43, 44], and other solutions based on, e.g., time division multiple access [45, 46], multiple channels [47, 48], and token passing [49, 50]. As mentioned earlier, we believe that any realistic QoS proposal must be compatible with IEEE 802.11, that is, the de facto standard for WLANs. Among the studies that are based on IEEE 802.11 and focus on distributed solutions, most proposed solutions are based on random medium access [51, 52, 53, 54, 55]; thus, they cannot provide QoS guarantees. Below we present some of these works but also some related works that aim at providing QoS guarantees through resource reservation.

4.1.1 Service Differentiation Proposals

To improve the performance of EDCA, the authors in [51] propose to adjust the value of the CW taking into account both application requirements and network conditions. The authors in [52] follow another, but similar, approach by dynamically adapting the priority class instead of the CW value also taking into account both application requirements and network conditions. The improvements of EDCA are still based on service differentiation so it is not possible to guarantee QoS.

The Extended EDCA (E^2DCA) is a distributed and dynamic bandwidth allocation scheme proposed in [53]. As the name implies, the scheme is based on EDCA and is compatible with IEEE 802.11. Using ideas from control theory, E^2DCA aims to provide delay guarantees to real-time traffic. More specifically, the goal is to drive the queuing delay experienced by each frame to a desired target delay. The scheme is shown to perform better than EDCA regarding average delay of real-time traffic and goodput of best-effort traffic. However, since E^2DCA is a scheme providing service differentiation, just like EDCA it suffers from performance degradation as the traffic load increases.

In [54], the authors present the Dynamic Contention Control (DCC) scheme, which is a modified version of EDCA with the aim to support real-time traffic in multi-hop ad hoc networks. Stations use per-hop delay estimations to dynamically adjust the contention window for each user priority.

Moreover, DCC can alleviate frame delay and jitter by generating a non-uniform random backoff timer for retransmitted frames. More specifically, a retransmitted frame with the smallest remaining time constraint and the largest residual hop count is intended to use the shortest backoff time among frames with the same number of retransmission retries. The authors show that, compared to EDCA, DCC can reduce the average packet delay and increase the amount of packets meeting the end-to-end delay requirements. However, just like previous contention-based mechanisms, the performance of DCC degrades with increasing traffic load.

A few distributed MAC schemes based on IEEE 802.11 and designed for providing QoS are studied in [55]. The schemes are classified into priority-based and fair-scheduling-based approaches. The priority-based schemes, like EDCA, provide service differentiation by allowing faster access to the channel to traffic classes with higher priority. The authors do not consider these schemes in their simulations since they are unfair: as the number of high-priority streams increase, they tend to grab the channel, preventing fair access for low-priority streams. Thus, the authors make a simple comparison between the three approaches using fair scheduling: Distributed Weighted Fair Queuing (DWFQ) [56, 57], Distributed Fair Scheduling (DFS) [58], and their own proposal Distributed Deficit Round Robin (DDRR) [59], which is based on the concept of Deficit Round Robin (DRR) [60]. In DDRR, each traffic class determines its allotted *service quantum rate* based on its throughput requirements and maintains a deficit counter of accumulated quanta. The deficit counter is decreased by the size of the transmitted frame and a traffic class can transmit only when the counter is positive. As the authors note themselves, the proposed mechanisms only provide service differentiation; none of them can guarantee QoS since they do not have any mechanism for admission control or resource allocation.

4.1.2 Resource Reservation Proposals

Even though there are many proposals for QoS provisioning, most of them provide service differentiation only. Among the few studies that have the potential to provide QoS guarantees, are based on IEEE 802.11, and offer distributed solutions, we can mention the Distributed Reservation Request Protocol (DRRP) [61], which is a decentralized MAC scheme based on EDCA. Whenever a station (A) needs to reserve medium access for communication with another station (B), it sends a data frame containing reservation request information. Upon reception of such a reservation request, B sends an ACK frame that also contains information about the reservation request. The ACK frame is overheard by the neighbors of B; thus, the 2-hop neighborhood of A is also informed of the reservation request. The reservation request includes information regarding the duration and repetition interval of the next transmission. All neighbors receive the reservation information by overhearing the transmissions between A and B. However, since the neighbors do not acknowledge the overheard frames and these can be lost, the reservation request is transmitted periodically so the information about its periodicity is also included in the reservation request. The scheme is similar to our scheme EDCA/RR (which will be presented in Section 4.2), allowing stations to reserve access to the medium. However, as opposed to EDCA/RR, DRRP has no distributed admission control mechanism, cannot handle reservation collisions caused by uninformed stations (see Section 4.2.4) that lie outside the transmission range of both the transmitter and the receiver, and requires applications to specify in advance how many reservation slots they need. Finally, although multi-hopping is one of the advantages of DRRP, there is no mechanism for the routing protocol to consider the QoS requirements of the requested reservation during the route discovery process. Thus, the routing protocol might very well find a route that cannot support the requested service. On the other hand, a multi-hop extension of EDCA/RR for IEEE 802.11s [62] wireless mesh networks, which are based on EDCA as well, could easily collaborate with the mesh routing protocol operating at the MAC sublayer just like EDCA/RR.

The Distributed end-to-end Allocation of time slots for REal-time traffic (DARE) [63] is another distributed MAC scheme that allows stations to reserve periodic time slots. In particular, DARE extends the RTS/CTS reservation concept of IEEE 802.11 DCF to a multi-hop end-to-end perspective. To reserve resources for a real-time flow over several hops, the routing protocol at the source must first find a route to the destination. The route is assumed to be symmetric. Once such a route is established, the source sends a Request-To-Reserve (RTR) frame, which includes the requested duration and periodicity of a time slot as well as the address of the destination. When an intermediate station receives the RTR frame, it checks whether the request is conflicting with already existing reservations. If the intermediate station can make the requested reservation, it processes the RTR frame and forwards it; otherwise, the request is rejected. Once the destination receives the RTR frame, it responds with a Clear-To-Reserve (CTR) frame. When the source receives the CTR frame, it can start transmitting real-time traffic at the next reserved interval. DARE is also able to repair and release reservations. One of the main disadvantages with DARE is the very complex and inefficient method for multiple reservations. A requested reservation may conflict with existing ones so stations might have to re-schedule reservations and send messages back and forth trying to find a suitable reservation slot; this can happen at every hop between the source and the destination! The authors mention that slot shifting becomes necessary more frequently as the number of reservations increases. Thus, new reservations can only be admitted if they can squeeze in between existing ones and the problems get worse as the network grows. Obviously, this is a scalability problem. As opposed to EDCA/RR, DARE is based on DCF instead of EDCA, has no distributed admission control mechanism, cannot handle uninformed stations and reservation collisions but relies on real-time applications being robust to packet loss, wastes resources by transmitting dummy packets during silent periods to prevent a false reservation release instead of having an explicit reservation deletion process, and last but not least, has a very complex and inefficient method for multiple reservations. Similar to DRRP, DARE supports multi-hopping but since the routing protocol does not take into account the QoS requirements, the discovered route might very well not support the requested service.

As opposed to the previous work, in this thesis we propose a MAC scheme that i) is based on EDCA and is *compatible* with IEEE 802.11; ii) operates in a *fully distributed* manner offering distributed admission control, scheduling, and medium access; iii) provides *QoS guarantees* by allowing applications with strict QoS requirements to reserve TXOPs for contention-free medium access; iv) provides all the existing favorable features of EDCA (which performs very well during light traffic load), i.e., in addition to *contention-free* medium access and *parameterized QoS*, it also provides *contention-based* medium access and *prioritized QoS*; and v) offers a solution that *handles uninformed stations* that lie outside the transmission range of both the transmitter and the receiver. In other words, EDCA/RR is a *realistic* approach that can be implemented into existing wireless systems to *fill the gap* between the distributed but contention-based EDCA/WMM and the contention-free but centralized and ignored HCCA/WMM-SA.

4.2 EDCA with Resource Reservation

In the beginning of this chapter, we argued for distributed medium access mechanisms. We discussed that the distributed EDCA is being implemented by the majority of the vendors, replacing the older DCF as the dominant medium access mechanism. At the same time, it seems the centralized HCCA is being neglected just as PCF; despite the fact that HCCA is a great improvement compared to its predecessor. This development was explained by simple and fast installation for the distributed techniques and high complexity for the centralized ones. As a result of this reasoning, we claimed that any realistic MAC proposal for IEEE 802.11-based networks should be distributed and compatible with the new EDCA standard for medium access. The problem with EDCA is, however, that it can provide service differentiation only, whereas we would like to have a solution that can provide QoS guarantees as well. Hence, the motivation of our work is to find a *dis-*

tributed QoS solution that offers both *contention-based* and *contention-free* medium access. In other words, we would like to combine the advantages of EDCA, being distributed, with the advantages of HCCA, being able to provide QoS guarantees through resource reservation. To achieve this, our strategy was to incorporate the favorable features of HCCA into EDCA, resulting in EDCA with Resource Reservation (EDCA/RR). EDCA/RR provides all existing features of EDCA and, in addition, gives applications with hard QoS requirements the possibility to reserve transmission time for guaranteed medium access. In other words, EDCA/RR provides both *prioritized and parameterized QoS*.

4.2.1 Distributed Scheduling and Admission Control

As mentioned earlier in Chapter 2, the IEEE 802.11e standard amendment provides guidelines for the design of a simple scheduler and ACU. Since the scheduler and ACU are rather simple, there have been efforts proposing more sophisticated, efficient, and flexible solutions.

One problem with the ACU is that it is based on the minimum physical rate, from which the actual physical rate could be quite different. As a consequence, the ACU could be somewhat conservative and pessimistic. Therefore, the Physical Rate-based Admission Control (PRBAC) [64] tries to overcome this problem by considering physical rate variance due to wireless medium characteristics and station mobility. The key point of PRBAC is to use the long-term average physical rate for admission control and the instantaneous physical rate for calculating the TXOP duration. In this way, more traffic streams can be admitted thanks to the more optimistic algorithm. However, being more optimistic can result in over-reserved resources and, consequently, packet losses. Therefore, the authors propose a simple packet-dropping method to alleviate this problem.

Another, perhaps more severe, problem with the ACU (and PRBAC, which it is based on), is that it performs well for CBR applications but not for Variable Bit Rate (VBR) applications. The reason for this is that the ACU and PRBAC only consider the mean data rate and the mean frame size. This is not suitable for VBR traffic where the instantaneous data rate and frame size can vary a lot from the corresponding mean values. Therefore, a new admission control scheme for VBR traffic is presented in [65]. The authors propose a method to calculate the TXOP duration such that it can provide statistical guarantee on the packet loss probability.

Yet another problem with the sample scheduler is that each station can only schedule TXOPs of fixed length at constant intervals. In [66], the authors extend their proposed admission control scheme in [65], by using variable SIs resulting in more admitted traffic streams.

The issue of fairness is considered in [67], proposing Fair HCF (FHCF), which is a scheduling algorithm that aims to be fair to both CBR and VBR traffic. Basically, the scheme is composed of two schedulers: the QAP scheduler and the node scheduler. The QAP scheduler estimates the queue length for each station before the next SI. This estimated value is compared to the ideal queue length and gives the estimation error, which is used by the QAP scheduler to adapt the calculation of the TXOP duration. The node scheduler has to redistribute the unused reserved time to traffic streams within the station.

The scheduler in [68], which is called Scheduling based on Estimated Transmission Times - Earliest Due Date (SETT-EDD), has extended the sample scheduler by allowing the stations to schedule TXOPs of variable length at variable SIs. SETT-EDD uses ideas from real-time scheduling theory to schedule TXOPs based on earliest deadlines to reduce delay and packet loss due to expiration.

One drawback of the sample scheduler, which has not been considered by SETT-EDD, is that the duration of the TXOP to be reserved is calculated based on average traffic rates or estimations. In [69] the calculation is instead based on actual requirements. This is achieved by using two fields with information about the queue size and the requested duration of the TXOP to be reserved.

In the current implementation of EDCA/RR, we have implemented the sample scheduler and ACU, rather than any enhanced algorithms, because we wanted to minimize their effect on the reported results. However, since EDCA/RR is not dependent on any specific scheduling or admission control algorithm, any proposed enhancement (such as those mentioned above) can be used together with EDCA/RR; perhaps with some minor modifications. Thus, EDCA/RR does not only provide distributed resource reservation, but also distributed admission control and scheduling. The distributed admission control and scheduling is realized simply by implementing the algorithms in the stations instead of in the QAP only. For this idea to hold, stations must broadcast their reservation requests to all neighbors instead of unicasting them to the QAP alone. A more detailed description of the operation of EDCA/RR and its reservation setup procedure is presented in the following sections.

4.2.2 EDCA vs. EDCA/RR

As mentioned earlier, EDCA/RR has all the existing functionalities of EDCA plus the capability to reserve resources for high-priority[1] traffic with strict QoS needs. Hence, if we prevent stations to reserve TXOPs, EDCA and

[1] In this thesis, we sometimes use the term “high-priority traffic” when referring to “traffic with QoS requirements”. In the same way, we sometimes use the term “low-priority traffic” when referring to “traffic with no QoS requirements”.

EDCA/RR work exactly the same - except for delivering important control messages. While studying EDCA, we noticed that connections between two stations could be heavily delayed. This was because important control messages such as management frames (e.g., ADDTS Requests/Responses), ARP frames (e.g., ARP Requests/Replies), routing packets (e.g., AODV RREQs/RREPs), and other important higher-layer packets, shared their transmission queue with data frames. This could result in long connection setup times due to control frames being transmitted after all queued data frames having been serviced. In order to avoid such situations, one could optionally add a new AC, especially configured for important control messages. As an alternative approach to adding a new AC, the existing ACs could be implemented such that important control messages were inserted at the front of the queue ahead of any data frames. If this optional feature is not preferred, EDCA and EDCA/RR will have exactly the same behavior when there are no reserved TXOPs. In our current EDCA/RR implementation, however, we chose to add a new AC, called AC_CO for control messages, with the same high-priority access parameters as AC_VO except for TXOP limit, which was set to zero. This is necessary in order to remove the possibility of sending more than one frame during each successful medium access, as control messages are usually transmitted one at a time.

4.2.3 Basic Reservation Setup

When a station determines that it needs to reserve TXOPs for one of its high-priority traffic streams, it requests admission for its traffic stream. More specifically, when the first frame of a traffic stream with QoS requirements reaches the MAC sublayer, the admission control algorithm checks whether the traffic stream can be admitted. This check is done locally within the station and not by sending a message to a central device, such as a QAP. In case the traffic stream is rejected, the application can either try to lower its QoS demands or fall back to EDCA for contention-based medium access. However, if the traffic stream can be admitted, the reserving station broadcasts an ADDTS Request to start the reservation setup process. In order to decrease the reservation delay, the ADDTS Request is sent from AC_CO.

The broadcasted ADDTS Request contains a TSPEC element with information such as nominal MSDU size, mean data rate, SI, service start time, and minimum PHY rate. This information is stored and used by the neighbors to schedule the reservation exactly as the reserving station. After storing the TSPEC information, the neighbors send an ADDTS Response back to the reserving station.

Usually, when an IEEE 802.11 frame is broadcasted, stations do not start a timer to wait for a response. However, since we want to make sure that all neighbors receive information about the TXOP reservation, we need a mechanism for reliable broadcasting. To realize such a mechanism, the reserving station starts a timer when the ADDTS Request is broadcasted and waits for all neighbors to send their ADDTS Responses. If the reserving station does not receive ADDTS Response from all neighbors, the timer will expire and the ADDTS Request will be retransmitted, up till a specified number of times, e.g., RESERVATION_RETRIES times. With n neighbors and r received ADDTS Responses, the time a station has to wait before retransmitting the ADDTS Request is calculated as follows:

$$\begin{aligned} timeout &= txtime\{ADDTS\ Request\} + (n - r) \times \\ &\quad (BACKOFF\_DELAY + txtime\{ADDTS\ Response\} + \\ &\quad txtime\{ACK\}), \end{aligned}$$

where BACKOFF_DELAY = CWmax × aSlotTime and txtime{frame f} is the transmission time of frame *f*. Note that we must give the neighbors a chance to respond to the ADDTS Request, so, in addition to the transmission time of the ADDTS Request, the ADDTS Responses and their associated ACKs, we must also consider the backoff delay of the neighbors.

When the reserving station receives an ADDTS Response, it stores the address of the neighbor. Once it has received an ADDTS Response from all neighbors, the reservation setup is done, i.e., the reserving station has successfully reserved periodic TXOPs with a specified duration every SI. The first TXOP starts at the requested service start time, whereas the rest continue periodically every SI:

$$TXOP_n = service\ start\ time + n \times SI, \quad n = 0, 1, 2...$$

If the time when all responses are received has already passed the requested service start time, the station waits until the next TXOP to start its transmission. During the reserved TXOPs, the station can access the medium without needing to start a backoff procedure, knowing that the neighbors will refrain from transmitting. In other words, the station has contention-free access to the medium. The MSDUs belonging to a traffic stream with reserved TXOPs are not allowed to be transmitted at time instants other than during the reserved TXOPs.

Multiple reservations are managed rather easily by EDCA/RR; Figure 4.1 illustrates a case where there are two existing TXOP reservations, whereas a third is being setup. When a traffic stream finishes and has no more frames to send, it can broadcast a Delete Traffic Stream (DELTS) frame notifying other stations to delete the TXOP reservation belonging to that traffic stream and to reschedule the TXOPs of any remaining traffic stream.

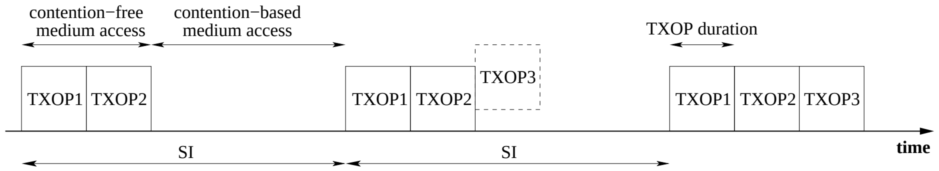


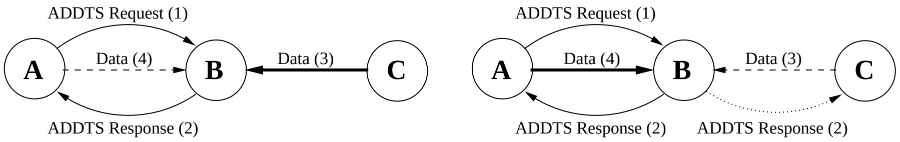
Figure 4.1: Scheduling multiple TXOP reservations.

Whereas some QoS demanding applications are somewhat robust to packet loss and do not need to retransmit lost frames, others might benefit from it. On that ground, EDCA/RR supports retransmitting lost high-priority data frames as an optional feature, and lets the application decide whether retransmission should be used. For example, the delay bound of an application might be such that a frame can be retransmitted and still not exceed the delay bound. In our current implementation, high-priority data frames lost within a TXOP are handled immediately by retransmitting the frames after SIFS duration, in case the remaining time in the TXOP is enough for the retransmission. Thus, it is not necessary to start a back-off procedure after every transmission failure. To support retransmission of high-priority frames, the scheduler must calculate a sufficiently long TXOP duration.

4.2.4 The Uninformed Station Problem

In the original version of EDCA/RR, there was no mechanism to handle the *uninformed station problem*. An uninformed station is a station that is not informed of the existing TXOP reservations in the network. The problem that could occur is that uninformed stations could cause *reservation collisions*. Stations that are informed of the existing reservations in the network do not start a transmission unless it finishes before a TXOP starts. But unfortunately, uninformed stations do not receive any ADDTS Request containing a TSPEC element, which contains information about the reservation in progress, so they will stay uninformed of the TXOP reservations. Therefore, they might start a transmission that extends across a reserved TXOP and collides with the reservation.

To illustrate the problem caused by uninformed stations, suppose there are three stations in a row (see Figure 4.2a): A, B and C, where A and B as well as B and C are within each other's transmission range, but A and C cannot hear each other. Assume further that A is about to send high-priority traffic to B so it has broadcasted an ADDTS Request (1) and B has replied with an ADDTS Response (2). However, C is not informed of A's TXOP reservation since it has not received A's ADDTS Request. Thus, there is a chance that C starts transmitting just before a TXOP reserved by A is



(a) Station C is uninformed

(b) Station C is informed

Figure 4.2: (a) C is not informed of a A's TXOP reservation so it can start transmitting just before a reserved TXOP of A is about to start; (b) C is informed of the TXOP reservation of A and defers during A's reserved TXOPs.

about to start (3). In that case a collision would occur during A's reserved TXOP (4), meaning that A would no longer have collision-free access to the medium. In order to prevent these kinds of situations, C must be informed of A's TXOP reservation.

4.2.4.1 Solution by Overhearing

There are different ways of achieving the goal of spreading the TSPEC to stations outside the reserving station's transmission range. One possible approach is to rebroadcast the ADDTS Request sent by the reserving station during the reservation setup process. Hence, in our example, B would rebroadcast the ADDTS Request of A to let also C receive the request containing the TSPEC. However, there are many problems related to this approach. One question that comes to one's mind is whether C should send an ADDTS Response to B just like B has to send an ADDTS Response to A. Before answering this question, we must consider that A might have more 1- and 2-hop neighbors; let us call them $B_1, B_2, ..., B_n$ and $C_1, C_2, ..., C_n$. This means that if C has to respond to B, then every other station two hops away from A $(C_1, C_2, ..., C_n)$ should also respond to B because those are also uninformed stations. Moreover, this procedure would continue until all stations one hop away from A $(B_1, B_2, ..., B_n)$ rebroadcast the ADDTS Request from A, and all stations two hops away from A send back an ADDTS Response. Obviously, this would lead to a lot of overhead and a significant increase in the reservation delay.

Ruling out the option where C has to send an ADDTS Response to B, B could just rebroadcast the ADDTS Request without requiring any response back from C. In other words, we could rely on C overhearing the ADDTS Request. Although this approach would require significantly less extra signaling compared to the previous option, there is an even better approach.

Instead of rebroadcasting the ADDTS Request, we could extend the

ADDTS Response frames to include TSPEC elements and let all stations overhear the ADDTS Responses to become informed of the reservation in progress. This way, the TSPEC will be known to all stations within two hops from the reserving station, without any additional signaling and with a limited increase of overhead. Thus, in our example (see Figure 4.2b), when B receives the ADDTS Request (1), it stores the reservation information included in the TSPEC element, copies it into the TSPEC of the extended ADDTS Response, and transmits the response back to A (2). Station C will overhear this frame (2), store the information included in the TSPEC element, and refrain from transmitting frames that do not finish before the start of a reserved TXOP (3). Consequently, A can access the medium (4) knowing that stations within its 2-hop neighborhood will refrain from transmitting during its reserved TXOPs. This approach is much less complex and results in very little overhead compared to previous methods mentioned above.

4.2.4.2 Solution by Reactive Notification

Studying the uninformed station problem further, we noticed that the solution based on overhearing works fine as long as all uninformed stations lie in the transmission range of either the source or destination. However, this might not always be the case in reality. Thus, even though the goal is to inform all 2-hop neighbors of the existing reservations in the network, it is unavoidable that sometimes a station might not be able to overhear an ADDTS Response; as a consequence, there is a chance that its transmissions collide with reserved TXOPs. Here, we present a solution to this problem, which is not necessarily specific to EDCA/RR, but could also be applied to other reservation-based MAC protocols for wireless networks based on IEEE 802.11.

A TXOP owner that senses the medium busy at the start of its TXOP realizes that the ongoing transmission causing the busy medium must belong to a station that has not yet been informed of the reservation of the TXOP owner. To prevent a transmission collision, the TXOP owner may choose to skip the collided TXOP and wait until the next TXOP. Meanwhile, the station causing the collision must be notified of the reservation of the TXOP owner. In order to do that, we use those neighbors of the colliding station that are informed of the existing reservations. If the informed neighbors sense the medium busy just before a TXOP is about to start, just like the TXOP owner they realize that the transmitting station cannot be aware of the upcoming TXOP reservation. However, just by sensing a busy medium, the informed neighbors cannot know the address of the colliding station; but once the colliding transmission is finished, the neighbors can decode the frame header to get the address of the colliding station. Then, the neighbors

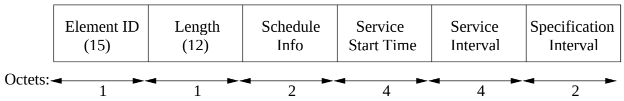


Figure 4.3: The Schedule element format.

wait until the end of the ongoing TXOP and send a *Schedule frame* to notify the colliding station about the existing reservations in the network.

The Schedule frame contains the Schedule element, which is illustrated in Figure 4.3. Both the Schedule frame and the Schedule element are defined in IEEE 802.11e. Here, we have extended the schedule element with three fields from the TSPEC element, namely, *nominal MSDU size*, *mean data rate*, and *minimum PHY rate*. These fields are necessary as they contain information about a TXOP reservation. Moreover, the Schedule element has been extended with another field, called *TXOP Owner Address*, containing the MAC address of the station whose TXOP was corrupted. This information is needed by the destination of the Schedule frame, i.e., the colliding station, so it can update its TSPEC information for the correct TXOP owner. Thus, extended with these four fields, the Schedule frame is used to reactively notify colliding stations of the TXOP reservation that was corrupted.

4.2.5 Reservation Setup - Alternative Approach

With the introduction of this mechanism solving the uninformed station problem, the reservation setup could alternatively be implemented in another simpler and faster, but less reliable way. Depending on the level of desired reliability and QoS guarantee, one approach might be preferable over another. Instead of requiring all 1-hop neighbors sending back an ADDTS Response to the reserving station, we could rely on overhearing and use Schedule frames to inform uninformed neighbors, as presented above.

Thus, if a traffic stream can be admitted, the reserving station broadcasts an ADDTS Request, but as opposed to the reliable reservation approach discussed in Section 4.2.3, it does not wait for neighbors sending back ADDTS Responses. Since this broadcasted frame does not need to be reliable, the reserving station does not need to start a timer to retransmit the ADDTS Request in case all ADDTS Responses have not been received. The TSPEC information is used by neighbors to schedule the reservation exactly as the reserving station. The reserving station assumes that the ADDTS Request has been received by all neighbors and starts transmitting at the advertised service start time. Whenever the transmission of an uninformed station, whether 1-hop or 2-hop neighbor, collides with a reserved TXOP, it will be informed of the existing reservations using the reactive notification method discussed in Section 4.2.4.2. This alternative reservation scheme will be less reliable but also more simple and have less reservation delay.

4.3 Distributed Deterministic Channel Access for Wireless Mesh Networks

Since the idea behind EDCA/RR was shown to be promising, we decided to explore the idea to use a similar concept for WMNs. This multi-hop extension of EDCA/RR is called *Distributed Deterministic Channel Access* (DDCA).

In order to provide end-to-end QoS guarantees in a wireless, multi-hop network, we need a routing protocol that is able to find a multi-hop route between two communicating devices. For MANETs, there have been lots of proposed routing protocols, e.g., AODV, DYMO, OLSR, and OLSRv2. In addition, the upcoming IEEE 802.11s [62] standard for mesh networking suggests a routing protocol called Hybrid Wireless Mesh Protocol (HWMP). As the name indicates it is a hybrid reactive/proactive routing protocol. The reactive part is called Radio Metric AODV (RM-AODV), whereas the proactive part is based on tree-based routing.

Since IEEE 802.11s will have a reactive routing protocol based on AODV, we integrated DDCA with AODV in order to provide deterministic medium access in a WMN. Later, when the standard becomes ready, it should not be difficult to combine DDCA with HWMP. In fact, since DDCA operates at the MAC sublayer, it should even be easier to make DDCA collaborate with HWMP also operating at the MAC sublayer compared to AODV operating at the network layer. Furthermore, the fact that IEEE 802.11s is based on EDCA makes the integration of EDCA/RR into IEEE 802.11s rather simple.

DDCA is a multi-hop extension of EDCA/RR. Consequently, the two schemes have many similarities. The main difference between DDCA and EDCA/RR is the reservation setup procedure. In DDCA, stations do not use ADDTS Requests or ADDTS Responses. Instead, stations use a modified version of the alternative and less reliable reservation approach (presented in Section 4.2.5), which is more suitable for multi-hop wireless networks, such as WMNs.

4.3.1 Route Discovery and Reservation Setup

Instead of first starting a route discovery process and then a reservation setup process, our idea is to combine the route discovery and reservation setup procedures. The advantage of this approach is that the route discovery is aware of the QoS requirements of the application so the reserving station will search for a route that fulfills those QoS requirements. Other advantages are faster route discovery and reservation setup and less overhead.

When the first packet of a traffic stream with QoS requirements reaches the network layer, the ad hoc routing protocol checks its routing table for a route to the destination. Since this is the first packet of a QoS-requiring traffic stream, it must trigger a route discovery/reservation setup process by generating a *Route and Reservation Request* (RRQ). Thus, the routing protocol must be modified not to return a route to the destination if that route is not known to support the requested QoS requirements. The RRQ message is a RREQ extended with some fields copied from the high-priority packet: frame size, data rate and priority class. These parameters are set by the application for the TSPEC and are used during the calculation of SI and TXOP duration. After creating the RRQ message, it is sent down in the protocol stack to the data link layer. The MAC sublayer starts the reservation setup procedure by checking if the new traffic stream can be admitted. Note that DDCA only checks whether to admit or reject the traffic stream - it does not reserve any TXOPs yet. The reason for this is that at this point, the station cannot know whether a (multi-hop) route between the source and the destination really exists - and even if such a route exists, the station cannot know whether the route has enough resources to be reserved. Each station has knowledge about the reservations of its 2-hops neighborhood, but not about the reservations of those further away. Therefore, the actual reservation is made once the destination sends a response back to the source (after receiving an RRQ indicating that a route with enough resources exists). If the admission control fails, the application might prefer an ordinary non-QoS route (not taking into account the QoS requirements of the application) to the destination than not having any route at all. Although some applications need a certain minimum level of QoS to work properly (e.g., multiplayer online games), others might be able to function tolerably despite insufficient QoS levels (e.g., VoIP applications). The preferred option shall be indicated by the application. Thus, if the traffic stream is rejected, depending on the QoS requirements of the application, either it is notified to abort because there are not enough resources to reserve or the routing protocol is notified to generate a RREQ instead.

On the other hand, if the admission control is successful, the RRQ is broadcasted as usual, i.e., as if it was a normal RREQ message. A station that receives the message, stores the information about the reservation at the MAC sublayer and then checks at the network layer if it is the destination of the RRQ. If it is not the destination, it will start the reservation setup procedure at the MAC sublayer to check whether the new traffic stream can be admitted. Once again, if the admission control fails, depending on the QoS requirements of the application, either it is notified to abort because there are not enough resources to reserve or the routing protocol is notified to search for an ordinary non-QoS route to the destination. However, if the traffic stream can be admitted, the station marks the status of the route as "admitted" and rebroadcasts the RRQ.

One question that arises here is if the rebroadcasted RRQ should contain the reservation information of the originating station or that of the intermediate station. To clarify the problem, let us assume a simple scenario with four stations in a row, similar to the scenario illustrated in Figure 4.2 but extended with another station called D. Each station is within the transmission range of its direct neighbor(s) and cannot hear stations two hops away. Moreover, we assume that A is about to send traffic with QoS requirements to D so it has broadcasted an RRQ, which B is about to rebroadcast since it is an intermediate station and not the destination. Now the question is whether B's RRQ should contain information about the reservation by the originating station A or by the intermediate station B. Since C must refrain from transmitting during the reservations of both A and B, it needs to be informed of the reservation of both stations. Therefore, the RRQ must carry the reservation information of both A and B. Similarly, when C rebroadcasts the RRQ, it will contain the reservation information of both B and C in order to prevent D from transmitting during B's and C's reservations.

The RRQ is forwarded by intermediate stations until it is received by the destination. The destination responds with a *Route and Reservation Reply* (RRP), which is an ordinary RREP message extended with information about the TXOP reservation. This message confirms the reservation request by changing the status of the route from "admitted" to "reserved". Moreover, on the way back from the destination to the source, the RRP will inform intermediate stations and the source of the reservations that have been completed in a later phase of the route and reservation discovery process. Thus, D unicasts an RRP to its next hop C, which changes the status of the route from "admitted" to "reserved". Next, C appends information about D's and its own reservation to the RRP before forwarding it to B. When B receives this RRP, it will be informed of the reservations of C and D. This way, the RRP is forwarded until it is received by A. Upon receiving the RRP, the source A becomes informed of other reservations in the network and has deterministic contention-free access to the medium. Stations that overhear the RRP will also get informed of the reservations and thus refrain from transmitting during the reserved TXOPs.

If a station along the route between the source and the destination cannot reserve the requested resources, the destination will never generate an RRP, which may result in some stations keeping the preliminary status of the route as "admitted". Moreover, it is possible that stations temporarily reserve resources during the dissemination of the RRQ, but the RRP is sent along another path. Therefore, we need a timer to release the temporary reservations. This timer is started once the status of a route is marked as “admitted”. When the timer expires, the routes that are still marked as “admitted” will be reset to release the temporary reservations.

4.4 Evaluation

In order to evaluate the performance of EDCA and EDCA/RR, we used the popular network simulator ns-2 [19]. Although many EDCA studies using ns-2 are based on the TKN EDCA model [70], our implementation of EDCA/RR is based on an accurate and detailed EDCA implementation by Mike Moreton [71] for ns-2.26. The main reason for this choice is that the TKN model is based on the legacy IEEE 802.11 implementation in ns-2, which is reported to contain many errors, some of which are still remaining in the TKN model; instead, Moreton's model is reported to be correct [72].

The presented results are averages over 100 simulation runs, each ran for 300 simulated seconds. We were interested in studying the behavior of the network in steady state, i.e., after the transient state during which the connections are set up. After some testing, we concluded that it took somewhat less than 30 seconds until all connections had been set up, so in our simulations, the first 30 seconds are ignored.

4.4.1 Simulation Setup

The simulated scenario, illustrated in Figure 4.4, consists of 25 wireless stations, a gateway, a router, an FTP server, an HTTP server, and a video streaming server. The size of the simulation area is 1000 m × 1000 m. The gateway is placed in the middle of the scene, with x- and y-coordinates in meters at (500,500) and connected to the three servers via a router on the wired network. All wireless stations can communicate directly with the gateway. The placement of the wireless stations is chosen such that they are within the interference range of each other. The transmission range is 250 m and the carrier sense range is 550 m, i.e., the default ns-2 values.

In order to evaluate the effectiveness of EDCA and EDCA/RR in protecting real-time VoIP traffic from interfering transmissions by uninformed stations, we deliberately placed four stations outside the transmission range of all wireless stations involved in VoIP communication, i.e., outside the transmission range of *both transmitters and receivers.* Throughout this text, we will refer to these stations as “uninformed stations” for simplicity[2]. More-

[2] We say for simplicity because these stations may be able to correctly overhear the ADDTS Responses from the gateway. However, since these stations rely on overhearing from one single device (the gateway), there is a significant chance that the ADDTS Responses might not be correctly overheard. Thus, these stations are in fact possible uninformed stations, but we will refer to them as uninformed stations.

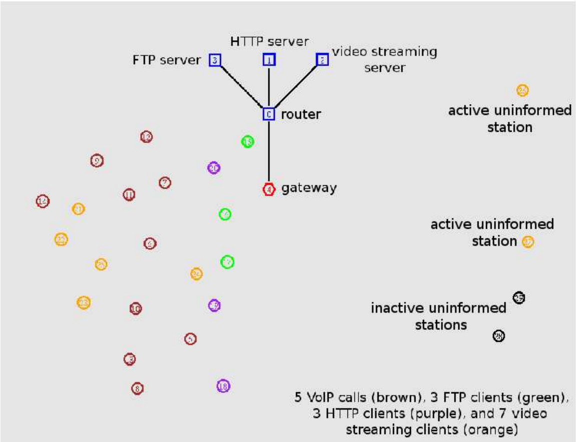

HTTP server
FTP server
video streaming server
router
gateway
active uninformed station
active uninformed station
inactive uninformed stations
5 VoIP calls (brown), 3 FTP clients (green), 3 HTTP clients (purple), and 7 video streaming clients (orange)

Figure 4.4: A snapshot of the scenario showing the case with five VoIP calls.

over, to disturb the VoIP traffic even further, we deliberately chose to use applications with different characteristics; these are modeled as follows:

- VoIP: The bi-directional VoIP traffic is modeled according to a G.711 voice codec generating 160 bytes every 20 ms, resulting in 64 kbps. This kind of traffic is given high priority and its frames are sent using AC_VO. When EDCA/RR is used, these streams reserve TXOPs for contention-free medium access.

- FTP: The FTP application represents a bulk data transfer of large size, sending TCP segments equal to 1000 bytes. The application has always something to send and runs throughout the whole simulation. FTP is given low priority and its frames are sent using AC_BE.

- HTTP: The HTTP traffic is modeled according to NSWEB/SURGE [73]. HTTP is given low priority and its frames are sent using AC_BE.

- Video: The video traffic is modeled as an H.261 video codec generating 30 frames per second; each with a size equal to 1600 bytes, resulting in 384 kbps. Video is given high priority and its frames are sent using AC_VI, but these streams do not reserve TXOPs. The fragmentation threshold for UDP packets, equal to 1000 bytes by default in ns-2, is increased to prevent video packets becoming fragmented.

In our simulations, between 0 and 12 of the 25 stations are involved in VoIP communication, that is, there are 0-6 VoIP calls[3]. The choice of

[3] The notion of a VoIP call refers to two VoIP streams, one going from station A to station B, and the other going in the opposite direction.

Table 4.1: Some important simulation parameters.

Parameter	**Value**
Topology area	1000 m × 1000 m
Number of stations	25
Number of VoIP calls	0-6 (variable)
Number of FTP clients	3
Number of HTTP clients	3
Number of video streaming clients	7
VoIP packet size	160 bytes
VoIP data rate	64 kbps
Video packet size	1600 bytes
Video data rate	384 kbps
FTP packet size	1000 bytes
Transmission range	250 m
Carrier sense range	550 m
Simulation time	300 s
Warmup time	30 s

varying the number of real-time VoIP calls was made to demonstrate the ability of EDCA/RR to handle *multiple reservations*. Three stations are downloading files from the FTP server on the Internet, whereas three others are surfing the Web; i.e., they communicate with the HTTP server. Finally, seven stations are involved in downlink video streaming sessions with the video streaming server. When there are five or six VoIP calls, the uninformed stations are active and involved in video streaming transmission. More specifically, two of the four uninformed stations are active when there are five VoIP calls, whereas all four are active when there are six VoIP calls. Figure 4.4 illustrates the case with five VoIP calls (10 VoIP stations), three FTP clients, three HTTP clients, and seven video streaming clients.

Since the operation of EDCA/RR is basically the same as that of EDCA when there are no TXOP reservations, we deliberately configured the admission control unit to allow for many reservations in order to be able to see the differences between the two schemes. The beacon interval is assumed to be 100 ms and we made 80% of the time available for TXOP reservations. The scheduler and admission control unit calculate an SI equal to 25 ms and TXOPs equal to 1.23 ms for the VoIP calls under EDCA/RR. Thus, EDCA/RR can admit $\lfloor 0.8 \times 25/1.23 \rfloor = 16$ TXOP reservations or, equivalently, eight VoIP calls. The simulation and TSPEC parameters are summarized in Table 4.1 and Table 4.2 respectively. A description of the TSPEC parameters is provided in Section 2.2.2.2.

Table 4.2: Some important TSPEC parameters used to schedule TXOPs and perform admission control for the VoIP traffic.

Parameter	**Value**
SI	25 ms
ρ	80 kbps
L	200 bytes
M	200 bytes
R	11 Mbps
O	640 μs
TXOP	1.23 ms

The VoIP and video messages are encapsulated in RTP/UDP/IP packets, whereas the FTP messages are encapsulated in TCP/IP packets. At the network layer, *AODV+* is used as the ad hoc routing protocol (see Section 3.3), using reactive gateway discovery to access the wired network via the gateway. At the MAC sublayer the stations use either EDCA or EDCA/RR, depending on the MAC scheme under evaluation. Finally, at the physical layer they use IEEE 802.11b, or more specifically HR/DSSS using the short preamble and header mode (HR/DSSS/short).

The traffic sources are started randomly between 1.0 and 1.5 s from each other, according to a uniform distribution. There is no mobility in network. The VoIP calls are made within the ad hoc network, whereas the FTP, HTTP and video streaming clients communicate with the corresponding wired server.

As discussed in the IEEE 802.11e standard amendment, the unpredictable and error-prone nature of wireless media in general and unlicensed spectra in particular, may make it impossible to provide absolute QoS guarantees. However, in a controlled environment free of external interference, it is possible to provide techniques that can provide guaranteed medium access and thus QoS guarantees [1]. Studying EDCA/RR in both error-free and lossy media, allows us to see whether it really is capable of providing true QoS guarantees in controlled environments free of external interference and how well it fulfills the task in error-prone media. Therefore, we use the error model provided by ns-2 to simulate packet loss. The simulations are run both for the case when the medium is error-free and when 5% of the packets are lost.

4.4.2 Performance Metrics

In comparing the ability of EDCA and EDCA/RR to provide QoS, the evaluation is done according to the following metrics:

- **The average end-to-end delay** together with its **99% confidence interval** and **Complementary Cumulative Distribution Function (CCDF)**: the end-to-end delay is calculated as the time when a frame is received at the destination's application layer minus the time when the same frame was generated at the application layer of the source.
- **The jitter**: calculated as the variance of the end-to-end delay.
- **The packet delivery ratio**: calculated as the number of data frames received at the destination's application layer divided by the number of data frames generated at the application layer of the source.
- **The average throughput**: calculated as the number of data bits received at the destination's application layer divided by the time the considered traffic type (VoIP, FTP/HTTP, or video) is active.

4.4.3 Simulation Results

In this section, we sometimes use the term "contending traffic" when referring to FTP, HTTP, and video traffic since these always contend for medium access as opposed to VoIP traffic, which gets contention-free medium access under EDCA/RR. Also, the results of the TCP-based traffic, i.e., FTP and HTTP, are presented together.

4.4.3.1 Average End-to-End Delay Analysis

Table 4.3a and 4.3b show the average end-to-end delay and its 99% confidence interval experienced by the VoIP calls. Regarding EDCA, both tables show that the average end-to-end delay increases to very high levels as the traffic load increases. This is a typical behavior for contention-based medium access schemes like EDCA and it is this kind of behavior that we would like to avoid. Another typical, but more advantageous, behavior for random-access schemes is that they have very low medium access delays when the network load is light. This is also shown in the tables.

An interesting observation that needs to be commented is the sharp increase of the average end-to-end delay as the number of VoIP calls increases from four to five. This is because the uninformed stations are active in video streaming transmissions when there are five or six VoIP calls. Moreover, we recall that these stations lie outside the transmission range of all wireless stations involved in VoIP communication, i.e., outside the transmission range of both transmitters and receivers. Hence, the results show that these stations have a great negative impact on EDCA.

For the contention-free EDCA/RR, on the other hand, the average end-to-end delay is rather constant when the medium is error-free, whereas we

Table 4.3: The average end-to-end delay and its 99% confidence interval of the VoIP traffic.

(a) 0% packet error

Number of	**Delay (ms)**		**Confidence interval (ms)**	
VoIP calls	EDCA	EDCA/RR	EDCA	EDCA/RR
1	5.16	12.59	(5.11, 5.21)	(12.33,12.84)
2	7.07	12.64	(6.99, 7.14)	(12.44,12.84)
3	11.33	12.61	(11.20,11.47)	(12.44,12.77)
4	17.93	12.63	(17.69,18.16)	(12.49,12.78)
5	88.81	12.60	(87.27,90.35)	(12.47,12.72)
6	93.51	12.60	(91.80,95.21)	(12.49,12.71)

(b) 5% packet error

Number of	**Delay (ms)**		**Confidence interval (ms)**	
VoIP calls	EDCA	EDCA/RR	EDCA	EDCA/RR
1	9.26	16.84	(9.01, 9.50)	(16.60,17.08)
2	14.30	17.00	(13.89,14.70)	(16.81,17.19)
3	22.41	17.16	(21.89,22.92)	(16.98,17.34)
4	32.00	17.21	(31.06,32.93)	(17.06,17.36)
5	60.98	17.53	(59.57,62.39)	(17.38,17.69)
6	77.01	18.03	(75.56,78.47)	(17.84,18.23)

see a slight increase when the medium is lossy. Moreover, we can see that EDCA/RR can handle uninformed stations since the average end-to-end delay does not increase sharply when the number of VoIP calls increases from four to five. As the results show, EDCA/RR is clearly a technique that achieves the goal of providing guaranteed medium access within the limitations of error-prone wireless media.

Let us continue to analyze the results, focusing on the average end-to-end delay for EDCA when the medium is error-free compared to when it is lossy. It is interesting to note that, when the traffic load is high (5-6 VoIP calls), the average end-to-end delay for EDCA is lower when the medium is lossy compared to when it is error-free. The main reasons for this behavior are:

a) Dropped frames are not considered in the end-to-end delay calculations. As the packet delivery ratio analysis will show in Section 4.4.3.5, more VoIP frames are dropped in lossy compared to error-free media, as one would expect. This effect becomes more notable when the traffic load is high, causing the difference of the delivery ratio between error-free and lossy media to increase significantly from below 1% to 6-7%. The low packet delivery ratio for VoIP in lossy media with high traffic load shows that the combination of lossy media and high traffic load results in, not only retransmissions, but also packet drops (UDP-based VoIP frames are dropped after four transmission attempts at

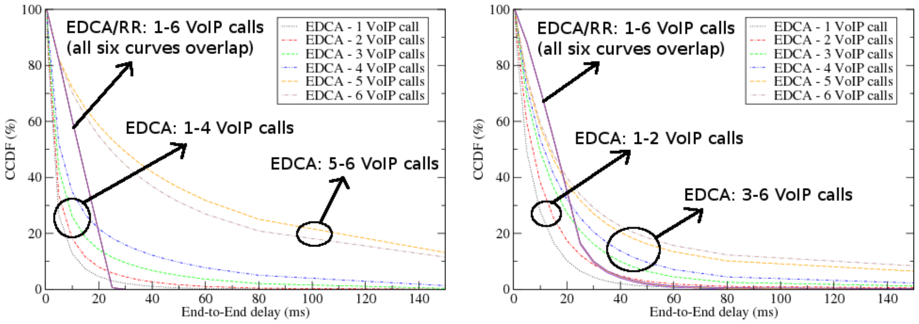
Table 4.4: Default EDCA parameter set for IEEE 802.11b PHY

AC	CWmin	CWmax	AIFSN	TXOP limit
AC_BK	31	1023	7	0
AC_BE	31	1023	3	0
AC_VI	15	31	2	6.016 ms
AC_VO	7	15	2	3.264 ms

the MAC sublayer). The dropped VoIP frames would likely have the highest end-to-end delays if they would have been received successfully. However, since they are not considered in the delay calculations, when the medium is lossy and the traffic load is high, the average end-to-end delay becomes lower for those frames that are successfully transmitted.

b) Transmission failures (as a consequence of lossy media and high traffic load) have a greater negative impact on low-priority ACs (AC_BK and AC_BE) than on high-priority ACs (AC_VI and AC_VO), thereby making the network appear less loaded to VoIP traffic when the medium is lossy and especially when the traffic load is high. This is because the CW, which is doubled after each transmission failure, becomes much larger for low-priority ACs than for high-priority ACs. As we can see in Table 4.4 (the same table presented in Section 2.2.2.1 but repeated here for the sake of easier reading), the default CWmin and CWmax values are 31 and 1023 for AC_BE (used by FTP and HTTP), and 7 and 15 for AC_VO (used by VoIP). For example, after three transmission failures, CW is equal to 127 for AC_BE and 15 for AC_VO, resulting in much higher medium access delays for FTP and HTTP traffic compared to VoIP traffic. Since high traffic load, in combination with lossy media, increases the probability of transmission failures, the result is longer and longer medium access delays for low-priority traffic, and in effect, decreased low-priority traffic load. The throughput analysis in Section 4.4.3.4 will support this claim, showing that the throughput of low-priority FTP/HTTP traffic falls much more than that of high-priority VoIP traffic, when comparing EDCA in error-free and lossy media. For example, with six VoIP calls, the VoIP throughput falls from 593 kbps in error-free medium to 543 kbps in lossy medium giving 8% throughput fall, compared to the 98% throughput fall for FTP/HTTP falling from 306 kbps to 6 kbps!

To sum up, since the frames with largest end-to-end delays are dropped and not considered in the calculations, and since the backoff mechanism in IEEE 802.11e disfavors low-priority traffic after transmission failures, the



(a) 0% packet error

(b) 5% packet error

Figure 4.5: The CCDF of VoIP End-to-End Delay.

average end-to-end delay is reported to be lower for EDCA when the medium is lossy and the traffic load is high. This behavior is not seen in EDCA/RR thanks to periodic and contention-free medium access for the VoIP calls.

4.4.3.2 CCDF Analysis

Figure 4.5 shows the CCDF of the end-to-end delay experienced by the VoIP calls in error-free and lossy media. Although the average value, confidence interval and variance of the end-to-end delay reveal useful information to us, its CCDF will add to our understanding about the delay characteristics of the two MAC schemes under investigation. For example, Figure 4.5b shows that more than 8% of the VoIP frames have an end-to-end delay over 150 ms under EDCA in lossy medium, resulting in a significant impact on the voice quality [74]; the corresponding value for EDCA/RR is less than 0.3%. Moreover, we notice that all six curves representing 1-6 VoIP calls are overlapping under EDCA/RR, implying that the performance of EDCA/RR is independent of the traffic load in the network.

Continuing to study the EDCA/RR curves, we notice a "knee point" at 25 ms both in error-free and lossy media. This value has its origin in SI, which is calculated by the scheduler. Moreover, Figure 4.5a shows that more than 99% of the VoIP frames have an end-to-end delay less than 25 ms in error-free media, which once again shows that EDCA/RR keeps it promises of providing QoS guarantees in controlled environments.

For EDCA, on the contrary, the situation is totally different with an increasing amount of VoIP frames with very high end-to-end delays as the traffic load increases. Also here the results show that, when the traffic load is high (the curves representing five or six VoIP calls), the CCDF of the end-to-end delay for EDCA is lower when the medium is lossy compared to when it is error-free. The reasons for this behavior are the same as those

Table 4.5: The jitter of the VoIP traffic.

(a) 0% packet error

Number of	**Jitter** ($10^{-4}s^2$)	
VoIP calls	EDCA	EDCA/RR
1	0.7	0.50
2	1.5	0.51
3	4.8	0.51
4	14	0.51
5	390	0.51
6	560	0.51

(b) 5% packet error

Number of	**Jitter** ($10^{-4}s^2$)	
VoIP calls	EDCA	EDCA/RR
1	6.0	1.4
2	24	1.7
3	53	2.7
4	139	2.4
5	339	3.8
6	561	5.5

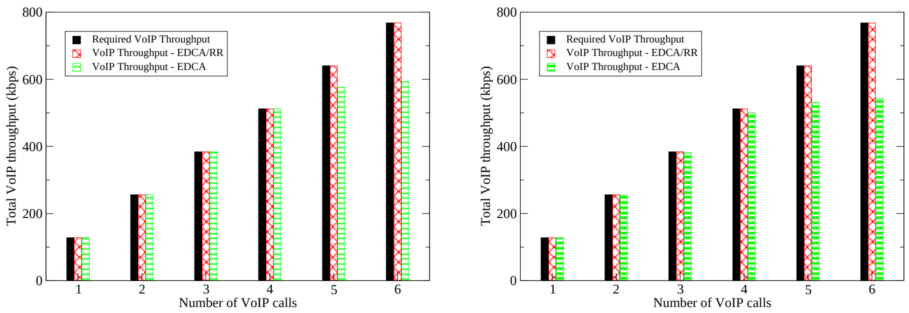
explained earlier in Section 4.4.3.1: since the frames with largest end-to-end delays are dropped and not considered in the calculations, and since the backoff mechanism in IEEE 802.11e disfavors low-priority traffic after transmission failures, the average end-to-end delay is reported to be lower for EDCA when the medium is lossy and the traffic load is high.

4.4.3.3 Jitter Analysis

Table 4.5a and 4.5b show the jitter experienced by the VoIP calls. Regarding EDCA, the tables show that the jitter starts from very low values and increases by 2-3 orders of magnitude as the number of VoIP calls increases. Considering EDCA/RR, the jitter is constant low in error-free media, whereas it increases very slowly in lossy media. Moreover, we can see that high traffic load (that is, when there are five or six VoIP calls) have a great negative impact on EDCA, whereas their impact on EDCA/RR is very limited. To sum up, again the results show that EDCA/RR is able to provide QoS to high-priority traffic even during high traffic load.

4.4.3.4 Average Throughput Analysis

Figure 4.6a and 4.6b show the total required and actual VoIP throughput for all VoIP calls as the number of VoIP calls increases. First, let us concentrate on whether the two schemes are able to provide the required throughput to the VoIP calls. Since we consider bi-directional VoIP communication, each VoIP call requires 2 × 64 kbps = 128 kbps. As the results show, EDCA/RR fully manages to give the required throughput to the VoIP applications both in error-free and lossy media, whereas EDCA fails to do so when the traffic load increases. With the given traffic load (three FTP clients, three HTTP clients and seven video streaming clients), EDCA can provide the required throughput to one VoIP call only in lossy media. Two VoIP calls require 256 kbps, which EDCA is close to fulfill, whereas six VoIP calls require 768 kbps, which EDCA is far from being able to provide. In fact, EDCA



(a) 0% packet error

(b) 5% packet error

Figure 4.6: Average VoIP Throughput in (a) error-free and (b) lossy media.

fails to fulfill the throughput requirements, not only in lossy, but also in error-free media when the traffic load is high. This will of course have consequences on the voice quality that the users experience.

Let us now move on to the analysis of the FTP/HTTP and video traffic. Table 4.6a and Table 4.6b show the average throughput for FTP/HTTP and video as the number of VoIP calls increases. The VoIP throughput is also there for the sake of easier comparison. The general view revealed by the results is expected: the throughput of the FTP/HTTP and video traffic decreases with increasing traffic and error rate for both EDCA and EDCA/RR. Leaving the general view, to focus on the FTP/HTTP throughput in lossy media, it is worth to note that the FTP/HTTP throughput drops to extremely low levels for EDCA when the traffic load is high. Obviously, the TCP-based FTP/HTTP flows are starved by the UDP-based VoIP and video streams. For EDCA/RR, on the other hand, the throughput does not decrease as dramatically as for EDCA (64 kbps compared to 6 kbps when there are six VoIP calls). The reason for this is that, due to the contention-free medium access for VoIP traffic in EDCA/RR, the TCP-based flows have to contend for medium access with UDP-based video streams only; whereas in EDCA, they have to contend with UDP-based VoIP streams as well. The more streams contending for medium access, the higher is the probability for collisions and retransmissions resulting in lower throughput.

Next, we notice that EDCA/RR performs better than EDCA even though there are no VoIP calls, that is, there are no reserved TXOPs. One might have expected a comparable or similar performance since both schemes have contending traffic only. However, from Section 4.2.2 we recall that, except for the resource reservation capability of EDCA/RR, the two schemes differ in another way: EDCA/RR has an extra AC used for important control messages. Thanks to this extra AC, ARP frames, AODV packets, and ADDTS

Table 4.6: The average throughput of the VoIP, FTP/HTTP, and video traffic.

(a) 0% packet error

	Throughput (kbps)					
Number of	VoIP		FTP/HTTP		Video	
VoIP calls	EDCA	EDCA/RR	EDCA	EDCA/RR	EDCA	EDCA/RR
0	0	0	1283	1358	2684	2684
1	128	128	1105	1102	2683	2685
2	256	256	836	841	2681	2685
3	384	384	592	574	2672	2685
4	512	512	414	358	2658	2606
5	576	640	399	300	2507	2211
6	593	768	306	204	2453	1829

(b) 5% packet error

	Throughput (kbps)					
Number of	VoIP		FTP/HTTP		Video	
VoIP calls	EDCA	EDCA/RR	EDCA	EDCA/RR	EDCA	EDCA/RR
0	0	0	386	612	2670	2678
1	128	128	255	306	2651	2680
2	255	256	106	171	2591	2680
3	381	384	32	115	2274	2459
4	500	512	14	89	1748	2083
5	531	640	8	77	1247	1702
6	543	768	6	64	1138	1326

Request/Response frames are delivered faster to their destinations, resulting in faster address resolution, route discovery and connection setup and thus higher throughput. The observant reader might now wonder why the impact of this enhancement is seen in the throughput of FTP/HTTP only, and not in that much in the throughput of the video traffic. For example, when the medium is error-free, the video throughput is exactly the same (2684 kbps) for both EDCA and EDCA/RR, whereas there is a significant difference in the FTP/HTTP throughput of the two schemes: 1358 kbps compared to 1283 kbps. To explain this behavior, we recall that each video stream requires 384 kbps so seven video streams require 2688 kbps, which is basically what they receive[4]. The TCP-based FTP and HTTP flows, on the other hand, adapt their data rate according to the flow and congestion control mechanism of TCP and depending on the condition of the network, they try to transmit as fast as possible while avoiding congestion. To sum up, EDCA/RR performs better than EDCA, even when there are no TXOP reservations, thanks to the introduction of an extra AC used for control messages.

[4] The reason why the video streams do not get exactly 2688 kbps, but “only” 2684 kbps, is a small amount of packet loss due to collisions on the wireless medium. The packet delivery ratio analysis in the next section confirms this claim.

Another point worth to comment is that, comparing the throughput of FTP/HTTP with that of video, the throughput fall is much larger for FTP/HTTP than that for video. For example, in the case of EDCA in error-free media, the throughput of FTP/HTTP falls with 76% from 1283 kbps to 306 kbps as the number of VoIP calls increases, whereas the video throughput decreases with only 9% from 2684 kbps to 2453 kbps. One reason for this is of course the higher priority given to video traffic compared to FTP/HTTP traffic. Another reason is the flow and congestion control mechanism of TCP slowing down the sending rate of FTP and HTTP traffic when the traffic load is high, whereas UDP continues to aggressively send packets at the same rate without caring about the condition of the network.

Yet another interesting observation is made by studying the throughput of the contending traffic, i.e., FTP/HTTP and video, when the medium is error-free compared to when it is lossy. With 0% packet error, the throughput of the contending traffic becomes higher for EDCA compared to EDCA/RR as the number of VoIP calls increases. On the other hand, with 5% packet error the throughput is higher for EDCA/RR. In the case of EDCA/RR and error-free medium, the throughput of the contending traffic is negatively affected by the increasing part of the medium being reserved by VoIP applications. However, in the case of lossy medium, the effect of the capacity reservation by EDCA/RR is not that negative anymore. This is because at the same time as decreasing the available resources for contending traffic, capacity reservation results in fewer traffic streams contending for medium access as the VoIP streams in EDCA/RR are not allowed to transmit at time instants other than during their reserved TXOPs. Thus, only when the medium is error-free and the amount of reserved capacity starts becoming significant (after 2-3 VoIP calls), the negative impact of not having access to the whole capacity affects the performance of the contending traffic more than the positive impact of less number of contending traffic streams. In all other cases, the contending traffic benefits from the fact that, in EDCA/RR, VoIP applications with reserved TXOPs do not contend for medium access causing collisions, backoff and retransmissions.

4.4.3.5 Packet Delivery Ratio Analysis

Table 4.7a and 4.7b show the packet delivery ratio (or equivalently, one minus the packet loss) experienced by the VoIP, FTP/HTTP and video traffic. Let us start the analysis by studying the packet delivery ratio for VoIP in error-free and lossy media. The results show that, the delivery ratio decreases when EDCA is used. For the contention-free EDCA/RR, on the other hand, the packet loss is negligible. Once more we see that EDCA suffers from high traffic load, with up to 29% of the VoIP frames being lost when the medium is lossy.

Table 4.7: The packet delivery ratio of the VoIP, FTP/HTTP, and video traffic.

(a) 0% packet error

	Packet delivery ratio (%)					
Number of	VoIP		FTP/HTTP		Video	
VoIP calls	EDCA	EDCA/RR	EDCA	EDCA/RR	EDCA	EDCA/RR
0	-	-	99.03	99.08	99.87	99.87
1	100	100	98.78	98.80	99.85	99.88
2	99.99	100	98.59	98.65	99.77	99.89
3	99.98	100	98.59	98.41	99.51	99.91
4	99.95	100	98.76	98.17	98.97	97.01
5	90.48	100	99.07	99.37	93.71	82.32
6	77.91	100	99.09	99.84	91.90	68.12

(b) 5% packet error

	Packet delivery ratio (%)					
Number of	VoIP		FTP/HTTP		Video	
VoIP calls	EDCA	EDCA/RR	EDCA	EDCA/RR	EDCA	EDCA/RR
0	-	-	99.28	99.25	99.36	99.64
1	99.92	99.98	99.03	98.87	98.81	99.69
2	99.79	99.97	98.66	98.56	96.73	99.71
3	99.35	99.97	97.25	98.35	85.49	91.54
4	99.01	99.97	94.37	98.10	65.58	77.56
5	83.23	99.96	83.37	98.12	47.19	63.40
6	71.35	99.96	69.94	97.77	43.42	49.40

Let us now study the delivery ratio of the video traffic when the medium is error-free compared to when it is lossy. Here we observe that, in the case of error-free medium, the packet loss becomes higher for EDCA/RR compared to EDCA as the number of VoIP calls increases, whereas the opposite behavior is seen in lossy media. To explain this, once more we look at the delivery ratio for VoIP traffic, and note that its delivery ratio is equal or very close to 100% for EDCA/RR, whereas it was significantly lower in EDCA. Thus, it is clear that EDCA/RR takes capacity from video and gives it to VoIP; in other words, the price of nearly loss-free VoIP transmission is lower performance for the video traffic. However, despite this cost, we can see that EDCA/RR has lower packet loss than EDCA when the medium is lossy. The reason for this behavior is the same as for EDCA/RR reporting higher video throughput than EDCA in lossy media, but lower video throughput in error-free media with increasing traffic load. This was discussed in the previous section analyzing the throughput: in error-free media, where there is no external source of error and, consequently, a very low probability of packet loss, the performance of the video traffic is negatively affected by an increasing part of the medium being reserved by VoIP traffic. In lossy media with higher packet loss probability, on the contrary, reserving TXOPs for VoIP transmission actually helps improving the performance of the video traffic. This is because the consequence of TXOP reservations is, not only less time available for contending traffic, but also less traffic contending for medium access, and thus, lower probability of collisions and packet loss. The less traffic contending for medium access is a result of the policy that traffic streams with TXOP reservations are allowed to transmit only during their reserved TXOPs.

Studying the packet delivery ratio of FTP/HTTP in error-free and lossy media, we see a very high delivery ratio in error-free media for both EDCA and EDCA/RR. This is due to the reliable delivery service provided by TCP. In lossy media, on the other hand, the delivery ratio decreases for EDCA, whereas it remains rather high for EDCA/RR. This effect has a common reason as for the dramatic throughput fall of FTP/HTTP when EDCA is used in lossy media: whereas the TCP-based traffic in EDCA/RR has to contend for medium access with UDP-based video traffic only, in EDCA it has to contend with UDP-based VoIP streams as well. Hence, using EDCA/RR, the less number of contending stations in lossy media results in lower probability of collisions and retransmissions and, consequently, better performance for the contending traffic.

Moreover, the results show that, as the traffic load increases, the packet loss becomes much larger for video than for FTP/HTTP, in spite of higher priority given to video traffic. This behavior is expected and explained by the reliable delivery service offered by TCP. FTP and HTTP use the connection-oriented TCP, which retransmits dropped packets, whereas video uses the connection-less UDP, which provides an unreliable delivery service. Thus, when video packets are dropped by IEEE 802.11 MAC after four transmission attempts, FTP and HTTP packets will continue to be retransmitted by TCP, resulting in higher packet delivery ratio. Also note that the higher priority for video traffic has a slightly noticeable impact on the results: when the medium is rather reliable (low traffic load, no packet error), the higher priority makes up for the unreliable service provided by UDP, resulting in a slightly higher packet delivery ratio for the video traffic compared to that of the FTP/HTTP traffic. However, as soon as the medium becomes unreliable, this positive impact of higher priority can no longer match the positive impact of reliable delivery service of TCP.

4.4.3.6 Instantaneous Throughput Analysis

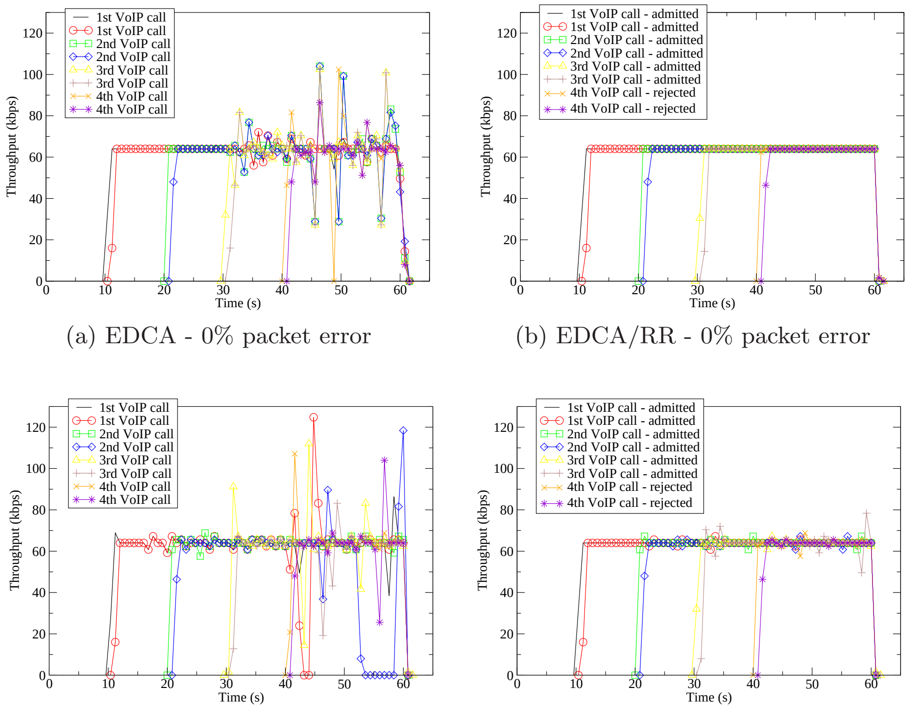
In the experiments done so far, 80% of the time has been available for TXOP reservations, which means that the admission control algorithm in EDCA/RR has been able to admit 8 VoIP calls. Consequently, all six VoIP calls have been admitted. Now we would like to study how EDCA/RR performs when a VoIP call is rejected. Therefore, we decrease the amount of time available for TXOP reservations from 80% to 30%. Thus, with an SI equal to 25 ms and TXOPs equal to 1.23 ms for the VoIP calls, EDCA/RR can admit $\lfloor 0.3\times 25/1.23 \rfloor = 6$ VoIP streams or, equivalently, three VoIP calls.

Since EDCA/RR can admit only three VoIP calls, we will use four VoIP calls in this experiment because we want the fourth VoIP call to be rejected. Let us now study the instantaneous VoIP throughput when starting new VoIP calls in the network. The parameters and their corresponding values used in this experiment are the same as those reported in Section 4.4.1, unless otherwise stated.

As mentioned earlier, a VoIP call consists of two VoIP streams, each transmitting 64 kbps. In this scenario we also have a high-quality video stream transmitting at 6.4 Mbps. This video stream is running in the background from the start until the end of the simulation, which lasts for 60 seconds. The purpose of the video stream is to add traffic to the network and disturb the VoIP traffic. The two VoIP streams of a VoIP call are started one second after each other, whereas the four VoIP calls are started 10 seconds from each other. Hence, the 8 VoIP streams are started at the 10th and 11th, 20th and 21st, 30th and 31st, and 40th and 41st second of the simulation. Once the streams are started, they continue transmitting until the end of the simulation.

The outcome of this experiment is shown in Figure 4.7. The figure shows the throughput of each VoIP stream measured at the application layer. Consequently, we must keep in mind that the bit rate on the wireless medium is much higher due to the overheads at the transport, network, data link, and physical layer. Since we are interested in the performance of the VoIP calls only, the throughput of the video stream is not included in the figure.

Figure 4.7a shows the case with EDCA when the medium is error-free. We can see that the first and second VoIP calls have a constant throughput, but as soon as the third call is started after 30 seconds, the throughput of all VoIP calls start to fluctuate. When the fourth VoIP call starts after 40 seconds, the situation gets even worse. Under EDCA/RR, on the other hand, the VoIP calls try to reserve TXOPs and if the reservation requests are admitted by the admission control algorithm, the VoIP calls get the amount of bandwidth they require. With 30% of the resources available for TXOP reservations, we know that the first three VoIP calls will be admitted, whereas the fourth will be rejected and contend for medium access. This can be seen in Figure 4.7b, which shows a constant throughput both for the three admitted VoIP calls and for the rejected VoIP call. It is interesting to note that, even though the fourth VoIP call is rejected, thus contending for medium access, its throughput is constant. The reason for this behavior is that, as opposed to the case with EDCA, the fourth (rejected) VoIP call



(a) EDCA - 0% packet error

(b) EDCA/RR - 0% packet error

(c) EDCA - 5% packet error

(d) EDCA/RR - 5% packet error

Figure 4.7: Instantaneous VoIP Throughput under (a) EDCA and (b) EDCA/RR in error-free media, and under (c) EDCA and (d) EDCA/RR in lossy media.

does not have to contend for medium access with three other VoIP calls; instead, it has a smaller amount of time reserved for contention, but during that contention period, it is the only VoIP call contending for medium access with the video stream. Hence, we notice that EDCA/RR does not only have a positive impact on the admitted VoIP calls, but also on the contending VoIP traffic.

Studying the same scenario when the medium is lossy, Figure 4.7c shows that EDCA performs extremely poorly when the fourth VoIP call is started. This overloaded network affects not only the fourth VoIP call, but also all other existing traffic in the network. Thus, the overall performance of the network is rather poor. It is also worth to notice that the throughput of the VoIP calls drops to very low levels, sometimes as low as 0 kbps! This has of course a significant impact on the voice quality experienced by the end users. Under EDCA/RR, on the other hand, Figure 4.7d shows a small throughput variation, but this variation is negligible compared to that experienced by the VoIP calls under EDCA. Again, we can see that also the rejected VoIP call has a rather constant throughput since it does not have to contend for medium access with many other streams.

To sum up, after this experiment we can point out three advantages of EDCA/RR compared to EDCA. First, the variance of the throughput of the admitted VoIP calls is very low; i.e., the throughput is rather constant around 64 kbps. Second, the throughput of the admitted VoIP calls is not decreased when the traffic increases. Third, resource reservation seems beneficial for not only the admitted streams, but also for the rejected ones. A similar conclusion was drawn in the previous throughput and packet delivery ratio analyzes in Section 4.4.3.4 and Section 4.4.3.5, showing that resource reservation was beneficial for not only admitted VoIP traffic, but also for contending traffic.

4.5 Conclusion

In this chapter we have presented EDCA/RR, which is a distributed MAC scheme based on EDCA and *compatible* with IEEE 802.11. One advantage with this solution is that it operates in a *fully distributed* manner offering distributed admission control, scheduling, and medium access. Moreover, it provides *QoS guarantees* by allowing applications with strict QoS requirements to reserve TXOPs for contention-free medium access. However, EDCA/RR does not only provide *contention-free* medium access and *parameterized QoS*; it also provides *contention-based* medium access and *prioritized QoS*. This is possible since EDCA/RR provides all existing features of EDCA as well.

We have evaluated the ability of EDCA and EDCA/RR to provide QoS guarantees. Our results show that EDCA/RR is clearly a technique that can provide guaranteed QoS within the limitations of error-prone wireless media. In particular, we have seen that, whereas EDCA suffers from severe performance degradation with increased network load, EDCA/RR succeeds providing low and controlled end-to-end delay and jitter, the throughput as required by the real-time application, and negligible packet loss. In addition, we would like to stress that, not only does EDCA/RR provide better service than EDCA in lossy wireless media regarding real-time traffic, but also when it comes to contending non-real-time traffic.

Since EDCA/RR is based on existing and commonly used protocols, it is a *realistic* approach that can be implemented into existing wireless systems to *fill the void* left by the ignored HCCA/WMM-SA. In other words, since HCCA/WMM-SA will most probably not find their way into our IEEE 802.11-based wireless equipment, EDCA/RR can be considered as a good compromise keeping the preferable distributed medium access approach of EDCA/WMM but extending it with guaranteed QoS provisioning capabilities of HCCA/WMM-SA.

Although EDCA/RR can be used in a multi-hop environment, it is based on EDCA, which is mainly designed for single-hop networks. Since multi-hop mesh networks are expected to offer new communication possibilities and since the idea behind EDCA/RR showed to be promising, we also presented a distributed and reservation-based MAC scheme for WMNs. This scheme is called DDCA, and has the same advantageous properties of EDCA/RR. In other words, it is also based on EDCA, it operates in a fully distributed way, provides QoS guarantees via resource reservation, and provides both parameterized and prioritized QoS.

Chapter 5

Experiments on a Wireless Mesh Network

A WMN[1] is a multi-hop wireless network consisting of stationary devices that are usually not power-constrained. The aim of WMN technology is to provide capabilities that can facilitate the deployment of multi-hop wireless networks with access to the Internet. The architecture of a WMN makes it a hybrid wireless network between a WLAN and a MANET. In essence, a WMN is able to extend the coverage of the network infrastructure by multi-hop wireless connections between APs. The APs in a WMN can hence be detached from any wired infrastructure while being connected to each other through wireless links. As shown in Figure 5.1, nodes in a WMN can be categorized into:

- Mesh Point (MP): A device that provides mesh services; it can relay messages in an ad hoc fashion to other MPs in the WMN.
- Mesh Access Point (MAP): A special MP that also provides AP services, i.e., it provides wireless connectivity.
- Mesh Point collocated with a mesh Portal (MPP): A special MP that also serves as a gateway to a wired network, i.e., it provides wired connectivity.
- Station (STA): A mobile user device that does not participate in mesh services; instead it can communicate with other stations via an AP, a MAP, or an MPP.

In essence, WMNs are composed of MPs that facilitate the connectivity and intercommunication of wireless clients through multi-hop wireless

[1] In this chapter we consider IEEE 802.11-based WMNs.

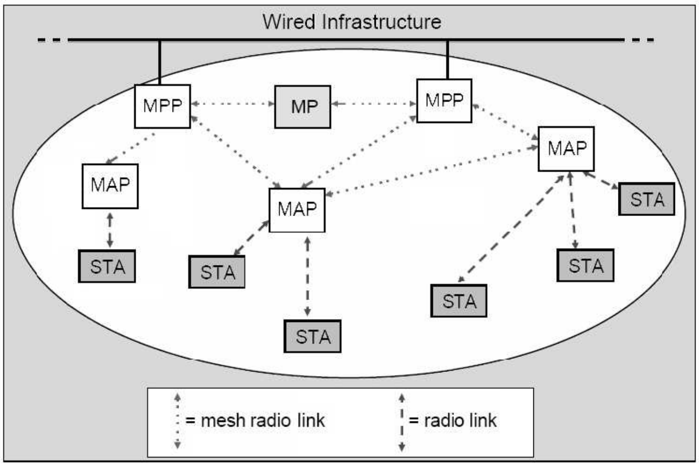


Figure 5.1: Wireless mesh network architecture.

paths. WMNs may be connected to the Internet through an MPP, whereas the MPs function as wireless bridges within the WMN. Therefore, if we compare WMNs with MANETs, end hosts and routing devices are distinct in a WMN, whereas every device can play both roles in MANETs. Furthermore, MPs in WMNs are often stationary and not power-constrained; thus, routing protocols designed for WMNs are free from the burden of dealing with mobility and power constraints.

A peculiar feature of WMNs is that, if the source and the destination are not in the same Basic Service Set (BSS) domain, the source MAP does not forward packets to all the MAPs in the Extended Service Set (ESS); instead, the packets are sent via MPs to reach the destination. A WMN can be viewed as a multi-hop, ad hoc, packet switching, and forwarding network between MPs in the same ESS. The Wireless Distribution System (WDS) uses an extension of IEEE 802.11, named IEEE 802.11s, to provide a protocol for self-configuring paths among MPs in a multi-hop topology, supporting broadcast, multicast, and unicast traffic.

Generally, the primary purpose of the WMN is to create a low-cost, easily deployable, high performance wireless coverage. WMNs can be useful network architecture where Ethernet cabling does not exist or its installation is economically prohibitive. Examples include small and large offices, manufacturing plants, university campuses government buildings, and health care centers/hospitals.

In this chapter, we present the idea behind Augmented Reality (AR), which is born as a variation of Virtual Reality (VR), but differs from its ancestor as it does not put the user into an exclusively virtual environment. Instead, AR supplements reality by superimposing digital objects upon the real world, which remains visible for the user. After a brief overview of AR, we give a few examples of application scenarios where collaborative AR applications and WMNs form an attractive combination of technologies. Next we present the tool used to set up a WMN testbed that we used for some real-world experiments. Finally, we present the results from our off-the-shelf testbed and conclude the chapter.

5.1 Related Work

Delving into the scientific literature, we can find works that relate to our case study such as the preliminary WMN testbed in [75]. However, most works focus on issues such as network capacity, transmission reliability, packet routing, and security [76, 77]. Although important and inspiring, each of them represents only a part of the scenario we are considering, which involves a real testbed assessment of WMNs to support transmissions (especially real-time ones) in a department-wide AR environment.

In this context, real-time applications represent an important source of traffic in the WMN; hence, [78] evaluates the possibility of aggregating packets to improve VoIP performance over a WMN. The hidden station problem, the exposed station problem, and the binary exponential backoff scheme are indicated by [79] as main causes for transmission delay over multi-hop wireless links, such as in a WMN. The authors hence propose to reserve at least one path having enough bandwidth before starting to transmit real-time multimedia contents so as to reduce this delay.

Analyzing a home WMN scenario, [80] shows that the classic shortest-path selection algorithm with minimum hop counts to search for a gateway can easily lead to load imbalance in the network and thus negatively affect both the throughput and the per-packet delay of transmitted data flows. Therefore, the authors suggest adopting a wireless home mesh router selection mechanism based on a QoS-driven selection metric that takes into account also the residual capacity on each link. Finally, [81] presents a theoretical study of a G/G/1 queuing model to characterize the average delay and maximum throughput in WMNs, given certain network parameters and assuming intra-mesh communication.

Strongly characterized by a practical aim, our work differs from the preceding ones as it is a real testbed evaluation of networking issues and solutions related to a specific and challenging application instance: collaborative services for AR environments. Yet, some of the aforementioned solutions may be compatible with our tested architecture and contribute in enhancing its performance.

Regarding work related to AR technology, most of the studies about AR environments have focused on the problem of promptly aligning digital objects over the real world. To this aim, most of the experimental testbeds reported in scientific literature consider an off-line single user with pre-stored information about digital objects to be visualized depending on the user's position and scene in front of her/him and thus require efficient technologies in terms of position tracking, image recognition, rendering, and alignment [82, 83].

To the best of our knowledge, no study has been performed that analyzes *collaborative* AR environments, where communication among participants plays a fundamental role. Collaborative AR environments represent a very interesting case study, both for the appealing applications that can be deployed (e.g., team coordination for first aid operations) and for the research challenges that involve (e.g., prompt delivery of data generated by any participant to the whole team through multi-hop wireless connectivity).

To this aim, the main contributions of this study are related to providing practical directions for the networking support of collaborative AR environments and can be summarized by the following list:

- analysis of networking issues related to the practical deployment of collaborative AR environments;
- proposal of a networking architecture, based on WMN technology, to support communication among participants in the considered context;
- creation of a real testbed to evaluate the proposed architecture;

5.2 Augmented Reality

Augmented Reality, also known as *Mixed Reality*, is the technology that enables superimposing digital data (e.g., images, links to web pages, and 3D objects) upon a user's view of the real world [84, 85]. It is clear how AR descended from *Virtual Reality*, however, whereas the latter completely immerses a user in a digital environment, the former combines digital and physical world. In other words, AR is perfectly suited to help us in perceiving and managing both our physical world and our information world altogether.

In recent years, AR has been subject by a raising interest from both researchers and practitioners. This is due to progressive advancements in this field and to the infinite possibilities for AR technology to complement and improve the way we interact with our favorite digital services. Indeed, whereas sociologists agree that we are living in an *information society*, it is also evident that we are not naturally equipped to continuously manage all the information that is available to us. Simply, our real world is not anymore made of only physical objects: also all the information that is somehow related to us belongs to our real world, even if our natural senses are not capable to handle it.

5.2.1 Collaborative AR Applications

From a research point of view, AR is particularly interesting both for its technical challenges and for its appealing applications; this is especially true if we consider collaborative AR applications. Indeed, collaborative AR is potentially able to enhance the way users perceive the world and interact with/through it. By overlaying digital data over the physical view it is possible to provide users with a shared, synthetic, information-based "sixth sense". Possible applications for this technology are limited only by our imagination. In the following we provide a representative list of possible employments for collaborative AR technology.

Medical applications. AR can be used to enhance a doctor's view of a patient, especially for non-invasive surgery or for remote operations. Data generated by magnetic resonance, computed tomography scans, X-rays, and ultrasounds could be directly projected over the patient's body or over a remote manikin, allowing the doctor to see inside the patient and perform precise operations without the need for large incisions, and wherever the doctor and the patient are located with respect to each other [82], [86].

Maintenance and assembly. Assembling, maintaining and repairing could be tough tasks when regarding complex machineries. To ease these tasks, AR can project online instructions, drawings, step-by-step animated examples, known issues, and previously performed reparations over the operator's view of the machinery [87], [88]. Furthermore, to help any operator that may be in front of the broken machinery, suggestions could be prepared in real-time by remote highly specialized operators and projected over the machinery, along with instructions and requests simultaneously exchanged by voice communication.

Annotations. People use notes as reminders or to leave messages for others. These notes could be replaced by digital ones left in an AR environment [89], [90]. As a major advantage, digital notes could be easily customized to be public or specifically destined to a certain user (and existing only in the AR environment as seen by this user); moreover, they could also be automatically generated from databases (e.g., labels in a store) and instantaneously modified over the entire AR environment with just one click/event in a remote location.

Safety ensuring applications. Virtual lines and objects, even through Head-Up Displays (HUDs), can be used to aid the navigation, especially in conditions of limited visibility (e.g., under water, in outer space, with adverse meteorological conditions), or to support and coordinate first aid squads in an emergency area after a crisis (e.g., earthquake, flooding, major accident) [91]. Indeed, it is not hard to imagine a scenario where first aid squads in an emergency area utilize HUDs with superimposed information about dangers and people's health conditions while coordinating through voice communication.

Entertainment applications. Entertainment applications can exploit AR technology in several ways. For instance, merging real actors with virtual ones over real or virtual landscapes has become a regular practice in Hollywood movies allowing great visual effects at a reduced production cost. This technology is soon going to be used also for gaming applications bringing real people (maybe organized in squads) into a virtual or mixed-reality arena populated by both digital creatures and humans [92], [93].

Cultural heritage applications. Presentations based on AR technologies provide museum visitors with the possibility to enrich their visit, interact with (the digital representation of) a piece of art, and choose the level of reconstruction of artifacts and historical sites [94], [95]. Furthermore, investigators can use digital notes superimposed on archaeological sites or paintings to attach information to the object of study in a non-invasive way and make it available to other researchers to facilitate research in collaboration [96].

In essence, because of the current proliferation of collaborative applications, the advancements of AR technology, and the growing availability of wireless devices, it is interesting to study how these technologies can be integrated to create effective collaborative AR applications based on wireless communication in a department-wide environment. The use of WMNs is ideal for a collaborative AR environment as they are composed by a collection of routers so as to extend the coverage area of a WLAN from a *hot spot* to a *hot zone* composed by various hot spots [79]. Thus, WMN architecture enables wireless communication among participants in a hot zone, such as a department, in a quick and simple way. However, we also need to evaluate the performance efficiency of WMNs in supporting the involved services, e.g., control messages and VoIP communication among several users.

5.2.2 Networking in Collaborative AR Environments: Provided Services

The applications discussed in the previous section can be run off-line, by simply pre-storing on the adopted device all the information that will be superimposed on the real world to create the AR environment. Yet, this method sensibly limits the potentiality of AR applications. Revising the list of presented applications, we can envision various appealing services that can be enabled only by networking capabilities. First, any of the applications could be run from a remote location, e.g., remote surgery, remote maintenance, and remote annotation. Actions performed by a remote operator could be transmitted to be locally executed by a machine or just superimposed on the HUD of a local operator in order to assist her/him.

Second, when operating in groups, actions performed by a certain person may have effects even for other team members. For instance, think of online game players immersed in an AR arena competing with each other: information about “shooting” at a certain player has to be transmitted to the target player and information about decreased points of the hit player has to be transmitted to all participants. Another example is represented by an employer that has to leave her/his office temporarily and leaves a virtual note on the door that automatically reports her/his current location within the building in case somebody urgently needs to get in touch with her/him.

Last but not least, voice communication among users may be of prominent importance for many collaborative AR applications. Indeed, whereas sending control messages and assigning virtual notes represent important features, voice communication is often necessary, or desirable, or just the fastest way to coordinate a group of users. To this aim, think again of the doctor remotely assisting another one, or of the game players (or first aid responders) organized in teams where members of the same team can communicate with each other. This kind of communication has to be an integral part of the software architecture supporting the collaborative AR environment. In this sense, it may be deployed as a VoIP service integrated within the system.

5.2.3 Networking in Collaborative AR Environments: Challenges

Networking services depicted in the previous subsection may be roughly summarized as:

1. transmitting information about virtual notes digitally attached to certain real object to users walking by the area where the object is located;
2. transmitting control messages (e.g., game events, machinery movements) to users belonging to a certain group or located in a certain area;
3. establishing VoIP communication among users or groups of users.

Providing these services passes through enabling a continuous coverage in the whole AR area and ensuring a certain performance level. Focusing on the first of these two requirements, we have to keep in mind that users could be moving in an area wider than a single hot spot, thereby, providing seamless connectivity becomes a non-trivial challenge. However, mesh networks may represent a perfect answer to this challenge thanks to their ability in merging various hot spots into a unified hot zone. Yet, having several wireless nodes moving around the AR area transmitting, receiving, and relaying data through different APs may generate interference and congestion that cause the loss of several transmitted packets. Even if the considered applications (e.g., control messages, VoIP) are generally resilient to (few) packet losses, having a highly unreliable channel may negatively affect the performance of the AR system as perceived by end-users, for instance, by losing movements remotely commanded by an operator, or critical game events such as a shooting, or a position update about a virtual object with respect to a real one.

The second requirement has to be interpreted through the considered applications. Delivering control messages and providing VoIP support have different performance requirements/metrics than, for instance, downloading files.

The aforementioned network services mostly fall into the realm of real-time applications; it is reasonable to expect that at any moment there will be several people talking and various control message streams going on, whereas file transfers will only happen seldom. We are hence more concerned with the per-packet delay and jitter as these represent performance metrics for real-time applications. Jitter is strictly linked to per-packet delay; they are usually present together in a system. Yet, jitter could be even more annoying than delays in the considered context. In fact, even if message delivery delays represent a problem for real-time applications, yet, when these delays are constant, some applications may be built so as to anticipate the delay and correct the effect (e.g., by superimposing the virtual object on the real one while calculating its position few tens or hundreds of milliseconds ahead in time). However, this prediction technique can clearly not be applied in presence of highly variable delays, i.e., a high jitter.

For these reasons, in our experimental evaluation (reported in Section 5.4) we have built a real mesh network in a department-wide area, generating traffic representing the aforementioned AR applications: control messages, VoIP, and background FTP flow. In this context, we have monitored the packet loss, jitter, and packet reception rate performance with different network traffic configurations so as to study the behavior of the system when more services and/or more users are simultaneously exploiting the WMN-based collaborative AR environment.

5.3 Mesh Connectivity Layer

Microsoft's Mesh Connectivity Layer (MCL) allows for the deployment of a WMN using any wireless card [97]. Simply, as a native Windows driver, upon installation the host system can see a virtual network adapter that allows for direct connectivity to the wireless mesh network. More specifically, MCL is an interlayer protocol, architecturally located between the network layer and the data link layer. Its position allows it to complement surrounding layers in a transparent way. The need of modifications of existing systems is hence minimized, allowing for the use of regular technologies and protocols while introducing the mesh connectivity feature.

As its routing protocol, MCL uses a modified version of DSR, which is called Link-Quality Source Routing (LQSR) [98]. LQSR implements all the basic DSR functionality, including route discovery and route maintenance. The main differences between LQSR and DSR are related to LQSR's implementation at layer 2.5 instead of layer 3 and its support for link-quality metrics. Upon reception of a RREQ, an MP appends not only its own address but also the metric for the link over which the RREQ arrived. When an MP sends a RREP, the reply carries back the complete list of link metrics for the route. LQSR uses two metric maintenance mechanisms to handle link metric variations because link metrics vary a lot, even when nodes are static. Thus, LQSR uses a reactive mechanism to maintain the metrics for the links that it is actively using and a proactive background mechanism to maintain the metrics for all links. The basic operation of LQSR is as follows:

a) it discovers all the neighbors and assigns weights to the links;

b) it determines the channel, the bandwidth, and the loss rate for every possible link and spreads this information to other MPs;

c) based on this information, LQSR computes a routing metric called Weighted Cumulative Expected Transmission Time (WCETT) [99], which defines the best data transmission path from a given source to a given destination;

d) if the optimum path between a particular source-destination pair changes, the route is modified accordingly.

Surprisingly, MCL is provided by Microsoft as an open source tool, allowing anyone to modify its code and testing alternative solutions to any of its components. For our testbed, we intentionally used the default settings since the goal was to use a tool for quick and easy deployment of WMNs, that is, an off-the-shelf option for mesh networking.

5.4 Evaluation

In order to evaluate the considered scenario of a WMN supporting collaborative AR applications, we used the MCL software to build our WMN testbed. The mesh backbone is implemented with Dell Latitude D610 Review laptops (Pentium M 760, 2.0 GHz, 512 MB RAM) with ZyXEL AG-225H Wireless Network Interface Cards (WNICs). Instead, the stations are Dell laptops with Pentium III and 128 MB of RAM. The presented results are averages over ten experiments, each ran for 30 seconds.

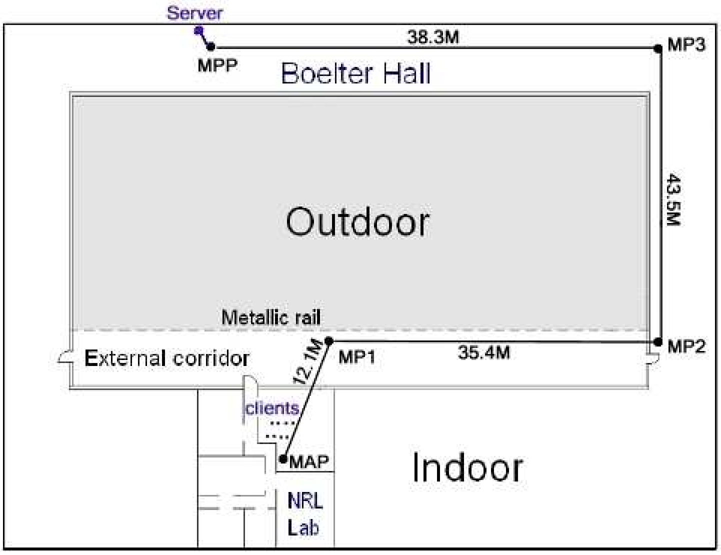


Figure 5.2: A map of the testbed.

5.4.1 Experiment Setup - The Testbed

A total of five mesh-capable devices are part of our WMN; three MPs, one MAP, and one MPP. When not differently stated, an FTP and video streaming server is connected to the MPP, whereas a variable number of stations (clients) are connected to the MAP. The MPs on the mesh backbone are operating on channel 11, whereas the stations are communicating with the MAP on channel 1. The rationale behind this choice is that of keeping these two channels far from each other so as to decrease the inter-carrier interference.

The WMN was set up at Boelter Hall, UCLA campus, in the immediate vicinity of the Network Research Lab. Boelter Hall is a square-shaped building with an open area in the middle. The map in Figure 5.2 provides a bird-eye view of our network topology setup in its most general configuration, whereas Figure 5.3 shows two pictures of the actual testbed taken from different angles.

In order to determine the communication range of each MP in the WMN, we ran preliminary tests using tools such as ping and traceroute. Then we carefully positioned the nodes so that non-neighboring MPs could not communicate with each other unless they communicated via an intermediate MP. For instance, MP1 in Figure 5.2 cannot communicate directly with MP3 unless the packets are routed through MP2. This way, data packets were prevented from using shortcuts among MPs as these would have affected the accuracy and the clarity of collected results by unpredictably decreasing the

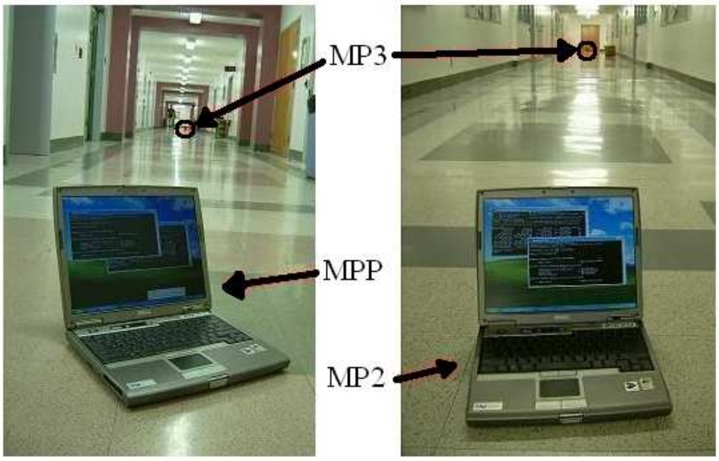


Figure 5.3: A snapshot of the testbed.

actual number of hops with respect to our experimental intentions.

In our experiments, different real-time and collaborative AR applications (e.g., UDP-based streams of control and VoIP messages, and TCP-based FTP flows) are run solo or together to evaluate their performance on the WMN. In particular, we considered VoIP streams, each corresponding to 64 kbps of voice traffic generated according to the G.711 voice codec [74]. Each generated voice packet carries two samples of 80 bytes, i.e., the size of the payload is 160 bytes. Moreover, we have run experiments with a real video stream: an MPEG video with a bit rate varying between 218 and 456 kbps. Finally, we also had an FTP flow.

The VoIP traffic was generated using the Distributed Internet Traffic Generator (D-ITG) [100], whereas we used VLC media player [101] for video streaming using RTP, and FileZilla [102] for file transfer using FTP. Even if we consider both elastic and real-time applications, we keep the focal point on the performance achieved by real-time applications, as they represent the main service for the considered scenario.

5.4.2 Performance Metrics

In order to evaluate the performance of the WMN testbed, we used the following metrics:

- **The average jitter**: the average of delay differences between consecutive packets; if S_i is the sending time and R_i is the receiving time of packet i and n is the total number of packets, then:

$$average\ jitter = \frac{\sum_1^n |\Delta D_i|}{n},$$

where

$$\Delta D_i = (R_i - S_i) - (R_{i-1} - S_{i-1}).$$

- **The average packet loss**: the amount of data packets (in %) generated by D-ITG that are not successfully received by the destination.
- **The average packet reception rate**: the goodput in terms of packets/s successfully received by the destination

5.4.3 Experiment Results

In this section we report on the experimental outcomes of the presented WMN testbed. The first subsection regards a scenario with competing elastic and real-time flows, whereas the second subsection considers different real-time applications simultaneously sharing the WMN.

5.4.3.1 Elastic Flows vs. Real-time Flows

As our first experiment, we run a single elastic application (i.e., an FTP/TCP download session) over our WMN, varying the number of hops that packets have to traverse from the source (the FTP server) to the destination (the FTP client). Just to provide a couple of examples, in the considered AR scenario, this data flow could represent a digital note that has to be superimposed on a real object, or a reparation manual for a certain component.

The client was positioned as depicted in Figure 5.2 and engaged with the MAP, whereas the position of the server is varied: on MAP to have the 1-hop evaluation, on MP1 to have the 2-hops evaluation, on MP2 to have the 3-hops evaluation, etc[2]. The purpose of this experiment is both to evaluate the performance of a single elastic flow in a WMN and to verify the reliability of the outcomes produced by our testbed.

Results for this experiment are reported in Figure 5.4. As the figure shows, the time to download the file increases linearly with the number of

[2] Of course, when moving the server from MPP to, for example, MP3, the role of MPP changes to an MP, whereas MP3 becomes an MPP. However, this change of roles of the mesh devices is not of any importance for us as it does not have any impact on our results.

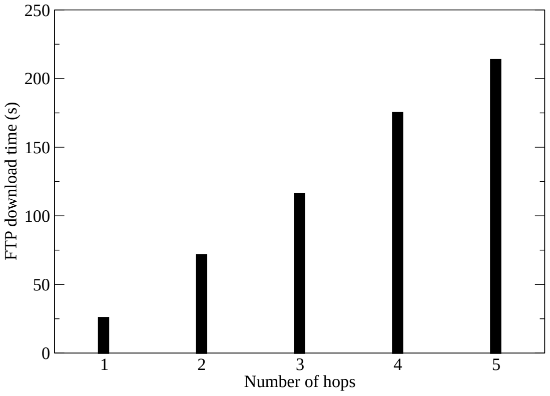
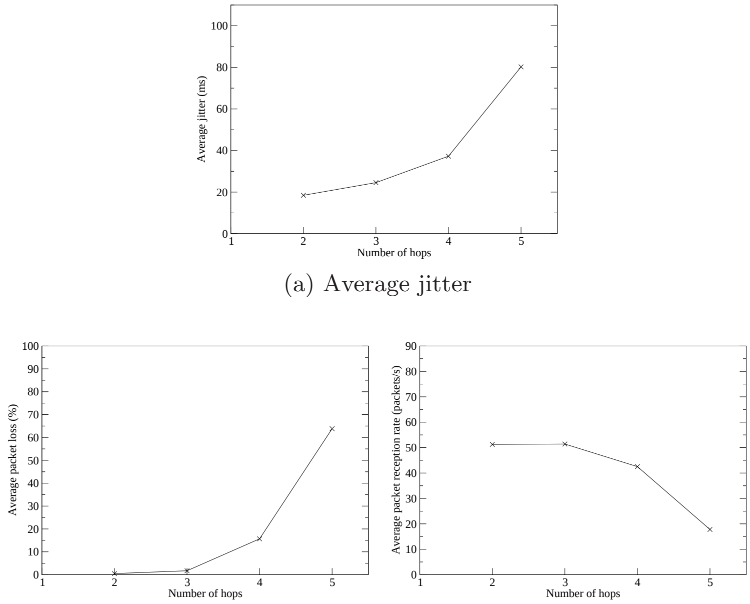


Figure 5.4: FTP download time for a 17.3 MB file considering a connection exploiting several hops in the WMN.

hops the flow has to traverse. This is not surprising and indicates that our testbed is performing correctly. Indeed, it is well-known in scientific literature that the available data rate for TCP-based flows decreases for each wireless hop until becoming unable to support any application after a certain “ad hoc horizon” [103, 104].

As a second experiment, we consider a scenario with several users simultaneously voice chatting with each other and study the impact of changing the number of hops traversed by the VoIP streams; the configuration also includes one ongoing FTP session run in the background. The FTP flow always traverses two hops, whereas ten simultaneous VoIP streams traverse from two to five hops. Just like the previous experiment, we had to change the position of the server to evaluate the performance of the considered scenario under different number of hops. Clearly, the more hops the VoIP streams traverse, the more is the impact on their performance. This is confirmed by Figure 5.5, which shows the average jitter, packet loss, and packet reception rate experienced by the ten VoIP streams when varying the number of hops. In particular, the performance of the VoIP streams follows an exponential trend when the number of hops grows linearly. This has a devastating effect on the perceived quality of the VoIP applications: voices will result severely scattered if the distance between the participants is larger than four hops.

As a third experiment, we consider the case where the background traffic is either represented by an elastic FTP flow or by a video stream. In both cases, this background traffic is traversing two hops between the FTP/video



(a) Average jitter

(b) Average packet loss

(c) Average packet reception rate

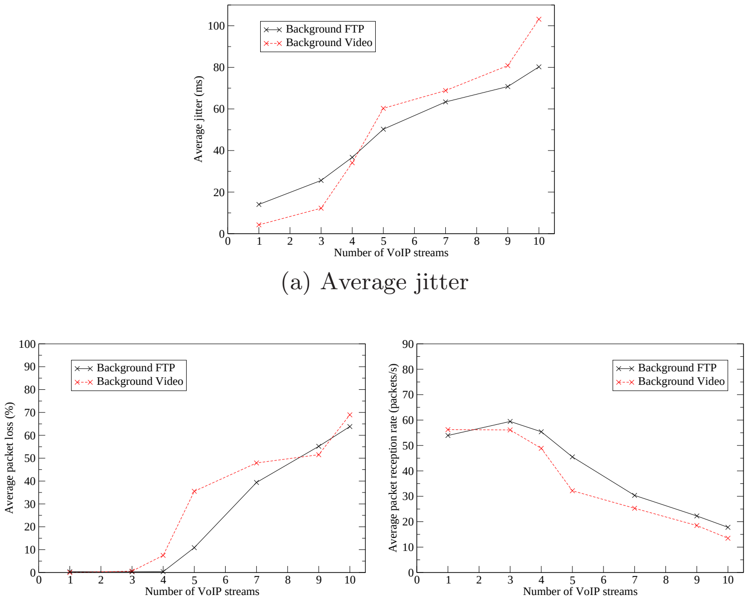
Figure 5.5: (a) Average jitter, (b) average packet loss, and (c) average packet reception rate experienced by ten VoIP streams

streaming server (connected to MPP) and MP2. Simultaneously, between one and ten VoIP streams traverse all five hops of the WMN. The average jitter, packet loss, and packet reception rate for the VoIP streams are reported in Figure 5.6. As expected, the performance of the VoIP streams worsens when increasing the number of simultaneous VoIP sessions as they interfere with each other (and with the background traffic).

As the charts show, when the background traffic is represented by the FTP flow, the jitter grows linearly with the number of simultaneous VoIP streams, whereas the packet loss is negligible from one to four simultaneous VoIP streams and then grows very quickly. At that point, however, even the jitter starts to be too high to ensure good performances. Instead, with the video stream set as background traffic, a negligible packet loss is ensured only up to three simultaneous VoIP streams. We can hence conclude that with the considered configuration, only up to 3-4 simultaneous VoIP streams can be effectively supported over the WMN.

5.4.3.2 Real-time Flows vs. Real-time Flows

Since real-time applications represent the main source of traffic in the considered WMN scenario, it is important to evaluate how they would affect each other if running simultaneously. To this aim, we analyze how the perfor-



(a) Average jitter

(b) Average packet loss

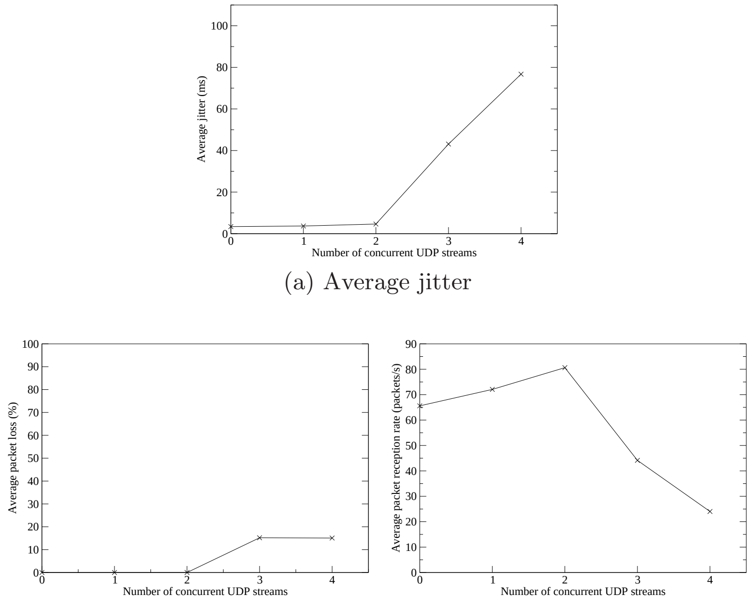
(c) Average packet reception rate

Figure 5.6: (a) Average jitter, (b) average packet loss, and (c) average packet reception rate experienced by a variable number of VoIP streams with one concurrent FTP or video flow.

mance of a VoIP session is affected by other generic UDP-based streams, e.g., how the quality of an ongoing conversation would be affected by messages automatically sent to synchronize the alignment between digital objects and the real world.

Results for this analysis are reported in Figure 5.7, showing the average jitter, packet loss, and packet reception rate of a VoIP stream while the background UDP-based traffic (carrying control messages for the AR appearance) is progressively augmented. Both the VoIP stream and the UDP streams traverse all five wireless hops through the WMN, i.e., from the stations to the server connected to MPP. Each UDP traffic source generates 320 kbps. The charts demonstrate a sudden performance worsening when the number of the background UDP-based streams is increased from two to three. Indeed, even if both the considered applications are not particularly bandwidth-consuming, yet, they continuously send out packets and thus keep the wireless channel busy. When these transmissions involve multiple hops, they consume their portion of the channel on each of the involved hops, multiplying their congestion and interference effects until causing the sudden deterioration of performances seen in the charts.

It becomes hence fundamental to differentiate among the various data flows and provide adequate performance to the most relevant among them.



(a) Average jitter

(b) Average packet loss

(c) Average packet reception rate

Figure 5.7: (a) Average jitter, (b) average packet loss, and (c) average packet reception rate experienced by a single VoIP stream when competing with several concurrent UDP streams.

For instance, when deploying the WMN to support the collaborative AR environment, the designer should decide whether to privilege VoIP streams over alignment-control messages or vice-versa. Probably, this decision will be based on cognitive analysis on the impact of providing certain QoS levels to final application users, rather than by networking or computer science studies. Our responsibility as computer scientists is that of providing instruments to be able to enforce this differentiation, once decided its rules.

As demonstrated in this chapter, intense network traffic conditions makes it impossible to adequately support all simultaneous flows (i.e., VoIP services, synchronization, virtual and physical objects alignment, file download, etc.). In these situations, real-time flows should be privileged as the functioning of the collaborative AR world is mostly based on prompt responsiveness. Among the various real-time services that could be simultaneously present, we assume that an intrinsically interactive application such as VoIP has a greater impact on the global performance of the AR system perceived by the user than control messages for the alignment of the synthetic objects over the real world. Once the QoS requirements and priorities for each kind of service are decided, our purpose is to factually enforce the chosen policy and provide an adequate performance level at least to the most important application(s).

5.5 Conclusion

AR technology represents a great complement for many applications. Yet, current studies generally focus on graphical enhancements or problems related to the alignment between digital objects and the real world. Instead, networking emerges as crucial, even if currently overlooked, when considering AR for collaborative applications.

To this aim, we have proposed to employ WMN technology to quickly set up and support AR-based collaborative applications in department-wide, indoor-outdoor environments. We have reported on the deployment of a real WMN and used our testbed to provide an evaluation of the actual capability of this technology in supporting collaborative AR applications. Based on our results, we can claim that today's WMN technology is promising even if performance quickly degrades when increasing the number of wireless hops between the source and the destination, becoming hardly satisfying after three wireless hops. Yet, a wise design of the network architecture that limits the number of consecutive wireless hops may result in a WMN that is able to satisfy the users' needs. Also, a QoS mechanism providing medium access reservation and/or service differentiation could improve the performance of the applications. For example, by using EDCA or DDCA (presented in Chapter 4) we are able to provide different priorities to traffic flows or reserve resources for the most time-critical applications.

Chapter 6

Contributions and Conclusions

In this thesis we have studied topics related to distributed wireless computer networks. More specifically, we have tackled the problems of providing Internet access and QoS guarantees in these networks.

Among the scientific achievements of this thesis, we can mention a successfully designed and implemented ad hoc routing protocol that is capable of routing packets not only within an autonomous MANET, but also between a wireless MANET and the wired Internet. This way, mobile stations in a MANET can have access to the Web with the broad range of services that it offers. Using the popular ns-2 network simulator, we have shown that our solution, called *AODV+*, succeeds in this network interconnection between a MANET and the Internet. As its name implies, our solution is based on the widely used and popular AODV ad hoc routing protocol. Our implementation of *AODV+* has been contributed to the ns-2 community in order to help others benefit from the advantages of our solution. Sure enough, *AODV+* has shown to be rather popular and we have found it being used by researchers and students all around the world. It is worth to mention that, besides the obsolete Destination-Sequenced Distance Vector (DSDV) ad hoc routing protocol, *AODV+* is the only solution for simulations of wired-cum-wireless scenarios in ns-2.

Furthermore, we have studied the de facto medium access standard for IEEE 802.11-based wireless networks, namely EDCA, and identified its limitations regarding its ability to provide QoS guarantees. The behavior of EDCA was studied in a real ad hoc network, showing its capabilities of differentiating traffic. However, as the network traffic increases, the overall performance will decrease for all applications since there is no admission control mechanism to regulate access to the shared medium. In order to continue prioritizing real-time traffic even during high traffic load, we have proposed a reservation-based MAC scheme that is able to provide QoS guarantees.

Since EDCA has been reported not to satisfy time-critical applications, there was a hope that the introduction of HCCA/WMM-SA into our wireless devices was going to solve the problems of EDCA. However, as a centralized MAC scheme, HCCA/WMM-SA is not getting any attention, thus clearing the way for other solutions. In this context, EDCA/RR is proposed as a realistic and conceivable solution with advantageous features from both worlds: it is both distributed and provides contention-free medium access through resource reservation. In addition, we would like to stress that our MAC scheme is distributed and compatible with EDCA, which we believe are two important features that any realistic proposed enhancement for IEEE 802.11 networks should have.

Although EDCA/RR can be used in a multi-hop environment, it is based on EDCA, which is mainly designed for single-hop networks. Since WMNs are expected to offer new communication possibilities and since the idea behind EDCA/RR showed to be promising, we also presented a distributed and reservation-based MAC scheme for WMNs. This scheme is called DDCA, and has the same advantageous properties as EDCA/RR.

Furthermore, we have complemented our simulation studies with real-world experiments on an off-the-shelf WMN. Among other things, we studied the impact of network traffic and number of wireless hops in a multi-hop WMN. Using real applications together with traffic generators that we could control in an off-the-shelf WMN, we could gain a clear insight into the limitations as well as opportunities of WMNs.

Finally, we shall mention that both AODV and EDCA are used in the upcoming IEEE 802.11s amendment for mesh networking. Since *AODV+*, EDCA/RR, and DDCA, are based on and propose enhancements to these protocols, our solutions can be considered as attractive solutions for future WMNs. Thus, the solutions presented in this thesis can be implemented into existing wireless systems to provide Internet access, service differentiation and resource reservation.

Bibliography

[1] *IEEE Std 802.11-2007 Part11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, 2007.

[2] The MANET working group website. `www.ietf.org/html.charters/manet-charter.html`, Page accessed April 2009.

[3] C. Perkins, E. M. Belding-Royer, and S. Das. Ad hoc On-Demand Distance Vector (AODV) Routing. Experimental RFC 3561.

[4] D. B. Johnson, Y. Hu, and D. A. Maltz. The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4. Experimental RFC 4728.

[5] T. Clausen and P. Jacquet. Optimized Link State Routing Protocol (OLSR). Experimental RFC 3626.

[6] R. Ogier, F. Templin, and M. Lewis. Topology Dissemination Based on Reverse-Path Forwarding (TBRPF). Experimental RFC 3684.

[7] I. Chakeres and C. Perkins. Dynamic MANET On-demand (DYMO) Routing. IETF Internet-Draft.

[8] T. Clausen, C. Dearlove, and P. Jacquet. The Optimized Link State Routing Protocol version 2. IETF Internet-Draft.

[9] *IEEE Std 802.11e-2005 Part11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications - Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements*, 2005.

[10] *IEEE Std 802.11-1999 Part11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, 1999.

[11] W. Stallings. IEEE 802.11: Wireless LANs from a to n. IEEE Computer Society, Sep/Oct 2004.

[12] Wikipedia, the free encyclopedia.
`http://en.wikipedia.org/wiki/802.11`, Page accessed April 2009.

[13] D. Lang. A comprehensive overview about selected Ad Hoc Networking Routing Protocols. Master's thesis, Department of Computer Science, Technische Universität München, München, Germany, 2003.

[14] X. Zou, B. Ramamurthy, and S. Magliveras. Routing Techniques in Wireless Ad Hoc Networks - Classification and Comparison. In *Proc. of the 6th World Multi Conference on Systemics, Cybernetics and Informatics (SCI)*, Orlando, USA, Jul 2002.

[15] X. Hong, K. Xu, and M. Gerla. Scalable Routing Protocols for Mobile Ad Hoc Networks. *IEEE Network*, 16(4):11–21, Jul/Aug 2002.

[16] L. M. Feeney. A Taxonomy for Routing Protocols in Mobile Ad Hoc Networks. Technical Report T99/07, Swedish Institute of Computer Science, Oct 1999.

[17] T. Clausen, C. Dearlove, J. Dean, and C. Adjih. Generalized Mobile Ad Hoc Network (MANET) Packet/Message Format. Standard RFC 5444.

[18] T. Clausen, C. Dearlove, and J. Dean. MANET Neighborhood Discovery Protocol (NHDP). IETF Internet-Draft.

[19] S. McCanne and S. Floyd. The Network Simulator - ns-2.
`http://www.isi.edu/nsnam/ns`.

[20] C. Perkins. IP Mobility Support for IPv4. Standard RFC 3220.

[21] D. Johnson, C. Perkins, and J. Arkko. Mobility Support in IPv6. Standard RFC 3775.

[22] U. Jönsson, F. Alriksson, T. Larsson, P. Johansson, and G. Q. Maguire. MIPMANET - Mobile IP for Mobile Ad Hoc Networks. In *Proc. of the 1st Workshop on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Boston, USA, Aug 2000.

[23] E. M. Belding-Royer, Y. Sun, and C. Perkins. Global Connectivity for IPv4 Mobile Ad Hoc Networks. IETF Internet-Draft.

[24] J. Xi and C. Bettstetter. Wireless Multihop Internet Access: Gateway Discovery, Routing and Addressing. In *Proc. of International Conference on Third Generation Wireless and Beyond (3Gwireless)*, San Francisco, USA, May 2002.

[25] R. Wakikawa, J. Malinen, C. Perkins, A. Nilsson, and A. J. Tuominen. Global Connectivity for IPv6 Mobile Ad Hoc Networks. IETF Internet-Draft.

[26] A. Weyland, T. Staub, and T. Braun. Comparison of motivation-based cooperation mechanisms for hybrid wireless networks. *Computer Communications*, 29(13-14):2661–2670, Jan 2006.

[27] F. P. Setiawan, S. H. Bouk, and I. Sasae. An Optimum Multiple Metrics Gateway Selection Mechanism in MANET and Infrastructured Networks Integration. In *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*, Las Vegas, USA, Mar 2008.

[28] E. H.-K. Wu, W.-L. Chang, C.-W. Chen, and K. C. Hsu. Gateway Zone Multi-path Routing in Wireless Mesh Networks. In *Proc. of the 4th international conference on Ubiquitous Intelligence and Computing (UIC)*, Hong Kong, China, Jul 2007.

[29] R. N. B. Rais, T. Turletti, and K. Obraczka. Coping with Episodic Connectivity in Heterogeneous Networks. In *Proc. of the 11th International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, Vancouver, Canada, Oct 2008.

[30] A. Trivino-Cabrera, B. Ruiz-Villalobos, and E. Casilari. Adaptive Gateway Discovery in Hybrid MANETs. In *Proc. of the 7th international Workshop on Applications and Services in Wireless Networks (ASWN)*, Santander, Spain, May 2007.

[31] U. Javaid, F. Rasheed, D.-E. Meddour, and T. Ahmed. Adaptive Distributed Gateway Discovery in Hybrid Wireless Networks. In *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*, Las Vegas, USA, Mar 2008.

[32] R. Kumar, M. Misra, and A. K. Sarje. An Efficient Gateway Discovery in Ad Hoc Networks for Internet Connectivity. In *Proc. of International Conference on Computational Intelligence and Multimedia Applications (ICCIMA)*, Sivakasi, India, Dec 2007.

[33] B.-N. Park, W. Lee, and C. Lee. QoS-aware Internet access schemes for wireless mobile ad hoc networks. *Computer Communications*, 30(2):369–384, Jan 2007.

[34] A. J. Yuste, F. D. Trujillo, A. Trivio, E. Casilari, and A. Daz-Estrella. An Adaptive Gateway Discovery for Mobile Ad Hoc Networks. In *Proc. of the 5th ACM International Workshop on Mobility Management and Wireless Access (MobiWac)*, Chania, Crete Island, Greece, Oct 2007.

[35] M. Bernard. Gateway Detection and Selection for Wireless Multihop Internet Access. Master's thesis, Olching, Germany, 2002.

[36] ns-2 Contributed Code.
`http://nsnam.isi.edu/nsnam/index.php/Contributed\_Code`.

[37] J. Yoon, M. Liu, and B. Noble. Random Waypoint Considered Harmful. In *Proc. of IEEE Conference on Computer Communications (INFOCOM)*, San Francisco, USA, Mar 2003.

[38] W. R. Stevens. *TCP/IP Illustrated, Volume 1 - The Protocols*. Addison Wesley, 1994.

[39] J. Broch, D. Maltz, D. B. Johnson, Y. Hu, and J. Jetcheva. A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols. In *Proc. of the 4th Annual International Conference on Mobile Computing and Networking (MobiCom)*, Dallas, USA, Oct 1998.

[40] C. Carter, S. Yi, and R. Kravets. ARP Considered Harmful: Manycast Transactions in Ad Hoc Networks. In *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*, New Orleans, USA, Mar 2003.

[41] S. Perur, L. Wadia, and S. Iyer. Improving the Performance of MANET Routing Protocols using Cross-Layer Feedback. In *Proc. of the 6th International Conference on Information Technology (CIT)*, Bhubaneshwar, India, Dec 2003.

[42] H. Koga, S. Kashihara, Y. Fukuda, K. Iida, and Y. Oie. Quality-Aware VoWLAN Architecture and its Quantitative Evaluation. *IEEE Wireless Communications, Special Issue on Voice over Wireless Local Area Network*, 13(1):52–59, Feb 2006.

[43] Y. Yang and R. Kravets. Distributed QoS Guarantees for Realtime Traffic in Ad Hoc Networks. In *Proc. of the 1st IEEE Sensor and Ad Hoc Communications and Networks (SECON)*, Santa Clara, USA, Oct 2004.

[44] S. H. Shah, K. Chen, and K. Nahrstedt. Dynamic Bandwidth Management in Single-Hop Ad Hoc Wireless Networks. *Mobile Networks and Applications*, 10(1-2):199–217, Feb 2005.

[45] C. W. Ahn, C. G. Kang, and Y. Z. Cho. Soft Reservation Multiple Access with Priority Assignment (SRMA/PA): A Novel MAC Protocol for QoS-Guaranteed Integrated Services in Mobile Ad-Hoc Networks. In *Proc. of the 52nd Vehicular Technology Conference (VTC 2000)*, Boston, USA, Sep 2000.

[46] C. R. Lin and M. Gerla. Adaptive Clustering for Mobile Wireless Networks. *IEEE Journal on Selected Areas in Communications*, 52(7):1265–1275, Sep 1997.

[47] S. Sivavakeesar and G. Pavlou. Quality of Service Aware MAC Based on IEEE 802.11 for Multihop Ad-Hoc Networks. In *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*, Atlanta, USA, Mar 2004.

[48] Z. Ningyu, Z. Dongfeng, and D. Hongwei. Analysis of Multi-Channel and Slotted Random Multi-Access Protocol with Two-Dimensional Probability for Ad Hoc Network. *Tsinghua Science and Technology*, 13(6):747–753, Dec 2008.

[49] D. Maniezzo, G. Pau, M. Gerla, G. Mazzini, and K. Yao. T-MAH: A Token Passing MAC protocol for Ad Hoc Networks. In *Proc. of the 1st Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, Chia, Sardegna, Italy, Sep 2002.

[50] C.-H. Lin. *Analysis of Multi-Channel and Slotted Random Multi-Access Protocol with Two-Dimensional Probability for Ad Hoc Network*. PhD thesis, University of California, Los Angeles, 1996.

[51] L. Romdhani, Q. Ni, and T. Turletti. Adaptive EDCF: Enhanced Service Differentiation for IEEE 802.11 Wireless Ad-Hoc Networks. In *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*, New Orleans, USA, Mar 2003.

[52] A. Iera, A. Molinaro, G. Ruggeri, and D. Tripodi. Improving QoS and Throughput in Single- and Multihop WLANs through Dynamic Traffic Prioritization. *IEEE Network*, 19(4):35–44, Jul/Aug 2005.

[53] L. Blasi, G. Boggia, P. Camarda, L. Carbone, and L. A. Grieco. Extended EDCA for Providing Bounded Delay Services in 802.11e WLANs. In *Proc. of ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, Torremolinos, Spain, Oct 2006.

[54] Y.-J. Wu, J.-H. Chiu, and T.-L. Sheu. A Modified EDCA with Dynamic Contention Control for Real-Time Traffic in Multi-hop Ad Hoc Networks. *Journal of Information Science and Engineering*, 24(4):1065–1079, Jul 2008.

[55] W. Pattara-Atikom and P. Krishnamurthy. Distributed Mechanisms for Quality of Service in Wireless LANs. *IEEE Wireless Communications Magazine*, 10(3):26–34, Jun 2003.

[56] A. Banchs and X. Perez. Providing Throughput Guarantees in IEEE 802.11 Wireless LAN. In *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*, Orlando, USA, Mar 2002.

[57] A. Banchs and X. Perez. Distributed Weighted Fair Queuing in IEEE 802.11 Wireless LAN. In *Proc. of IEEE International Conference on Communications (ICC)*, New York, USA, Apr 2002.

[58] N. H. Vaidya, P. Bahl, and S. Gupta. Distributed Fair Scheduling in a Wireless LAN. In *Proc. of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom)*, Boston, USA, Aug 2000.

[59] W. Pattara-Atikom, S. Banerjee, and P. Krishnamurthy. Starvation Prevention and Quality of Service in Wireless LANs. In *Proc. of the 5th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, Pittsburgh, USA, Oct 2002.

[60] M. Shreedhar and G. Varghese. Efficient Fair Queuing Using Deficit Round-Robin. *IEEE/ACM Transactions on Networking*, 4(3):375–385, Jun 1996.

[61] G. R. Hiertz and J. Habetha. A new MAC Protocol for a wireless multi-hop broadband system beyond IEEE 802.11. In *Proc. of Wireless World Research Forum (WWRF)*, Zurich, Switzerland, Jul 2003.

[62] *IEEE P802.11s/D1.07 Part11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications - Amendment: Mesh Networking*, 2007.

[63] E. Carlson, C. Prehofer, C. Bettstetter, and A. Wolisz. A Distributed End-to-End Reservation Protocol for IEEE 802.11-based Wireless Mesh Networks. *Journal on Selected Areas in Communications (JSAC), Special Issue on Multi-Hop Wireless Mesh Networks*, 24(11):2018–2027, Nov 2006.

[64] D. Gao, J. Cai, and L. Zhang. Physical Rate Based Admission Control for HCCA in IEEE 802.11e WLANs. In *Proc. of the IEEE 19th International Conference on Advanced Information Networking and Applications (AINA)*, Taipei, Taiwan, Mar 2005.

[65] W. F. Fan, D. Gao, D. H. K. Tsang, and B. Bensaou. Admission Control for Variable Bit Rate traffic in IEEE 802.11e WLANs. In *Proc. of the 13th IEEE Workshop on Local and Metropolitan Area Networks (LANMAN)*, Mill Valley, USA, Apr 2004.

[66] W. F. Fan, D. H. K. Tsang, and B. Bensaou. Admission Control for Variable Bit Rate traffic using variable Service Interval in IEEE 802.11e WLANs. In *Proc. of the 13th International Conference on Computer Communications and Networks (ICCCN)*, Rosemont, USA, Oct 2004.

[67] P. Ansel, Q. Ni, and T. Turletti. FHCF: A Simple and Efficient Scheduling Scheme for IEEE 802.11e Wireless LAN. *Mobile Networks and Applications*, 11(3):391–403, Jun 2006.

[68] A. Grilo, M. Macedo, and M. Nunes. A Scheduling Algorithm for QoS Support in IEEE 802.11e Networks. *IEEE Wireless Communications Magazine*, 10(3):36–43, Jun 2003.

[69] D. Skyrianoglou, N. Passas, and A. Salkintzis. ARROW: An Efficient Traffic Scheduling Algorithm for IEEE 802.11e HCCA. *IEEE Transactions on Wireless Communications*, 5(12):3558–3567, Dec 2006.

[70] S. Wiethölter, M. Emmelmann, C. Hoene, and A. Wolisz. TKN EDCA Model for ns-2. Technical Report TKN-06-003, Telecommunication Networks Group, Technische Universität Berlin, Jun 2006.

[71] M. Moreton. IEEE 802.11e patch for ns-2. Original download location is offline:
`http://cvs.sourceforge.net/viewcvs.py/ns2-wlan-patch/patch\_802\_11/`.

[72] J. William and T. Robinson. An Analytical Model for the Service Delay Distribution of IEEE 802.11e Enhanced Distributed Channel Access. Master's thesis, Simon Fraser University, Vancouver, Canada, 2005.

[73] J. Wallerich. Design and Implementation of a WWW Workload Generator for the NS-2 Network Simulator. Master's thesis, Saarland University, Saarbrücken, Germany, 2001.

[74] *ITU-T Recommendation G.711 - Pulse Code Modulation (PCM) of Voice Frequencies*, 1988.

[75] S. Heecheol, B. C. Kim, J. Y. Lee, and S. Hwang. IEEE 802.11-based Wireless Mesh Network Testbed. In *Proc. of the 16th IST Mobile and Wireless Communications Summit*, Budapest, Hungary, Jul 2007.

[76] I. F. Akyildiz, X. Wang, and W. Wang. Wireless Mesh Networks: A Survey. *Computer Networks*, 47(4):445–487, Aug 2005.

[77] B.-N. Cheng, M. Yuksel, and S. Kalyanaraman. Directional Routing for Wireless Mesh Networks: A Performance Evaluation. In *Proc. of IEEE Workshop on Local and Metropolitan Area Networks (LANMAN)*, Princeton, USA, Jun 2007.

[78] S. Ganguly, V. Navda, K. Kim, A. Kashyap, D. Niculescu, R. Izmailov, S. Hong, and S. R. Das. Performance Optimizations for Deploying VoIP Services in Mesh Networks. *IEEE Journal on Selected Areas in Communications*, 24(11):2147–2158, Nov 2006.

[79] T.-J. Tsai and J.-W. Chen. IEEE 802.11 MAC Protocol over Wireless Mesh Networks: Problems and Perspectives. In *Proc. of the 19th International Conference on Advanced Information Networking and Applications (AINA)*, Taipei, Taiwan, Mar 2005.

[80] B.-N. Park, W. Lee, S. Ahn, and S. Ahn. QoS-Driven Wireless Broadband Home Networking Based on Multihop Wireless Mesh Networks. *IEEE Transactions on Consumer Electronics*, 52(4):1220–1228, Nov 2006.

[81] N. Bisnik and A. Abouzeid. Delay and Throughput in Random Access Wireless Mesh Networks. In *Proc. of IEEE International Conference on Communications (ICC)*, Istanbul, Turkey, Jun 2006.

[82] A. State, M. Livingston, G. Hirota, W. Garrett, M. Whitton, H. Fuchs, and E. Pisano. Techniques for Augmented-Reality Systems: Realizing Ultrasound-Guided Needle Biopsies. In *Proc. of the 23rd International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, New Orleans, USA, Aug 1996.

[83] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre. Recent Advances in Augmented Reality. *IEEE Computer Graphics and Applications*, 21(6):34–47, Nov/Dec 2001.

[84] W. Piekarski and B. Thomas. ARQuake: The Outdoor Augmented Reality Gaming System. *Communications of the ACM*, 45(1):36–38, Jan 2002.

[85] J. Bulman, B. Crabtree, A. Gower, A. Oldroyd, M. Lawson, and J. Sutton. Mixed Reality Applications in Urban Environments. *BT Technology Journal*, 22(3):84–94, Jul 2004.

[86] M. Bajura, H. Fuchs, and R. Ohbuchi. Merging Virtual Reality with the Real World: Seeing Ultrasound Imagery within the Patient. In *Proc. of the 19th International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, Chicago, USA, Jul 1992.

[87] S. Feiner, B. MacIntyre, and D. Seligmann. Knowledge-based Augmented Reality. *Communications of the ACM*, 36(7):52–62, Jul 1993.

[88] M. Tuceryan, D. Greer, R. Whitaker, D. Breen, C. Crampton, E. Rose, and K. Ahlers. Calibration Requirements and Procedures for Augmented Reality. *IEEE Transactions on Visualization and Computer Graphics*, 1(3):255–273, Sep 1995.

[89] S. Feiner, B. MacIntyre, M. Haupt, and E. Solomon. Windows on the World: 2D Windows for 3D Augmented Reality. In *Proc. of the 6th Annual ACM Symposium on User Interface Software and Technology (UIST)*, Atlanta, USA, Nov 1993.

[90] J. Rekimoto and K. Nagao. The World through the Computer: Computer Augmented Interaction with Real World Environments. In *Proc. of the 8th Annual ACM Symposium on User Interface Software and Technology (UIST)*, Pittsburgh, USA, Nov 1995.

[91] D. Drascic, J. Grodski, P. Milgram, K. Ruffo, P. Wong, and S. Zhai. ARGOS: A Display System for Augmenting Reality. In *Proc. of International Conference on Human-Computer Interaction (INTERCHI)*, Amsterdam, Netherlands, Apr 1993.

[92] P. Maes. Artificial Life Meets Entertainment: Lifelike Autonomous Agents. *Communications of the ACM*, 38(11):108–114, Nov 1995.

[93] A. D. Cheok, S. W. Fong, K. H. Goh, X. Yang, W. Liu, and F. Farzbiz. Human Pacman: A Sensing-Based Mobile Entertainment System with Ubiquitous Computing and Tangible Interaction. In *Proc. of the 2nd Workshop on Network and System Support for Games (NetGames)*, Redwood City, USA, May 2003.

[94] H. Benko, E. Ishak, and S. Feiner. Collaborative Mixed Reality Visualization of an Archaeological Excavation. In *Proc. of the 3rd IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, Arlington, USA, Nov 2004.

[95] R. Wojciechowski, K. Walczak, M. White, and W. Cellary. Building Virtual and Augmented Reality Museum Exhibitions. In *Proc. of the 9th International Conference on 3D Web Technology*, Monterey, USA, Apr 2004.

[96] R. Mazzeo, C. E. Palazzi, M. Roccetti, and G. Sciutto. Computer-assisted Pigment Identification in Artworks. In *Proc. of European Conference on Internet and Multimedia Systems and Applications (EuroIMSA)*, Chamonix, France, Mar 2007.

[97] Microsoft Research. Mesh Connectivity Layer (MCL).
`http://research.microsoft.com/mesh/`.

[98] R. Draves, J. Padhye, and B. Zill. Comparison of Routing Metrics for Static Multi-Hop Wireless Networks. In *Proc. of the annual conference of the Special Interest Group on Data Communication (SIGCOMM)*, Portland, USA, Aug 2004.

[99] R. Draves, J. Padhye, and B. Zill. Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks. In *Proc. of the 10th Annual International Conference on Mobile Computing and Networking (MobiCom)*, Philadelphia, USA, Sep 2004.

[100] A. Botta, A. Dainotti, and A. Pescape. Multi-protocol and Multi-platform Traffic Generation and Measurement. In *Proc. of the 26th IEEE International Conference on Computer Communications (INFOCOM) DEMO Session*, Anchorage, USA, May 2007.

[101] VLC media player.
`http://www.videolan.org/vlc/`.

[102] FileZilla - The free FTP solution.
`http://filezilla-project.org/`.

[103] M. Gerla, K. Tang, and R. Bagrodia. TCP Performance in Wireless Multi-hop Networks. In *Proc. of the 2nd Workshop on Mobile Computing Systems and Applications (WMCSA)*, New Orleans, USA, Feb 1999.

[104] C. Tschudin, P. Gunningberg, H. Lundgren, and E. Nordström. Lessons from Experimental MANET Research. *Ad Hoc Networks Journal*, 3(2):221–233, Mar 2005.