Ad hoc qs for traffic differentiation

Amir Mowlaei

2008

Amir, M. (2008). Ad hoc qs for traffic differentiation. Master's thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/40553>

<https://doi.org/10.32657/10356/40553>

AD HOC QS FOR TRAFFIC DIFFERENTIATION

AMIR MOWLAEI

SCHOOL OF COMPUTER ENGINEERING

A thesis submitted to the Nanyang Technological University
in fulfilment of the requirement for the degree of
Master of Engineering

2008

Abstract

The prevalence of wireless network access and the explosion of new multimedia applications running across the network have made it a challenge to provide some form of Quality of Service (QoS) across the wireless network. This is reflected in the activity in the Institute of Electrical and Electronics Engineers (IEEE) to come out with the 802.11e. However, it will take a while before it becomes available for commercial deployment.

We have proposed an innovative mechanism called Ad Hoc Qs, which is a MAC-Independent traffic differentiation mechanism for QoS provisioning in an ad hoc wireless network. Ad Hoc Qs enables stations in a shared wireless network to prioritize the transmission across the media, i.e. traffic differentiation, using existing 802.11a/b/g. The mechanism resides in the Logical Link layer where packets are scheduled for transmission based on the network traffic in the wireless network channel. Based on the priority of the arriving packets from the Network layer and the time and priority of the last heard packet on the wireless channel, they are sent immediately or delay for a specified time before they are passed to the Medium Access Control (MAC) layer. The scheduler delays the low priority traffics to provide more bandwidth and less delay for the High priority traffic.

To proof the implementation viability, a prototype test-bed is set up with Ad Hoc Qs mechanism incorporated as a patch in the kernel. The experimental results show the Ad Hoc Qs mechanism is able to support service differentiation across the wireless network. This is unlike Host Qs, which is also a service differentiation mechanism residing in the Logical Link layer, which could only support service differentiation within the node.

An analytical model of the Ad Hoc Qs is developed. Based on the model, we develop an adaptive mechanism, based on the number of active stations and the type of service they request. Adaptive Ad Hoc Qs varies the slot size to minimize the delay when likely loaded while at the same time provide high utilization and service differentiation under heavy load. The Adaptive Ad Hoc Qs mechanism is simulated using QualNet and its results are compared with 802.11e. The simulation results show that the Adaptive Ad Hoc Qs surpasses the 802.11e in terms of preserving the low priority traffic from "starvation" as well as high utilization of the shared medium.

Acknowledgements

I would like to express my thanks to my supervisor, Associate Professor Lee Bu Sung, for his helps, invaluable guidance and patience during this research. I am grateful to Dr. Lim Teck Meng, whose helps and discussions helped to develop ideas put forward. I am also grateful to Agency for Science, Technology and Research (A*STAR) for all its supports.

I am very thankful to my wife, Soraya Mowlaei, and my father, Gholamhossein Mowlaei, and my mother, Naaztala Madhani, and other family members for their love and supports.

I thank God for all the blessings He has given me.

2.1	Service Differentiation Mechanisms	9
2.1.1	Enhanced Distributed Coordination Function (EDCF)	10
2.1.2	Persistent Factor DCF (P-DCF)	12
2.1.3	Distributed Weighted Fair Queue (DWFQ)	13
2.1.4	Distributed Fair Scheduling (DFS)	13
2.1.5	Distributed Deficit Round Robin (DDRR)	14
2.2	MAC-Independent Service Differentiation Mechanisms	14
2.2.1	Virtual Quality of Service MAC Layer (VQML)	14
2.2.2	Host Qs	17
2.2.3	Ad Hoc Qs	19
2.2.4	Adaptive Ad Hoc Qs	21
2.3	Concluding Remarks	22
3	Ad Hoc Qs	24
3.1	Ad Hoc Qs Algorithm	24
3.2	Illustration of Ad Hoc Qs Algorithm	25
3.2.1	Scenario 1: One slot size delay	25
3.2.2	Scenario 2: Immediate transmission	28
3.3	Experiments: Set up and Results	28
3.3.1	Set up	28
3.3.2	Results	32
3.4	Concluding Remarks	43
4	Analytical Models	44
4.1	Analytical Model of Host Qs	44

4.2	Analytical Model of Ad Hoc Qs	47
4.3	Concluding Remarks	59
5	Adaptive Ad Hoc Qs	61
5.1	Adaptive Service Differentiation	61
5.2	One Application of Adaptive Ad Hoc Qs	68
5.3	Concluding Remarks	73
6	Performance Evaluation	74
6.1	Host Qs	74
6.1.1	Simulation Environment and Settings	75
6.1.2	Simulation Results	75
6.2	Adaptive Ad Hoc Qs	77
6.2.1	Simulation Environment	78
6.2.2	Results	80
6.3	Comparison between Adaptive Ad Hoc Qs and IEEE 802.11e	86
6.3.1	IEEE 802.11e EDCA	86
6.3.2	Performance Comparison in Single-hop & Multi-hop	88
6.4	Concluding Remarks	92
7	Conclusion and Future Work	94
7.1	Future Work	96
A	Publications	98
	Bibliography	99

List of Tables

2.1	VCW Priority Settings	19
2.2	Priority Mapping of Host Qs to a 3-bits Binary Field	21
3.1	Hardware used in experiments	29
3.2	Software configuration	29
3.3	Parameters are used for IEEE 802.11b MAC Layer	34
3.4	Priority traffic throughput (Mbps) for stations with 2 queues	37
3.5	Priority traffic throughput for stations with 1 queue	39
3.6	Stations throughput as slot size increases	40
3.7	Throughput of three stations each transmits 3 priority traffic	41
5.1	List of Parameters and Assumptions	69
6.1	Host Qs: Different Traffic Ratios	76
6.2	Effect of VCW_{max} on Traffic Ratios	76
6.3	Simulation Settings	78
6.4	Scenarios Settings (1 to 3 priority queue(s) per station)	79
6.5	Settings for Different Traffic Ratio Values - Scenario $S8$	82
6.6	Simulation Settings	87

6.7 Simulations Settings for Adaptive Ad Hoc Qs & 802.11e 88

6.8 Scenarios Settings (1 priority queue per station) 89

List of Figures

1.1	Infrastructure-Based Wireless Network	2
1.2	Wireless Ad Hoc Network	2
2.1	Service differentiation mechanisms in WLAN	11
2.2	VQML resides between MAC and Network layer	15
2.3	Internal architecture of QoS management unit	16
2.4	Block diagram of the Host Qs's LLC layer	18
2.5	Block diagram of the Ad Hoc Qs's LLC Layer	20
3.1	Pseudo code of the Ad Hoc Qs algorithm	25
3.2	Ad Hoc Qs Scenario 1: One slot size delay	26
3.3	Head of Line (HOL) blocking	27
3.4	Ad Hoc Qs Scenario 2: Immediate transmission	27
3.5	Test-bed setup	30
3.6	RTS/CTS handshaking	32
3.7	Each station sends only one type of priority traffic	35
3.8	Total throughput of different priority traffic	36
3.9	Each station sends two types of priority traffics	37
3.10	Throughput ratio of two stations with two different traffic priorities	39

3.11	Station throughput vs slot size (msec)	41
3.12	Throughput of three stations each has all 3 priority queues	42
4.1	Average transit delay: 1500 bytes/packet and 802.11b with RTS/CTS	50
5.1	Pseudo code of the Adaptive Ad Hoc Qs	64
5.2	Flowchart of the Adaptive Ad Hoc Qs	65
5.3	Block diagram of the Adaptive Ad Hoc Qs	66
5.4	Number of 80 Kbps traffic flows which can be supported when $\theta = 4$	72
5.5	Number of 80 Kbps traffic flows which can be supported when $\theta = 8$	72
6.1	Single-hop scenario	79
6.2	Average throughput of each priority queue per station: $\theta = 4$ and $\lambda = 2$	81
6.3	Average throughput of each priority queue per station: $\theta = 7$ and $\lambda = 3.5$	81
6.4	Average throughput of each priority queue per station for scenario $S8$	82
6.5	Average throughput of each priority queue per station for scenario $S8$, $\theta = 8$ and $\lambda = 4$	83
6.6	Average End-to-End delay in under-load situation	85
6.7	Ad Hoc Qs and 802.11e service differentiation comparison - (single-hop)	90
6.8	Multi-hop scenario	90
6.9	Ad Hoc Qs and 802.11e service differentiation comparison - (5 hops)	91

List of Symbols

D^{slot}	slot size
CW_{min}	minimum contention window size
CW_{max}	maximum contention window size
P	persistent factor
T_i	actual throughput of the i th station
W_i	weight of the i th station
R_i	ratio of actual throughput of the i th station to its weight
BI_i	backoff interval of the i th flow
L_i	packet length of the i th flow
φ_i	weight of the i th flow
ρ_i	a random variable uniformly distributed in (0.9,1.1)
$Q_{i,j}$	service quantum rate of the i th throughput class at the j th station
$DC_{i,j}$	deficit counter of the i th throughput class at the j th station
$IFS_{i,j}$	InterFrame space of the i th throughput class at the j th station
VCW_{min}	minimum virtual contention window size
VCW_{max}	maximum virtual contention window size
VCW_i	virtual contention window size of the i th queue
t_{b_i}	backoff time of the i th queue

P_l	the last heard packet's priority on the wireless channel
A_l	the time when the last heard packet on the wireless channel is heard
D_k	the delay imposed to packet k by the Ad Hoc Qs's scheduler
P_k	the priority of packet k
T^C	the current time
T_k^D	the departure time of packet k to the MAC layer
S_0	the station that transmits packet with priority 0
S_1	the station that transmits packet with priority 1
$VCW_{min,i}$	minimum virtual contention window size of the priority i
$VCW_{max,i}$	maximum virtual contention window size of the priority i
α	probability that the High priority queue wins the virtual contention
β	probability that the Medium priority queue wins the virtual contention
γ	probability that the Low priority queue wins the virtual contention
τ	probability that a station transmits a packet in a randomly chosen slot time
p	probability of a collision seen by a packet being transmitted on the channel
W	minimum contention window
m	the maximum backoff stage
n	the number of active stations
d_j	the delay of a packet been successfully transmited in the j backoff stage
q_j	probability of a packet been successfully transmited in the j backoff stage
$E[d_j]$	the average packet delay in the j backoff stage
$E'[slot]$	the average length of a slot time when the remaining $n - 1$ stations compete for the channel
T_C	The time duration the channel is sensed busy during a collision

T_s	the time duration the channel is sensed busy during a successful transmission
σ	the period of an empty slot
P'_{tr}	the probability that at least one station out of $n - 1$ transmit in the considered slot time
P'_s	the probability that a transmission occurring on the channel is successful
$E_N[D]$	the average packet delay
N_i	the number of stations that transmit priority i
D_H	the average transit delay of the High priority packets
D_M	the average transit delay of the Medium priority packets
D_L	the average transit delay of the Low priority packets
D_N	the average transit delay of the packets of N stations contend for the channel
ρ_1	the probability for the High priority packets contention for the channel
ρ_2	the probability for the Medium priority packets contention for the channel
ρ_3	the probability for the Low priority packets contention for the channel
ρ_4	the probability for the Medium or Low priority packets contention for the channel
N'_H	equivalent number of the High priority stations
N'_M	equivalent number of the Medium priority stations
N'_L	equivalent number of the Low priority stations
H^x_P	the total number of the High priority packets are transmitted on the channel by station x in time duration of T
M^x_P	the total number of the Medium priority packets are transmitted on the channel by station x in time duration of T

L_P^x	the total number of the Low priority packets are transmitted on the channel by station x in time duration of T
θ	the ratio of each High priority queue throughput to each Low priority queue throughput
λ	the ratio of each High priority queue throughput to each Medium priority queue throughput
H_P	the total number of the High priority packets are transmitted on the channel in time duration of T
M_P	the total number of the Medium priority packets are transmitted on the channel in time duration of T
L_P	the total number of the Low priority packets are transmitted on the channel in time duration of T
B_H	the share of the bandwidth consumed by the High priority packets within a station
B_M	the share of the bandwidth consumed by the Medium priority packets within a station
B_L	the share of the bandwidth consumed by the Low priority packets within a station
H_{BW}	the required bandwidth for transmitting a 64 Kbps Voice stream over IP
$AC[i]$	the Access Category i
$aSlotTime$	the duration of one slot time

List of Abbreviations

AC	Access Point
AIFS	Arbitrary InterFrame Space
AIFSN	Arbitrary InterFrame Space Number
API	Application Programming Interface
A*STAR	Agency for Science, Technology and Research
BI	Backoff Interval
BS	Base Station
CAC	Call Admission Control
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CTS	Clear To Send
CW	Contention Window
DCF	Distributed Coordination Function
DDRR	Distributed Deficit Round Robin

DiffServ	Differentiated Service
DFS	Distributed Fair Scheduling
DWFQ	Distributed Weighted Fair Queue
EDCA	Enhanced Distributed Channel Access
EDCF	Enhanced Distributed Coordination Function
ESSID	Extended Service Set Identifier
FTP	File Transfer Protocol
HCF	Hybrid Coordination Function
HTTP	Hypertext Transfer Protocol
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IFG	Inter-Frame Gap
IntServ	Integrated Service
LLC	Logical Link Control
MAC	Medium Access Control
OS	Operating System
PCU	Packet Classifying Unit
P-DCF	Persistent Factor DCF

PHB	Per-Hop-Behavior
PHY	Physical
QoS	Quality of Service
RSVP	Resource ReSerVation Protocol
RTS	Ready To Send
SIFS	Short Inter-Frame Space
TCU	Transmission Control Unit
ToS	Type of Service
UDP	User Datagram Protocol
VCW	Virtual Contention Window
VoIP	Voice over Internet Protocol
VQML	Virtual Quality of Service MAC Layer

Chapter 1

Introduction

1.1 Motivation

Deployment of Wireless Local Area Networks (WLANs) based on the IEEE 802.11 standard are growing day by day due to their convenience in implementation. WLAN has two modes of operation: Infrastructure-Based Wireless Networks and Ad Hoc Wireless Networks. The former uses a Base Station (BS) or Access Point (AP) to coordinate communication between stations as well as with wired network as shows in Figure 1.1. In the later mode of operation each station forwards packets for other stations in the network. Figure 1.2 depicts an example of Ad Hoc Wireless Network.

Ad Hoc mode of operation is gaining more acceptance. The advantages include ease of deployment and independency from infrastructure which makes it attractive for some applications like disaster relief and battle field and where network infrastructure is not available.

On the other hand, the demand for multimedia traffics like VoIP (Voice over Internet

Figure 1.1: Infrastructure-Based Wireless Network

Figure 1.2: Wireless Ad Hoc Network

Protocol) and Video Conferencing are consistently increasing. This type of traffic is delay sensitive and consumes more bandwidth compared to the best effort traffic like HTTP (Hypertext Transfer Protocol) and FTP (File Transfer Protocol). To provide the required bandwidth and/or delay for these traffics, proper QoS mechanism is essential.

Currently, most of the WLANs (both Infrastructure-based and Ad Hoc) are using the IEEE 802.11 standard. With various extensions, like 802.11a/b/g [8, 9, 10] that provide a variety of features, it has been able to satisfy the customers diverse demands. These standards do not support network QoS that are necessary for multimedia traffics. Consequently, the IEEE established the task group E to provide QoS support for the IEEE 802.11 standard. The last draft of IEEE 802.11e [11] was released in September 2005. At the moment only a few WLAN cards from some manufacturers like Atheros and EDiMAX comply with the IEEE 802.11e standard.

Meanwhile, a MAC-Independent mechanism that can provide QoS requirements for multimedia traffics independent of WLAN cards is a big advantage. A software-based approach introduces more flexibility as the design can be easily upgraded or changed. In addition, such a mechanism can enhance the operation of QoS-enabled MAC protocols like the IEEE 802.11e by providing more adjustable parameters.

1.2 Quality of Service in WLAN

Quality of service in our context refers to the guarantee from the network to provide a desired level of services. These guarantees typically take the form of minimum bandwidth or maximum delay guarantees. QoS parameters in WLAN include the followings:

- **Access delay:** It is the time from when a packet reaches a MAC layer until it is

successfully transmitted.

- **Throughput:** It is the amount of data transferred from one place to another in a specified amount of time.
- **Channel utilization:** It presents the amount of time in which the medium are effectively used for data transmission.
- **Collision rate:** It is the amount of packet collisions that occur in a network per time unit

In ad-hoc networks, it is extremely difficult to provide guarantees to the services due to the variety of reasons. First, variations in physical medium causes the link characteristics of an ad-hoc network to change frequently. Second, it is difficult to maintain the stability in the routing due to the dynamic nature of the network. Third, interference from nearby stations affects the existing services. Finally, limited bandwidth and no centralized control make it difficult to provide QoS in an ad-hoc network.

However, for providing multimedia services such as video sharing and gaming over ad-hoc networks QoS is still a desired characteristic. Mechanisms for providing QoS in traditional networks especially Internet are categorized into two main groups: Integrated Services (IntServ) [6] and Differentiated Services (DiffServ) [4].

The main idea of IntServ is to provide end-to-end per-flow QoS guarantees. It was targeted both unicast and multicast applications. The main IETF (Internet Engineering Task Force) protocol for the IntServ architecture is Resource ReSerVation Protocol (RSVP) [7]. It is used for making the reservations; other protocols are used for sending the data. All stations on the network capable of sending QoS data send a PATH message periodically, which spread out through the network. The PATH message carries the TSpec (Traffic

Specifications) and AdSpec (Advertise Specifications) from source towards the receivers. The TSpec includes the description of the traffic characteristics (peak rate, minimum unit and maximum size), while the AdSpec describes the properties of the data path such as availability of a specific QoS and the requirement of sending applications.

The receiver uses RESV message for setting up reservations in the routers along the data path towards the sender. The RESV includes a FlowSpec and a FilterSpec. The FlowSpec describes the expected level of traffic to be supported and the QoS service level expected on each router in terms of bandwidth, packet delay, or packet loss. The routers between the sender and receiver have to decide if they can support the reservation being requested, and, if they cannot, they send a reject message to the receiver. IntServ does not scale well due to enormous router memory requirement and great processing complexity; consequently, the interest in using IntServ decreased gradually. This has led the IETF to introduce the DiffServ model which is simpler and more scalable.

The main idea of DiffServ is to provide service differentiation based on specified service types. Each data frame is placed into a limited number of traffic classes, rather than differentiating network traffic based on the requirements of an individual flow. Routers on the network are configured to differentiate traffic based on its class, assuring prioritized treatment for higher-priority traffic on the network. In DiffServ, the Edge Router is responsible for traffic classification and conditioning and the Core Router forward packets on a Per-Hop-Behavior (PHB) manner. DiffServ advantages include simplicity, easy implementation, and high scalability.

WLANs bandwidth is much lower than traditional wired LANs, 54 Mbps compared to 10 Gbps. QoS support in WLANs becomes important due to limited bandwidth in WLANs and increasing demands for delay and bandwidth sensitive traffics over wireless links.

The desired characteristics of a WLAN QoS mechanism are as follows [26]:

- ***Fairness***: To prevent starvation of traffic classes, specially when QoS mechanism provides relative throughput support.
- ***Compatible with the IEEE 802.11 MAC***: Consequently the proposed algorithm is compatible with these dominant MAC protocols and therefore easier to implement and adopt.
- ***Scalability***: It ensures that for a large number of active stations the QoS support will not degrade.
- ***Low variation of throughput***: The multimedia applications are sensitive to variation of throughput.
- ***Minimum requirements of admission controls***: A simple admission control is needed to ensure that the QoS is maintained.
- ***Distributed***: One of the benefits that a distributed mechanism has is that it can continue to operate even if some of the participant stations are missing.
- ***Ability to adapt to unexpected network load***: Two scenarios of network load are under-load and overload. In under-load scenario the requested throughput of a traffic class that demands only relative throughput should be fully satisfied. In overload scenario the mechanism should be able to allocate fair throughput to the traffic classes which require relative throughput and fully satisfies the traffic classes which demand absolute throughput.
- ***Minimum computational complexity***: Computational complexity should be kept to the minimal to ensure practicality of implementation.

Much works have been carried out in the area of QoS provisioning in ad-hoc networks by designing QoS aware routing protocols [29] and QoS-enabled MAC protocols [22]. In our research we have focused on a MAC-Independent traffic differentiation mechanism for QoS provisioning in the ad-hoc network.

1.3 Research Contributions

Most of the proposed mechanisms to provide QoS support in wireless ad-hoc networks require some modifications in the MAC layer. Our contribution is to develop and evaluate a novel MAC-Independent adaptive service differentiation mechanism for wireless ad-hoc networks, called Adaptive Ad Hoc Qs. Adaptive Ad Hoc Qs is implemented in LLC layer and has the following main features:

- Adaptive Ad Hoc Qs provides consistent differentiation amongst 3 traffic priorities and also prevents Low priority traffic from starvation in overload conditions.
- Adaptive Ad Hoc Qs is a MAC-Independent mechanism which is implemented just by software upgrade which is more desirable due to its simplicity and cost saving.

1.4 Thesis Structure

The rest of this thesis is organized as follows:

Chapter 2 presents some traffic differentiation mechanisms to support QoS in WLANs and ad-hoc networks. This chapter also includes previous works on MAC-Independent traffic differentiation in WLAN: VQML [21], Host Qs [13], and Ad Hoc Qs [24]. Adaptive Ad Hoc Qs is briefly introduced.

Chapter 3 introduces Ad Hoc Qs algorithm and a test-bed which was developed in order to investigate the characteristics of the Host Qs and Ad Hoc Qs algorithms.

Chapter 4 presents two analytical models each for Host Qs and Ad Hoc Qs. Host Qs's analytical model enables us to calculate the values of the Minimum and Maximum Virtual Contention Windows for the desired service differentiation within each wireless station. Ad Hoc Qs's analytical model enables us to calculate an appropriate slot size value of each wireless station based on the number of the active station and type of service they request in order to maintain the desired service differentiation among multiple wireless stations.

Chapter 5 introduces the Adaptive Ad Hoc Qs. Using the Ad Hoc Qs's and Host Qs's analytical models, I have proposed the Adaptive Ad Hoc Qs, which selects the appropriate value of the Minimum and Maximum Virtual Contention Windows and slot size of each wireless station based on the number of active stations and the type of service requested, i.e., traffic type. Also, we use the Ad Hoc Qs's analytical model to perform some analysis on the characteristics of the Adaptive Ad Hoc Qs.

Chapter 6 presents the simulation results of Host Qs and Adaptive Ad Hoc Qs. In this chapter, we also compare the performance of the Adaptive Ad Hoc Qs and the IEEE 802.11e in terms of service differentiation.

Chapter 7 concludes the thesis and presents some possible future works.

Chapter 2

Background

In this chapter, we introduce the service differentiation mechanisms for WLANs based on the IEEE 802.11 technology, which is the dominant technology in WLANs and ad-hoc networks. Then we introduce three MAC-Independent mechanisms for traffic differentiation in ad-hoc networks: VQML [21], Host Qs [13], and Ad Hoc Qs [24]. Also, we briefly introduce the Adaptive Ad Hoc Qs, which is the enhanced version of the Ad Hoc Qs.

2.1 Service Differentiation Mechanisms

Most existing QoS mechanisms for 802.11 can be classified into three categories [31]:

- Service Differentiation
- Admission control and bandwidth reservation
- Link adaptation

We only introduce Service Differentiation mechanisms in this section because our research is mostly related to these mechanisms for QoS provisioning. Figure 2.1 shows

hierarchical taxonomy of service differentiation mechanisms in WLAN.

Basically, traffic differentiation is achieved by using two main methods: priority and fair scheduling [14, 15, 20, 25, 27, 30]. The former uses prioritized contention parameters for different traffic classes for channel access and the latter divides the channel bandwidth fairly by adjusting wait times of traffic classes in proportion according to given weights. The adjustable parameters (Objects in Figure 2.1) for both methods are Contention Window (CW) size, backoff algorithm, InterFrame space, Virtual Contention Window (VCW), Slot Size (D^{slot}), and Inter-Frame Gap (IFG). In the following sections each of these mechanisms will be introduced briefly.

2.1.1 Enhanced Distributed Coordination Function (EDCF)

The basic 802.11 MAC access scheme is the Distributed Coordination Function (DCF), which is based on the well-known Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA).

When a station has a packet to transmit, if it senses the channel has been idle for a period of time longer than or equal to a Distributed InterFrame Space (DIFS) the packet transmission may commence at the start of the immediately following slot. If the channel is still idle after this waiting period, the node transmits; otherwise, the station back off as follows. The station waits until the channel is idle for DIFS and then sets its backoff timer. The backoff timer is set to a random integer number that is selected from a contention window $[0, CW - 1]$. Initially, contention window size (CW) is set to its minimum (CW_{min}) and is doubled for the next transmission attempt every time the packet is involved in a collision. However, the contention window size cannot exceed its maximum (CW_{max}). The station decreases the counter by one every $aSlotTime$ (the duration of one slot time) after

Figure 2.1: Service differentiation mechanisms in WLAN

the channel has been idle for a DIFS, but freezes the counter when the channel becomes busy. Once the backoff timer becomes zero, the station transmits the packet.

The EDCF is an extension of DCF to include different priorities to packets. The EDCF is the main part of the IEEE 802.11e standard [11]. It uses different parameters to prioritize traffic categories: Arbitrary InterFrame Space (AIFS), maximum and minimum contention window size (CW_{min} , CW_{max}) and a multiplication factor for expanding the backoff window. The IEEE 802.11e uses separate queues for 4 defined Access Categories (ACs) corresponding to voice, video, best effort, and background traffics. In the IEEE 802.11e each queue has its own values for three parameters: Arbitrary Inter-Frame Space Number ($AIFSN[AC]$), CW_{min} , and CW_{max} . $AIFSN[AC]$ is used to calculate the Arbitrary $IFS[AC]$. Queues with smaller values for $AIFSN[AC]$ and contention windows has higher probability of winning the contention to access the channel.

In EDCF, the contention window is $[1, CW]$ instead of $[0, CW - 1]$ as in DCF. Furthermore, in EDCF the backoff counter will be decreased one slot before the end of AIFS while in DCF the backoff counter will be decreased exactly after the end of DIFS.

Virtual collision happens, if the backoff timers of two or more queues expire at the same time. In this situation, the data frame with the highest priority amongst the collided frames is transmitted and the others perform backoff stage with an enlarged CW value.

2.1.2 Persistent Factor DCF (P-DCF)[20]

In this mechanism, each traffic class has a persistent factor P (high-priority classes have smaller P). In each backoff stage, a uniformly distributed random number r is generated in every slot time. Each flow stops the backoff and starts transmission only if ($r > P$) in the current slot time, given no transmission occurs in previous slot times. Consequently,

the backoff interval is a geometrically distributed random variable with parameter P .

2.1.3 Distributed Weighted Fair Queue (DWFQ)[14, 15]

Two different algorithms using this strategy have been proposed. In the first algorithm, based on the difference between the actual and expected throughput, the backoff window size CW of any traffic flow is adjusted. If the actual throughput is lower than the expected throughput, CW will be decreased in order to increase the flow's priority, and vice versa. In the second algorithm, a ratio ($R_i = T_i/W_i$) is calculated. Where T_i is the actual throughput and W_i the corresponding weight of the i th station. By comparing its own R_i with those of others, a station can adjust its CW . For example, if its R_i is larger than those of others, it will increase its CW . However, the randomness associated with using the CW increases the variability of throughput and delay, particularly in overloaded situation. In addition, DWFQ needs an additional field in the frame header in the MAC layer to exchange the values of R_i among stations. Finally, the appropriate values of W_i for each traffic classes or methods of mapping this value to different types of traffic class are not clear [26].

2.1.4 Distributed Fair Scheduling (DFS)[15, 30]

Differentiating the backoff interval (BI) based on the packet length and traffic class is the main idea of DFS. The station with smaller BI transmits first. For the i th flow, $BI_i = \rho_i \times scaling \times factor \times L_i / \varphi_i$ where BI_i is the backoff interval, L_i is the packet length, φ_i is the weight (the weights of high priority classes are larger than that of low priority classes), and ρ_i is a random variable uniformly distributed in $[0.9, 1.1]$. ρ_i is inserted in order to minimize the collision among stations with the same backoff interval. One disadvantage is that the throughput is very sensitive to the choice of packet lengths and correct value of

weights that make it complicated to map the QoS requirement into the weight [12].

2.1.5 Distributed Deficit Round Robin (DDRR)[27]

In this algorithm, a service *quantum rate* ($Q_{i,j}$) equal to the required throughput, and a deficit counter ($DC_{i,j}$) that accumulates at the rate of $Q_{i,j}$ and is decreased by the packet length whenever a packet is transmitted are assigned to the i th throughput class at the j th station. $DC_{i,j}$ is used to figure out the InterFrame space ($IFS_{i,j}$), which is the wait time before transmission or backoff starts. A larger $DC_{i,j}$ results in a smaller $IFS_{i,j}$. In order to minimize the collision between stations with the same deficit counter, randomization of $IFS_{i,j}$ will be adopted if a backoff scheme is eliminated.

2.2 MAC-Independent Service Differentiation Mechanisms

Most of the proposed mechanisms for QoS provisioning in wireless networks [11, 14, 15, 20, 25, 27, 30] need some modifications in the legacy MAC protocols. We name the mechanisms that do not propose any modification in the MAC layer MAC-Independent mechanisms. In this section, we first introduce three MAC-Independent service differentiation mechanisms: VQML, Host Qs, and Ad Hoc Qs. Finally, we briefly introduce the Adaptive Ad Hoc Qs, which is the enhanced version of the Ad Hoc Qs.

2.2.1 Virtual Quality of Service MAC Layer (VQML) [21]

Virtual QoS MAC layer is implemented between MAC and networking layer to provide QoS. The VQML architecture was implemented in a Linux platform and tested in an experimental wireless network test-bed.

Figure 2.2 shows the VQML architecture. A VQML layer resides between Network and MAC layers, and offers software-based QoS support for networking layer. The VQML does not interrupt the operations of normal protocol stack by using a bypass connection between networking layer and MAC layer. The VQML only was deployed at the wireless stations. Therefore, wireless stations will only be able to *transmit* QoS traffic across the wired backbone via WLAN access point. This means that, only up-link part of the wireless connection will become QoS-capable.

Figure 2.2: VQML resides between MAC and Network layer [21]

QoS management unit is the heart of the VQML. Figure 2.3 shows the internal architecture of QoS management unit. QoS management unit consists of a number of units:

1. The Packet Classifying Unit (PCU): This unit receives network layer packets and

Figure 2.3: Internal architecture of QoS management unit [21]

- maps “user-level”, “application-level” or “flow-level” priority to “access-level” priority.
2. Parameter bank: It provides different settings for parameters that control the operation of PCU and queue pool. For example PCU may be forced to look for priority of packets in fields other than Type of Service (ToS) or the number of priority queues in queue pool can be changed.
 3. Queue pool: Based on the assigned priority, PCU stores the packet in one of the queues in the queue pool.
 4. Transmission Control Unit (TCU): This unit selects the packets from different queues

and transmits them to the wireless channel based on a specific algorithm. This algorithm is selected from the algorithm bank.

5. Algorithm bank: This bank provides multiple algorithms which can be implemented by TCU.

Application Programming Interface (API) Unit provides application level interface so that the functionality of VQML can be easily changed or tuned. The VQML itself has to deal with and support different wireless MAC protocols. Therefore, MAC Adaptation Unit is responsible for adapting to various MAC protocols.

The Inter-Frame Gap (IFG) [21] scheduling algorithm was implemented at TCU. It controls the duration of the inter-frame gap depending on the priority of the traffic. Higher priority traffic is assigned a smaller gap.

Hassan et al [21] used a test-bed to evaluate the performance of the mechanism. The test-bed consists of two wireless stations which send packets to an access point. The results show that VQML can achieve good QoS for the video application in the presence of background file transfer traffic. The authors did not report any experiment for large number of active stations. However, it seems that the mechanism fails to maintain service differentiation in dense networks due to the lack of adaptation mechanism.

2.2.2 Host Qs [13]

Host Qs is an emulation of IEEE 802.11e which uses 4 separate queues for 4 defined access categories corresponding to voice, video, best effort, and background. In IEEE 802.11e each queue has its own values for two parameters: CW_{min} and CW_{max} . Queues with smaller values for contention windows with higher probability will win the contention to access to the channel because with higher probability they select smaller random values for

their backoff timers. Host Qs implements these priority queues at the Logical Link Control (LLC) layer (Figure 2.4).

Figure 2.4: Block diagram of the Host Qs's LLC layer

In Host Qs [13], each queue, i , has a different priority expressed by:

- Minimum Virtual Contention Windows (VCW_{min})
- Maximum Virtual Contention Windows (VCW_{max})

Each queue, i , keeps a virtual contention window, VCW_i , and a backoff time, t_{b_i} , whereby t_{b_i} draws an integer value from the uniform distribution between 0 and VCW_i .

$$t_{b_i} = \text{uniform_variate}(0, VCW_i) \quad (2.1)$$

VCW_i is initialized with VCW_{min} .

$$VCW_i = VCW_{min} \tag{2.2}$$

The queue with the lowest t_b wins the contention and its Head-Of-Line packet will be sent to the MAC layer while the t_b of the other queues are decremented correspondingly by the value of t_b from the winning queue. After a successful packet transmission, the winning queue resets its VCW to its assigned VCW_{min} and draws a new t_b using (2.1). If the t_b of more than one queue expires at the same time (virtual collision), the queue with the highest priority wins the contention while other collided queues back off and draw a new t_b from a larger contention window is calculated as follows [13]:

$$VCW_i = \min(2 \times VCW_i + 1, VCW_{max,i}) \tag{2.3}$$

The assigned VCW values to each priority are shown in Table 2.1.

Table 2.1: VCW Priority Settings

	High	Medium	Low
VCW_{min}	3	15	31
VCW_{max}	7	255	1023

2.2.3 Ad Hoc Qs [24]

Host Qs fails to provide service differentiation when it is used in ad-hoc WLAN [24]. To achieve service differentiation among multiple stations, Lim et al [24] proposed a slotted-delay scheduler below the Host Qs’s scheduler (Figure 2.5), which ensures that service differentiation can be achieved among multiple stations in the ad-hoc WLAN. The slotted-delay scheduler schedules packets in a slotted manner to delay packets of lower priority,

thus, allowing high priority packets to have a higher share of the channel bandwidth.

Figure 2.5: Block diagram of the Ad Hoc Qs's LLC Layer

A packet that wins the contention at the Host Qs's scheduler will be passed into the slotted-delay scheduler. Each slotted-delay scheduler keeps the value of two parameters:

- The last heard packet's priority (P_l) on the WLAN channel
- The time when this packet is heard on the WLAN channel (A_l)

When a packet, k , arrives from the Host Qs's scheduler, a delay (D_k) at the Ad Hoc Qs's scheduler is calculated using (2.4).

$$D_k = D^{slot} \times \max(0, (P_l - P_k)) \quad (2.4)$$

Where D^{slot} is the time slot duration that is pre-defined and P_k is the priority of the packet k . Table 2.2 shows the mapping of priorities of Host Qs into a 3-bits binary field (precedence field of ToS). This 3-bits binary field is used to identify the priority of the packet in the MAC header. Based on the time of the last heard packet, A_l , the departure time (T_k^D) to the MAC layer for this packet is calculated using (2.5) where T^C is the current time.

$$T_k^D = \max(T^C, A_l + D_k) \quad (2.5)$$

Table 2.2: Priority Mapping of Host Qs to a 3-bits Binary Field

Priority	3-bits Binary Filed
High	011
Medium	010
Low	001

In the slotted-delay scheduler, all High priority packets will be passed immediately to the MAC layer. Lower priority packets (Medium and Low) will be passed immediately to the MAC layer if the last heard packet on the channel has the same or lower priority, or the last heard packet on the channel is heard longer than the delay (D_k). Otherwise, a packet will be delayed by the number of slot duration based on the difference between the last heard packet's priority and its own priority.

2.2.4 Adaptive Ad Hoc Qs

The adjustable parameters used by Ad Hoc Qs are Minimum and Maximum Virtual Contention Windows (VCW_{min}, VCW_{max}) and slot size (D^{slot}). In order to achieve the desired

service differentiation amongst the three priority traffics, these parameters should be adaptively changed based on the number of active stations and the type of service they request.

The original Ad Hoc Qs [24] does not provide the adaptation mechanism, which is essential to maintain the service differentiation while the number of active stations and the requested service vary.

We propose Adaptive Ad Hoc Qs (Chapter 5) based on two analytical models each for Ad Hoc Qs and Host Qs (Chapter 4). In Adaptive Ad Hoc Qs, based on the number of active stations and requested service, the values of VCW_{min} , VCW_{max} and D^{slot} are adaptively changed in order to maintain the desired service differentiation.

2.3 Concluding Remarks

In this chapter, we first surveyed some service differentiation mechanisms based on the IEEE 802.11. These mechanisms use some adjustable parameters like contention window size, backoff algorithm and InterFrame space to implement priority channel access and fair scheduling in order to provide service differentiation in WLAN. All these mechanisms need some modification of legacy 802.11 standard.

Furthermore, we introduced three MAC-Independent service differentiation mechanisms: VQML, Host Qs and Ad Hoc Qs. VQML resides between network layer and MAC layer and provides QoS for WLANs. Its Packet Classifying Unit (PCU) assigns an access-level priority to the packets according to the upper-layer priorities and inserts them to the corresponding queues in the queue pool unit. Its Transmission Control Unit (TCU) selects the packets from different queues and transmits them to the wireless channel based on a specific algorithm. Host Qs is able to differentiate services in one wireless station. It uses contention window size to prioritize the channel access of different priorities. Ad Hoc Qs

is able to provide service differentiation amongst multiple wireless stations. It uses a slotted manner scheduler to delay the lower priority packets and therefore give more channel bandwidth to higher priority packets.

Original Ad Hoc Qs dose not provide adaptation mechanism to maintain the desired service differentiation while the number of active stations and the requested service vary. We will propose Adaptive Ad Hoc Qs based on the Ad Hoc Qs's analytical model to fulfill this essential requirement in Chapter 5.

Chapter 3

Ad Hoc Qs

In this chapter, we present the Ad Hoc Qs [24] algorithm in more details. We will evaluate the characteristics of the mechanism through an experimental test-bed.

3.1 Ad Hoc Qs Algorithm

Ad Hoc Qs is proposed to provide service differentiation among multiple stations in WLANs. It uses a slotted-delay scheduler below the Host Qs's scheduler as shown in Figure 2.5. This scheduler schedules packets in a slotted manner to delay packets of lower priority, thus, allowing High priority packets to have a higher share of the channel bandwidth.

In the slotted-delay scheduler, all High priority packets will be passed immediately to the MAC layer, but lower priority packets will be passed immediately to the MAC layer if the last heard packet on the WLAN channel has the same or lower priority, or the last heard packet on the WLAN channel is heard longer than the delay (D_k). Otherwise, a packet will be delayed by the number of slot duration based on the difference between the last heard packet's priority and its own priority. Figure 3.1 shows the pseudo code of the Ad Hoc Qs

algorithm.

```

Procedure: Ad Hoc Qs
Begin
  While (packet  $k$  with priority  $P_k$  arrives from the Host Qs's scheduler) do
    Begin
      Keep the last heard packet's priority ( $P_l$ ) on the WLAN channel
      Keep the time when this packet is heard on the WLAN channel ( $A_l$ )
      If ( $P_l \leq P_k$ ) then
        Begin
          Send the packet  $k$  immediately to the MAC layer
        End
      Else
        Begin
          Calculate delay  $D_k$  :  $D_k = D^{slot} \times \max(0, (P_l - P_k))$ 
          If ( $(A_l + D_k) \leq T^C$  (Current Time)) then
            Begin
              Send the packet  $k$  immediately to the MAC layer
            End
          Else
            Begin
              Send the packet  $k$  to the MAC layer at time  $(A_l + D_k)$ 
            End
          End
        End
      End
    End
  End

```

Figure 3.1: Pseudo code of the Ad Hoc Qs algorithm

3.2 Illustration of Ad Hoc Qs Algorithm

3.2.1 Scenario 1: One slot size delay

Assume that there are 2 stations: S_1 and S_0 . S_1 transmits packets with priority 1 (Medium Priority) and S_0 transmits packets with priority 0 (Low Priority). Further, assume that the slot size (D^{slot}) is equal to the time for transmitting three packets as shown in Figure 3.2.

At time T^C (current time), the low priority station S_0 has packet to send. The last

heard packet on the channel (S_1-P1) has priority 1. Therefore according to the Ad Hoc Qs's scheduler, the packet from the low priority station will be sent to the MAC layer with a delay. The delay is equal to one slot size (D^{slot}), because the difference between the priorities is only 1. As shown in Figure 3.2, the low priority packet will be sent to the MAC layer at $A_l + D^{slot}$ to compete with other station for the channel.

Figure 3.2: Ad Hoc Qs Scenario 1: One slot size delay

In this scenario, assume that station S_0 transmits priority 0 and priority 1 packets and it has a priority 1 packet (S_0-P1) for transmission right after (S_0-P0) packet as shown in Figure 3.3. Packets are sent to the MAC layer in the order they are received by Ad Hoc Qs's scheduler, first-in-first-out (FIFO), irrespective of a packet's priority. The priority 1 packet can not be sent to the MAC layer until packet (S_0-P0) is transmitted. Hence, a higher priority packet which has come into the queue after a low priority packet, is blocked from transmitting to the MAC layer before the lower priority packet. This problem is called the Head of Line (HOL) blocking [16].

Figure 3.3: Head of Line (HOL) blocking

Figure 3.4: Ad Hoc Qs Scenario 2: Immediate transmission

3.2.2 Scenario 2: Immediate transmission

Figure 3.4 shows scenario 2. In this scenario, the low priority station S_0 is going to transmit a packet at time T^C . The last heard packet is a priority 1 packet that has been heard on the channel at A_l . Because the difference between two priorities is only 1, this packet transmission time is calculated as $A_l + D^{slot}$. But this time is behind the current time T^C , therefore this packet is sent to the MAC layer immediately.

3.3 Experiments: Set up and Results

In order to investigate the characteristics of Ad Hoc Qs algorithm, a test-bed was developed by Bramandia [18]. Table 3.1 and Table 3.2 show the hardware and the software used in experiments respectively.

3.3.1 Set up

In all the experiments, the laptops are placed so that they are in the transmission range of each other. In all experiments, one laptop acts as a receiver (Station R). The receiver laptop runs three instance of *Iperf* [1], one for each priority. *Iperf* is a tool to measure maximum TCP bandwidth. It also allows the tuning of various parameters and UDP characteristics. *Iperf* reports bandwidth, delay jitter, and packet loss. One to three laptops act as sender in the experiments. They send packets using *Iperf* to the receiver laptop at the same time.

Iperf is used as packet generator and bandwidth measurement tool. In all cases, a stream of data is pumped with the rate of 7Mbit/s for duration of 60 seconds. All of the generated packets are UDP packets with size of 1470 bytes. *Ethereal* [2], as packet sniffer,

Table 3.1: Hardware used in experiments [18]

No	Hardware	Remarks
1	Acer TravelMate 290	PC laptop acting as an ad-hoc station
2	Zyair Zyxel 802.11b Wireless LAN Card	It is used to participate actively in the channel as an ad-hoc station
3	Built in Centrino 802.11b/g	It is used to monitor the channel

Table 3.2: Software configuration [18]

No	Software	Remarks
1	Linux Mandrake 10.0 Official	Operating System (OS)
2	Kernel 2.6.3-7mdk	The default kernel provided by the OS
3	Patched Hostap 0.2.0	Driver for Zyxel Card. Patched by Queue Manager
4	Modified Queue Manager 0.6.16	It is necessary to modify the queue manager to work in kernel 2.6 and to dump packets to Prioritizer instead of MAC layer
5	Patched ipw2200 1.0.4 and ipw2200 firmware 2.3	Centrino driver and firmware. It is necessary to patch the driver to actively monitor the channel and update last Priority and last Priority Time
6	Prioritizer 3.2.1	Kernel module that implements the proposed algorithm
7	Iperf 1.7.0	It is used as bandwidth measurement tool
8	Ethereal 2.1	It is used to monitor and capture packets from a channel.

was executed in the receiver laptop (Station R) to capture the packets during some experiments for more details. The RTS/CTS threshold was set to 1000 bytes. The MAC layer fragmentation is turned off. Figure 3.5 shows the test-bed setup.

Figure 3.5: Test-bed setup

The Ad Hoc Qs algorithm is implemented as a kernel module called *Prioritizer*. The *Prioritizer* provides a *proc* file system interface to let the user communicate with the module to adjust certain parameters such as slot size, verbose debugging or print out internal information. The verbose debugging of *Prioritizer* is turned off during most experiments.

The default kernel-source provided by Mandrake 10.0 is used for the experiments. No significant changes were made in the kernel configuration; however, a recompilation is necessary to ensure that the kernel source is configured properly.

The latest firmware (ver 2.3) and the latest driver (ver 1.0.4) of ipw2200 were used to do the experiments. The Centrino wireless card is configured to be at monitor mode (*mode* = 2). The ESSID (Extended Service Set Identifier) and the channel of all the laptops are configured to be the same. The Centrino driver was patched to capture the priority and time of the last heard packet on the channel whenever a packet is detected.

A lot of changes were introduced in Host Qs implementation [5] because the Queue Manager only works with 2.4 kernel. These modifications were applied in order to upgrade Queue Manager to work with 2.6 kernel that we were using. The modifications still keep the original functionality of the Queue Manager. Also, Queue Manager was modified to redirect the packet to the *Prioritizer* instead of the MAC layer. The 2.6 kernel was used, because the version of Centrino driver (ipw2200) that supports monitor mode, only works in 2.6 kernel. No modification is done on the *hostap* driver [5] except the patches that comes with Queue Manager.

3.3.2 Results

In these experiments RTS/CTS is turned on. RTS/CTS is defined in the IEEE 802.11 standard and proposed to solve the hidden terminal problem in wireless ad-hoc networks. Figure 3.6 shows the RTS/CTS handshaking to solve the hidden terminal problem in wireless ad-hoc networks. Hidden terminal problem happens when a wireless terminal (Terminal C) is not in the transmission range of sender (Terminal A) but is in the transmission range of receiver (Terminal B). In this case, when the sender A transmitting to receiver B, the hidden terminal C can not sense the channel as busy and may start to send data to receiver B and therefore the collision happens.

Figure 3.6: RTS/CTS handshaking

To avoid collision due to hidden terminals, sender terminal A sends a short Ready To Send (RTS) packet to the receiver B. This packet is received by the receiver B and all terminals in the transmission range of the sender. The Receiver B sends a short Clear To Send (CTS) packet after receiving RTS and inform the sender A that it is ready to receive the data from the sender A. The CTS packet is received by the sender A and terminals in the transmission range of receiver B, eg terminal C. Stations which receive RTS and CTS will defer their transmission until the end of current session. This mechanism prevents collision

due to hidden terminals.

Using the above mentioned test-bed, we performed extensive experiments and measurements to investigate the characteristics of Host Qs and Ad Hoc Qs in terms of service differentiation in wireless network.

Host Qs and Ad Hoc Qs Service Differentiation Capability

We set up several experiments using the test-bed to evaluate the service differentiation characteristic of the Host Qs and Ad Hoc Qs. In experiments 1 to 3 we show that Host Qs is only able to differentiate the services within a wireless station; while the Ad Hoc Qs is capable to provide service differentiation both inside and amongst multiple wireless stations. There is only one case that Host Qs differentiates services amongst multiple wireless stations: each wireless station is transmitting all the three (High, Medium, and Low) priority traffics. In experiments 4 and 5, we investigate the effect of slot size (D^{slot}) on the performance of the Ad Hoc Qs in terms of service differentiation.

Experiment 1: One priority traffic per station

In this experiment, each station is sending only one of the three (High, Medium, and Low) priorities. The experiment is set up using four wireless stations, three senders and one receiver, within a one hop ad-hoc WLAN coverage. Station A, Station B, and Station C generates High priority, Medium priority, and Low priority UDP (User Datagram Protocol) packets respectively. These packets are sent to Station R as the unique receiver. Each simulation lasts for 60 s. For Host Qs, we used parameters as shown in Table 2.1 and for Ad Hoc Qs, we used a 7 ms slot size (D^{slot}). For the MAC layer we used IEEE 802.11b with RTS/CTS. Table 3.3 shows the parameters used for the MAC layer.

Table 3.3: Parameters are used for IEEE 802.11b MAC Layer

<i>Slot time</i>	VCW_{min}	m
$20\mu s$	32	5

Figure 3.7 shows that the three stations share the channel bandwidth equally even when the Host Qs is enabled. Host Qs sends the packets to the MAC layer upon the expiration of the backoff timer. In this case, no virtual collision happens because each station transmits only traffic of the same priority. Therefore, the backoff time is always drawn from the primary interval $[0, VCW_{min}]$. The delay is caused by Host Qs's scheduler is negligible compared to the access delay when three stations are contending for the channel. At the MAC level, the packets are contending for the channel based on the legacy IEEE 802.11 standard where the channel bandwidth is equally shared amongst the contending stations. The results of the experiment shown in Figure 3.7 verify this interpretation where the bandwidth is about equal for all 3 stations. Based on this experiment, we conclude that Host Qs fails to provide service differentiation amongst multiple stations in ad-hoc wireless network.

Figure 3.7 shows that the throughput of all 3 stations is about equal for 802.11b and Host Qs. On the other side, when Ad Hoc Qs is enabled in stations, the throughput of Station A (High priority) is more than the throughput of Station B (Medium priority) and Station C (Low priority) and throughput of Station B (Medium priority) is more than the throughput of Station C (Low priority). Therefore, we conclude that Ad Hoc Qs is capable of providing service differentiation amongst multiple wireless stations in contrast with both 802.11b and Host Qs, which fail to provide such service differentiation.

Figure 3.7: Each station sends only one type of priority traffic

Experiment 2: Three priority traffics per station

In the second experiment, each of the three sender stations (Station A, Station B, and Station C) sends all the three (High, Medium, and Low) priority traffics. Figure 3.8 shows the total throughput obtained for each type of priority traffic on the channel. It is apparent from Figure 3.8 that Host Qs is able to provide service differentiation amongst multiple stations in this scenario. In this scenario, all the stations are alike in terms of the type and number of priority traffic queues. Therefore, in each station service differentiation is performed by Host Qs’s scheduler amongst three available queues. In this scenario, Ad Hoc Qs provides similar service differentiation and throughput as Host Qs. Therefore, the scheduling added by Ad Hoc Qs does not affect the original performance of Host Qs.

Figure 3.8: Total throughput of different priority traffic

Experiment 3: Two priority traffics per station

In the third experiment, Station A sends High and Medium priority traffics, Station B sends Medium and Low priority traffics, and Station C sends High and Low priority traffics. Therefore, there are two stations for each priority traffic in the network. Figure 3.9 shows the throughput achieved by each priority traffic when Host Qs/Ad Hoc Qs is implemented in each station. Figure 3.9 shows that the throughput of each priority traffic deviates from the throughput achieved from the previous experiment when Host Qs is implemented in each station, ie each station sends all 3 traffic types. Ad Hoc Qs throughput results is close to the intended results due to the slotted-delay scheduler imposed at the LLC layer.

The throughput achieved by each queue of a station is shown in Table 3.4. Table 3.4 shows that within each station the throughput of the higher priority is more than the throughput of the lower priority regardless of the implemented mechanism, Host Qs or Ad Hoc Qs. Therefore, both Host Qs and Ad Hoc Qs are able to provide service differentiation

Figure 3.9: Each station sends two types of priority traffics

Table 3.4: Priority traffic throughput (Mbps) for stations with 2 queues

Station	Implemented Mechanism	High Priority	Medium Priority	Low Priority	Station Throughput
Station A	Host Qs	1.292579	0.296779	0.000000	1.589358
	Ad Hoc Qs	1.566300	0.364638	0.000000	1.930939
Station B	Host Qs	0.000000	0.925550	0.654148	1.579698
	Ad Hoc Qs	0.000000	0.593872	0.417252	1.011124
Station C	Host Qs	1.436738	0.000000	0.145899	1.582637
	Ad Hoc Qs	1.602115	0.000000	0.167980	1.770095

between different priority queues within each station. Table 3.4 shows that when Host Qs is used for service differentiation all the stations have equal share of the total bandwidth (Station Throughput) regardless of the type of priority traffics they transmit. This is expected, because the delay is caused by Host Qs's scheduler is negligible compared to the access delay. The forwarded packets to the MAC layer by Host Qs's scheduler are contending for the channel based on the legacy 802.11b standard. Since 802.11b is a fair MAC protocol, all the stations have equal share of the total bandwidth. Host Qs is only differentiate traffic priorities inside each station. As a result, Host Qs is not able to provide service differentiation amongst wireless stations. On the other hand, when Ad Hoc Qs is used for service differentiation the order of stations from high to low throughput is: Station A (High and Medium), Station C (High and Low), and Station B (Medium and Low). It means that the stations that transmit traffics with higher priority have higher share of the total bandwidth. Therefore, Ad Hoc Qs is able to provide service differentiation amongst wireless stations.

The Effect of Slot Size (D^{slot}) on Service Differentiation

In order to examine the effect of slot size on the performance of Ad Hoc Qs, we performed several experiments using the test-bed.

Experiment 4: Service Differentiation amongst Two Stations

In these experiments Station A sends only High priority traffic, Station B sends only Medium priority traffic and Station C sends only Low priority traffic. We perform 3 experiments. In each experiment, only 2 out of 3 stations send packet to Station R (Figure 3.5). Table 3.5 shows the throughput and throughput ratio of two stations in each experiment for various slot sizes from $7ms$ to $14ms$.

Table 3.5: Priority traffic throughput for stations with 1 queue

Slot Size (ms)	B & C			A & B			A & C		
	Low (Mbps)	Medium (Mbps)	Ratio	Medium (Mbps)	High (Mbps)	Ratio	Low (Mbps)	High (Mbps)	Ratio
7	2.31	2.29	0.99	2.25	2.33	1.04	1.31	2.61	1.99
8	2.42	2.21	0.91	2.34	2.32	0.99	1.16	2.72	2.34
9	2.08	2.27	1.09	2.09	2.33	1.11	0.889	3.02	3.40
10	1.97	2.29	1.16	1.85	2.39	1.29	0.861	3.08	3.58
11	1.57	2.47	1.57	1.64	2.49	1.52	0.743	3.23	4.35
12	1.52	2.53	1.66	1.55	2.5	1.61	0.643	3.39	5.27
13	1.31	2.6	1.98	1.49	2.49	1.67	0.565	3.53	6.25
14	1.17	2.69	2.30	1.2	2.76	2.30	0.51	3.65	7.16

Figure 3.10: Throughput ratio of two stations with two different traffic priorities

Figure 3.10 illustrates the throughput ratio of two priority traffic queues in two stations when RTS/CTS is turned on. As shown in Figure 3.10, increasing the slot size will increase the delay for lower priority traffics and therefore decrease the bandwidth allocated to lower priorities.

Table 3.6: Stations throughput as slot size increases

Slot Size (ms)	Low & Medium & High		
	Low (Mbps)	Medium (Mbps)	High (Mbps)
8	0.977	1.8	1.8
9	0.885	1.8	1.81
10	0.794	1.76	1.87
11	0.697	1.69	1.95
12	0.674	1.68	1.96
13	0.624	1.61	1.99
14	0.562	1.48	2.07
15	0.536	1.39	2.1

Experiment 5: Service Differentiation amongst Three Stations

(a) Each station transmits only one of the three priority traffics

In this experiment Station A sends only High priority traffic, Station B sends only Medium priority traffic and Station C sends only Low priority traffic. Table 3.6 shows the throughput of each station.

Figure 3.11 illustrates the traffic differentiation for three stations with different priorities. The larger the slot size, the more bandwidth will be allocated to the highest priority and the lesser bandwidth will be allocated to the other two lower priorities. This is expected as the Low and Medium priority packets must delay longer in the presence of the High priority packets on the channel. Increasing the slot size will increase the delay of

Figure 3.11: Station throughput vs slot size (msec)

Table 3.7: Throughput of three stations each transmits 3 priority traffic

Slot Size (ms)	Station A			Station B			Station C		
	Low (Mbps)	Medium (Mbps)	High (Mbps)	Low (Mbps)	Medium (Mbps)	High (Mbps)	Low (Mbps)	Medium (Mbps)	High (Mbps)
9	0.127	0.169	1.51	0.109	0.167	1.47	0.123	0.168	1.56
10	0.134	0.168	1.55	0.112	0.172	1.46	0.115	0.191	1.5
11	0.141	0.2	1.38	0.145	0.223	1.43	0.135	0.194	1.35
12	0.111	0.174	1.51	0.133	0.196	1.34	0.138	0.215	1.51
13	0.132	0.182	1.5	0.111	0.192	1.41	0.127	0.233	1.49
14	0.134	0.153	1.51	0.126	0.164	1.36	0.133	0.203	1.52
15	0.121	0.211	1.48	0.127	0.197	1.24	0.143	0.282	1.44
100	0.0883	0.147	0.669	0.091	0.138	0.679	0.0859	0.159	0.696
Prioritizer disabled	0.102	0.177	1.51	0.118	0.172	1.49	0.12	0.187	1.52

lower priorities and therefore decrease the throughput achieved by lower priorities.

(b) Each Station has Three Queues with Different Priorities

Table 3.7 shows the throughput achieved by three stations each has three queues with different priorities. In this case, since all stations send packets with the same set of priorities, all stations are identical in respect of traffic priorities. Thus, the presence of Prioritizer does not have any impact on the traffic differentiation. The bandwidth distribution among stations is independent of the slot size. The traffic differentiation in the experiment results is solely caused by the Queue Manager (Host Qs) that provides a traffic differentiation within a station.

Figure 3.12: Throughput of three stations each has all 3 priority queues

As shown in Figure 3.12, for big slot sizes (here 100ms) the bandwidth of High priority queues drop noticeably due to Head of Line (HOL) blocking problem. The interpretation

of this phenomena is as follows: For big slot sizes, the High priority packets in each station experience more delay because they are waiting for lower priority packets at the same station to be sent out and these lower priority stations delayed due to existence of higher priority packets from other stations. When the access delay of High priority traffic increasing, its throughput decreases as shown in Figure 3.12. When Prioritizer (a kernel module that implements Ad Hoc Qs's scheduler) is disabled, Host Qs's scheduler inside each station is responsible for traffic differentiation. In this case, three traffic priorities inside each station is differentiated by Host Qs's scheduler and then compete for the channel based on 802.11b standard.

3.4 Concluding Remarks

In this chapter, we introduced Ad Hoc Qs test-bed for examining the characteristics of Ad Hoc Qs and Host Qs in terms of service differentiation in ad-hoc WLAN.

The experiments verify that Host Qs is only able to provide service differentiation within one wireless station. There is only one case that Host Qs is able to provide service differentiation amongst multiple stations: each station is transmitting all the three priorities.

In contrast, regardless of the type and number of priority traffic queues in each station, Ad Hoc Qs is able to differentiate services amongst multiple wireless stations. The experiments illustrate that the service differentiation using Ad Hoc Qs is sensitive to the slot size. The larger the slot size (D^{slot}), the longer the delay is experienced by the low priority packets based on the Ad Hoc Qs algorithm and therefore the lower the throughput of low priority traffic. These experiments verify the effectiveness of Ad Hoc Qs for service differentiation in wireless network between multiple stations.

Chapter 4

Analytical Models

In chapter 2, we introduced Host Qs and Ad Hoc Qs for traffic differentiation in wireless ad-hoc networks. These mechanisms are both MAC-Independent and therefore easy to implement. The former mechanism was introduced for traffic differentiation in downlink from Access Point (AP) to network stations and cannot differentiate traffic flows between stations. Ad Hoc Qs was introduced to provide traffic differentiation in wireless media among multiple stations and at the same time uses Host Qs for traffic differentiation in each station.

In this chapter, we introduce two analytical models each for Host Qs and Ad Hoc Qs. In chapter 5, we use these analytical models to make the Ad Hoc Qs adaptive to the number of active stations and type of service they request.

4.1 Analytical Model of Host Qs

The VCW_i is initialized with $VCW_{min,i}$ and in each virtual collision its value (except for the highest priority) [13] doubles till reach the $VCW_{max,i}$.

We will show through simulation results, in Chapter 6, that service differentiation carried out by Host Qs is not sensitive to the value of $VCW_{max,i}$ if this value is greater than or equal to four times the value of Minimum Virtual Contention Window of the Low priority as shown in (4.1).

$$\begin{aligned} VCW_{max,i} &\geq 4 \times VCW_{min,L} \\ i &\in \{H, M, L\} \end{aligned} \tag{4.1}$$

Therefore, we ignore the backoff stages and only consider $VCW_{min,i}$. The queue with the smallest t_b wins the contention and sends its Head-of-Line packet to the MAC layer upon the expiration of the backoff timer. We assumed that only three priorities (High, Medium and Low) exist. $VCW_{min,H}$, $VCW_{min,M}$ and $VCW_{min,L}$ are the values of the Minimum Virtual Contention Window of these three priority queues respectively:

$$VCW_{min,H} \leq VCW_{min,M} \leq VCW_{min,L} \tag{4.2}$$

Let us define α , β and γ as the probability that the High, Medium, and Low priority queues win the virtual contention and send their Head-of-Line packet to the MAC layer.

α is the probability that the random number is drawn from uniform distribution $[0, VCW_{min,H}]$ be smaller than or equal to the random numbers are drawn from uniform distributions $[0, VCW_{min,M}]$ and $[0, VCW_{min,L}]$. Therefore, α can be calculated as follow:

$$\begin{aligned}
\alpha &= \frac{1}{VCW_{min,H} + 1} \\
&+ \frac{1}{VCW_{min,H} + 1} \times \left(1 - \frac{1}{VCW_{min,M} + 1}\right) \left(1 - \frac{1}{VCW_{min,L} + 1}\right) \\
&\dots \\
&+ \frac{1}{VCW_{min,H} + 1} \times \left(1 - \frac{VCW_{min,H}}{VCW_{min,M} + 1}\right) \left(1 - \frac{VCW_{min,H}}{VCW_{min,L} + 1}\right)
\end{aligned} \tag{4.3}$$

The first term of (4.3) is the probability that the value *zero* is drawn from $[0, VCW_{min,H}]$ and therefore regardless of the values are drawn from $[0, VCW_{min,M}]$ and $[0, VCW_{min,L}]$ the High priority wins the contention. The second term of (4.3) is the probability that the value *one* is drawn from $[0, VCW_{min,H}]$ and the value one or bigger are drawn from $[0, VCW_{min,M}]$ and $[0, VCW_{min,L}]$. The last term of (4.3) is the probability that the value $VCW_{min,H}$ is drawn from $[0, VCW_{min,H}]$ and the value $VCW_{min,H}$ or bigger are drawn from $[0, VCW_{min,M}]$ and $[0, VCW_{min,L}]$. Using the same method we can find equations for β and γ . We can calculate α , β and γ using (4.4), (4.5) and (4.6) respectively.

$$\alpha = \frac{1}{VCW_{min,H} + 1} \sum_{i=0}^{VCW_{min,H}} \left(1 - \frac{i}{VCW_{min,M} + 1}\right) \left(1 - \frac{i}{VCW_{min,L} + 1}\right) \tag{4.4}$$

$$\beta = \frac{1}{VCW_{min,M} + 1} \sum_{i=0}^{VCW_{min,H}} \left(1 - \frac{i}{VCW_{min,H} + 1}\right) \left(1 - \frac{i}{VCW_{min,L} + 1}\right) \tag{4.5}$$

$$\gamma = \frac{1}{VCW_{min,L} + 1} \sum_{i=0}^{VCW_{min,H}} \left(1 - \frac{i}{VCW_{min,H} + 1}\right) \left(1 - \frac{i}{VCW_{min,M} + 1}\right) \quad (4.6)$$

These equations enable us to calculate the appropriate Minimum Virtual Contention Window value for each priority queue in order to achieve the desired traffic ratios within each station. Correctness of the analytical model is verified through the simulation results that will be presented in chapter 6.

4.2 Analytical Model of Ad Hoc Qs

In this section, we describe the characteristics of Ad Hoc Qs through an analytical model in order to find the appropriate slot size in which the desired traffic differentiation is achieved. Each station in the ad-hoc wireless network has three types of priority packets (High, Medium, and Low) to send. Host Qs differentiates these traffics within each station and Ad Hoc Qs is responsible for traffic differentiation among multiple stations.

Bianchi [17] introduced an analytical model to compute the 802.11 DCF throughput, in the assumption of finite number of stations and ideal channel conditions.

In this model τ is the stationary probability that the station transmits a packet in a randomly chosen slot time. p will be referred to as conditional collision probability, meaning that this is the probability of a collision seen by a packet being transmitted on the channel [17]. τ and p are calculated using (4.7) and (4.8) respectively [17].

$$\tau = \frac{2(1 - 2p)}{(1 - 2p)(W + 1) + pW(1 - (2p)^m)} \quad (4.7)$$

$$p = 1 - (1 - \tau)^{n-1} \quad (4.8)$$

In these equations, W is the minimum contention window ($W = CW_{min}$), m is the maximum backoff stage ($CW_{max} = 2^m W$) and n is the number of active stations. Equations (4.7) and (4.8) represent a nonlinear system in the two unknown τ and p , which can be solved using numerical techniques.

Based on Bianchi's model, Raptis et al [28] introduced a formula for calculating average packet delay. We use this model to calculate the average transmission delay for a station when contending with other active stations using the IEEE 802.11b and RTS/CTS access scheme.

The average packet delay is calculated as follow [28]:

$$E_N[D] = \sum_{j=0}^m (E[d_j] \cdot q_j) \quad (4.9)$$

Where d_j represents the delay of a packet been successfully transmitted in the j back-off stage and q_j is the probability of this successful transmission in j backoff stage. The probability q_j is calculated as follow [28]:

$$q_j = \frac{p^j}{1 - p^{m+1}} \cdot (1 - p) \quad j \in [0, m] \quad (4.10)$$

Where $(1 - p)$ is the probability that a packet is successfully transmitted. p^j is the probability that a packet reaches the backoff stage j . $(1 - p^{m+1})$ is the probability that the packet is not dropped.

The average delay $E[d_j]$ is calculated as follow [28]:

$$E[d_j] = T_s + jT_c + E'[\text{slot}] \sum_{i=0}^j \frac{W_i - 1}{2} \quad j \in [0, m] \quad (4.11)$$

Where $(W_i - 1)/2$ is the average number of slot times the station delays in i backoff stage, $E'[\text{slot}]$ is the average length of a slot time when the remaining $n - 1$ stations compete for the channel, jT_c is the time that the packet spends in collisions until it reaches the j stage, and T_s is the time to send successfully from the j th stage. $E'[\text{slot}]$ is calculated as follows [28]:

$$E'[\text{slot}] = (1 - P'_{tr})\sigma + P'_{tr}P'_sT_s + P'_{tr}(1 - P'_s)T_c \quad (4.12)$$

Where σ is the duration of an empty slot time. T_s and T_c are the time durations the channel is sensed busy during a successful transmission and a collision, respectively. P'_{tr} is the probability that at least one out of $n - 1$ stations transmits in a randomly chosen slot time and is calculated as follows [28]:

$$P'_{tr} = 1 - (1 - \tau)^{n-1} \quad (4.13)$$

And P'_s is the probability that a transmission occurring on the channel is successful and is given by the probability that only one station of the $n - 1$ remaining stations transmits, with the condition that a transmission occurs on the channel [28]:

$$P'_s = \frac{(n - 1)\tau(1 - \tau)^{n-2}}{P'_{tr}} = \frac{(n - 1)\tau(1 - \tau)^{n-2}}{1 - (1 - \tau)^{n-1}} \quad (4.14)$$

Finally, the average packet delay is given by [28]:

$$E_N[D] = \sum_{j=0}^m \left(\left(T_s + jT_c + E'[\text{slot}] \sum_{i=0}^j \frac{W_i - 1}{2} \right) \frac{p^j(1-p)}{1-p^{m+1}} \right) \quad (4.15)$$

We define *transit* delay as the time from the packet being ready for transmission (in MAC layer) until it is received at the receiver correctly. This time consists of delay for accessing to the channel, transmission delay and propagation delay. Figure 4.1 shows the average transit delay of a station to transmit a packet of 1500 bytes using IEEE 802.11 with RTS/CTS.

Figure 4.1: Average transit delay: 1500 bytes/packet and 802.11b with RTS/CTS

We assume that each station can send any combination of the three priority traffics (High, Medium, and Low). Let N_i be the number of the stations that transmit priority i traffic.

$$i \in \{H, M, L, HM, HL, ML, HML\} \quad (4.16)$$

For example, N_{HL} represents a station that always has data to transmit from each of the High and Low priority queues. These stations are in saturated mode (always have packets to send). We considered that all these stations are transmitting packets to a specific receiver station and all stations are in the transmission range of each other.

Let us define D_H , D_M and D_L as the average transit delay of the High, Medium and Low priority packets respectively. Ad Hoc Qs's scheduler sends the High priority packets immediately to the MAC layer. For the Medium priority packets, depending on the last heard packet on the channel, they might be sent immediately or with one slot size delay (which has been defined in Ad Hoc Qs algorithm) to the MAC layer. For the Low priority packets, depending on the last heard packet on the channel, they might be sent immediately or with one or two slot size delay to the MAC layer. Consequently, we can consider that the average transit delay of the High priority packets consists of three terms as are shown in (4.17):

$$D_H = \rho_1 D_{N'_H} + \rho_2 D_{N'_H+N'_M} + \rho_3 D_{N'_H+N'_M+N'_L} \quad (4.17)$$

- First term: The delay which is experienced by the High priority packets when only High priority stations contend for the channel. Medium and Low priority stations do not have packets in their MAC layers due to the delay caused by the Ad Hoc Qs's scheduler.
- Second term: The delay which is experienced by the High priority packets when the High priority and Medium priority stations contend for the channel.

- Third term: The delay which is experienced by the High priority packets when the High priority, Medium priority and Low priority stations contend for the channel simultaneously.

Where $D_{N'_H}$, $D_{N'_H+N'_M}$ and $D_{N'_H+N'_M+N'_L}$ are the average transit delay of N'_H , $N'_H+N'_M$ and $N'_H+N'_M+N'_L$ stations respectively. ρ_1 , ρ_2 and ρ_3 are the probability for the High, Medium and Low priority packets contention for the channel respectively. N'_H , N'_M and N'_L represent the effect of all stations in terms of pure High, Medium and Low priority stations. For example, N_H is the number of stations that send only High priority traffic, while N'_H brings into consideration all the stations that transmit High priority traffic including N_H , N_{HL} , N_{HM} and N_{HML} . These values are calculated using (4.51), (4.52) and (4.53) respectively.

Bianchi [17] presents an analytical model under a saturated traffic assumption. The delay forced by the Ad Hoc Qs's scheduler to the Medium and Low priority packets make Bianchi's model [17] inappropriate for analyzing Ad Hoc Qs. But as apparent from (4.17) there are three terms for three situations. These situations happen with different priorities, but in each situation the Bianchi's model is still valid because each situation represents a saturated condition for specific number of stations.

Similarly, the average transit delay of the Medium priority packets (D_M) consists of two terms as are shown in (4.18):

$$D_M = \rho_1 D^{slot} + \rho_4 D_H \quad (4.18)$$

- First term: The delay of one slot size is caused by the Ad Hoc Qs's Scheduler (when the last heard packet on the channel is a High priority packet).

- Second term: The delay when the Medium priority packet is sent immediately to the MAC layer and therefore experiences the same delay as a High priority packet (when the last heard packet on the channel is a Medium/Low priority packet).

Where D^{slot} is the slot size and ρ_4 is the probability for the Medium or Low priority packets contention for the channel so $\rho_4 = \rho_2 + \rho_3$. When the last heard packet on the channel is Medium or Low, Medium priority packets are sent immediately to the MAC layer. The probability of this situation is ρ_4 . In this case, Medium priority packets experience same delay as High priority packets (D_H).

For low priority stations, the average transit delay (D_L) consists of three terms as are shown in (4.19):

$$D_L = \rho_1(2D^{slot}) + \rho_2(D^{slot}) + \rho_3D_H \quad (4.19)$$

- First term: Two slot size delay is caused by the Ad Hoc Qs's Scheduler (when the last heard packet on the channel is a High priority packet).
- Second term: One slot size delay is caused by the Ad Hoc Qs's Scheduler (when the last heard packet on the channel is a Medium priority packet).
- Third term: The same delay as the High priority packets while the Low priority packet is sent immediately to the MAC layer (when the last heard packet on the channel is a Low priority packet).

When the last heard packet on the channel is Low, Low priority packets are sent immediately to the MAC layer. The probability of this situation is ρ_3 . In this case, Low priority packets experience same delay as High priority packets (D_H).

Let us define H_P^x , M_P^x and L_P^x the total number of the High, Medium, and Low priority packets respectively, which are transmitted in time duration of T on the channel by station x :

$$x \in \{H, M, L, HM, HL, ML, HML\} \quad (4.20)$$

We define the desired traffic ratios as follow:

$$\theta = \frac{\text{Average Throughput of Each High Priority Queue}}{\text{Average Throughput of Each Low Priority Queue}} \quad (4.21)$$

$$\lambda = \frac{\text{Average Throughput of Each High Priority Queue}}{\text{Average Throughput of Each Medium Priority Queue}} \quad (4.22)$$

According to the definition of the *transit* delay, in average, there is one successful packet transmission in each average transit delay duration. Average throughput is the average number of packets that are delivered successfully per time unit:

$$\text{Average Throughput} = \frac{\text{One Packet}}{\text{Average Transit Delay}} \quad (4.23)$$

As a result, average throughput of each priority queue is reversely proportional to the average transit delay of the same priority. Therefore average transit delay of each Low and Medium priority packet is θ and λ times average transit delay of each High priority packet respectively:

$$\theta = \frac{D_L}{D_H} \quad (4.24)$$

$$\lambda = \frac{D_M}{D_H} \quad (4.25)$$

The next step is to calculate the total number of the High, Medium and Low priority packets each station sends to the channel in time duration of T . For stations that send only High, Medium or Low priority packets, they are calculated as follows:

$$H_P^H = \frac{T}{D_H} \quad (4.26)$$

$$M_P^M = \frac{T}{D_M} \quad (4.27)$$

$$L_P^L = \frac{T}{D_L} \quad (4.28)$$

For HM stations, in time duration of T :

$$T = H_P^{HM} \cdot D_H + M_P^{HM} \cdot D_M \quad (4.29)$$

$$H_P^{HM} = \lambda M_P^{HM} \quad (4.30)$$

Therefore

$$H_P^{HM} = \frac{T}{2D_H} \quad (4.31)$$

$$M_P^{HM} = \frac{T}{2D_M} \quad (4.32)$$

For HL stations, in time duration of T :

$$H_P^{HL} = \frac{T}{2D_H} \quad (4.33)$$

$$L_P^{HL} = \frac{T}{2D_L} \quad (4.34)$$

For ML stations, in time duration of T :

$$M_P^{ML} = \frac{T}{2D_M} \quad (4.35)$$

$$L_P^{ML} = \frac{T}{2D_L} \quad (4.36)$$

For HML stations, in time duration of T :

$$H_P^{HML} = \frac{T}{3D_H} \quad (4.37)$$

$$M_P^{HML} = \frac{T}{3D_M} \quad (4.38)$$

$$L_P^{HML} = \frac{T}{3D_L} \quad (4.39)$$

H_P is the total number of the High priority packets are transmitted on the channel in time duration of T :

$$H_P = N_H \cdot H_P^H + N_{HM} \cdot H_P^{HM} + N_{HL} \cdot H_P^{HL} + N_{HML} \cdot H_P^{HML} \quad (4.40)$$

M_P is the total number of the Medium priority packets are transmitted on the channel in time duration of T :

$$M_P = N_M \cdot M_P^M + N_{HM} \cdot M_P^{HM} + N_{ML} \cdot M_P^{ML} + N_{HML} \cdot M_P^{HML} \quad (4.41)$$

L_P is the total number of the Low priority packets are transmitted on the channel in time duration of T :

$$L_P = N_L \cdot L_P^L + N_{HL} \cdot L_P^{HL} + N_{ML} \cdot L_P^{ML} + N_{HML} \cdot L_P^{HML} \quad (4.42)$$

ρ_1 , ρ_2 , ρ_3 and ρ_4 are calculated as follows:

$$\rho_1 = \frac{H_P}{H_P + M_P + L_P} \quad (4.43)$$

$$\rho_2 = \frac{M_P}{H_P + M_P + L_P} \quad (4.44)$$

$$\rho_3 = \frac{L_P}{H_P + M_P + L_P} \quad (4.45)$$

$$\rho_4 = \frac{M_P + L_P}{H_P + M_P + L_P} \quad (4.46)$$

$$\rho_1 = \frac{N_H + \frac{N_{HM}}{2} + \frac{N_{HL}}{2} + \frac{N_{HML}}{3}}{N_H + \frac{N_M}{\lambda} + \frac{N_L}{\theta} + \frac{N_{HM}}{2} \left(1 + \frac{1}{\lambda}\right) + \frac{N_{HL}}{2} \left(1 + \frac{1}{\theta}\right) + \frac{N_{HML}}{2} \left(\frac{1}{\lambda} + \frac{1}{\theta}\right) + \frac{N_{HML}}{3} \left(1 + \frac{1}{\lambda} + \frac{1}{\theta}\right)} \quad (4.47)$$

$$\rho_2 = \frac{\frac{N_M}{\lambda} + \frac{N_{HM}}{2\lambda} + \frac{N_{ML}}{2\lambda} + \frac{N_{HML}}{3\lambda}}{N_H + \frac{N_M}{\lambda} + \frac{N_L}{\theta} + \frac{N_{HM}}{2} \left(1 + \frac{1}{\lambda}\right) + \frac{N_{HL}}{2} \left(1 + \frac{1}{\theta}\right) + \frac{N_{HML}}{2} \left(\frac{1}{\lambda} + \frac{1}{\theta}\right) + \frac{N_{HML}}{3} \left(1 + \frac{1}{\lambda} + \frac{1}{\theta}\right)} \quad (4.48)$$

$$\rho_3 = \frac{\frac{N_L}{\theta} + \frac{N_{HL}}{2\theta} + \frac{N_{ML}}{2\theta} + \frac{N_{HML}}{3\theta}}{N_H + \frac{N_M}{\lambda} + \frac{N_L}{\theta} + \frac{N_{HM}}{2}(1 + \frac{1}{\lambda}) + \frac{N_{HL}}{2}(1 + \frac{1}{\theta}) + \frac{N_{ML}}{2}(\frac{1}{\lambda} + \frac{1}{\theta}) + \frac{N_{HML}}{3}(1 + \frac{1}{\lambda} + \frac{1}{\theta})} \quad (4.49)$$

By dividing (4.18) to D_H and using (4.25) we obtain:

$$D^{slot} = \left(\frac{\lambda - \rho_4}{\rho_1} \right) D_H \quad (4.50)$$

N'_H, N'_M and N'_L are calculated as follows:

$$N'_H = N_H + \frac{\lambda}{\lambda + 1} \cdot N_{HM} + \frac{\theta}{\theta + 1} \cdot N_{HL} + \frac{\lambda\theta}{\lambda\theta + \theta + \lambda} \cdot N_{HML} \quad (4.51)$$

$$N'_M = N_M + \frac{1}{\lambda + 1} \cdot N_{HM} + \frac{\theta}{\theta + \lambda} \cdot N_{ML} + \frac{\theta}{\lambda\theta + \theta + \lambda} \cdot N_{HML} \quad (4.52)$$

$$N'_L = N_L + \frac{1}{\theta + 1} \cdot N_{HL} + \frac{\lambda}{\theta + \lambda} \cdot N_{ML} + \frac{\lambda}{\lambda\theta + \theta + \lambda} \cdot N_{HML} \quad (4.53)$$

If there is no High priority packets on the channel ($N_H = N_{HM} = N_{HL} = N_{HML} = 0$), then $\rho_1 = 0$ and $\rho_2 + \rho_3 = 1$. In this case, the Medium priority is the highest priority on the channel and the average transit delay of Medium priority packets is calculated as follow:

$$D_M = \rho_2 D_{N'_M} + \rho_3 D_{N'_M + N'_L} \quad (4.54)$$

In this case, the average transit delay of the Low priority packets is calculated as follow:

$$D_L = \rho_2 (D^{slot}) + \rho_3 D_M \quad (4.55)$$

By dividing (4.55) to D_M and using (4.22) and (4.21) we obtain:

$$D^{slot} = \left(\frac{\theta/\lambda - \rho_3}{\rho_2} \right) D_M \quad (4.56)$$

As a result of the rules of the Ad Hoc Qs's scheduler, which are expressed in equations (2.4) and (2.5), the traffic ratio of Medium to Low priority traffic approaches to 2 when the desired service differentiation takes place. Using (4.18) and (4.19) :

$$\frac{D_L}{D_M} = \frac{\rho_1(2D^{slot}) + \rho_2 D^{slot} + \rho_3 D_H}{\rho_1 D^{slot} + \rho_4 D_H} \quad (4.57)$$

By increasing the slot size, more bandwidth will be allocated to High priority traffics, therefore ρ_1 increases and ρ_2 , ρ_3 and ρ_4 decreases. Consequently, the dominant terms in (4.57) numerator and denominator are $\rho_1(2D^{slot})$ and $\rho_1 D^{slot}$ respectively, therefore (4.57) approaches to 2 by increasing the slot size. To change this ratio, the Ad Hoc Qs's formulas should be modified.

4.3 Concluding Remarks

In this chapter, we proposed two analytical models each for Host Qs and Ad Hoc Qs. Host Qs's analytical model enables us to calculate the appropriate values of the Minimum Virtual Contention Window for each one of the three traffic priorities to obtain desired traffic ratios within each station.

Ad Hoc Qs's analytical model enables us to calculate the appropriate slot size for each station as a function of the number of active stations and the traffic priorities. The proposed analytical model for Ad Hoc Qs is based on the Bianchi's analytical model [17] for IEEE 802.11 DCF mode. Bianchi's model assumptions include ideal channel condition and saturated mode (stations always have packet for transmission). Although, Ad Hoc Qs imposes

some delays to lower priority packets, but there are three situations that the stations can be considered in saturated mode from the MAC layer viewpoint. First, when all the High priority stations contend for the channel. Second, when all the High and Medium priority stations contend for the channel and finally, when all the High, Medium, and Low priority stations contend for the channel. These situations happen with different probabilities.

In chapter 5, we propose Adaptive Ad Hoc Qs based on the Host Qs's and Ad Hoc Qs's analytical models. The Adaptive Ad Hoc Qs selects the appropriate values of VCW_{min} , VCW_{max} and D^{slot} based on the number of active stations and the type of service they request to maintain the desired service differentiation among multiple stations in the ad-hoc wireless network.

Chapter 5

Adaptive Ad Hoc Qs

In chapter 4, we proposed two analytical models each for Ad Hoc Qs and Host Qs. These analytical models enable us to make Ad Hoc Qs adaptive to the number of active stations and the type of service they request. In this chapter, we propose the Adaptive Ad Hoc Qs algorithm and furthermore analyze its characteristics using the Ad Hoc Qs's analytical model.

5.1 Adaptive Service Differentiation

Host Qs is responsible for traffic differentiation among traffics within a station. On the other hand, Ad Hoc Qs provides traffic differentiation in wireless media between multiple stations.

Let us consider B as the total bandwidth available for the three priority packets within each station. Based on the Host Qs's analytical model, the share of the bandwidth consumed by the High, Medium and Low priority packets are $B_H = \alpha B$, $B_M = \beta B$ and $B_L = \gamma B$, respectively. The desired traffic ratios of High to Low and High to Medium are

θ and λ respectively, which are calculated using (4.21) and (4.22).

$$\theta = \frac{B_H}{B_L} = \frac{\alpha B}{\gamma B} = \frac{\alpha}{\gamma} \quad (5.1)$$

$$\lambda = \frac{B_H}{B_M} = \frac{\alpha B}{\beta B} = \frac{\alpha}{\beta} \quad (5.2)$$

$$\alpha + \beta + \gamma = 1 \quad (5.3)$$

Using equations (5.1), (5.2) and (5.3), α , β and γ are calculated in terms of θ and λ as follows:

$$\alpha = \frac{\lambda\theta}{\lambda\theta + \theta + \lambda} \quad (5.4)$$

$$\beta = \frac{\theta}{\lambda\theta + \theta + \lambda} \quad (5.5)$$

$$\gamma = \frac{\lambda}{\lambda\theta + \theta + \lambda} \quad (5.6)$$

Given the desired traffic ratios, θ and λ , the values of α , β and γ are calculated using (5.4), (5.5) and (5.6), respectively. We use (4.4), (4.5), (4.6) and (4.1) for calculating the values of VCW_{min} and VCW_{max} of each priority queue in each station and (4.50) for calculating the value of slot size of each station. The equation (4.50) expresses the value of slot size as a function of the number of active stations and type of service they request.

Each station always monitors the packets on the physical channel and decodes the source IP address and the priority of the packets. An array (M) keeps the number and

the priority of the packets that are heard from each individual station on the channel. We assumed that all the stations are within each other transmission range. Three priorities are the High, Medium and Low priorities.

Every 5 seconds interval, the total number of packets has been heard from each station is checked and it is assumed inactive if this value has not changed in this period. This interval is long enough to discover that a station is inactive.

Using the array M , the number of each of the seven types of stations, which send High, Medium, Low, High & Medium, High & Low, Medium & Low, and High & Medium & Low are determined. Using these seven values and the desired traffic ratios ($\theta = \frac{High}{Low}$, $\lambda = \frac{High}{Medium}$), the probability for the High, Medium and Low priority packets contention for the channel ρ_1, ρ_2 and ρ_3 , respectively and the equivalent number of the High, Medium and Low priority stations N'_H, N'_M and N'_L , respectively are calculated.

For example, if there are 2 stations which only send High priority packets and 5 stations which send High and Medium priority packets and the desired throughput ratio of the High to Medium priority traffic is λ then: $N'_H = N_H + \frac{\lambda}{1 + \lambda} \times N_{HM} = 2 + \frac{\lambda}{1 + \lambda} \times 5$.

Using the above probabilities (ρ_1, ρ_2 and ρ_3) and the equivalent number of stations for each priority, the average transit delay incurred by the packets with the highest priority is calculated. Using this average transit delay the appropriate slot size (D^{slot}) for desired traffic ratios is calculated. Figures 5.1 and 5.2 illustrate the pseudo code and flowchart of this algorithm, respectively. Figure 5.3 shows the block diagram of the Adaptive Ad Hoc Qs.

```

Procedure: Adaptive Ad Hoc Qs
Begin
  Set the appropriate values of Minimum and Maximum Virtual Contention Windows
  based on the desired traffic ratios,  $\theta$  and  $\lambda$ 
  While (There is any packet on the channel) do
    Begin
      Decode the header of the IP packet to detect source IP address (srcIP)
      and priority of packet (TOS)
      If ( $\theta \geq 2$ ) & (packet is a Data packet) then
        Begin
          Increase the number of packets received from srcIP by one;
          Increase the number of detected priority from srcIP by one
          If (In the last 5 seconds NO packet has been detected from srcIP) Then
            Begin
              consider srcIP as an inactive station
            End
          For  $i = 1$  to Number of Active Stations do
            Begin
              Based on the priority of the heard packets from each station
              determine the number of each of the seven types of stations ( $N_H, N_M, \dots, N_{HML}$ )
            End
          If (There is NO active station) Then
            Begin
               $\rho_1 = \rho_2 = \rho_3 = 0$ 
            End
          Else
            Begin
              Calculate  $\rho_1, \rho_2$  and  $\rho_3$  using (4.47), (4.48) and (4.49) respectively
            End
          Calculate  $N'_H, N'_M$  and  $N'_L$  Using (4.51), (4.52) and (4.53) respectively
          If (There is any High priority packet on the channel) Then
            Begin
              Calculate  $D_H$  using (4.17)
              Calculate  $D^{\text{slot}}$  using (4.50)
            End
          Else
            If (There is any Medium priority packet on the channel) Then
              Begin
                Calculate  $D_M$  using (4.54)
                Calculate  $D^{\text{slot}}$  using (4.56)
              End
            Else
               $D^{\text{slot}} = 0$ 
            End
          End
        End
      End
    End
  End

```

Figure 5.1: Pseudo code of the Adaptive Ad Hoc Qs

Figure 5.2: Flowchart of the Adaptive Ad Hoc Qs

Figure 5.3: Block diagram of the Adaptive Ad Hoc Qs

An example is presented to explain how the slot size (D^{slot}) is calculated. There are seven stations, one for each member of set $\{H, M, L, HM, HL, ML, HML\}$:

$$N_H = N_M = N_L = N_{HM} = N_{HL} = N_{ML} = N_{HML} = 1 \quad (5.7)$$

The desired service differentiation is: $\theta = 4$ and $\lambda = 2$. It means that the average throughput of each High and Medium priority queue is 4 and 2 times the average throughput of each Low priority queue, respectively. using (4.47), (4.48) and (4.49), the ρ_1 , ρ_2 and ρ_3 are calculated respectively: $\rho_1 = \frac{4}{7}$, $\rho_2 = \frac{2}{7}$ and $\rho_3 = \frac{1}{7}$. The next step is calculation of N'_H , N'_M and N'_L using (4.51), (4.52) and (4.53) respectively: $N'_H = \frac{319}{105}$, $N'_M = \frac{16}{7}$ and $N'_L = \frac{38}{21}$.

Using the Bianchi's model [17] the average transit delay of N'_H , $N'_H + N'_M$ and $N'_H + N'_M + N'_L$ stations, ie $D_{N'_H}$, $D_{N'_H+N'_M}$ and $D_{N'_H+N'_M+N'_L}$, are calculated. For simplicity, we use linear regression for calculating the average transit delay based on Bianchi's model:

$$D_{N'_H} = 2.1(ms) \times N'_H = 2.1(ms) \times \frac{319}{105} \quad (5.8)$$

$$D_{N'_H+N'_M} = 2.1(ms) \times (N'_H + N'_M) = 2.1(ms) \times \left(\frac{319}{105} + \frac{16}{7}\right) \quad (5.9)$$

$$D_{N'_H+N'_M+N'_L} = 2.1(ms) \times (N'_H + N'_M + N'_L) = 2.1(ms) \times \left(\frac{319}{105} + \frac{16}{7} + \frac{38}{21}\right) \quad (5.10)$$

By inserting the above values into (4.17), the average transit delay of High priority packets (D_H) is calculated:

$$\begin{aligned}
D_H &= \rho_1 D_{N'_H} + \rho_2 D_{N'_H+N'_M} + \rho_3 D_{N'_H+N'_M+N'_L} \\
&= \frac{4}{7} \times 2.1(ms) \times \frac{319}{105} + \frac{2}{7} \times 2.1(ms) \times \left(\frac{319}{105} + \frac{16}{7}\right) + \\
&\quad \frac{1}{7} \times 2.1(ms) \times \left(\frac{319}{105} + \frac{16}{7} + \frac{38}{21}\right) \approx 8.98
\end{aligned} \tag{5.11}$$

The last step is calculation of slot size (D^{slot}) using (4.50):

$$\begin{aligned}
D^{slot} &= \left(\frac{\lambda - \rho_2 - \rho_3}{\rho_1} \right) D_H \\
&\approx \left(\frac{2 - \frac{2}{7} - \frac{1}{7}}{\frac{4}{7}} \right) 8.98 \\
&\approx 24.7ms
\end{aligned} \tag{5.12}$$

5.2 One Application of Adaptive Ad Hoc Qs

In this section, we calculate the number of High priority stations that can be supported in the presence of various number of Medium and Low priority stations. The results of this calculation can be used for developing a distributed Call Admission Control mechanism.

The Number of High Priority Stations which can be supported

Table 5.1 shows the parameters and assumptions that are used for this calculation. Based on the above mentioned assumptions and using (4.51), (4.52) and (4.53): $N'_H = N_H$, $N'_M = N_M$ and $N'_L = N_L$. By inserting these values into (4.17) the average transit delay of the High priority packets is calculated as follow.

$$D_H = \rho_1 D_{N_H} + \rho_2 D_{N_H+N_M} + \rho_3 D_{N_H+N_M+N_L} \tag{5.13}$$

Table 5.1: List of Parameters and Assumptions

Parameter & value	Description
Packet Size=1500 Bytes	-
$N_{HM} = N_{HL} = N_{ML} = N_{HML} = 0$ $N_H \neq 0, N_M \neq 0, N_L \neq 0$	number of active stations. each station has only one type of priority traffic
$H_{BW}=80$ Kbps	required bandwidth for transmitting a 64 Kbps Voice stream over IP network
D_H	average transit delay of High priority packets
ρ_1, ρ_2, ρ_3	probability for the High, Medium and Low priority packets contention for the channel respectively
D_{N_H}	average transit delay of N_H stations
$D_{N_H+N_M}$	average transit delay of $N_H + N_M$ stations
$D_{N_H+N_M+N_L}$	average transit delay of $N_H + N_M + N_L$ stations
λ	ratio of average throughput of each High priority queue to average throughput of each Medium priority queue
θ	ratio of average throughput of each High priority queue to average throughput of each Low priority queue

Using (4.47), (4.48) and (4.49) and the above assumption, $N_{HM} = N_{HL} = N_{ML} = N_{HML} = 0$, the ρ_1, ρ_2 and ρ_3 are calculated as follows.

$$\rho_1 = \frac{\lambda \theta N_H}{\lambda \theta N_H + \theta N_M + \lambda N_L} \tag{5.14}$$

$$\rho_2 = \frac{\theta N_M}{\lambda \theta N_H + \theta N_M + \lambda N_L} \quad (5.15)$$

$$\rho_3 = \frac{\lambda N_L}{\lambda \theta N_H + \theta N_M + \lambda N_L} \quad (5.16)$$

Where θ and λ , the desired traffic ratios, are calculated using (4.21) and (4.22) respectively.

For simplicity we used linear regression for calculating the average transit delay based on Bianchi's model (Figure 4.1). Slope of linear regression of the graph in Figure 4.1 is about $2.1(ms)$.

$$D_{N_H} = 2.1(ms) \times N_H \quad (5.17)$$

$$D_{N_H+N_M} = 2.1(ms) \times (N_H + N_M) \quad (5.18)$$

$$D_{N_H+N_M+N_L} = 2.1(ms) \times (N_H + N_M + N_L) \quad (5.19)$$

For transmitting a 64 Kbps Voice stream over IP network it requires 80 Kbps bandwidth. We name this required bandwidth H_{BW} . The size of data frames is 1500 bytes:

$$H_{BW} = \frac{1500 \times 8}{D_H} \quad (5.20)$$

$$D_H = \frac{1500 \times 8}{H_{BW}} \quad (5.21)$$

Using (5.14),(5.15),(5.16) ,(5.17),(5.18), (5.19) and (5.21) the equation (5.13) can be rewritten as:

$$\begin{aligned} \frac{12000(bit)}{H_{BW}(kbps)} &= \frac{\lambda\theta N_H}{\lambda\theta N_H + \theta N_M + \lambda N_L} \times 2.1(ms) \times N_H + \\ &\frac{\theta N_M}{\lambda\theta N_H + \theta N_M + \lambda N_L} \times 2.1(ms) \times (N_H + N_M) + \\ &\frac{\lambda N_L}{\lambda\theta N_H + \theta N_M + \lambda N_L} \times 2.1(ms) \times (N_H + N_M + N_L) \end{aligned} \quad (5.22)$$

Finally we reach the following quadratic equation for calculating the number of High priority stations:

$$\begin{aligned} &(\lambda\theta) N_H^2 + \left(\theta N_M + \lambda N_L - \frac{12000\lambda\theta}{2.1H_{BW}} \right) N_H + \\ &\left(\theta N_M^2 + \lambda N_L N_M + \lambda N_L^2 - (\theta N_M + \lambda N_L) \left(\frac{12000}{2.1H_{BW}} \right) \right) = 0 \end{aligned} \quad (5.23)$$

Figures (5.4) and (5.5) illustrate the number of High priority stations which can be supported in the presence of Low and Medium priority stations when $\theta = 4$ and $\theta = 8$, respectively and $H_{BW} = 80$ kbps. By choosing the larger values for θ , we restrict the maximum bandwidth of Medium and Low priority stations in lower amounts. Therefore, in the presence of the same number of Medium and Low priority stations the larger θ values result in the larger number of High priority stations that can be supported. For example, for $\theta = 4$ the maximum bandwidth that is allocated to Medium and Low priority stations are 40 kbps and 20 kbps respectively; While, for $\theta = 8$ the maximum bandwidth that is allocated to Medium and Low priority stations are 20 kbps and 10 kbps respectively.

The above mentioned analysis can be used as the basis for developing a distributed

Figure 5.4: Number of 80 Kbps traffic flows which can be supported when $\theta = 4$

Figure 5.5: Number of 80 Kbps traffic flows which can be supported when $\theta = 8$

measurement-assisted model-based Call Admission Control (CAC). The CAC mechanism is implemented in each wireless station. Each time a station is going to initiate a connection; it monitors the channel and detects the number and priority of the active stations. Using the above analysis, the station is able to calculate the number of the High priority stations that can be supported. If the number of supported High priority stations is more than the number of active High priority stations, then the station initiates its connection.

5.3 Concluding Remarks

In this chapter, we proposed the Adaptive Ad Hoc Qs algorithm based on two proposed analytical models each for Host Qs and Ad Hoc Qs. The Adaptive Ad Hoc Qs adaptively changes the values of the Minimum and Maximum Virtual Contention Windows ($VCW_{min,i}$, $VCW_{max,i}$) of the three priority queues and slot size (D^{slot}) of each wireless station to maintain the desired service differentiation while the number of active stations and requested service vary.

Furthermore, we introduced an application of the Adaptive Ad Hoc Qs which enables us to calculate the number of High priority stations that can be supported in the presence of various number of Medium and Low priority stations. This analysis can be used for possible CAC mechanisms for the Adaptive Ad Hoc Qs.

Chapter 6

Performance Evaluation

In this chapter, we evaluate the proposed analytical models and Adaptive Ad Hoc Qs algorithm through some extensive simulations using QualNet v3.8. QualNet software is a product of Scalable Network Technologies (SNT) [3]. It is used to model and simulate networks.

6.1 Host Qs

Host Qs is able to provide service differentiation within a station. Host Qs uses distinct queues for each traffic category with different values of Minimum and Maximum Virtual Contention Windows. We proposed an analytical model for Host Qs, which enables us to calculate the appropriate values of Minimum and Maximum Virtual Contention Windows for desired traffic ratios amongst the three defined priorities. In this section, we evaluate the accuracy of the proposed analytical model for Host Qs through simulations.

6.1.1 Simulation Environment and Settings

We used QualNet v3.8 to perform our simulations. For evaluating Host Qs's analytical model, the scenario consists of two stations: One station as the sender and one station as the receiver. These two stations are in the transmission range of each other. The sender station always has data to transmit from each of the three priority queues, High, Medium and Low.

6.1.2 Simulation Results

Let us consider B as the total bandwidth available. The share of the bandwidth consumed by the High, Medium and Low priority packets are $B_H = \alpha B$, $B_M = \beta B$ and $B_L = \gamma B$, respectively.

Using (4.4), (4.5) and (4.6) we can calculate the appropriate values of $VCW_{min,i}$ for the desired traffic ratios of the three priority traffics.

Traffic ratios are $B_H/B_M = \alpha/\beta$ and $B_M/B_L = \beta/\gamma$. Using (4.4), (4.5) and (4.6) we can calculate $VCW_{min,H}$, $VCW_{min,M}$ and $VCW_{min,L}$. Table 6.1 shows the analytical and simulation results for the values of α/β and β/γ . The first two columns are the target traffic ratios. Using these values and using (4.4), (4.5) and (4.6) we calculate the appropriate values of Minimum Virtual Contention Window of the three priority queues. The last two columns show the traffic ratios obtained from simulations. The results verify the accuracy of the proposed analytical model for calculating the values of Minimum Virtual Contention Windows.

The simulation results show that if the value of $VCW_{max,i}$ is greater than or equal to four times the value of $VCW_{min,L}$ as shown in (4.1), ie two backoff stages, the traffic ratios

Table 6.1: Host Qs: Different Traffic Ratios

Desired Traffic ratios		Calculated Using (4.4), (4.5), (4.6)			Simulation Results ($VCW_{max,i} = 400$)	
B_H/B_M	B_M/B_L	$VCW_{min,H}$	$VCW_{min,M}$	$VCW_{min,L}$	B_H/B_M	B_M/B_L
2	2	12	18	32	2.1	2.1
3	2	12	24	44	2.8	2.1
4	3	9	22	60	4.0	3.0
5	4	11	38	144	4.8	4.0
6	3	7	24	67	6.3	2.9

Table 6.2: Effect of VCW_{max} on Traffic Ratios

Desired Traffic Ratios		For all Priorities	Simulation Results	
B_H/B_M	B_M/B_L	$VCW_{max,i}$	B_H/B_M	B_M/B_L
2	2	40	2.0	1.7
		64	2.1	2.0
		100	2.1	2.0
		400	2.1	2.1
3	2	50	2.7	1.7
		88	2.7	1.9
		150	2.8	2.0
		400	2.8	2.1
4	3	80	3.7	2.4
		120	3.8	2.7
		200	3.7	3.0
		400	4.0	3.0
6	3	90	5.6	2.4
		134	5.7	2.7
		200	6.0	2.8
		400	6.3	2.9

are controlled by only the Minimum Virtual Contention Window ($VCW_{min,i}$) values of the priority queues.

We can interpret it as follow: In the case of collision only VCW of lower priorities are doubled. When VCW of lower priorities doubled twice, the probability that a collision occurs is negligible. Note that after a successful packet transmission, the winning queue resets its VCW to its assigned VCW_{min} .

Table 6.2 shows the traffic ratios for various $VCW_{max,i}$ values. In these simulations the values of $VCW_{min,H}$, $VCW_{min,M}$ and $VCW_{min,L}$ are the values shown in Table 6.1. The results verify the above mentioned interpretation.

6.2 Adaptive Ad Hoc Qs

Ad Hoc Qs proposed to provide service differentiation amongst multiple stations. This mechanism uses a slotted-delay scheduler which delays the low priority traffics to provide more bandwidth and less delay for High priority traffic. To achieve a consistent service differentiation for various numbers of active stations with various service requirements the value of slot size should be changed accordingly.

Based on two proposed analytical models each for Host Qs and Ad Hoc Qs, we proposed the Adaptive Ad Hoc Qs, which adaptively changes the values of $VCW_{min,i}$, $VCW_{max,i}$, and D^{slot} to maintain the desired service differentiation among multiple wireless stations. In this section, we evaluate the accuracy of the proposed analytical model for Ad Hoc Qs as well as the Adaptive Ad Hoc Qs algorithm through simulations.

6.2.1 Simulation Environment

We used QualNet v3.8 to simulate the Adaptive Ad Hoc Qs. Table 6.3 shows the simulation settings. We carried out several simulations with various numbers of active stations (seven types of stations) and various traffic ratios requirement. The seed parameter generates random numbers as necessary for the simulation. Using the same parameters and seed during two simulations assures that the simulator’s result will be exactly the same no matter how many times the specific scenario is executed. In order to reduce the number of outliers in the statistical analysis of network behavior, we vary the seed and average the resulting metrics over five runs. Table 6.4 shows the scenarios setting for our simulations. In these simulations, all the stations are in the transmission range of each other (single-hop) and all the transmitters send packets to a single station, the receiver. Figure 6.1 illustrates this single-hop scenario.

Table 6.3: Simulation Settings

Type of Settings	Value
Terrain Dimension	Flat Space Rectangular $250 \times 250m^2$
Channel Data Rate	11 Mbps
Packet size	1500 bytes
Simulation time	60 seconds
Seed values	1,2,3,4 and 5
Communication model	Constant Bit Rate (CBR)
Sending rate	12 Mbps
MAC and Physical protocol	IEEE 802.11b with RTS/CTS
MAC propagation delay	$1\mu s$
Stations’ mobility	Fixed

Table 6.4: Scenarios Settings (1 to 3 priority queue(s) per station)

Scenario	N_H	N_M	N_L	N_{HM}	N_{HL}	N_{ML}	N_{HML}	$D^{slot}(ms)$	
								$\theta = 4, \lambda = 2$	$\theta = 7, \lambda = 3.5$
S1	1	1	1	0	0	0	0	9.6	14.6
S2	1	1	1	1	0	0	0	13.2	21.6
S3	1	1	1	1	1	0	0	16.8	28.6
S4	1	1	1	1	1	1	0	21.2	33.5
S5	1	1	1	1	1	1	1	25.2	40.8
S6	1	2	1	2	1	2	1	37.1	62.2
S7	2	1	2	1	2	1	2	39.1	60.8
S8	1	2	3	3	2	1	3	54.1	88.2

Figure 6.1: Single-hop scenario

6.2.2 Results

Figure 6.2 shows the average throughput of each priority queue per station for $\theta = 4$ and $\lambda = 2$. There is a close match between the analytical and simulation results. By increasing the number of stations (from scenario $S1$ to $S8$) the bandwidth quota of each priority queue per station decreases, but the traffic ratios (B_H/B_M and B_H/B_L) remain unchanged and on average each High priority queue per station receives four times bandwidth of each low priority queue per station. The traffic ratios are maintained because of the Adaptive Ad Hoc Qs, which adaptively selects the correct slot size to maintain the desired traffic ratios. Figure 6.3 shows the average throughput of each priority queue for $\theta = 7$ and $\lambda = 3.5$. There is a close match between the analytical and simulation results. Although there are different number of active stations in each scenario, but the traffic ratios (B_H/B_M and B_H/B_L) remain unchanged and on average each High priority queue per station receives seven times bandwidth of each low priority queue per station.

Using Scenario $S8$, which has the largest number of stations and priority queues amongst the scenarios, and different traffic ratios several simulations were performed to evaluate the accuracy of the proposed analytical model. Table 6.5 shows the settings applied to these simulations.

For each desired traffic ratio in each station, the appropriate values of the Maximum and Minimum Virtual Contention Window for each priority traffic queue in each station should be selected. The proposed analytical model for Host Qs is used for this purpose.

Figure 6.4 shows the average throughput of each priority queue per station for Scenario $S8$ with eight different traffic ratio values. This graph shows a close match between the analytical and simulation results for various traffic ratios.

To evaluate the robustness of the proposed algorithm for calculating the appropriate slot

Figure 6.2: Average throughput of each priority queue per station: $\theta = 4$ and $\lambda = 2$

Figure 6.3: Average throughput of each priority queue per station: $\theta = 7$ and $\lambda = 3.5$

Table 6.5: Settings for Different Traffic Ratio Values - Scenario *S8*

θ	λ	$VCW_{min,H}$	$VCW_{min,M}$	$VCW_{min,L}$	D^{slot}
3	1.5	15	19	33	39.6
4	2	12	18	32	54.1
5	2.5	12	21	38	66.2
6	3	14	28	51	77.5
7	3.5	10	22	41	88.2
8	4	6	14	26	98.7
9	4.5	7	18	34	108.6
10	5	12	35	66	118.6

Figure 6.4: Average throughput of each priority queue per station for scenario *S8*

Figure 6.5: Average throughput of each priority queue per station for scenario $S8$, $\theta = 8$ and $\lambda = 4$

size, some simulations with different packet sizes were performed. Figure 6.5 illustrates the average throughput of each priority queue per station for five different packet sizes. In these simulations, the scenario is $S8$ and we set $\theta = 8$ and $\lambda = 4$. The simulation results verify the robustness of the adaptive mechanism to the variation of packet size. For smaller packet size, the throughput of each priority queue decreases because of the overhead of the header of the packets.

All the simulation results verifies the effectiveness of the Adaptive Ad Hoc Qs in service differentiation in overload situation amongst multiple stations in wireless ad-hoc network. In under-load situation, all the stations achieve the required bandwidth because the available bandwidth is more than the requested bandwidth from all the stations. To examine the effect of the Adaptive Ad Hoc Qs on the End-to-End delay of the packets in under-load

traffic, we carry out some simulations. In these simulations, three transmitting stations are sending packets with three different priorities to a receiver station. All the stations are in the transmission range of each other. We carry out several experiments for various number of offered traffic per station. When Adaptive Ad Hoc Qs is disabled, all the stations contend for the channel based on the 802.11b, which is a fair MAC protocol. Therefore, all the stations have the same access delay and therefore End-to-End delay. Figure 6.6 verifies this interpretation and the packets of all the stations have equal End-to-End delay, which is about 5 ms. This value is close to the value is obtained using Bianchi's model [17] which is about 6 ms.

When the Adaptive Ad Hoc Qs is enabled, all the stations still have the same throughput because the total traffic load is less than the total available bandwidth. Therefore, in one-third the time the low priority packets hear a High priority packet on the channel and are delayed for two slot size and in one-third the time hear a Medium priority packet on the channel and are delayed for one slot size. In these simulations, we set $\theta = 4$ and $\lambda = 2$. For Scenario *S1*, $\theta = 4$ and $\lambda = 2$ (Table 6.4), the calculated slot size is equal to 9.6 ms. Therefore the average delay of a low priority packet to reach to the MAC layer is: $1/3 \times 2 \times 9.6 + 1/3 \times 9.6 = 9.6$ ms. The average End-to-End delay of the low priority packets is calculated by adding the access delay and transmission delay to 9.6 ms. Figure 6.6 shows that in under-load situation, the average End-to-End delay of the Low priority packets is about 13 ms. The average End-to-End delay of the High and Medium priority packets is less than 5 ms, because these stations occasionally contend for the channel with only one station due to delay are imposed to the Low priority packets by Ad Hoc Qs's scheduler.

Based on the above analysis, in under-load situation the End-to-End delay of the Low

priority packets is about the slot size value, which is imposed by the Adaptive Ad Hoc Qs. For example, for Scenario $S8$, $\theta = 4$ and $\lambda = 2$ (Table 6.4) the slot size is 54.1 ms . Therefore, in under-load situation the End-to-End delay of the Low priority packets is about 60 ms .

Figure 6.6 shows that in under-load situations (offered traffic per station below 1 Mbps) the Adaptive Ad Hoc Qs does not increase the End-to-End delay of the Low priority packets significantly. In overload situations (offered traffic per station above 1 Mbps), the Adaptive Ad Hoc Qs increases the End-to-End delay of the Low priority packets noticeably.

Figure 6.6: Average End-to-End delay in under-load situation

6.3 Comparison between Adaptive Ad Hoc Qs and IEEE 802.11e

In this section, we briefly introduce the IEEE 802.11e EDCA and then compare the performance of the Adaptive Ad Hoc Qs and IEEE 802.11e EDCA in terms of service differentiation in single-hop and multi-hop wireless ad-hoc network.

6.3.1 IEEE 802.11e EDCA

The IEEE 802.11e uses a new access method called the Hybrid Coordination Function (HCF). Two medium access mechanisms are defined in HCF: controlled channel access (a centralized control mechanism) and contention-based channel access (a distributed control mechanism). In this section, we focus only on HCF contention-based channel access, known as Enhanced Distributed Channel Access (EDCA).

The EDCA of 802.11e which provides priority-based QoS support is an enhanced version of 802.11 DCF. Four Access Categories (ACs) corresponding to voice, video, best effort, and background are defined in EDCA.

Each AC is associated with one backoff entity and the EDCA parameter set including Arbitrary Inter-Frame Space Number ($AIFSN[AC]$), minimum contention window ($CW_{min}[AC]$), and maximum contention window ($CW_{max}[AC]$). $AIFSN[AC]$ is used to determine the Arbitrary $IFS[AC]$:

$$AIFS[AC] = SIFS + AIFSN[AC] \times aSlotTime \quad (6.1)$$

Where $aSlotTime$ is the duration of one slot time and $AIFSN[AC] \geq 2$. Therefore the

earliest access time for an EDCA station is after a DIFS (Distributed InterFrame Space). The higher priority traffics have the smaller $AIFS[AC]$ or $CW_{min}[AC]$, and consequently have the higher chance to send packets to the channel and thus have the higher throughput. In EDCA, the backoff interval for an AC is randomly selected from $[1,CW]$, instead of $[0,CW - 1]$ as in DCF. Also in EDCA the backoff counter will be decrease by one slot before the end of AIFS while in DCF the backoff counter will be decrease just after the end of DIFS. The 802.11e EDCA operates as follows. Each data frame arrives in the MAC layer from the higher layer has a specific priority value. Based on the specified priority, each data frame is mapped to the corresponding AC. Access Point (AP) periodically announces the values of the EDCA parameter set via beacon frames. Each AC has its own queue, AIFS, backoff interval, and contention window. After each unsuccessful transmission, the contention window is doubled until a retry limit or the maximum contention window is reached. When the backoff timers of two or more queues expire at the same time virtual collision happens. In this case, the data frame with the highest priority among the collided frames is transmitted and the others carry out a backoff with an enlarged CW value. There is no priority among EDCA stations. Table 6.6 shows the default values of the IEEE 802.11e parameters.

Table 6.6: Simulation Settings

Access Category	CW_{min}	CW_{max}	$AIFSN$
AC[0]	15	1023	7
AC[1]	15	1023	3
AC[2]	7	15	2
AC[3]	3	7	2

Table 6.7: Simulations Settings for Adaptive Ad Hoc Qs & 802.11e

Type of Settings	Value
Terrain Dimension	Flat Space Rectangular $250 \times 250m^2$
Channel Data Rate	11 Mbps
Packet size	1500 bytes
Simulation time	60 seconds
Seed values	1,2,3,4 and 5
Communication model	Constant Bit Rate (CBR)
Sending rate	12 Mbps
MAC and Physical protocol	IEEE 802.11b/e with RTS/CTS
Ad Hoc Qs's parameters	$\theta = 6$ & $\lambda = 3$
IEEE 802.11e	Default (Standard) settings [11]
MAC propagation delay	$1\mu s$
Nodes' mobility	Fixed

6.3.2 Performance Comparison in Single-hop & Multi-hop

For performance comparison we set two simulation scenarios: single-hop and multi-hop. In the single-hop scenario all the stations are in the transmission range of each other and all the transmitters send packets to a receiver station (Figure 6.1). Table 6.7 shows the simulations settings and Table 6.8 shows the scenarios setting for single-hop scenario. We used QualNet v3.9.5 network simulator to simulate the IEEE 802.11e.

Figure 6.7 illustrates the throughput of each station in each scenario. By increasing the number of High priority stations the Low priority stations suffer from starvation when the IEEE 802.11e is used for service differentiation. In contrast, Adaptive Ad Hoc Qs is able to protect the low priority stations from starvation and at the same time provide desired service differentiation. In single-hop scenario, the channel utilization in both mechanisms is closely matched as shown in Figure 6.7. In 802.11e, the values of $AIFSN$, CW_{min} , and CW_{max} are fixed based on the priority class. Fixing these values with a priority class

Table 6.8: Scenarios Settings (1 priority queue per station)

Scenario	No of HP or <i>AC</i> [2] Stations	No of MP or <i>AC</i> [1] Stations	No of LP or <i>AC</i> [0] Stations
S1	1	1	1
S2	2	1	1
S3	3	1	1
S4	3	2	1
S5	3	2	2
S6	3	3	3
S7	4	6	5
S8	6	5	4

makes EDCF difficult to adapt when the amount of traffic, traffic priorities, and the number of stations can change with time [19]. As the number of stations increases, High priority traffic with very small values of CW_{min} and CW_{max} may cause high probability of collision, especially in highly contentious conditions [12, 23]. In addition, the priority of a traffic class is statically integrated with the right to access the wireless channel leading to starvation of low priority traffic [26]. In contrast, Adaptive Ad Hoc Qs adaptively changes the proper parameters according to the number of active stations and requested service to provide consistent differentiation amongst 3 traffic priorities and as a result prevents Low priority traffic from starvation.

The Scenarios settings for multi-hop simulation is the same as single-hop (Table 6.7) but the terrain dimension is a flat space rectangular $140 \times 900m^2$. In this scenario, three transmitting stations which are in the transmission range of each other are sending packets with three different priorities to three destinations which are five hops away (4 intermediate stations). Figure 6.8 depicts multi-hop scenario.

Figure 6.9 shows the service differentiation results for both Adaptive Ad Hoc Qs and

Figure 6.7: Ad Hoc Qs and 802.11e service differentiation comparison - (single-hop)

Figure 6.8: Multi-hop scenario

Figure 6.9: Ad Hoc Qs and 802.11e service differentiation comparison - (5 hops)

802.11e in the multi-hop scenario. By increasing the offered traffic per AC/priority both mechanism give more channel bandwidth to higher priority traffic by delaying the transmission of the lower priority traffics. But, as apparent from Figure 6.9 the channel utilization of Adaptive Ad Hoc Qs is much more than 802.11e. For example when the offered traffic is 12 Mbps, the total throughput of Adaptive Ad Hoc Qs and 802.11e are 0.895 Mbps and 0.593 Mbps respectively. When offered traffic is increasing, the AC[2] throughput decreases since smaller contention window leads to higher collision probability. In single-hop scenario, there is only one collision zone, while in the multi-hop scenario there are multiple collision zones. Consequently, the total throughput of 802.11e decreases notably in multi-hop scenario. On the other hand, Adaptive Ad Hoc Qs uses the legacy 802.11 MAC and therefore prevents collisions and throughput degradation compared to 802.11e. As a result, the throughput of each priority traffic in Adaptive Ad Hoc Qs is higher than its counterpart

in 802.11e.

6.4 Concluding Remarks

In this chapter, we evaluated the accuracy of the proposed analytical models for Host Qs and Ad Hoc Qs through the extensive simulations using QualNet v3.8. Host Qs's analytical model enables us to calculate the appropriate contention window values for the desired traffic ratios amongst the three priority traffics within each station. The simulation results verified the accuracy of the proposed analytical model for Host Qs.

Furthermore, using the Ad Hoc Qs's analytical model the appropriate slot size value is calculated as a function of the number of active stations, the type of service they request, and the desired traffic ratios. The simulation results verifies the accuracy of the proposed analytical model and its effectiveness for making Ad Hoc Qs adaptive to the number of active stations and the type of service they request. We performed extensive simulations with various traffic ratios, various number of active stations, and various packet sizes to verify the robustness of the Adaptive Ad Hoc Qs algorithm to all these variations.

Based on the Ad Hoc Qs's analytical model, we are able to calculate the number of the High priority traffic flows in the presence of Medium and Low priority traffics. This analysis can be the base for a Call Admission Control mechanism for the Adaptive Ad Hoc Qs.

Finally, we compared the performance of the IEEE 802.11e and Adaptive Ad Hoc Qs in terms of service differentiation in single-hop and multi-hop static wireless network. In single-hop scenario, although both mechanisms have the same channel utilization but the low priority traffic suffers from starvation when 802.11e is used and the amount of High priority traffic is increasing. In contrast, Adaptive Ad Hoc Qs will protect low priority

traffic from starvation regardless of the amount of High priority traffic.

Furthermore, in multi-hop scenario the Adaptive Ad Hoc Qs surpasses 802.11e in channel utilization. In a 5 hops simulation, Adaptive Ad Hoc Qs's total throughput is as much as 1.5 times more than 802.11e's total throughput.

By considering all the above mentioned advantages of Adaptive Ad Hoc Qs along with this fact that Adaptive Ad Hoc Qs is implemented just by software upgrade, it is a desirable and applicable mechanism for service differentiation in ad-hoc WLAN.

Chapter 7

Conclusion and Future Work

The increasing demand for multimedia applications such as VoIP and video streaming running across the network has created new challenges to provide some form of QoS across the wireless network.

The dominant standard in WLAN's MAC and physical (PHY) layers is the IEEE 802.11. The popular extension of this standard are a/b/g which do not support QoS requirements. Due to the need for proper QoS support, the IEEE established the task group E to provide QoS support for 802.11 standards. The last draft of the IEEE 802.11e was released in September 2005. Nonetheless, the number of WLAN cards which support 802.11e is still limited. Also, users have to change their old WLAN cards to the new one (Hardware upgrade).

Meanwhile, a MAC-Independent service differentiation mechanism is of great interest. It can also enhance the operation of QoS-enabled MAC protocols by providing additional adjustable parameters.

Host Qs and Ad Hoc Qs together provide a MAC-Independent framework for traffic differentiation in wireless ad-hoc networks. While Host Qs is responsible for traffic differentiation among traffics within a station, Ad Hoc Qs provides traffic differentiation in wireless media between multiple stations.

In order to make both algorithms adaptive and adjustable we introduced two analytical models. Host Qs's analytical model enables us to select the appropriate values of Minimum and Maximum Virtual Contention Windows for the three priority traffics to achieve the desired traffic ratios within each station. Using Ad Hoc Qs's analytical model, we can calculate the appropriate value of slot size to obtain the desired traffic ratios among multiple stations.

Based on theses two analytical models, we proposed the Adaptive Ad Hoc Qs. It adaptively changes the values of the Minimum and Maximum Virtual Contention Windows (VCW_{min}, VCW_{max}) and slot size (D^{slot}) of each active wireless station and maintains the desired service differentiation while the number of active stations and the type of service requested vary. The simulation results verify the effectiveness of the Adaptive Ad Hoc Qs to maintain the desired service differentiation.

Based on the simulation results, the Adaptive Ad Hoc Qs surpasses the IEEE 802.11e in some features:

- When IEEE 802.11e is used for service differentiation in the ad-hoc WLAN, by increasing the amount of the High priority traffic, the low priority traffic suffers from

starvation. On the other hand, Adaptive Ad Hoc Qs protects the Low priority traffic from starvation when the amount of the High priority traffic is increasing.

- In the multi-hop ad-hoc WLAN, Adaptive Ad Hoc Qs provides more channel utilization compared to the IEEE 802.11e.
- For implementing IEEE 802.11e the hardware upgrade from the legacy 802.11 WLAN cards is essential. In contrast, Adaptive Ad Hoc Qs is a MAC-Independent mechanism which requires no changes to the IEEE 802.11 MAC. Adaptive Ad Hoc Qs is implemented just by software upgrade.

All the above mentioned Adaptive Ad Hoc Qs's characteristics make it a desirable and applicable mechanism for providing QoS support in ad-hoc WLANs.

7.1 Future Work

In this thesis, we proposed an algorithm to make Ad Hoc Qs adaptive to the number of active stations and the type of service they request. We verified the accuracy of the algorithm through simulations. In this section, we identify some future research:

- **Implementation of Adaptive Ad Hoc Qs in the test-bed:**

The next step to validate the functionality of Adaptive Ad Hoc Qs is its implementation in the test-bed. Hopeful results from the research in the real test-bed will confirm the attractiveness of the Adaptive Ad Hoc Qs while any unexpected results will help to make the necessary adjustments.

- **A distributed measurement-assisted model-based Call Admission Control (CAC) for Adaptive Ad Hoc Qs:**

Furthermore, it is desirable to implement a Call Admission Control mechanism based on the analytical models to guarantee an absolute throughput for the High priority traffics. In a distributed measurement-assisted model-based CAC, each station actively monitors the channel and keeps the number of active stations and the type of service requested. Then, using the Ad Hoc Qs's analytical model, each station calculates the number of the High priority queues which can be supported. The station initiates new transmissions if this value is less than the number of active High priority stations.

- **Multi-hop networks and End-to-End QoS:**

In this study, we proposed Adaptive Ad Hoc Qs based on the single-hop analysis. Extending the work to multiple hops toward end-to-end QoS support is quite desirable. In end-to-end QoS support, we may expect delay and not throughput to be the major constraint.

Appendix A

Publications

1. Mowlaei, A.; Lee, B.S.; Lim, T.M., "Service Differentiation In Wireless Networks: Adaptive Ad Hoc Qs Versus IEEE 802.11e," In Proceedings of IEEE International Conference on Networks (ICON), Nov. 19-21, Adelaide, South Australia, 2007.
2. Mowlaei, A.; Lee, B.S.; Lim, T.M., "Enhanced Ad Hoc Qs: MAC-Independent Adaptive Traffic Differentiation in IEEE 802.11-based Ad Hoc Networks," In Proceedings of IEEE Consumer Communications and Networking Conference, January 11-13, Harrah's Las Vegas, Nevada, 2007.
3. Lim, T.M.; Lee, B.S.; Yeo, C.K.; Mowlaei, A., "Ad Hoc Qs: MAC-Independent traffic differentiation in IEEE 802.11-based ad-hoc WLAN," In *Proceedings of 13th IEEE International Conference on Networks*, 2005, Vol.1, pp. 510-514.

Bibliography

- [1] [Online]. Available: <http://dast.nlanr.net/Projects/Iperf/>
- [2] [Online]. Available: <http://www.ethereal.com/docs/man-pages/tethereal.1.html>
- [3] [Online]. Available: <http://www.scalable-networks.com/>
- [4] “Differentiated services working group,” IETF, Tech. Rep.
- [5] “Host ap driver for intersil prism2/2.5/3, hostapd, and wpa supplicant.” [Online]. Available: <http://hostap.epitest.fi/>
- [6] “Integrated services working group,” IETF, Tech. Rep.
- [7] “The rsvp working group,” IETF, Tech. Rep.
- [8] *Supplement to IEEE standard for information technology telecommunications and information exchange between systems - local and metropolitan area networks - specific requirements. Part 11: wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: high-speed physical layer in the 5 GHz band.* IEEE Std 802.11a-1999, Vol., Iss., 1999.

- [9] *Supplement To IEEE Standard For Information Technology- Telecommunications And Information Exchange Between Systems- Local And Metropolitan Area Networks- Specific Requirements- Part 11: Wireless LAN Medium Access Control (MAC) And Physical Layer (PHY) Specifications: Higher-speed Physical Layer Extension In The 2.4 GHz Band.* IEEE Std 802.11b-1999, Vol., Iss., 2000.
- [10] *IEEE standard for information technology- telecommunications and information exchange between systems- local and metropolitan area networks- specific requirements Part II: wireless LAN medium access control (MAC) and physical layer (PHY) specifications.* IEEE Std 802.11g-2003 (Amendment to IEEE Std 802.11, 1999 Edn. (Reaff 2003) as amended by IEEE Std 802.11a-1999, 802.11b-1999, 802.11b-1999/Cor 1-2001, and 802.11d-2001), Vol., Iss., 2003.
- [11] *IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements.* IEEE Std 802.11e-2005 (Amendment to IEEE Std 802.11, 1999 Edition (Reaff 2003)), 2005.
- [12] A. A. Anders Lindgren and O. Schelen, "Quality of service schemes for IEEE 802.11 wireless lans - an evaluation," *Special Issue of the Journal on Special Topics in Mobile Networking and Applications (MONET) on Performance Evaluation of QoS Architectures in Mobile Networks*, vol. 8, no. 3, pp. 223–235, June 2003.
- [13] S. Annese, C. Casetti, C.-F. Chiasserini, and A. Ghittino, "Host qs: Mac-independent traffic differentiation in 802.11-based wlans," in *Proceedings of the 38th Annual*

- Hawaii International Conference on System Sciences, 2005. HICSS '05, 2005.*
- [14] A. Banchs and X. Perez, "Distributed weighted fair queuing in 802.11 wireless lan," in *IEEE International Conference on Communications, ICC'02, 2002.*
- [15] —, "Providing throughput guarantees in ieee 802.11 wireless lan," in *2002 IEEE Wireless Communications and Networking Conference, WCNC2002, 2002.*
- [16] C. S. N. Bennett, J.C.R.; Partridge, "Packet reordering is not pathological network behavior," *Networking, IEEE/ACM Transactions on*, vol. 7, no. 6, pp. 789–798, Dec 1999.
- [17] G. Bianchi, "Performance analysis of the ieee 802.11 distributed coordination function," *Selected Areas in Communications, IEEE Journal on*, vol. 18, no. 3, pp. 535–547, March 2000.
- [18] R. Bramandia, "Diffserv based ad hoc network testbed and evaluation," Nanyang Technological University, School of Computer Engineering, URECAproject report, Tech. Rep., 2005.
- [19] A. Duric, "Voip on wlan - qos issues and voip specifics," in *ETSI Workshop on Speech Processing, Transmission and Quality Aspects (STQ)*, Sophia Antipolis, France, 2003.
- [20] Y. Ge and J. Hou, "An analytical model for service differentiation in ieee 802.11," in *IEEE International Conference on Communications, ICC'03, 2003.*
- [21] M. Hassan, K. Lee, and M. Rezvan, "Design and implementation of a virtual quality of service mac layer (vqml) for wireless lans," School of Computer Science and Engineering, The University of New South Wales, Tech. Rep., 2002.

- [22] S. Kumar, V. S. Raghavan, and J. Deng, "Medium access control protocols for ad hoc wireless networks: a survey," *Ad Hoc Networks*, vol. 4, pp. 326–58, 2006.
- [23] Q. N. L. Romdhani and T. Turletti, "Adaptive edcf: enhanced service differentiation for ieee 802.11 wireless adhoc networks," in *IEEE WCNC*, vol. 2. IEEE, March 2003, p. 1373 1378.
- [24] T. M. Lim, B. S. L. Francis, C. K. Yeo, and A. Mowlaei., "Ad hoc qs: Mac-independent traffic differentiation in ieee 802.11-based ad hoc wlan," in *Networks, 2005. Jointly held with the 2005 IEEE 7th Malaysia International Conference on Communication., 2005 13th IEEE International Conference on*, vol. 1, 16-18 Nov. 2005, p. 5pp.
- [25] W. Pattara-Atikom, P. Krishnamurthy, and S. Banerjee, "Distributed mechanisms for quality of service in wireless lans," *IEEE Wireless Communications*, vol. 10, pp. 26–34, 2003.
- [26] W. Pattara-Atikom, "Quality of service support in ieee 802.11 wireless lan," Ph.D. dissertation, University of Pittsburgh, School of Information Sciences and Telecommunications, 2005.
- [27] W. Pattara-Aukom, S. Banerjee, and P. Krishnamurthy, "Starvation prevention and quality of service in wireless lans," in *The 5th International Symposium on Wireless Personal Multimedia Communications*, 2002.
- [28] P. Raptis, V. Vitsas, K. Paparrizos, P. Chatzimisios, A. C. Boucouvalas, , and P. Adamidis, "Packet delay modeling of ieee 802.11 wireless lans," in *Proceedings of*

- the International Conference on Cybernetics and InformationTechnologies, Systems and Applications (CITSA 2005)*, 2005.
- [29] T. B. Reddy, I. Karthigeyan, B. S. Manoj, and C. S. R. Murthy, "Quality of service provisioning in ad hoc wireless networks: A survey of issues and solutions," *Ad Hoc Networks*, vol. 4, pp. 83–124, 2006.
- [30] N. Vaidya, A. Dugar, S. Gupta, and P. Bahl, "Distributed fair scheduling in a wireless lan," *IEEE Transactions on Mobile Computing*, vol. 4, pp. 616–629, 2005.
- [31] H. Zhu, M. Li, I. Chlamtac, and B. Prabhakaran, "A survey of quality of service in ieee 802.11 networks," *IEEE Wireless Communications*, vol. 11, pp. 6–14, 2004.