EXPERIMENTAL EVALUATION OF 802.11E QUALITY OF SERVICE IN A LARGE-SCALE NETWORK

A Thesis by

Pui See Chung

Bachelor of Science, University Tenaga Nasional, 2007

Submitted to the Department of Electrical Engineering and Computer Science
and the faculty of the Graduate School of
Wichita State University
in partial fulfillment of
the requirement of the degree of
Master of Science

August 2010
EXPERIMENTAL EVALUATION OF 802.11E QUALITY OF SERVICE IN A LARGE-SCALE NETWORK

The following faculty members have examined the final copy of this thesis for form and content, and recommend that it be accepted in partial fulfillment of the requirement for the degree of Master of Science with a major in Electrical Engineering.

_____________________________________
Vinod Namboodiri, Committee Chair

_____________________________________
Bin Tang, Committee Member

_____________________________________
Michael Jorgensen, Committee Member
DEDICATION

To my parents and my dear friends
ACKNOWLEDGMENTS

First of all, I would like to thank my thesis advisor, Dr. Vinod Namboodiri, for his endless guidance and patience in assisting in my research work. He has provided me with extensive suggestions and advice, all of which helped enhance my understanding of computer networking behavior.

I would also like to express my gratitude to my committee members, Dr. Bin Tang and Dr. Jorgensen, for their readiness in helping me complete my graduate research. I must not forget my supportive research group members—Suresh, Babak, Naeem, and Wajid—who provided useful information in my research area during our group discussions.

Special thanks to Mr. Stephen Copeland for providing the technical support and resources needed in my research work.
IEEE 802.11e is a fairly recent amendment to the IEEE 802.11 WLAN standard that supports quality of service (QoS) (or service differentiation) based on medium-access control (MAC) and contention-based channel access. While prior research has been done regarding the performance of this enhancement, the focus here is on two novel aspects: performance of this standard in a large testbed, and energy consumption of an individual node that relies on this service differentiation. Results demonstrate that to get the best benefits of service prioritization, there should be an even distribution of nodes in each service class to prevent intra-class contention. Furthermore, results also demonstrate the high correlation between energy consumption of a node with its traffic priority class.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>2. BACKGROUND...</td>
<td>5</td>
</tr>
<tr>
<td>2.1 IEEE 802.11 QoS Limitations</td>
<td>5</td>
</tr>
<tr>
<td>2.1.1 QoS Limitations of 802.11 Distributed Coordination Function</td>
<td>5</td>
</tr>
<tr>
<td>2.2 IEEE 802.11e Standards</td>
<td>7</td>
</tr>
<tr>
<td>2.2.1 EDCA</td>
<td>7</td>
</tr>
<tr>
<td>2.2.2 Wireless Multimedia</td>
<td>11</td>
</tr>
<tr>
<td>3. LITERATURE REVIEW</td>
<td>12</td>
</tr>
<tr>
<td>3.1 Adaptive EDCF</td>
<td>12</td>
</tr>
<tr>
<td>3.1.1 Scheme Description</td>
<td>12</td>
</tr>
<tr>
<td>3.2 IEEE 802.11e MAC-Level FEC Performance Evaluation and Enhancement</td>
<td>14</td>
</tr>
<tr>
<td>3.3 Experimental Evaluation of 802.11e EDCA for Enhanced Voice Over WLAN Performance</td>
<td>15</td>
</tr>
<tr>
<td>3.4 IEEE 802.11e Enhancement for Voice Service</td>
<td>18</td>
</tr>
<tr>
<td>3.5 TCP Fairness 802.11e</td>
<td>21</td>
</tr>
<tr>
<td>3.6 MAC Parameter Tuning for Best Effort Traffic in 802.11e Contention-Based Networks</td>
<td>25</td>
</tr>
<tr>
<td>4. EXPERIMENTAL SETUP AND IEEE 802.11E PRIORITIZATION</td>
<td>29</td>
</tr>
<tr>
<td>4.1 Testbed Setup</td>
<td>29</td>
</tr>
<tr>
<td>4.2 IEEE 802.11e Prioritization</td>
<td>31</td>
</tr>
<tr>
<td>5. ENERGY IMPLICATIONS OF IEEE 802.11E STANDARD</td>
<td>36</td>
</tr>
<tr>
<td>5.1 Introduction</td>
<td>36</td>
</tr>
<tr>
<td>5.2 Energy Consumption Measurement Methodology</td>
<td>36</td>
</tr>
<tr>
<td>5.3 Basic QoS Environment</td>
<td>37</td>
</tr>
<tr>
<td>5.4 Normal QoS Environment</td>
<td>38</td>
</tr>
<tr>
<td>5.5 High QoS Environment</td>
<td>39</td>
</tr>
<tr>
<td>5.6 Extreme QoS Environment</td>
<td>39</td>
</tr>
<tr>
<td>6. CONCLUSION AND FUTURE RECOMMENDATIONS</td>
<td>41</td>
</tr>
<tr>
<td>6.1 Challenges</td>
<td>41</td>
</tr>
<tr>
<td>Chapter</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>------</td>
</tr>
<tr>
<td>6.2</td>
<td>42</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>44</td>
</tr>
<tr>
<td>APPENDICES</td>
<td>47</td>
</tr>
<tr>
<td>A. Frescor Installation Guide in Linux</td>
<td>48</td>
</tr>
<tr>
<td>B. Frescor Commands and Usage</td>
<td>49</td>
</tr>
</tbody>
</table>
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Default EDCA Parameters</td>
<td>9</td>
</tr>
<tr>
<td>2. Number of Laptops Sending Types of Traffic</td>
<td>31</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>AC Mapping</td>
<td>8</td>
</tr>
<tr>
<td>2.</td>
<td>IEEE 802.11e MPDU Format Without and With the Optional FEC</td>
<td>15</td>
</tr>
<tr>
<td>3.</td>
<td>VOIP Architecture Over WLAN</td>
<td>18</td>
</tr>
<tr>
<td>4.</td>
<td>Proposed Structure of HCF Service Interval</td>
<td>20</td>
</tr>
<tr>
<td>5.</td>
<td>Comparison of Six TCP Uploads (STA 1-6) and Six TCP Downloads (STA 7-12)</td>
<td>23</td>
</tr>
<tr>
<td>6.</td>
<td>Comparison of Six TCP Uploads (STA 1-6) and Six TCP Downloads (STA 7-12) with 802.11e Prioritization</td>
<td>24</td>
</tr>
<tr>
<td>7.</td>
<td>Overall Throughput in Standard (Fixed) CWmin and Adaptive CWmin</td>
<td>27</td>
</tr>
<tr>
<td>8.</td>
<td>Throughput Repartition in Presence and Absence of Best-Effort Traffic in Adaptive CWmin Setting</td>
<td>28</td>
</tr>
<tr>
<td>9.</td>
<td>Real-Environment Testbed Setup</td>
<td>29</td>
</tr>
<tr>
<td>10.</td>
<td>Percentage of Successful Packets Received in Basic QoS</td>
<td>32</td>
</tr>
<tr>
<td>11.</td>
<td>Percentage of Successful Packets Received in Normal QoS</td>
<td>33</td>
</tr>
<tr>
<td>12.</td>
<td>Percentage of Successful Packets Received in High QoS</td>
<td>34</td>
</tr>
<tr>
<td>13.</td>
<td>Percentage of Successful Packets Received in Extreme QoS</td>
<td>34</td>
</tr>
<tr>
<td>14.</td>
<td>Basic QoS Total Energy Consumption</td>
<td>37</td>
</tr>
<tr>
<td>15.</td>
<td>Normal QoS Total Energy Consumption</td>
<td>39</td>
</tr>
<tr>
<td>16.</td>
<td>High QoS Total Energy Consumption</td>
<td>40</td>
</tr>
<tr>
<td>17.</td>
<td>Extreme QoS Total Energy Consumption</td>
<td>40</td>
</tr>
</tbody>
</table>
LIST OF ABBREVIATIONS

AC       Access Category
ACK      Acknowledgement
AEDCF    Adaptive Enhanced Distributed Control Function
AIMD     Additive Increase Multiplicative Decrease
AIFS     Arbitrary Interframe Spacing
AIFSN    Arbitrary Interframe Spacing Number
AP       Access Point
BE       Best Effort
BK       Background
bps      Bits Per Second
CA       Channel Access
CFP      Contention Free Period
CP       Contention Period
CSMA/CA  Carrier Sense Multiple Access with Collision Avoidance
CW       Contention Window
CWmax    Maximum Contention Window
CWmin    Minimum Contention Window
DCF      Distributed Coordination Function
EDCA/EDCF Enhanced Distributed Channel Access/Enhanced Distributed Control Function
FCS      Frame Check Sequence
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FEC</td>
<td>Frame Error Correction</td>
</tr>
<tr>
<td>FIFO</td>
<td>First In First Out</td>
</tr>
<tr>
<td>FTP</td>
<td>File Transfer Protocol</td>
</tr>
<tr>
<td>HCCA</td>
<td>Hybrid Control Channel Access</td>
</tr>
<tr>
<td>HCF</td>
<td>Hybrid Coordination Function</td>
</tr>
<tr>
<td>Kbps</td>
<td>Kilobits per second</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>MAC</td>
<td>Medium Access Control</td>
</tr>
<tr>
<td>Mbps</td>
<td>Megabits per second</td>
</tr>
<tr>
<td>MF</td>
<td>Multiplicator Factor</td>
</tr>
<tr>
<td>MPDU</td>
<td>MAC Protocol Data Unit</td>
</tr>
<tr>
<td>ms</td>
<td>milisecond</td>
</tr>
<tr>
<td>NIC</td>
<td>Network Interface Card</td>
</tr>
<tr>
<td>PCF</td>
<td>Point Coordination Function</td>
</tr>
<tr>
<td>PF[i]</td>
<td>Persistence Factor</td>
</tr>
<tr>
<td>PHY</td>
<td>Physical</td>
</tr>
<tr>
<td>PSTN</td>
<td>Public Switched Telephone Network</td>
</tr>
<tr>
<td>QAP</td>
<td>QoS Access Point</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>QSTA</td>
<td>Quality of Service Station</td>
</tr>
<tr>
<td>RS</td>
<td>Reed Solomon</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>RTS/CTS</td>
<td>Request to Send/Clear to Send</td>
</tr>
<tr>
<td>SIFS</td>
<td>Short Interframe Spacing</td>
</tr>
<tr>
<td>STA</td>
<td>Station</td>
</tr>
<tr>
<td>TCP</td>
<td>Transmission Control Protocol</td>
</tr>
<tr>
<td>TXOP</td>
<td>Transmit Opportunity</td>
</tr>
<tr>
<td>VI</td>
<td>Video Traffic</td>
</tr>
<tr>
<td>VO</td>
<td>Voice Traffic</td>
</tr>
<tr>
<td>VOIP</td>
<td>Voice Over Internet Protocol</td>
</tr>
<tr>
<td>VoWLAN</td>
<td>Voice Over WLAN</td>
</tr>
<tr>
<td>WLAN</td>
<td>Wireless Local Area Network</td>
</tr>
<tr>
<td>WMM</td>
<td>Wireless Multimedia Extension</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

1.1 Introduction

There has been an encouraging growth in the use of wireless media for communication everywhere, especially involving the wireless local area network (WLAN). WLAN usage supports not only basic Internet capabilities but also file sharing, voice calls, online gaming, and multimedia streaming. These applications are highly sensitive to delay and jitter, which affect the quality of the application output. For example, if delay or jitter is present during a voice over Internet protocol (VOIP) call, instead of a smooth voice, there may be interruptions. This is highly undesirable and results in loss of important information.

For a number of reasons, the IEEE 802.11 standard does not perform well when service differentiation is required. First, the IEEE 802.11 standard aims to give fairness to all users using the carrier sense multiple access with collision avoidance (CSMA/CA) mechanism, which ensures that all users have a chance to contend for a channel. Second, both the 802.11 medium access control (MAC) layer and the physical (PHY) layer are designed for best-effort data transmission. Therefore, the guarantee of bandwidth is not always achieved. When all users are given a fair chance of using a channel, we say there is no service differentiation. Simply put, there is no quality-of-service (QoS) support in the IEEE 802.11 standard. Multimedia applications require some QoS support such as guaranteed bandwidth, bounded delay jitter, and error rate.

Many researchers have proposed techniques to provide service differentiation in wireless local area networks (LANs). While researchers have worked on improving the current
802.11 features, at the same time the Internet has also been undergoing a huge growth and demand for multimedia applications. In wireless environments, resources are scarce, and channel conditions are unpredictable and susceptible to losses. Prior research work shows that schemes that have been successfully deployed in a wired network cannot be applied without modifications in a wireless network. A large number of 802.11 QoS enhancement schemes have been proposed, some of which have targeted modifications at the MAC layer [1,2] while others have targeted network behavior [3,4,5,6,7].

The proposal for an 802.11 QoS enhancement began as early as the year 2000 [2]. One of the proposed enhancements of the 802.11 standard is the 802.11e amendment, which serves as an add-on to 802.11 with QoS support. This MAC-layer enhancement provides up to eight levels of priority for users with different wireless needs. Also, additional parameters such as transmit opportunity (TXOP) is also introduced to further assist the smooth transmission of delay-sensitive packets. While a strong foundation for QoS in 802.11e has been established [6,8], a major concern that has rarely been addressed is the energy consumption behavior of nodes using this protocol.

For wireless mobile users, the battery life of their devices is one of the most essential factors to consider, other than mobility. It has been shown that an active WLAN interface is responsible for about 20 to 25 percent of the energy consumed by a laptop, while the figure is closer to 50 percent for a handheld device [9,10,11]. Thus, it is imperative that energy consumption is considered when studying the effectiveness of the IEEE 802.11e QoS standard.

The main contributions of this thesis are the following:
1. An experimental study of service prioritization in the IEEE 802.11e standard was conducted on a large scale (eight nodes). Most of the previous work with this standard has been based on small-scale (two to four nodes) analyses and simulations. Thus, this study helps to clearly understand relative prioritization across nodes under heavy contention for the medium in a practical setting.

2. The impact of prioritization on the successful packet reception of a node was investigated for a varying number of nodes in different service classes. While, the 802.11e standard prioritizes channel access (CA) for a higher-service class node, it was found that intra-class contention could in fact result in a higher-class node obtaining smaller throughput than a lower-class node.

3. While studying the energy consumed by a node based on its service class, it was found that a node with a higher-service class has a distinct advantage over a node with a lower-service class in terms of energy. This is an unintended consequence of QoS in the standard. For example, a node sending time-sensitive traffic, like VoIP, deserves better access to the channel compared to a node sending delay-insensitive traffic, such as transferring a file over a file transfer protocol (FTP). However, it is unfair for the node sending VoIP to also obtain benefits in terms of reduced energy consumption due to prioritized access to the medium.

The remainder of this thesis is organized as follows: Chapter 2 provides background for the IEEE 802.11 standard and its limitations in QoS, as well as further discussion about the need for service differentiation and enhancement of 802.11. Chapter 3 covers a literature review of previous work done in the 802.11e area. Chapter 4 discusses the experiments conducted and
802.11e prioritization in a large-scale network. Chapter 5 covers the energy implications when prioritization occurs in 802.11e, and Chapter 6 provides a conclusion and future recommendations.
CHAPTER 2
BACKGROUND

This chapter covers the fundamental concept in the 802.11 WLAN and its development involving various enhancements over the past decade. In the last few years, there has been a great demand for multimedia applications over the Internet, and it is difficult to accommodate the demand with the existing 802.11 WLAN with its lack of service differentiation. This chapter will provide some solutions, and detailed work will be discussed in later chapters.

2.1 IEEE 802.11 QoS Limitations

Providing the level of assurance for consistent data delivery by a network element is considered quality of service. Thus, the characteristics of the wireless link vary over time and location, and are specific, including high-loss rates, high latency, and jitter. The most challenging function of the medium access control layer is to maintain QoS.

Users that roam may cause path changes and connectivity issues. While mobility is the important consideration in mobile networking, users expect to receive an uninterrupted connection or QoS upon changing their point of attachment. In a network, QoS can be classified in several ways. QoS in a WLAN is classified in terms of data rate, delay, and jitter and is known as parametized QoS; while QoS in terms of delivery priority and queues within the MAC level is known as prioritized QoS.

2.1.1 QoS Limitations of IEEE 802.11 Distributed Coordination Function

In the IEEE 802.11 protocol [12], only best-effort services are supported by the distributed coordination function (DCF); it cannot provide quality-of-service guarantees. Time-
relativity issues, such as delay and jitter, are required by real-time or time-bound services such as VOIP; otherwise, online conferencing would not be able to tolerate losses. In the DCF, all stations (STAs) in the same network have equal opportunity to obtain access to the resources and the channel [13]. For multimedia applications that are running stations, there is no guarantee that these stations will have the desired bandwidth, delay, and jitter because there is no differentiation in data flows, data queuing, or backoff delay, since all stations have the same priority.

Collision avoidance in the DCF enables all stations to have the same opportunity to access the channel. This works well in a network with normal traffic demand. Problems occur when the network gets congested with high-traffic demand. The network becomes overloaded, thus affecting the overall performance of all devices. Also, in the modified 802.11b network, where the rate changes automatically among 11 Mbps, 5.5 Mbps, 2 Mbps, and 1 Mbps, stations with higher bit rates need to lower their data rate in order for all stations to have the opportunity to transmit. For example, in an 802.11b network where some stations have 11 Mbps and other have 1 Mbps, when the network is overloaded, stations with 11 Mbps do not get the maximum bit rate of 11 Mbps. Instead, their bit rate will be decreased accordingly because of equal opportunity property in the DCF.

When lower-bit-rate stations obtain access to the channel, this will also affect the overall performance of the network, as these stations need more time to transmit their frames, thus resulting in a higher waiting period and more energy consumption. Unless admission control is used, there is no guarantee that higher-priority stations can have “more” access to the
channel. While admission control seems to work, it introduces a series of overheads, including message exchange to inquire for the network condition and connection request.

2.2 IEEE 802.11e Standard

The 802.11e standard is a set of amendments to the legacy 802.11 and defines the hybrid coordination function (HCF) that replaces the DCF and the point coordination function (PCF) in a station implementing 802.11e on a QoS station. While the original 802.11 protocol has been modified, it still preserves the backward compatibility with legacy functions so that the quality of service station (QSTA) is able to communicate with a station without the QoS setting. The HCF involves two access mechanisms: enhanced distributed channel access (EDCA) and HCF hybrid controlled channel access (HCCA). Contrary to the legacy PCF, which used different frame exchange sequences in the contention period and the contention-free period, the HCF defines a uniform set of frame-exchange sequences that are usable at any time. To summarize: HCF = EDCA + HCCA [14]. The HCF allocates the transmission unit (the right to transmit or transmit opportunity that is granted to the QSTA [15].

2.2.1 EDCA

The EDCA contention-based access method is an extension of the legacy CSMA/CA DCF in order to introduce packet priorities. Applications assign each packet one of eight user-priority levels. Each priority is mapped to an access category (AC) corresponding to transmit queues with different 802.11e parameters (see Figure 1). These transmission queues are predefined in most recent wireless card drivers. As a result, packets are submitted to independent transmit queues (one per AC; i.e., voice, video, best effort, or background). As mentioned earlier relative to
interoperability between QoS stations and non-QoS stations, if a packet has not been assigned a priority, it will be mapped as a best effort AC by default [16].

Figure 1 shows the operation of 802.11e when each access category is assigned a separate 802.11 wireless card. The four defined access categories each represent different types of traffic with different priorities. The corresponding access categories are summarized in Table 1, where AC_VO, AC_VI, AC_BE, and AC_BK represent voice, video, best effort, and background traffic, respectively. AC_VO has the highest priority, as illustrated in the AC0 queue in Figure 1, while AC_BK has the lowest priority (AC3 queue in Figure 1). These parameters are only enabled when virtual collision occurs, that is, one or more stations try to obtain access to the channel at the same time. These parameters add extra measures in 802.11e in order to avoid future virtual collisions within the MAC layer since the backoff timers and contention window values are now different.
Table 1. Default EDCA Parameters

<table>
<thead>
<tr>
<th>AC</th>
<th>AIFSN</th>
<th>CWmin</th>
<th>CWmax</th>
<th>Burst</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC_VO</td>
<td>2</td>
<td>3</td>
<td>7</td>
<td>15</td>
</tr>
<tr>
<td>AC_VI</td>
<td>2</td>
<td>7</td>
<td>15</td>
<td>30</td>
</tr>
<tr>
<td>AC_BE</td>
<td>3</td>
<td>15</td>
<td>1023</td>
<td>0</td>
</tr>
<tr>
<td>AC_BK</td>
<td>7</td>
<td>15</td>
<td>1023</td>
<td>0</td>
</tr>
</tbody>
</table>

In 802.11e, all stations compete for the wireless channel, similar to the normal 802.11 function, and collisions between different nodes may occur [14]. When a collision occurs, the station with the highest priority that successfully gets access to the channel is also granted a transmit opportunity. The TXOP limit in which the node can send more than one data frame depends on the AC and the physical rate defined in 802.11a/b/g. For example, the TXOP limit ranges from 0.2 ms (background priority) to 3 ms (video priority) in an 802.11a/g network, and from 1.2 ms to 6 ms in an 802.11b network. The duration of this TXOP is specified for each AC [16].

The TXOP enables a station to transmit multiple frames within the specified TXOP time limit. When the first frame exchange sequence is finished, the TXOP limit does not expire; the station may send another frame after a period of short interframe spacing (SIFS). The probability that the EDCA function of the AC gains access to the medium (or transmission opportunity–TXOP) is given by adjustable EDCA parameters that are specific for each AC queue. The EDCA per AC queue specific parameters are as follows:

1. An arbitrary interframe spacing number (AIFSN) determines the arbitrary interframe spacing (AIFS), i.e., the interval during that medium is idle before the backoff algorithm is started.
2. The AIFS for each AC (AIFS[AC]) is computed as

\[ \text{AIFS}[AC] = \text{AIFSN}[AC] \times \text{slot time} + \text{SIFS} \]  

(2.1)

The slot time and SIFS are dependent on the physical layer; for 802.11a, the slot time is 9 us and the SIFS is 16 us.

The minimum value of AIFSN is 2 and maximum value of AIFSN is 15. Both values of the minimum contention window (CWmin) and maximum contention window (CWmax) are a power of two [16].

EDCA parameters are stored locally at the QSTA and can be dynamically updated by the QoS access point (QAP), which distributes them to stations in the management frames (the beacon, and in probe and reassociation response frames). This adjustment allows the STAs in the network to adjust to changing conditions, and gives the QAP the ability to manage overall QoS performance [1].

The previous parameters influence the probability of obtaining a transmit opportunity. The TXOP limit expresses the duration of the access. Table 1 shows the default settings of the AC parameters. A TXOP limit value of 0 indicates that only a single data frame (in addition to RTS/CTS exchange) can be transmitted at any rate for each TXOP.

Contention-based protocols are prone to serious performance degradation due to the heavy traffic demand in the network. During overload conditions, the stations must defer their transmissions by the exponential backoff algorithm and increase the contention window size. This results in greater waiting time before any station can actually send data. This significantly affects energy consumption, since more energy is consumed waiting to send data than actually transmitting data.
2.2.2 Wireless Multimedia

The Wi-Fi Alliance, called the Wi-Fi wireless multimedia (WMM) extension and a subset of the upcoming IEEE 802.11e quality of service (QoS) extensions for 802.11 networks, started a certification program for WMM to satisfy the most urgent needs of the industry in finding a QoS solution for Wi-Fi networks. WMM provides prioritized media access and is based on the enhanced distributed channel access method. The 802.11e draft includes additional capabilities and features that may be included later in the Wi-Fi CERTIFIED for WMM program as optional capabilities [15].

In an IEEE 802.11 network, WMM functionality requires that both the access point (AP) and the clients running applications that require QoS are Wi-Fi CERTIFIED for WMM and are WMM-enabled [15]. Up to this point, WMM is yet to be a common term because most wireless devices available today do not support QoS. However, WMM allows the coexistence of devices with QoS and without QoS in a network. If an AP is to be present in the same network, then it must be WMM-enabled. Nodes or stations that do not support WMM can only send data traffic with best-effort priority.
CHAPTER 3
LITERATURE REVIEW

Numerous reviews related to quality of service in 802.11 WLAN have been conducted. This chapter will examine some of the prominent contributions of researchers in their work to enhance, if not design, various protocols and schemes for better communication efficiency and network performance.

3.1 Adaptive EDCF

The adaptive EDCF (AEDCF) scheme was proposed by Romdhani et al. to provide enhancement of the original enhanced distributed channel access for service differentiation in 802.11 WLAN [2]. Their approach was to dynamically adjust the contention window size depending on each traffic class and the amount of load in the network. It is known that the size of the EDCF contention window is static and fixed at some values, depending on the traffic class, as shown in Table 1. Their scheme aims to provide efficiency in the transmission channel. Simulation work has been performed to prove the performance of the AEDCF, and the relative results are compared with the EDCF in 802.11e. Simulation results indeed reduced the collision rate by 50 percent, and channel utilization ratio was increased, which also shows that the overall output was improved up to 25 percent higher than the EDCF.

3.1.1 Scheme Description

In the EDCF scheme described in 802.11e [17], the CWmin[i] and CWmax[i] values are statically set for each priority level, and after each successful transmission, the CW[i] values are reset to CWmin[i]. In AEDCF, the CW[i] values are decreased and increased adaptively according to the collision rate while maintaining the norm of the priority class values. That is,
the minimum CW in each traffic class may or may not be the original CWmin[i]. It could be higher, depending on the network condition, but the attractiveness of this scheme is that it ensures that the highest priority class still maintains its lowest CW values among other classes so that it does not deprive its own channel access priority. In other words, each traffic class has its own factor to increase CWs.

The basic EDCF mechanism resets the CW value to its minimum predefined value after each successful transmission, regardless of network condition. When CW values are small within higher priority classes, collision is more likely and will definitely occur in the future. Therefore, the authors in [2] proposed a dynamic update of the contention window size slowly to avoid multiple collisions. One of the simplest schemes is to introduce a reducing factor to update the CW[i] values such as 0.5 * CWold. The authors also take into account that assigning static factors for each priority class does not deal with various network condition problems. To ensure an optimal solution for all network conditions, the authors proposed an adaptive way to update the CW values with parameters such as the estimated collision rate \( f_{\text{curr}}^{j} \) in each node.

The simplest method to determine the contention in a network is to know the collision rate. The value of \( f_{\text{curr}}^{j} \) is calculated using the number of collisions and the total number of packets sent during a constant period (i.e., a fixed number of slot times) as follows:

\[
f_{\text{curr}}^{j} = \frac{E(\text{collisions}_j[p])}{E(\text{data}_j\_sent_j[p])}
\]

(3.1)

As mentioned above, in order to preserve the priority relationship between different classes of traffic when the CW values are modified, a multiplicator factor (MF) is used. In AEDCF, the MF of class i and the update of the new CW value are defined as follows:
Equations (3.2) and (3.3) always result in a higher \( CW[i] \) compared to \( CW_{\text{min}}[i] \), and the priority for traffic class is maintained. The relationships between the equations are modeled in scenarios for each successful transmission as

\[
MF[i] = \min((1 + (i \times 2)) \times \frac{f^j}{\alpha_{\text{avg}}}, 0.8) \tag{3.2}
\]

\[
CW_{\text{new}}[i] = \max(CW_{\text{min}}[i], CW_{\text{old}}[i] \times MF[i]) \tag{3.3}
\]

For each collision that occurs in the AEDCF, after each unsuccessful transmission of packet of class \( i \), the new \( CW \) of this class is increased with a persistence factor (\( PF[i] \)), which ensures that high-priority traffic has a smaller value of \( PF[i] \) than low-priority traffic, as described in equation (3.4). The motivation to introduce the persistent factor is to decrease the probability of collision in the future and improve performance while minimizing delay.

3.2 IEEE 802.11e MAC-Level FEC Performance Evaluation and Enhancement

This research aims to evaluate the performance of the defined forward error correction (FEC) scheme in the IEEE 802.11e Medium Access Control protocol [18]. A novel retransmission-combining technique is proposed to enhance the performance of the MAC-level FEC scheme. The author uses the frame check sequence (FCS) to check for erroneous Reed Solomon (RS) blocks within a transmitted frame. If the MAC header is erroneous, i.e., the FCS failed, then the frame is retransmitted again. If the RS blocks are erroneous while the header is correctly decoded, the receiver will store the correct RS blocks along with the header information. Since the acknowledgement (ACK) is not received, the sender needs to retransmit the frame. If the MAC header is decoded correctly again, then the receiver should be able to determine whether
it is a retransmission or erroneous frame. If it is a retransmission, the RS blocks with the correct FEC are reconstructed again until all RS blocks are successfully constructed. Then, an ACK will be sent by the receiver indicating the successful completion of the frame. A MAC protocol data unit (MPDU) frame in 802.11e is depicted in Figure 2.

![Figure 2. IEEE 802.11e MPDU Format Without and With Optional FEC [18]](image)

**3.3 Experimental Evaluation of 802.11e EDCA for Enhanced Voice Over WLAN Performance**

The actual experimental evaluation performed on an 802.11e WLAN was performed in this research. The previous work mentioned in subsections 2.2 and 2.3 is simulation-based. Dangerfield et al. demonstrate a technique for measuring one-way delay in an 802.11e hardware testbed, thereby studying delay in the context of protecting a voice call competing against data traffic [19].

Experiments were performed on an 802.11b MAC setting capable of seizing bandwidth from a low-rate voice call. Test results show that the voice call loss rate was greatly affected, even if there were only five contending stations, thereby affecting voice quality and causing
calls to be dropped. By using 802.11e, voice data were assigned priority, thus delivering a desirable performance when prioritization was present.

In a network, voice traffic is highly sensitive to delay compared to data traffic. Delays in 802.11e can be determined by the waiting time before accessing the channel and backoff delay due to collision or successful transmission, which ranges from a few hundred microseconds to hundreds of milliseconds, depending on network conditions. Other delays include transmission delay and congestion control mechanism in the 802.11 WLAN.

The measurement technique used was to set a clock instead of observing synchronization on both sides. Indication of a successful transmission was the receipt of a MAC ACK from the receiver. Since the delay incurred by ACK is only about 10µs after a successful frame received, initiating an interruption at this time enables recording of the time that the ACK is received. Dangerfield et al. also mentioned that the time the packet was added to the hardware queue may be recorded, and by inverting the standard FIFO queuing recursion, the total processing time can be acquired.

Other than delay, the authors displayed the effect of setting the TXOP in the sending stations by limiting one packet per session in one station and variable TXOP (more than one packet per session) in another station. Results show that the mean delay in the station with variable TXOP is considerably lower between the saturated stations.

Another parameter studied was the effect of CW values. For the validation of this experiment one station as set with CWmin of 31, while another station was set to have varied CWmin in powers of two. Here, doubling the CWmin caused the doubling of the mean packet delay, while the average throughput is proportional to the CWmin values.
Other than that, Dangerfield et al. also studied the effect of interframe spacing in 802.11e on delay and throughput. They concluded that extending the AIFS decreased the overall throughput in the network since stations needed to wait longer to access the medium. Even so, the impact of AIFS was still lower compared to shifting the CW in terms of separation of throughput and delay.

Next, the authors tried to further prioritize voice traffic by increasing the AIFS value used by all other stations not sending voice traffic. As mentioned above, increasing a station's AIFS value means increasing the delay, since stations must wait longer before attempting transmission or decrementing the counter. As the network load increases, the effect of AIFS is more significant but still sufficient to cater to the service differentiation for voice traffic. A voice packet was emulated as a 64 Kbps stream with packets every 10 ms. Each packet had a payload of 80 bytes. This voice call shared the network with a number of stations that were saturated, transmitting a 1470 byte packet whenever the MAC allowed.

It is reasonable to infer that as more competing stations are added, the delays for non-prioritized traffics also increase. When the AIFS is set at some values, stations experience delay but throughput is stabilized and not affected. It can be concluded that the AIFS value is one of the useful parameters for setting priorities for voice traffic in a larger network.

Dangerfield et al. provided a useful experimental-based service difference and its performance in real WLAN that fully utilized the 802.11e parameters. They provided insight on how each parameter such as TXOP, AIFS, and CWmin affect throughput and delay in a network, therefore, giving useful tips for determining the optimal network size and parameter settings in order for voice traffic to perform best.
3.4 IEEE 802.11e Enhancement for Voice Service

Internet telephony, more commonly known as VOIP [20], is an emerging technology that uses a wireless network as the medium for real-time voice communication (see Figure 3). It has gained much attention because of its lower operational cost than traditional telephone service, the ever growing demand for multimedia applications in WLAN, and the feasibility of a voice-decoding scheme with a mature signal processing technique, which serves as the viable alternative for a public switched telephone network (PSTN).

![Figure 3. VOIP Architecture Over WLAN [20]](image)

Based on these facts, it is believed that there will be a very promising demand for VOIP technology, and it is expected that the wide deployment of WLAN will help to further boost voice applications over the Internet. Known as Voice Over WLAN (VoWLAN), this application is a real-time delay-sensitive application designed to tolerate less packet loss and jitter compared
to the normal data stream. If these two features are not controlled, voice quality may be degraded, which is undesirable.

The main element in QoS or service differentiation is to transmit a packet within a specified delay that will not affect the voice quality. Doing this is a considerably challenging task. One way to deal with delay and jitter is to use a playout buffer. Dangerfield et al. described a situation where the current standard in WLAN, such as DCF in 802.11 and EDCA in 802.11e, is not efficient enough to accommodate a large amount of simultaneous voice traffic.

Most of the time it takes to transmit a voice packet with a small payload has been used up because of handshaking overhead, preambles, and headers, which is one of the limitations for using VOIP in the WLAN.

Basically, the main focus of this study involved two main issues: voice multiplexing and reducing overhead. For voice multiplexing, the voice traffic was represented as an on/off model where active users transmit without interruption, while off state users do not transmit. As mentioned, overhead causes network inefficiency. Therefore, it must be suppressed as much as possible. To address these two issues, Dangerfield et al. proposed a modified hybrid coordination function (HCF) service interval where a beacon interval as set to be the same duration as the service interval. The modified intervals are shown in Figure 4.
Here, the contention free period was divided into the uplink voice and downlink voice. The downlink voice was used by the access point to send packets to mobile stations. After transmission of all the downlink voice packets, the AP sends a super CF poll frame, enabling the stations in its polling list to have a TXOP to eliminate the ACK transmission to reduce delay. In the CP, the AP and all the stations can contend for the channel used to transmit the first few voice packets of the station’s talk spurt. The length of the service interval is fixed and depends on the delay bound of voice traffic. The lengths of the CFP and CP depend on the voice traffic load. The QoS enhancement in the proposed service interval structure consists of four components: voice traffic multiplexing, deterministic access priority of voice, overhead reduction, and call admission control.

In deterministic access priority, it is important for the voice traffic to have access to the channel during a contention period. In order to achieve this, EDCA is used during the CP because of 802.11e parameters. Note that even though voice traffic should be guaranteed
access, this does not mean that voice traffic will ALWAYS obtain access to the channel. This property does not grant access to high-priority traffic in every contention but is rather long-term because lower-priority traffic will eventually have its backoff counters expire and thus gain access to the channel. However, there is a disadvantage in this scheme if the lower-priority traffic happens to have a heavy load to transmit, whereby it might obtain the channel for a long time and the high-priority traffic will suffer delay. To provide QoS guarantee for voice traffic regardless of the data traffic load in WLAN, data stations should not transmit in the CP until no voice traffic is contending for the channel.

To tackle the header overhead issue, Wang et al. [20] adopted a compression technique to reduce the overhead size to as small as 2 bytes. To reduce the overhead from frequent polling from the AP to a mobile station by sending the CF poll frame, 802.11e allows poll frames to be piggybacked together with the DATA or ACK frames.

Call admission control is also an important factor in guaranteeing QoS. This is performed by the AP in deciding which call to accept and which to reject based on available resources. The authors propose an admission control to calculate how many voice packets can be accommodated in a specific service interval and preset packet loss rate bound.

Overall, the main contribution of this work was to introduce a polling technique for voice packets to meet the delay requirements in 802.11e. In consideration of reducing overhead, this scheme can be easily implemented, as very little modification is needed.

3.5 TCP Fairness in 802.11e WLANs

The work in this area mainly investigated the use of 802.11e parameters to ensure interaction between MAC and the transport layer and, at the same time, to maintain fairness in
transmission control protocol (TCP) while congestion control is employed. Although extensive studies have been done on the MAC level to ensure fairness, very little attention has been paid to the transport level of the network. Although previous work has been done in this area, most is restricted to the the 802.11 MAC layer, and these approaches mainly did minimal changes to the MAC layer.

With the enhancement of 802.11, namely 802.11e, Leith et al. [22] investigated how the 802.11e parameters helped to resolve the fairness issue in the MAC layer itself in order to maintain transport-layer fairness. With more flexibility in 802.11e, the authors considered to handle both TCP uplinks and downlinks in their experiments.

Experiments were done on a rather large scale network of 12 stations with one PC as the access point. All stations, including the access points, used a MADWifi driver to enable modifications on the 802.11e parameters. The congestion TCP uses an algorithm called additive increase and multiplicative decrease (AIMD). In order to observe the behavior of AIMD, the authors relied solely on increasing the buffer size. Other than that, they also performed experiments to investigate the influence of interface and driver queue sizes on TCP fairness. Two main comparisons were made: between TCP upload flows, and between upload and download flows.

In an 802.11b WLAN, gross unfairness between throughputs occurred in the MAC layer due to the backoff algorithm nature before contention for the channel. In the transport layer, unfairness occurred as the result of the flow and congestion control mechanism (source rate matching in accordance with network capacity). Although all stations obtained access to the channel, it can be observed in Figure 5 that there is unfairness between TCP uplinks. The main
reason for this is because of the implicit assumption that the forward path (TCP transmission of data packets) and reverse path (AP ACK to stations) have the same transmission rate.

In the fair-channel access mechanism, for $n$ number of stations and one AP, each station assumed a transmission probability of $1/(n+1)$. The same transmission probability goes to the AP. It can be seen that the total transmission opportunity for $n$ stations is $n/(n+1)$, yet only $1/(n+1)$ share belongs to the AP ACK, which caused a forward path and reverse path asymmetry.

![Graph showing comparison of six TCP uploads and six TCP downloads for 180 seconds](image)

**Figure 5. Comparison of Six TCP Uploads (STA 1-6) and Six TCP Downloads (STA 7-12) [22]**

This path asymmetry for AP is the main reason for poor TCP performance, as it can lead to significant ACK packet drops and disrupt the ACK clocking mechanism as well as inducing repeated timeouts. Timeout occurred because of the infinite waiting time for the unreached ACK packets to the destination. If this situation persists, then flows will be starved for long periods.
Another unfairness mentioned earlier occurred between uplinks and downlinks. Since all download flows must go through the access point, it is as if all download flows are getting the rate of one upload flow. As the number of upload flows increases, this can eventually starve all download flows as well.

In order to address these two types of unfairness, Leith et al. made use of the per-class parameter settings in 802.11e and set priorities exclusively to the TCP ACKs. The rationale behind this approach was to match the volume of TCP ACK with the volume of data packets. By queuing the TCP ACKs in one queue rather than multiple queues (varying the packet size), there was unlimited access to the transport layer in the wireless channel, and therefore, fairness was restored when the forward and reverse paths were balanced.

The same approach was done to restore fairness between upload flows and download flows as well by setting priority to download packets. Stricter parameters such as smaller value
of AIFS and CWmin were used for TCP ACKs at the AP, while larger CWmin values were used on mobile stations to reduce the competition between TCP ACKs. TXOP was also used for TCP download data packets so that more packets per channel access could be transmitted. From Figure 6, it can be seen that fairness was evidently restored on the 802.11b network with 11 Mbps channel capacity.

Therefore, while 802.11e can prioritize certain traffic flows by giving higher transmit chances to traffic, it can also maintain fairness by using the same parameters when used appropriately. This experiment has further proven the flexibility achieved in 802.11e, which is very limited in 802.11 networks.

### 3.6 MAC Parameter Tuning for Best Effort Traffic in 802.11e Contention-Based Networks

Scalia et al. [23] investigated various optimizations of the 802.11e parameters and proposed an effective algorithm to maximize system throughput by adjusting the CWmin with respect to the network contention level. More attention was given to the best-effort traffic when it was deemed the traffic analogous to the legacy DCF traffic. Best-effort traffic resources were analyzed to determine their performance in the presence of QoS.

The collision avoidance nature of a wireless channel creates a tradeoff between channel access overheads and inefficiencies caused by time wasted due to backoff counter. The goal of Scalia et al. was to determine an optimized value for the CWmin value that maximized throughput by taking into account backoff times and collision reduction. This was highly dependent on network size and backoff time due to collision. Since the legacy distributed coordination function does not provide such room for modifications, it is almost impossible to find the optimal solution in throughput and transmission rate. Experiments were performed in...
a saturated condition for best-effort stations, where all stations will always have data to send to achieve a fine-tune the CWmin parameters.

However, in a real-life scenario, the above assumptions are slightly tolerated because it is not possible to estimate the number of per-class competing stations, and also it is not reasonable to assume that stations will always have data to send. Therefore, Scalia et al. inferred that the optimal CWmin is determined by comparing the derivatives of the backoff times with the derivatives of the collision times. They believe that when the two are almost the same, the optimum is found. Since collision decreases and backoff time increases with respect to higher CW size, the dynamic CWmin value can be found whenever the two derivatives are compared.

Also, in order to provide robustness in the proposed algorithm, it is kept as simple as possible. Each CW value updates occur at every beacon interval i, and this job is performed by the access point. The AP merely counts the overall time spent in backoffs (Bi) and the overall time spent in collisions (Ci) and updates the CWmin value with the formula below:

\[
\begin{align*}
CW_{\text{min}}(i) &= CW_{\text{min}}(i-1) \cdot 2 & \text{if } C_i > B_i \\
CW_{\text{min}}(i) &= CW_{\text{min}}(i-1) / 2 & \text{if } C_i \leq B_i
\end{align*}
\]

Tests were performed by comparing the throughputs between networks with fixed CWmin values and networks with proposed adaptive CWmin values. Also, in order to test the algorithm behavior in dynamic load conditions, the authors subsequently activated the new best-effort stations at certain time instants. Results showed that throughput degrades in a network with a fixed CWmin when new stations are introduced, while throughput remains stable in a network with adaptive CWmin, as depicted in Figure 7.
Figure 7. Overall Throughput in Standard (Fixed) CWmin and Adaptive CWmin [23]

Fine-tuning the best-effort traffic parameters does not grant a superior priority over other traffic. In 802.11e, best-effort traffic is only the third highest priority, below voice and video traffic. All tests conducted above did not take into account when higher-priority traffic is present. For further comparisons when other higher-priority traffic is present, a throughput repartition of bandwidth exists between traffic. The resources or bandwidth repartition is depicted in Figure 8.
Source repartition states that until higher-priority traffic is saturated, the remaining bandwidth will be obtained by best-effort traffic. This draws a very important conclusion: best effort traffic does not rob the resources meant for higher-priority traffic whenever adaptive CWmin is deployed. In fact, this adaptive scheme enables effective channel utilization without wasting it on backoff and collision.

Scalia et al. have successfully come up with a dynamic algorithm to adjust the CWmin values according to the network contention level, thanks to the ability to modify the parameters on a per-beacon basis. This has also provided elasticity to best-effort traffic, which is to regulate traffic using repartition according to available resources and the presence of high-priority traffic to satisfy the service differentiation requirements in 802.11e.
This chapter describes the experimental setup of eight laptops to study the behavior of traffic flow when prioritization is set for different classes of traffic.

4.1 Testbed Setup

The testbed consisted of nine Lenovo SL400 laptops, each with wireless connectivity. All laptops ran on an Ubuntu Linux 8.04 operating system. One laptop acted as a server, with the remaining ones acting as clients generating traffic (Figure 9). All laptops ran Frescor[16], a real-time wireless traffic emulator software. Frescor is a Linux application built for communications over IEEE 802.11 WLANs with 802.11e enabled. It is designed to work with network interface cards (NICs) conforming to WMM specification and employs an EDCA medium-access mechanism. All testbed laptops ran on Intel R PRO/Wireless 3945ABG wireless cards.

Figure 9. Real-Environment Testbed Setup
*Frescor* works according to the default 802.11e parameters shown in Table 1. When a specific traffic is defined, the packets are forwarded according to the transmission queues in Figure 1. The laptops were connected to a wireless router with the 802.11e feature enabled which forwarded the packets from client nodes to server and vice-versa. *Frescor* software enables a laptop to send four different types of traffic, each with different QoS priorities. The four different traffic classes are as follows: VO (voice traffic), VI (video traffic), BE (best-effort traffic) and BK (background traffic), in decreasing order of priority.

Voice traffic emulates real-time priority of VoIP traffic like Skype, online chat, etc. Video traffic emulates the real-time priority of video traffic such as YouTube streaming or online gaming. Both of these types of traffic are time critical. The best-effort traffic emulates real-time email and file transfer traffic, which are not so time-bound. Lastly, the background traffic tries to emulate all control traffic that does not involve sending useful data but rather sending data required to keep network connectivity.

Since the aim of these experiments was to investigate the behavior of 802.11e in a large-scale network, all outgoing traffic was kept as constant as possible in terms of packet size, running time, and sending bandwidth.\(^1\) In this way, the experiments utilized only the internal AC parameters shown in Table 1, in order to show prioritization. In the tests, all nodes began transmitting packets to the server at almost the same instant in order to allow channel contention resolution depending on the AC parameters.

Initial experiments involved only the *Frescor* WLAN application to see how the network behaved in terms of prioritization when eight laptops were present (Figure 9). This stage

---

\(^1\) That is, all traffic types send the same size data packets with similar intervals. This allows fair comparison among all types. This does not reflect actual traffic characteristics and will be the subject of future work. The aim of this work is only to study the impact of prioritization as the result of 802.11e on energy consumption.
verified that the 802.11e framework allows for prioritization when multiple nodes are competing for the medium. In the upcoming subsections, the actual behavior of the network in terms of energy efficiency is explained.

4.2 IEEE 802.11e Prioritization

Four testing scenarios, each depicting a real-time environment that could occur in practice, were simulated: **Basic QoS, Normal QoS, High QoS, and Extreme QoS.** Table 2 summarizes the types of traffic sent by a number of laptops in each scenario. The basic QoS setup used four laptops to observe the basic 802.11e behavior, while the remaining test scenarios used eight laptops to send different traffic to the server. Detailed configuration and results will be presented in the next subsections. Note that each laptop only sent one type of traffic. In each of these environments, all laptops were sending 800-byte packets. Overall transmission bit-rate was fixed to 1 Mbps on both sides to avoid automatic control. Although the setup for all experiments was the same, the streams produced by test clients were virtually different depending on the test scenarios.

<table>
<thead>
<tr>
<th>Traffic/Scenarios</th>
<th>Basic</th>
<th>Normal</th>
<th>High</th>
<th>Extreme</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voice (VO)</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>Video (VI)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Best Effort (BE)</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Background (BK)</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>
The packets mentioned above did not include any headers (UDP, IP, MAC). *Fresco* keeps packets a constant size to eliminate differences in transmission time. However, to investigate traffic behavior further, packet size can be varied for different types of priorities. The experiments in all these four categories were conducted with nodes sending a constant bit rate (CBR) traffic at 100 kbps, 150 kbps, 200 kbps, and 250 kbps. In basic quality of service, however, tests were performed starting from the lowest traffic data rate of 150 kbps instead of 100 kbps, in order to utilize more than half of the total channel capacity. Different data rates allow 802.11e prioritization to be studied under various loads for a fixed channel capacity of 1 Mbps.

Figures 10 to 13 show the percentage of successful packets received at the client side for the respective scenarios. Figure 10 shows the percentage of successful packets received by the sender in basic QoS. It can be seen clearly here that VO receives the highest priority, with about 63 percent of successfully received packets. Keep in mind that the application employs the user datagram protocol (UDP) entirely, where reliability is not the main concern.

![Figure 10. Percentage of Successful Packets Received in Basic QoS](image-url)
Figure 10 confirms the relative prioritization of all traffic types with 802.11e for the basic QoS scenario. Increasing the load on the network introduces heavy contention for the medium and reduces the success ratio of receiving packets for all traffic types, but still maintains relative priorities.

Figure 11 for the normal QoS scenario shows that increasing the number of low-priority stations brings down the packet-reception ratio for all nodes in the network but still maintains the relative prioritization.

![Percentage Packets Received for 802.11e Normal QoS](image)

Figure 11. Percentage of Successful Packets Received in Normal QoS

For the high QoS scenario, Figure 12 shows that the greater number of highest-priority nodes (VO nodes) in fact results in these nodes receiving lower priority on the medium due to mutual contention of the next highest-priority traffic of VI. It can be observed that the packet-
reception ratio for all traffic classes is reduced, due to the greater contention among the highest-priority traffic on the medium.

![Figure 12. Percentage of Successful Packets Received in High QoS](image)

Finally, Figure 13 shows that increasing the highest-priority traffic nodes even further results in a very low packet-reception ratio due to mutual contention. The next highest-priority traffic, VI, is not affected.
Results show that a higher priority does not guarantee a node getting adequate throughput from the network. The number of competing nodes with the same priority level greatly impacts this. The parameters in Table 1 show that the maximum contention window size for VO is 7, which is very small. During the backoff period when an initial collision happens, all nodes attempt to increase their CW size before trying to contend for the channel.

However, since the maximum contention window size is remarkably small, the nodes could encounter subsequent collisions even after numerous backoffs, and this is the reason why nodes with VO fail to get an adequate number of packets through, considering their high-priority levels. If the 802.11e standard was considered the saviour of real-time traffic, this may not hold if much of the traffic on the network is real-time traffic of the same high priority.

Another conclusion that can be drawn is relative to the severity of the impact of prioritization on low-priority traffic types. The goal of prioritization should be to give priority to
more delay-sensitive traffic but at the same time maintaining a balanced network in which lower-priority traffic gets an opportunity to send data. A good protocol with service differentiation should not only put emphasis on higher-priority traffic but also make ensure that lower-priority traffic does not get suppressed. Through this experimental evaluation, the limit where high priority traffic does not “rob” what should belong to lower-priority traffic can be determined.
This chapter shows the results of a study on the energy implications when the IEEE 802.11e was deployed on a real testbed. This is a novel approach for measuring the energy consumption of laptops when service differentiation takes place because most of the previous work has focused on the simulation of 802.11e. This chapter will cover the basic approach to measuring energy consumption when experiments are being conducted.

5.1 Introduction

In order to study the energy consumption behavior of laptops sending different types of traffic, the same scenarios in Table 2 were repeated and energy measurements were taken. The aim was to see how prioritization on the medium translated in terms of energy consumption for different nodes.

5.2 Energy Consumption Measurement Methodology

In the Linux environment, energy consumption was measured by acquiring battery levels before and after the tests. The metric used to compare energy consumption across different traffic types was per successful packet communicated. This was important because nodes from different classes had different packet-reception ratios. Since each packet’s reception consumed energy, it would not have been fair to penalize a class of nodes that had higher priorities and possibly a higher packet-reception ratio.
5.3 Basic QoS

In the basic QoS environment, only four laptops were used, each sending one type of traffic. The test on this setup was executed at bandwidths of 150 kbps and above for ten minutes each. As shown in Figure 14, the energy consumed by the voice data-sending laptop was the lowest among all other traffic, which agrees with the original hypothesis. As the bandwidth increased, there was more competition in the channel, and thus the BK traffic consumed a considerable amount of energy, while the VO traffic energy remained almost constant.

Figure 14. Basic QoS Total Energy Consumption
At the bandwidth was increased to 250 kbps (same run time as 150 kbps test), it can be seen that the difference in energy consumed by VO-, VI-, and BE-sending laptops becomes much less, thus making them almost similar, with the exception of BK traffic. The spike of energy consumption is the result of the total sending bandwidths of all laptops having reached the maximum channel capacity (1 Mbps), which caused the BK traffic to suffer serious suppression as well as excessive channel sensing without successfully received packets.

The almost constant energy consumed per packet by the higher-priority traffic when the sending-data rate was increased is an interesting trend. As medium contention increases, all nodes are able to get fewer packets through, which results in fewer packets received and, hence, reduced energy consumption due to packet reception.

However, energy consumption still increases as more attempts are made to send packets. The nodes that have higher priorities have a greater success ratio and, hence, waste less energy in contention. The lower-priority nodes waste considerably more energy in contention and are able to deliver fewer packets. This result implies that prioritization in 802.11e for channel access also results in a prioritization in terms of energy consumption for nodes.

5.4 Normal QoS Environment

In a normal QoS environment setup, similar tests as in the Basic QoS case were performed. After analyzing and graphing the results, as shown in Figure 15, it can be inferred that the same energy consumption pattern is observed as in the basic QoS environment setup. When the bandwidth is increased to 200 Kbps, a sudden increase in energy consumption is observed for BK traffic due to the same reasons as mentioned for the basic QoS case.
5.5 High QoS Environment

In a high QoS environment setup, three laptops were set to send VO traffic, 1 for sending VI traffic, and 2 each for sending BE and BK traffic. After analyzing and graphing the results, as shown in Figure 16, it can be observed that when the test was run at any bandwidth, there is an obvious difference in energy consumption for BK traffic compared to other traffic.

5.6 Extreme QoS Environment

This environment was perhaps the most important scenario considered in the prioritization experiments to counter the argument that high-priority traffic always gets better throughput. It can be clearly seen in Figure 17, VO traffic no longer has the highest throughput and, hence, lowest energy consumption per packet when five laptops were set to send the VO traffic. VI traffic consumed less energy due to less inter-class contention among nodes.
Figure 16. High QoS Total Energy Consumption

Figure 17. Extreme QoS Total Energy Consumption
CHAPTER 6

CONCLUSION AND FUTURE RECOMMENDATIONS

6.1 Challenges

In all experiments, laptops were manually started to begin sending traffic. This could have led to small errors in synchronization and could result in small errors. In future work, hopefully this process will be automated for all stations sending packets, beginning with a trigger packet from one station.

When all clients begin to send their respective traffic, the channel can get overloaded and result in some clients getting disconnected from the network. For this reason, in this study when one or more machines were disconnected, the entire test was restarted. Also, in order to obtain more precise results, each test was performed for up to five repetitions and the average values obtained before plotting the graphs.

In addition to human error, some machine errors occurred as well. Even after attempting to make all laptops run under similar conditions, it was observed that the LCD screens of a few of the laptops were a little brighter than others. LCD screens use a considerable amount of battery energy, thus making some readings not as precise and causing experiments to be repeated. In this study, the display brightness was set to be at the minimum.

Due to the large number of laptops sending high data rate traffic at the same time, connectivity issues were observed. Some laptops started losing their connectivity from the network due to overload when higher bandwidths were used to send data traffic. This limited the study to only a maximum of eight laptops contending for the medium.
In this study, each laptop sent only one type of traffic data, which is not the case in real life. An emulator that could make a laptop send different types of traffic data may be necessary in order to precisely observe the energy consumption pattern of laptops.

6.2 Conclusion

This study is a pioneering attempt to learn the energy-consumption behavior of laptops sending prioritized data using the IEEE 802.11e QoS standard. After considering the observations and results, it can be concluded that in any scenario, the laptops sending the highest-priority traffic consume the least amount of energy, except when a large number of high priority stations are present, which can introduce inter-class contention. To save energy, there should be an optimal number of laptops sending high-priority data and low-priority data so that the same class contention does not occur and no low-priority data gets suppressed.
REFERENCES


APPENDICES
APPENDIX A

FRESCOR INSTALLATION GUIDE IN UBUNTU

The following guide is the step by step guide to install Frescor application in a Linux Ubuntu 9.04 machine. Commands may be different if other Linux operating System is used.

1. The first thing to install is the git function because we will need to access the frescor repository to get the application. On your Linux machine go to Application --> Accessories ---> Terminal
2. At the command prompt type

[ user@localhost ~ ] # sudo apt-get install git-core

(Note: If prompted for password, type the admin password in the respective machine)

3. Type Y when prompted [y/n] to install the package.
4. To validate successful package installation, when 'git' is entered onto the command prompt, a series of option will appear. This means that the git package has been successfully installed.
5. To get the Frescor application type

    git clone git://rtime.felk.cvut.cz/frescor/fwp.git

6. The purpose of this command is to create a fwp folder in the home directory of 'user' and the application will be saved in the folder. Type Y when prompt. Application will be downloaded.
7. The author of the application will make modifications on the files occasionally. It is preferable for us to have the updated application. To update the Frescor Wireless Protocol, go to the home folder where fwp is located. cd to /Home/user/fwp/wme_test.
8. Type git pull for the updates.
9. Type 'make' at the command prompt to compile the application. You will find the compile error saying 'could not find -lncurses'. To solve this, install n curses package.
10. At command prompt, type

    [ user@localhost ~ ] # sudo apt-get install libncurses5-dev

11. Repeat step 9 after the ncurses package is installed. Compiling should return no error this time.
12. Here we go, ready to use the application. When the application is finished, we should expect a plot invoked when we used the ./run command. In order to see the graphical plot we need one more package for this which is gnuplot package. As usual, at the command prompt type

    [ user@localhost ~ ] # sudo apt-get install gnuplot
APPENDIX B

FRESCOR COMMANDS AND USAGE

Available Options to invoke Frescor

- **-B** default bandwidth for -b option [kbit]
- **-b** bandwidth of streams (VO|VI|BE|BK):<kbit>[@<msec> or /<bytes>]
- **-C** comment (added to header)
- **-c** count (number of seconds to run)
- **-g** histogram granularity [usec]
- **-G** show status in textual GUI
- **-I** <interface> send packets from this interface
- **-l** <loglevel> logging levels
- **-j** send jitter (0-100)
- **-o** output filename (.dat will be appended)
- **-q** gather statistics only after some queue becomes full
- **-Q** <bytes> set size for socket send buffers
- **-s** size of data payload in packets [bytes]
- **-T** default period for -b option [msec]

Usage:

**For computers running as clients:**

1. cd to the Linux directory in which Frescor is installed.
2. At the command prompt run the wcclient program using

```
sudo ./run -B<desired bandwidth> -b <stream type> -s <packet size> -c <running time> <server IP address>
```

(Note: more options can be added with reference to the above acceptable arguments)

3. Example;

```
sudo ./run -B 100 -b BK -c 600 192.168.0.198
```

The command above will produce an output data file ‘delay_stat.dat’ in the Frescor wme_test folder shown as below

```
# Invoked as: ./wcclient -B 100 -b BK -c 600 192.168.0.198
# Data gathered for 599 s.
# Stream 0: AC BK 100 kbps (800 bytes per 64.0 ms +- 0 us, 15 packets/s);
real: 86.5 kbps sent 9328 (15/s), received 911 (1/s)
## Format: msec csc% cs% sc%
```
Where “msec” is the time when reading is recorded and “csc%” is the percentage of successful packets received.

For computer running as server:

1. cd to the Linux directory in which Frescor is installed.
2. At the command prompt run the wserver program using
   ```
sudo ./wserver
   
   sudo ./wserver
   ```
3. The command above invoke the server program and loopback all the client packets back to the clients.

Energy Measurement Command

Basic battery reading command for Linux is used. To check current battery capacity, type the following at command prompt.

`cat /proc/acpi/battery/BAT0/state`

Sample Output:

```
present: yes
capacity state: ok
charging state: discharging
present rate: 4500 mAh
```
remaining capacity: 4268 mAh
present voltage: 14800 mV

General Notes

Sudo is the administrative command for normal users to execute some of the commands requiring admin privileges. If the user is a super user, “sudo” can be omitted when invoking the command lines mentioned above.