Power and Code Management in Wireless Networks

DISSERTATION

Presented in Partial Fulfillment of the Requirements for
the Degree Doctor of Philosophy in the
Graduate School of The Ohio State University

By

Ming Liu, B.E., M.S.

* * * * *

The Ohio State University

2005

Dissertation Committee:

Ming T. Liu, Adviser
Ten-Hwang Lai
Dong Xuan

Approved by

Adviser
Graduate Program in
Computer Science and
Engineering
ABSTRACT

Among different wireless technologies, wireless ad hoc networks and cellular networks account for a large part of wireless communications. In this work, we take an extensive study at the management of the two of very important resources in both wireless ad hoc networks and cellular networks: power and code (bandwidth).

In the first part, we focus on the power management in IEEE 802.11 ad hoc networks. We first aim at the mismatch between the power management protocol and the ever-increasing transfer rate. We propose modifications on the power management operations of IEEE 802.11 which introduce very little overhead but bring a big boost on the performance in both power efficiency and data throughput by combining a scheduling algorithm with the power management. Then, we concentrate on the new challenges faced by the power and synchronization protocols when pushing the IEEE 802.11 ad hoc mode into the multi-hop environment. We study the relationship between clock synchronization and power management in Mobile Multi-hop Ad Hoc Network (MANET). After reaching the conclusion that clock synchronization is vital for not only efficient power management, but many other network operations as well, we present a protocol to generate a globally synchronous system from synchronized sub-networks. We discuss the correctness of the protocol, and show the power efficiency brought by such a synchronous system in simulations. Lastly, we present a
framework, which takes advantage both of the two main approaches in energy conservation, i.e. power management and power control, to maximize power-saving. Because the goals that each scheme seeks contradict each other, we study the balance and the trade-off between them, and use them as the guideline on building our framework.

In the second part of this work, we shift to the code (bandwidth) assignment for multimedia traffic in Code Division Multiple Access (CDMA) networks. We propose several algorithms to handle the jitters of compressed video transmissions, and try to get a better playback quality while reducing the effects which video traffic has on other users in the system. Our simulations show the improved performance from those algorithms, especially at heavy traffic scenarios.
This is dedicated to those who love me and I love

“Are you going to Scarborough Fair?

Parsley, sage, rosemary and thyme

Remember me to one who lives there

She once was a true love of mine”

Scarborough Fair, Simon and Garfunkel.
ACKNOWLEDGMENTS

First of all, I must express my deep appreciation towards my advisor, Professor Ming T. Liu. For years of study and research at The Ohio State University, Professor Liu has always been there for me, encouraging me, guiding me, and helping me. This dissertation would not have been possible without his guidance. I am sincerely thankful to Professor Ten-Hwang Lai. Not only he serves in my reading committee, but also he enlightens and helps me generously during my dissertation research. I am also greatly indebted to Professor Emeritus Mervin E. Muller and Professor Wu-Chi Feng, their mentoring help me enormously, especially at the beginning of my study. My thanks also go to Professor Dong Xuan for serving as member of my reading committee.

I am grateful to the following friends: Min-Te Sun, Junfeng He, Cho-Yu Chiang, Jun Xu, Chong Kuo Chang, and Jian Huang. Most of them are senior students with who I shared precious years of enjoyable study and discussion at CIS/CSE. Their abundant supports make my research a unforgettable experience.

I thank my friends: Xinri Cong and Jing Xue, Yuming Ma and Jinmei Liao, Lining Yang, Yan Li, Shaoyong Wang, Yuan Wang, Jianping Jiang, Peg, Hilary, Sophia, Sue, and many, many more. . . . I do not want this to become a friends list, but I do want to say that I shall cherish the friendship with each one of them forever.
Last, but definitely not least, I can not express enough thanks and love to my family members, my dear father, Qinglian Liu; my beloved mother, Shulan Wang; and my sister Fang Liu. I owe my everything to them. They love me, care for me, and support me totally and unconditionally. This is dedicated to them and their love.
VITA

December 16, 1972 ....................... Born - Beijing, China

July 1996 ................................. B.E. Computer Science and Technology,
Tsinghua University, Beijing, China

December 1997 .......................... M.S. Computer Science,
The Ohio State University, Columbus, Ohio

October 1997–September 2004 .......... Graduate Associate,
The Ohio State University, Columbus, Ohio

PUBLICATIONS


FIELDS OF STUDY

Major Field: Computer Science and Engineering

Studies in:

Computer Networks Prof. Ming T. Liu
Theory and Algorithms Prof. Ten-Hwang Lai
Software Systems Prof. Anish Arora
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Abstract</strong></td>
<td>ii</td>
</tr>
<tr>
<td><strong>Dedication</strong></td>
<td>iv</td>
</tr>
<tr>
<td><strong>Acknowledgments</strong></td>
<td>v</td>
</tr>
<tr>
<td><strong>Vita</strong></td>
<td>vii</td>
</tr>
<tr>
<td><strong>List of Tables</strong></td>
<td>xii</td>
</tr>
<tr>
<td><strong>List of Figures</strong></td>
<td>xiii</td>
</tr>
<tr>
<td><strong>Chapters:</strong></td>
<td></td>
</tr>
<tr>
<td>1. <strong>Introduction</strong></td>
<td>1</td>
</tr>
<tr>
<td>1.1 Motivation</td>
<td>1</td>
</tr>
<tr>
<td>1.1.1 Power-Saving in Wireless Ad Hoc Networks</td>
<td>1</td>
</tr>
<tr>
<td>1.1.2 Multimedia in Multi-Code CDMA Networks</td>
<td>2</td>
</tr>
<tr>
<td>1.2 Our Contributions</td>
<td>4</td>
</tr>
<tr>
<td>1.3 Organization of the Dissertation</td>
<td>7</td>
</tr>
<tr>
<td>2. <strong>Background</strong></td>
<td>8</td>
</tr>
<tr>
<td>2.1 Data Networks and Wireless Ad Hoc Networks</td>
<td>8</td>
</tr>
<tr>
<td>2.1.1 Development of Data Networks</td>
<td>8</td>
</tr>
<tr>
<td>2.1.2 Wireless Ad Hoc Networks</td>
<td>17</td>
</tr>
<tr>
<td>2.2 Wireless Ad Hoc Network Technologies</td>
<td>19</td>
</tr>
<tr>
<td>2.2.1 IEEE 802.11 Standard</td>
<td>19</td>
</tr>
<tr>
<td>2.2.2 Bluetooth</td>
<td>27</td>
</tr>
<tr>
<td>2.2.3 Wireless Sensor Networks</td>
<td>30</td>
</tr>
</tbody>
</table>
2.2.4 A Survey on Power-Saving for IEEE 802.11 Based Wireless Ad Hoc Networks ............................................ 33
2.3 Multi-code CDMA and H.263 .......................................... 37
2.3.1 Multi-code CDMA .................................................. 37
2.3.2 H.263 Video Encoding ............................................. 39

3. Combining Packet Scheduling and Power Management in IEEE 802.11 Ad Hoc Networks ............................................. 41
3.1 Issues ........................................................................ 41
3.2 Proposed Algorithm .................................................... 46
3.3 Simulations and Results ................................................ 48
3.3.1 IEEE 802.11 and its enhancement algorithm results ........ 50
3.3.2 IEEE 802.11b and IEEE 802.11a Results ................. 56
3.4 Summary ................................................................... 62

4. Achieving Clock Synchronization for Power Management in Multi-Hop Ad Hoc Networks ............................................. 63
4.1 The Problem ............................................................. 64
4.1.1 Power Efficiency .................................................... 65
4.1.2 Broadcast and Multicast ........................................... 67
4.1.3 Beacon and Data Contention ................................... 70
4.2 The Protocol ............................................................. 71
4.2.1 Proposed Protocol .................................................. 72
4.2.2 Correctness of the Protocol ...................................... 75
4.3 Simulations and Results ................................................ 80
4.4 Summary ................................................................... 94

5.1 The Problem ............................................................. 99
5.2 Our Protocol ............................................................. 101
5.3 Simulations and Results ................................................ 104
5.4 Summary ................................................................... 110

6. Managing Code Assignment for H.263 Video in Multi-Code CDMA Networks112
6.1 The Problem ............................................................. 113
6.2 Our Approaches ........................................................ 114
6.3 Simulations and Results ................................................ 117
6.4 Summary ................................................................... 134
### LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 Internet usage statistics - the big picture</td>
<td>12</td>
</tr>
<tr>
<td>2.2 Slot time and minimum contention window size in IEEE 802.11</td>
<td>25</td>
</tr>
<tr>
<td>3.1 Transmission time for a 1KB data packet at different data rates in IEEE 802.11 standards (in $\mu s$)</td>
<td>43</td>
</tr>
<tr>
<td>4.1 Simulation configuration</td>
<td>82</td>
</tr>
<tr>
<td>5.1 WLAN card power consumption comparison</td>
<td>100</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>11</td>
</tr>
<tr>
<td>2.1</td>
<td>11</td>
</tr>
<tr>
<td>2.2</td>
<td>16</td>
</tr>
<tr>
<td>2.3</td>
<td>22</td>
</tr>
<tr>
<td>2.4</td>
<td>23</td>
</tr>
<tr>
<td>2.5</td>
<td>24</td>
</tr>
<tr>
<td>2.6</td>
<td>27</td>
</tr>
<tr>
<td>2.7</td>
<td>30</td>
</tr>
<tr>
<td>2.8</td>
<td>31</td>
</tr>
<tr>
<td>3.1</td>
<td>45</td>
</tr>
<tr>
<td>3.2</td>
<td>49</td>
</tr>
<tr>
<td>3.3</td>
<td>51</td>
</tr>
<tr>
<td>3.4</td>
<td>52</td>
</tr>
<tr>
<td>3.5</td>
<td>54</td>
</tr>
</tbody>
</table>
3.6 Results of enhanced IEEE 802.11 (II): (a) the data window size and (b) the average number of PS slots per PS station .......................... 55

3.7 IEEE 802.11b results: (a) the number of successful stations in ATIM window and (b) the number of successful stations in data window .......................... 57

3.8 IEEE 802.11a results: (a) the number of successful stations in ATIM window and (b) the number of successful stations in data window .......................... 58

3.9 The number of packets transferred: (a) IEEE 802.11b and (b) IEEE 802.11a ................................................................. 59

3.10 Results of enhanced IEEE 802.11b: (a) the number of successful stations in ATIM window and (b) the number of packets transferred .......................... 60

3.11 Results of enhanced IEEE 802.11a: (a) the number of successful stations in ATIM window and (b) the number of packets transferred .......................... 61

4.1 Discovery procedure of the new power-saving algorithm ................................................................. 75

4.2 Merging procedure of the new power-saving algorithm ................................................................. 76

4.3 Regular procedure of the new power-saving algorithm ................................................................. 77

4.4 Clock drifting between two nodes ................................................................................................. 79

4.5 Simulation results of Periodical-Fully-Awake-Interval protocol ................................................................. 84

4.6 Simulation results of power-saving scheme based on local synchronization (part I) .................................................................................. 86

4.7 Simulation results of power-saving scheme based on local synchronization (part II) .................................................................................. 87

4.8 Simulation results of our protocol (part I) .................................................................................. 89

4.9 Simulation results of our protocol (part II) .................................................................................. 90

4.10 Comparison of protocols at high mobility with 100 nodes ................................................................. 92

4.11 Comparison of protocols at high mobility with 20 nodes ................................................................. 93
6.10 Effects of code limit in algorithms 1 and 2 . . . . . . . . . . . . . . . 132
6.11 Effects of code limit in algorithm 3 . . . . . . . . . . . . . . . . . . . 133
CHAPTER 1

INTRODUCTION

1.1 Motivation

Wireless networks give the user the freedom of mobility, and in recent years, various wireless technologies have gained popularity. Along with the convenience provided by wireless networks come many new challenges. One of those challenges is the management of critical resources in wireless networks. In the mobile environment, there are more critical resources, which usually have a larger impact on system performance. In this work, we focus on the management of two of very important resources in both cellular and wireless data networks: power and code (bandwidth), study their effects on system performance, and try to provide efficient management algorithms for them.

1.1.1 Power-Saving in Wireless Ad Hoc Networks

Power is arguably the most limited resource for mobile devices. Faster processing speed, brighter display, further transmitting/receiving distance, and longer operating time between recharges can all be achieved by consuming more power. Even though the energy density has increased three times in the past 15 years, the field of battery technology still lags far behind that of semiconductors which has enjoyed an
exponential increase since its birth. Balancing power usage among different subsystems and reducing energy consumption as much as possible have always been major issues for the designers of mobile devices, wireless communication protocols and algorithms. Many power-saving schemes have been implemented or proposed for a variety of mobile devices. For example, Intel’s SpeedStep™ and AMD’s PowerNow!™ technologies can adjust the power consumption of the CPU by changing the operating frequency of the internal clock dynamically and shutting down unnecessary components, according to the demands of different applications. Many researchers are attempting to find better power conservation methods for network operations, ranging from Radio Frequency (RF) transceiver design to energy-aware routing protocols.\(^1\)

In a centralized system, we can let the central server, such as Base Station (BS) or Access Point (AP), handle the energy conservation. As a truly distributed system, the wireless ad hoc network\(^2\) has much more difficulties. Even IEEE 802.11, which standardizes the WLAN for both infrastructure networks and ad hoc networks, and enjoys a world-wide acceptance and a major share of WLAN market, leaves the power management issue for multi-hop ad hoc networks open. Thus, the first part of our work focuses on power-saving for multi-hop wireless ad hoc networks based on IEEE 802.11.

1.1.2 Multimedia in Multi-Code CDMA Networks

So far, we have mainly focused on wireless data networks, especially wireless ad hoc networks. Although they build on data networks technologies and have higher

\(^1\)We will have a detailed survey of power-saving algorithms in wireless ad hoc networks in Chapter 2.

\(^2\)We will talk more about infrastructure and ad hoc networks and other way to categorize wireless networks in Section 2.1.1.
transmission rates, most of those wireless data networks are aimed at small indoor environments and slow-moving users like those found in the office and in the home. For the capability of serving more diverse customers in a larger area, cellular networks are still a better choice, and enjoy a huge subscriber population and broad availability. Ever since AT&T Bell Laboratories developed the first U.S. cellular telephone system called the Advance Mobile Phone Service (AMPS) in the late 1970s, these systems have evolved from the first generation (1G) of analog technologies, to the second generation (2G) of digital networks, which are still mainly for voice service and have a data rate of no more than 14.4kbps, and then to the third generation (3G) of high speed data networks of at least 384kbps, which are being deployed world wide [26, 98]. With the proliferation of digital networks and the bandwidth increase in those cellular networks, they go beyond just delivering voice services to those on the move, and serve an important role the far-reaching, integrated, global data network. As the transmission rate gets faster and faster, new applications are introduced to cellular networks. Among them, multimedia traffic increases the fastest. This trend appeared in the Internet several years ago, and is clearly occurring in wireless networks as well.

There is more than one system even in cellular networks, and the differences among those systems also have impacts on the problems we try to solve. Time Division Multiple Access (TDMA) or Global System for Mobile Communications (GSM) are the most popular, but both have fixed channels and are thus inflexible on resource allocation. Code Division Multiple Access (CDMA) has soft capacity and many other advantages, like soft hand-off, better frequency efficiency, etc., so 3G wireless network standards converge on the CDMA technology. CDMA’s flexible resource allocation
responds well to the varied requirements of a video stream. Therefore, the second part of our work uses CDMA as its research environment.

However, how to deal with the high error rate in the wireless network remains one of the biggest problems that researchers have to solve. Although the bandwidth of wireless networks is comparable to that of wireline networks, wireless networks suffer from a very poor Bit Error Rate (BER). The typical error rate in wireline networks is $10^{-6}$, and is much lower in optical networks. However, it is not unusual to have a BER at the level of $10^{-2}$ in the wireless world, due to the variable transmission environment in wireless networks and the limited transmission power of Mobile Terminals (MTs). This error rate is acceptable for voice service, but because of the high compression rate, video traffic is very sensitive to errors. One erroneous bit might make the whole frame undecodable, and the effect of errors may spread into the following frames. Thus, the error rate is one of the problems we must solve in order to transfer video traffic over wireless networks.

1.2 Our Contributions

In this dissertation, we study the efficient management of critical resources in wireless networks, such as power and bandwidth. Our work can be divided in two parts. In the first part, we present the power-saving schemes aiming at the network functions above the physical layer, and focusing on Mobile Multi-hop Ad Hoc Networks (MANETs) which are based on IEEE 802.11 ad hoc mode. We observe the measurements presented by other research and provided by Wireless Local Area Network (WLAN) interface card manufacturers, and come the conclusion that
the power management approach\(^3\) plays a very important role in Multiple Access Control (MAC) layer power-saving. We also present a detailed study of the relationship between clock synchronization and power conservation, and this study leads us to take a synchronized approach in our power-saving algorithm just like in the standards. In the second part, we shift to cellular networks. Since in a centralized system like cellular networks, issues like power can be controlled by the base stations, we focus on another important resource, bandwidth. Particularly, we try to adapt the code management in multi-code CDMA networks to video traffic delivery.

- We first try to improve the energy performance of IEEE 802.11 based ad hoc networks by reducing multiple access contention. We make the management packets used for power management operations in IEEE 802.11 standard to piggyback extra information in order to establish contention-free scheduling at this stage. Because of this, we can also make the length of the Beacon Interval flexible to boost the networks’ efficiency. Then we modify the power management behavior to adapt to the faster transmission rate. All these improvements are done with only a little change to the standard and still under the original ad hoc environment on which the standard is based.

- In the second part of this work, we study clock synchronization and power management issues and their relationship with each other in MANET. IEEE 802.11 standard avoids these issues by assuming a single hop environment. We take a close look at some asynchronous protocols proposed by other researchers, and discuss the importance of clock synchronization to power management. We present a proposal which has the ability to discover unsynchronized nodes or

\(^3\)The discussion of different power-saving approaches and their categorization are in Chapter 2.
sub-networks and merge them into a synchronous network. And we compare
the performance of the power management protocols based on this synchronous
system to other existing schemes.

• In the third part of our power-saving research, we go beyond the MAC layer and
examine the two major approaches of power-saving algorithms, i.e. the power
management and the power control. The latter is usually coupled with the
routing function in the network layer. Many of the power control algorithms
ignore the power consumption of the idle mode and the effect of putting some
stations into sleep mode. After examining the interaction of the two approaches,
we present an algorithm combining both power management and power control
to achieve maximum power-saving for MANET.

• The last part of this work is on the code management in CDMA networks.
We study the problem of how to efficiently distribute network resources among
H.263 packet video streams in a multi-code CDMA network. Due to high flex-
bility of both the video encoding system and the CDMA system, it is not easy
to find an optimal solution for the problem. We present different algorithms
for transferring video in CDMA networks. We compare those algorithms in
our simulated environment with multiple video users in different situations and
show the results with various system loads. By observing those results, we pro-
duce some guidelines for designing code assignment algorithms for video traffic
in cellular networks.
1.3 Organization of the Dissertation

Following this introductory chapter on our motivations and contributions, this dissertation is organized as follows. In Chapter 2, we present background knowledge on general data and wireless networks, different wireless ad hoc technologies, CDMA networks, and H.263 video encoding schemes. The rest of the chapters discuss our contributions to the field. In Chapter 3, we present the improvements on the IEEE 802.11 standards, combining contention-free scheduling with power management. Chapter 4 displays a study of clock synchronization and power management in multi-hop ad hoc networks. Finally, in Chapter 5, we propose a power-saving algorithm which takes advantage of both power management and power control. While these three chapters represent the first part of our work, Chapter 6 comprises the second part, which deals with code management in multi-code CDMA networks for H.263 video traffic. At the end of this dissertation, we conclude our research and discuss possible future works in Chapter 7.
CHAPTER 2

BACKGROUND

In this chapter, we first review the development of general data network and wireless ad hoc networks. Then, we give background information on the IEEE 802.11 standard and other major wireless ad hoc networking technologies, and survey the existing energy conservation methods for wireless ad hoc networks. Lastly, we go over CDMA networks and H.263 video encoding.

2.1 Data Networks and Wireless Ad Hoc Networks

2.1.1 Development of Data Networks

There is nothing that has changed the life of people in such a way as computers, and there is nothing that has changed the usage of computers in such a way as networks. Ever since Alexander Graham Bell envisioned and finally achieved “transmitting vocal or other sounds telegraphically,” telecommunication has helped define our modern world. Twenty-three years after the birth of the first computer, ENIAC, researchers built the first computer network covering a long physical distance. That network has grown from connecting four computers at UCLA, UCSB, the Stanford Research Institute (SRI) and University of Utah to incorporating hundreds of millions of hosts around the world today [108]. Figure 2.1 shows the Internet host count
based on the Internet Domain Survey made by Internet Systems Consortium, Inc. [48]. The actual number of computers that are connected to the Internet should be even greater when we consider the common practice of Internet Protocol (IP) address sharing via Dynamic Host Configuration Protocol (DHCP) and Network Address Translation (NAT). Behind these computers, there are billions of users whose lives are changed by the Internet. We show the Internet usage statistics published by Nielsen//NetRatings and International Telecommunications Union (ITU) and their relationship to the world’s population in Table 2.1 [49]. The development of computer networks ultimately exposes the great potential of both computing and information exchanging. From Table 2.1, we can see that the network usage has been doubled in the past five years, while the world wide penetration rate is just over 10%, so there is still a lot of growing space.

Nowadays, not only do interconnected computers use the network to transfer bits of information around the globe, but networks change the way in which computing is done as well. Super computers that are built by interconnecting readily available PCs have comparable computing power to specially made mainframes, and computing power becomes a resource and commodity [31]. Sun Microsystems even adopted the slogan of “The Network is the Computer™”. Besides the advances in technical renovations, computer networks change our lives as well. More information than ever is just a click way for billions of people. More and more people realize the potential offered by computer networks and make use of it. Information flows smoothly along the networks. The ease and numerous ways of communication greatly reduce the necessity of physical closeness so that people are able to reach each other around the world just as easily as the next-door neighbor, give people greater freedom, and have
had a deep impact on our society [12, 70]. New network-centered business models change the look of today’s business world. E-business, Internet-based finance like online banking, Enterprise Resource Planning (ERP) tools, and others make today’s companies nimble, efficient, and able to interact closely with customers. We have never seen the world change so quickly and dramatically in such a short period of time, and the advance of data network technologies is the driving force behind most of these changes.

Accompanying all the freedom and convenience are new issues and challenges. As more computers and information are put into interconnected networks, information security becomes a major issue. Viruses, spywares, and other electronic junk, such as spam and adware, are also taking advantage of computer networks to spread more quickly and do more harm. Computer networks’ deeper penetration of our society demands wider bandwidth at both the backbones and Local Area Networks (LANs) to accommodate the fast-growing user population. On the other hand, faster networks give birth to newer applications. Realizing the importance of and problems associated with computer networks, researchers have made concentrated efforts to research those networks. Researchers try to improve the performance of the network, increase bandwidth, provide better service, and make it easier to access the network from any location.
Figure 2.1: Number of hosts advertised in the DNS (in millions)
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Africa</td>
<td>900,465,411</td>
<td>14.0%</td>
<td>12,937,100</td>
<td>186.6%</td>
<td>1.4%</td>
<td>1.6%</td>
</tr>
<tr>
<td>Asia</td>
<td>3,612,363,165</td>
<td>56.3%</td>
<td>266,742,420</td>
<td>133.4%</td>
<td>7.4%</td>
<td>32.6%</td>
</tr>
<tr>
<td>Europe</td>
<td>730,911,138</td>
<td>11.4%</td>
<td>230,923,361</td>
<td>124.0%</td>
<td>31.6%</td>
<td>28.3%</td>
</tr>
<tr>
<td>Middle East</td>
<td>259,499,772</td>
<td>4.0%</td>
<td>17,325,900</td>
<td>227.8%</td>
<td>6.7%</td>
<td>2.1%</td>
</tr>
<tr>
<td>North America</td>
<td>328,387,059</td>
<td>5.1%</td>
<td>218,400,380</td>
<td>102.0%</td>
<td>66.5%</td>
<td>26.7%</td>
</tr>
<tr>
<td>Latin America/Caribbean</td>
<td>546,917,192</td>
<td>8.5%</td>
<td>55,279,770</td>
<td>205.9%</td>
<td>10.1%</td>
<td>6.8%</td>
</tr>
<tr>
<td>Oceania/Australia</td>
<td>33,443,448</td>
<td>0.5%</td>
<td>15,838,216</td>
<td>107.9%</td>
<td>47.4%</td>
<td>1.9%</td>
</tr>
<tr>
<td><strong>World Total</strong></td>
<td><strong>6,412,067,185</strong></td>
<td><strong>100.0%</strong></td>
<td><strong>817,447,147</strong></td>
<td><strong>126.4%</strong></td>
<td><strong>12.7%</strong></td>
<td><strong>100.0%</strong></td>
</tr>
</tbody>
</table>

Table 2.1: Internet usage statistics - the big picture
In the traditional wireline network, on one side, research is pushing the transmission speed limit by replacing copper wires with fiber optic wires, and amazing work is being done to increase the capacity of each fiber [90]. The dream of an all-optical network is coming true [6], and the technologies are already there to replace even local access with fiber optics to provide huge bandwidths for the homes of millions of users. On the other side, since the Internet is today’s most important network, research focuses on improving its applications [22]. Multimedia services [115] and security [105] are a few examples of what today’s active networking research is interested in. Older applications are digitized and put onto the Internet platform while new applications are emerging such as Voice over Internet Protocol (VoIP) [35], digital TV, and on-line gaming [16]. With all these fronts, data networking research remains one of the most active research fields in computer science.

The development of network technology has also been moving into wireless networks [74, 100]. Starting with cellular networks in 1980s, wireless networks were originally deployed to provide analog voice services, and then evolved into digital networks for better utilization of bandwidth. Today, for wide area, there are the second generation cellular networks of digital services with enhanced data transmission support which provides a better signal, lower power consumption, and higher capacity for each cell. The third generation cellular networks are on the edge of being widely deployed by telecommunication companies, and provide customers with high-speed multimedia connections and new services like video messaging and watching TV on cell phones [32]. Yankee Group estimates that the number of global mobile users will reach 1.78 billion by the end of 2007, and in the US, wireless data revenue exceeded $4 billion in 2004. At the same time, in the local area, such as our home
or office, we have even more choices: 802.11b/g/n [44, 45], 802.11a [43], Home-RF, and High Performance Radio Local Area Network (HYPERLAN) are all the renovations and competing standards that have appeared in the last several years [102]. Among them, the IEEE 802.11 standards family dominates the market. In the few years since the technologies became standardized, products have already reached millions of customers and changed the way people access networks. WLAN “Hot spots” extend from downtown business districts to campus-wide and community-wide interconnected systems. With more and more people doing business or surfing the web in a coffee house, at the airport terminal, or in downtown city parks, WLANs are changing our lives.

Wireless networks also prove advantageous in fields hitherto unexplored. For example, in one new area, Personal Area Network (PAN), Bluetooth [9, 56] is gaining momentum quickly, because it can easily fit into many small devices, has a rather simple protocol stack, and requires low power. You might have a smart ID or smart key with Bluetooth chip in it. It can hold some personal information, letting you exchange electronic cards with others, or unlocking your workstation or car as you approach. Personal Digital Assistants (PDAs), cell phones, and notebook computers have communication capability and can be carried on your person. All of them can form a PAN and interact with other networking devices around you. Another rising star in this field is sensor networks [60, 92]. Hundreds of thousands of autonomous nodes, each composed of sensors, a transceiver, and a simple controlling unit, are deployed into an area to collect data and transfer them back to the collector. This enables automated around-the-clock monitoring. On other wireless frontiers, Radio Frequency Identification (RFID), which has already been used at toll booths on
highways and mass public transportation systems, is coming to retail stores. The biggest retail chain in America, Wal-Mart, Inc., has had it installed in 139 of its stores and three of its distribution centers. Soon, the lines at the cashiers will be history. Wireless networks do not always mean mobile and accessible only from a small area. The IEEE 802.16 standard (WirelessMAN) [46] aims at fixed broadband access for a metropolitan area. It provides an alternative for the “last mile” problem, which addresses the access line connecting end users to the local backbone of the telecommunication service provider. Taking advantage of the easy deployment of wireless access over wireline technology could let users at remote areas get connected to today’s information highway.

There are different ways to categorize wireless networks. One of them is to divide wireless networks into cellular networks and wireless data networks. Cellular networks are those, such as GSM, CDMA, and Wideband Code Division Multiple Access (W-CDMA), which are developed by telephone companies and based on the Public Switched Telephone Network (PSTN). They originally carried analog voice traffic, but have been evolving to digital service since the early 1990s. Wireless data networks are based on wireline computer networks, like WLAN. These networks are designed to carry data packets like those in the wireline computer networks. The categorization also could be based on the area covered by the networks. These range from global satellite networks, to cellular networks and WirelessMAN covering a whole city, to different kinds of WLAN technologies whose coverage area is anywhere from several hundred feet to a thousand feet, to the even smaller PAN, like a Bluetooth piconet, which has a very limited range of less than a hundred feet. Another way to divide wireless network technologies is based on the structure of the networks. Using this
scheme, wireless networks can be placed into two categories: infrastructure networks and ad hoc networks (Figure 2.2). All kinds of cellular networks are good examples of infrastructure networks. Wireless networks in this category have a central control point in each cell or the area that it covers. These central points are usually different from other nodes. They have more resources, and serve more functions. And very often they are fixed or semi-fixed, connecting to the wireline network. Inside the cell, communications happen solely or mainly between the central point and other stations. The wireless ad hoc network is also called the “peer-to-peer” network. It has no central point, and nodes inside the same network can talk directly with each other as long as they are in each other’s transmission range. All the nodes in the network share the responsibility of keeping the network up and running.
2.1.2 Wireless Ad Hoc Networks

The research of wireless ad hoc networks started back in 1972 as Packet Radio Network (PRNET) [57]. In the 1980s, it became Survivable Adaptive Radio Network (SURAN). At that time, wireless ad hoc networks were mainly utilized for the situations in which it was either impossible or too inefficient to put up a structural network and set up stations served as central points in advance, such as battle fields, emergency response or disaster relief. A wireless ad hoc network is formed by the nodes autonomously, and communication can happen between any pair of nodes within the range of each other [87, 97, 109]. All nodes in the network share the same responsibilities of discovering other stations and relaying information. With the dramatic increase in number of devices with wireless communication capability in recent years, a new picture of pervasive computing [101] has been drawn. This idea of computing and networking devices being everywhere around us calls for a very dynamic networking environment where nodes are moving and networks are forming and breaking all the time. Under such conditions, wireless ad hoc networks become a natural choice to connect devices in a quick and easy manner.

An ad hoc network is a highly distributed system with some desirable characteristics and unique limitations. Mobility is one of the most distinctive features of most wireless devices, and cause a highly dynamic network topology that could greatly affect network formation and routing operations. The heterogeneity of stations, ranging from laptop computers to tiny sensor “moles,” requires that protocols be highly adaptable. In wireless ad hoc networks, many traditional networking solutions need modification and adaptation, and numerous new challenges have to be addressed with
novel approaches. Below are some of these new challenges in wireless ad hoc network research:

**Network Formation** Unlike traditional wireline networks, where components such as computers and routers are up and running for the most of the time, stations in wireless ad hoc networks can be turned on and off pretty frequently and randomly. Limited transmission range and node mobility also make the topology highly unstable. Thus, many operations are built only on the knowledge of local topology and neighborhood information. Under such circumstances, network formation plays a much more important role than in traditional networks. Neighbor discovery [8, 19, 38], load balancing [2, 96], adaptability [68, 78] and efficiency [3, 79, 106] are common concerns in forming the networks.

**Clock Synchronization** Many networking operations are based on the assumptions that the differences among all stations’ clocks are within a certain limit. However, to find an efficient and scalable clock synchronization protocol in a distributed system has always been a daunting challenge. With wireless ad hoc networks, it is even harder to find a simple and power-efficient solutions to this issue [41, 66].

**Power-saving** Most mobile devices are battery powered, which makes energy conservation a prominent requirement for all operations. Translated to network protocol design, it means the less unnecessary packet receiving and sending and longer time spent in sleep mode, the better. The power-saving issue has a profound impact on many aspects of wireless ad hoc networks [28, 30, 52, 73].
Routing  The structureless and constant changing topology, the distributive nature that discourages the use of special routing stations, the fact that some nodes might not have enough computing power and/or storage space to handle complex routing algorithms, and the power efficiency requirements all have to be taken into consideration when designing routing protocols for wireless ad hoc networks [7, 36, 86, 89, 107]. Dynamic Source Routing Protocol (DSR) [11], and Destination-Sequenced Distance-Vector Routing Protocol (DSDV) [88] are examples of very successful routing protocols in wireless ad hoc networks.

Sometimes, these issues could be intertwined, and help resolve each other when considered together. Examples of issues that can be considered together are power and network formation [37], formation and routing [63, 114], and power and routing [21, 62, 110, 111, 113].

2.2 Wireless Ad Hoc Network Technologies

2.2.1 IEEE 802.11 Standard

The IEEE 802.11 working group is the IEEE working group for standardizing WLAN technologies. It published the first IEEE 802.11 standard in 1997, and updated it in 1999 [42]. The standard defines operation models for physical layer and MAC protocols in WLAN. The original IEEE 802.11 standard chooses the free Instrument, Scientific, and Medical (ISM) band at 2.4GHz, providing users up to 2Mbps bandwidth for wireless high speed data transmission. It supports three different transmission mechanisms in the physical layer: Frequency Hopping (FH) and Direct Sequence Spread Spectrum (DSSS), which use RF transmission, and another mechanism based on Infrared (IR) transmission. When using FH, Two-level Gaussian
Frequency Shift Key (2GFSK) modulation generates a data rate at 1Mbps, while Four-level Gaussian Frequency Shift Key (4GFSK) modulation can generate a higher data rate of 2Mbps. For DSSS, the modulation schemes for 1Mbps and 2Mbps are Differential Binary Phase Shift Keying (DBPSK) and Differential Quadrature Phase Shift Keying (DQPSK), respectively. IR uses Pulse Position Modulation (PPM), and 16-PPM results the 1Mbps rate, while 4-PPM results the 2Mbps rate. Under the umbrella of IEEE 802.11, the working group has published several amendments to take advantage of different physical layer technologies to provide higher transmission rates. In 1999, IEEE 802.11a [43] was introduced with Orthogonal Frequency Division Multiplexing (OFDM). IEEE 802.11a defines a group of transmission rates, including 6Mbps, 9Mbps, 12Mbps, 18Mbps, 24Mbps, 36Mbps, 48Mbps, and 54Mbps at 5GHz band. Among those rates, 6Mbps, 12Mbps, and 24Mbps are mandatory. Another physical layer extension, IEEE 802.11b [44], was also published in the same year using the original ISM band to remain compatible with IEEE 802.11, but giving users a 11Mbps transmission rate at a close range of 100 ft. – 150 ft., a 5.5Mbps rate between 150 ft. and 250 ft., or 2Mbps rate at 250 ft. – 350 ft. This rate improvement is acquired by employing 8-chip Complementary Code Keying (CCK) or Packet Binary Convolutional Coding (PBCC) (optional) as the modulation scheme. IEEE 802.11b also introduces a shorter Physical Layer Convergence Protocol (PLCP) preamble as an optional mode to further increase data throughput. In 2003, IEEE published another higher data rate amendment, IEEE 802.11g [45]. It defines the Extended Rate Physical Layer (ERP) with several different techniques. First, OFDM in IEEE 802.11a is adopted to 2.4GHz band as ERP-OFDM, and the supported rates are the same as in IEEE 802.11a. Second, PBCC in IEEE 802.11b is extended to
support 22Mbps and 33Mbps (ERP-PBCC). And last, a hybrid modulation combin-
ing a DSSS preamble and header with an OFDM payload is defined as DSSS-OFDM. It supports the same rate group as ERP-OFDM. All of these extensions only deal with physical layer technologies; MAC functions have stayed the same since the 1999 standard.

IEEE 802.11 provides two different system architectures for different situations. For home or office environment where users are mainly the same group of people, IEEE 802.11 works in the *infrastructure* mode, in which a special device called the AP serves as the center of a sub-network called a Basic Service Set (BSS) which is composed of and served by the AP and other stations in the communication range of AP. Inside a BSS, all the traffic is between the AP and the stations. There is no direct communication between stations. The AP is connected to the wireline network. Several BSSs interconnecting with each other via a local network form a Extended Service Set (ESS). Some basic functions, like roaming, are provided in ESS (Figure 2.3(a)). In this mode, by imitating computers that share a single uplink connection via a hub in an Ethernet LAN, the IEEE 802.11 standard tries to provide wireless access to wireline networks. For users gathered together temporarily or unexpectedly, like people in a conference room, IEEE 802.11 can also work in the *ad hoc* mode. In this mode, no special device is needed, and all the stations form an Independent Basic Service Set (IBSS) and talk to each other directly inside the IBSS (Figure 2.3(b)).

In the MAC layer, several schemes have been adopted in the IEEE 802.11 stan-
dard. The basic scheme is based on Carrier Sense Multiple Access with Collision
Avoidance (CSMA/CA). It is called Distributed Foundation Wireless Medium Access Control (DFWMAC) using CSMA/CA. In this MAC scheme, each node with data to send has to sense an idle medium for a period of time, called an Inter-Frame Spacing (IFS), before it can start a back-off count down, which has a length of a random number of slots bounded by the current contention window. This count down procedure will be waived only if the carrier sensing succeeds in its initial efforts in detecting an idle medium for an IFS. If the back-off procedure is broken by the detection of a busy medium, the count down will not resume until the medium goes back to idle for an IFS period (Figure 2.4). If packets collide with each other during the transmission, this back-off procedure will be repeated with an increased contention window. The increase rate of the contention window is exponential. Following every successful transmission, the contention window will be brought back to its initial value, designated as “aCWMin” in the standard. Three different IFSes are defined

Figure 2.3: Different architectures of IEEE 802.11 networks
to provide service differentiation in the system. Distributed Coordination Function Inter-Frame Spacing (DIFS) has the longest time, and is used before the transmissions of ordinary data packets with Distributed Coordination Function (DCF). Point Coordination Function Inter-Frame Spacing (PIFS) has a shorter length and is used in Point Coordination Function (PCF). In PCF, a central point (likely the AP) takes the scheduling responsibility so that time-sensitive traffic can be transmitted in a reservation or polling scheduling scheme. Short Inter-Frame Spacing (SIFS) is the shortest IFS. It is used by control packets to gain the highest priority in getting through the network. An example of several stations in a BSS accessing a shared medium is depicted in Figure 2.5.

The second MAC scheme also belongs to DFWMAC. However, to combat the hidden terminal problem\(^4\), it incorporates the RTS/CTS messages introduced in Multiple Access with Collision Avoidance (MACA) [61]. In this scheme, stations use RTS to contend for the medium and also request the receiver’s attention. If the receiver receives RTS correctly and is ready to receive information from the sender of RTS, it

\(^4\)The hidden terminal problem is the situation where a receiver’s neighbors are not always in the sender’s neighborhood, so some of the receiver’s neighbors can not hear the sender’s transmission and sense an idle medium. The following transmission then collides at the receiver’s end.
will send out a CTS packet as a response. Other stations overhearing this RTS/CTS exchange will defer their transmission according to the time indicated in the packet header. This RTS/CTS exchange will prevent nodes in the transmission range of the sender and the receiver from corrupting the session. Following a successful exchange of RTS and CTS, data frame and ACK will take place. Carrier sensing will still be used prior to all transmissions, but SIFS will be used for most cases, with the exception of DIFS being used to send RTS.

The last MAC scheme is DFWMAC with polling. It is used for PCF, while a special station in the BSS coordinates all the scheduling inside BSS. This scheme provides a reservation-based scheduling method rather than random access, so that it can provide some Quality of Service (QoS) for real time traffic.
Like most of the wireless communication systems, time is divided into time slots in IEEE 802.11, and communications are aligned with those slot boundaries. In Table 2.2, we show the minimum contention window size (aCWmin) and slot time (aSlotTime) defined in the standard. This requires a synchronization scheme to keep the clocks of all stations in sync. In the infrastructure mode, stations will synchronize their clocks with AP’s clock. In IBSS, the standard defines a Time Synchronization Function (TSF) to handle this. According to the TSF, at the beginning of every beacon interval, each station calculates a random delay time uniformly distributed between zero and $2 \times aCWmin \times aSlotTime$. If a beacon packet from any station is received during this period, the station cancels the pending beacon transmission. Otherwise, at the end of the delay, the station transfers a beacon packet containing its own timestamp. The collision of beacon packets is resolved by the same CSMA/CA method as in other transmissions. Upon receiving a beacon packet, a station will compare the packet’s timestamp with its own clock. It will adjust its clock only if the beacon packet shows a faster clock value.
The IEEE 802.11 standard also defines two modes for power-saving operations. When a station is in Active Mode (AM), it will stay active and ready to send or receive at any time. But if a station operates in Power-Saving Mode (PS), it will alternate between awake and doze states. In the doze state, most of the physical layer components of the interface card are shut down, and the power consumption is reduced to less than 4% of the peak transmission power. However, in the doze state, no packet can be sent, and neither is the station reachable, so the station has to wake up from time to time to keep communication going. Assuming all stations in an IBSS are synchronized by the TSF, IEEE 802.11 defines its power management in PS as shown in Figure 2.6. At each beacon interval, after a successful beacon broadcast, an Ad hoc Traffic Indication Message (ATIM) window\(^5\) of a fixed number of slots is dedicated to ATIM transmissions. All stations in the IBSS stay awake from the start of the beacon interval until the end of the ATIM window. If a station has data, it contends for ATIM transmission in the ATIM window, and waits for ACK after sending ATIM. At the end of the ATIM window, those stations which successfully finish ATIM/ACK exchanges stay awake for the rest of the beacon interval to send or receive data packets. Other stations, which either have no packets to send or receive, or for which the ATIM/ACK exchange fails, go to the doze state until the beginning of the next beacon interval. The same CSMA/CA access scheme is used for all packet transmissions including the beacon and ATIM packets.

\(^5\)Although both beacon interval and ATIM window lengths are adjustable parameter, stations in one IBSS should agree on the same beacon interval length and ATIM window length.
2.2.2 Bluetooth

Sweden mobile equipment manufacturer Ericsson Mobile Communications started research on cable replacement technology for its cell phones and their accessories in 1995. In 1998, many other leading computer and communication companies joined in, and it collectively became known as the Bluetooth Special Interest Group (Bluetooth SIG). The Bluetooth SIG published the first version of Bluetooth specification the next year [9]. The main goals of Bluetooth technology are low power and simple protocols, so that it can be adopted easily and for a variety of applications.

Bluetooth specification chooses a centralized topology for its basic networks. Nodes are grouped into piconets. Each piconet has, at most, eight active nodes, while there can be as many as 256 nodes in a low power mode called parked node. Parked nodes
do not participate in communication. Among those active nodes, one of them is master, while the others are slaves. Communications only occur inside the piconet between the master and slaves. Different piconets connect with each other through those nodes that belong to more than one piconet. These nodes participate in different piconets in Time Division Multiplex (TDM). Bluetooth specification defines a full stack of protocols for its operations, and these protocols are kept as simple as possible. The protocol stack is shown in Figure 2.7. The RF layer defines basic physical and electronic parameters, such as modulating method, timing, power emission etc. Similar to IEEE 802.11, Bluetooth chooses the free ISM band for its operation, and uses Frequency Hopping Spread Spectrum (FHSS) to reduce interference. The baseband layer specifies the encoding scheme, channel rate, packet types and formats. Each Bluetooth device has a unique address, called its Bluetooth Device Address (BD_ADDR). This is a 48-bit address. A Bluetooth piconet is a TDM system; time is divided into 625µs slots. Two kinds of links are defined in Bluetooth specification: Asynchronous Connection-less (ACL) links and Synchronous Connection Oriented (SCO) links. An ACL link is used for most of the connections. It is also the default link when a connection has just been established. SCO links provide a fixed bandwidth for real time traffic. Bluetooth supports several types of packets with lengths of one, three, or five slots and different Forward Error Correction (FEC) and Cyclic Redundancy Code (CRC) protection. The Link Controller layer configures and controls the link. It defines nine states for a Bluetooth node. When a node is inactive, and not connected with other nodes, it is in Standby state. Inquiry, Inquiry Scan, Page, and Page Scan are the states for link setup. Once a node is connected, it can be in either the Active, Hold, Sniff, or Park states. The difference between these
four states is how much power the node consumes. The Link Controller layer defines asymmetric neighbor discovery and link setup procedures, in which nodes assume either roles of “master” or “slave” before the procedure and act accordingly. The Link Manager layer controls the link setup, parameter negotiation. It also performs power control, deciding how the system will switch among the different low power modes defined in the Link Controller layer. Master/slave roll switch is also performed at this level. The Host Controller Interface is the interface between the hardware Bluetooth module and software programs residing in the host device. It encapsulates the low layer hardware implementation, and provides the upper layer with a unified interface. Layers beyond HCI are in the Bluetooth host, the device where the Bluetooth module is installed. The Logical Link Control and Adaptation Protocol (L2CAP) gives application a logical link for each pair of end to end communications. Although several logical links can be implemented in one ACL link, L2CAP will hide the multiplex from the applications. RFCOMM is an emulated serial port following the RS-232 standard, and is the basic function the Bluetooth designers trying to provide to users.

A lot of researchers try to improve the performance of Bluetooth in clustering [67, 77, 81, 99] and scheduling [5, 13, 15, 25, 33, 55, 53, 54, 94, 122]. However, since Bluetooth technology aims at “cable replacing” in a close-range environment, the power consumption for a Bluetooth device is low and its operations stay as simple as possible. Also, Bluetooth networks formed by Bluetooth devices are usually short-lived and simple. When a large network with hundreds of nodes and complex topology with multiple hops is needed, Bluetooth technology is not the best candidate for this situation. With the master-slave structure, power control can be done rather easily through the master node. So although Bluetooth is an up-and-coming technology in
wireless ad hoc networks, this dissertation does not include its power conservation issues.

### 2.2.3 Wireless Sensor Networks

With the advance of wireless and fabricating technologies, especially the introduction of the Micro-Electro-Mechanical System (MEMS), people can cheaply produce highly integrated systems on a large scale. These systems, called System-on-the-Chip (SoC), are tiny simple computers with limited storage. Their controlling units run small programs stored in the Erasable and Programmable Read-Only Memory (EPROM) on-board and carry out simple functions like collecting data at pre-defined intervals and doing some basic information processing based on the data collected. Putting such a SoC along with different kinds of sensing meters and a low power RF transceiver creates a “mote,” a node in a wireless sensor network (Figure 2.8).
A sensor network is composed of a large amount of nodes. Each of these, or at least most of these nodes has very limited computing power and storage. They can only perform some simple functions. The node carries sensors to monitor and record changes in its environment [24]. It could be light, movement, velocity, temperature, humidity, magnetic field and many, many other things. First, the nodes are deployed either strategically or just randomly. Once they are deployed, they will form a network autonomously. Then each node starts its sensing tasks, and report the data (the information collected by the sensors) to the system communication point using the network. The data could be processed either before sending or along the way to the final destination. Eventually, the collective information could be stored at the system’s communication point and wait to be picked up, or sent via other communication networks to the analyser. The low maintenance, high independence and low cost make the sensor network a very good candidate for collecting data automatically in a harsh or hazardous environment. In addition, the sensor network can do long-term monitoring that requires minimum human intervention. Due to the large amount of
nodes, their random deployment\textsuperscript{6}, the low processing power of each node, and the environment in which they are usually deployed, protocols for sensor networks need to be simple, highly adaptive, and highly tolerant.

Although the size limit of motes for wireless sensor networks makes the power issue as significant here as for any mobile networks, most sensor networks research is dedicated to network formation, routing, application, and system areas. Power conservation in wireless sensor networks is usually approached from a different angle \cite{117}. Because of the differences in many aspects of network operations between wireless sensor networks and general wireless ad hoc networks, discussed at the beginning of this dissertation, we leave the power issue for extended future research in Chapter 7.

Another contending technology for wireless LAN is HIPERLAN/HIPERLAN2, which is developed from the Broadband Radio Access Network (BRAN) and supported by the European Telecommunications Standards Institute (ETSI). However, it supports an infrastructure network only where an AP exists in a network and all the MTs are associated with that AP. Although unlike the infrastructure mode of IEEE 802.11 standards, HIPERLAN allows direct communications between MTs via a direct link phase inside each 2ms frame, its MAC protocol is based on the Time Division Multiple Access/Time Division Duplex (TDMA/TDD) scheme that is centralized to an AP. Thus, HIPERLAN is not an ad hoc network. With the existence of a central station to which every node listens, many MAC issues are handled much differently. Hence we will not include HIPERLAN in our study.

\textsuperscript{6}Although the sensor network does not require nodes to be deployed randomly, to make the generality of the protocol, the designer should not make the assumption of a certain topology.
2.2.4 A Survey on Power-Saving for IEEE 802.11 Based Wireless Ad Hoc Networks

As we point out in Chapter 1, power is one of the most important issues in wireless ad hoc network research, so there are many researchers devoting themselves to this area. They try to find innovative way to conserve energy from every aspect of network operations. In this section, we will survey this research.

A common method of reducing energy consumption in wireless stations is power management. It will put stations in a sleep (or doze) state like in the IEEE 802.11 standards which we describe in Section 2.2.1. The main challenge for power management is to have an efficient paging scheme to wake up stations when there are packets addressed to them. The power management algorithm used in IEEE standards is based on a synchronous system. Clock synchronization itself is a problem in a highly distributive system like an ad hoc network. So some researchers try to devise asynchronous power management schemes. In [29], the author proposes that each station shall stay as much as half of the beacon interval to ensure the overlap of awake states from stations with different clocks. The ATIM window is also doubled, being placed both at the beginning and end of the half-beacon-interval awake period, to speed up the asynchronous paging process. To compensate for the efficiency loss introduced by reducing the data window to only half of the beacon interval, stations try to track their neighbors’ “phase” (when a station will be up to listen for ATIM packet) and arrange a schedule based on this information to send as many data packets as possible. Tseng et al. borrow a term from distributed computing called “Quorum” to create a rendezvous pattern for stations [112, 52]. According to the “Quorum” schedule, each station stays up for the whole beacon interval for $2n - 1$ out of $n^2$ intervals. This
schedule will assure an overlap of awake time for any pair of stations. By increasing \( n \), the percentage of time a station has to stay awake will always be reduced, but the time to discover the other station’s paging increases. This is a common trade-off in this kind of asynchronous schemes. In [121], Zheng et al. use the same method in which stations stay awake for the whole beacon interval for paging, but with a different wake-sleep interval pattern. They prove mathematically the lower bound of percentage of such fully-awake intervals, which is \( k_u \times k_v \geq m \times T \) where \( k_u \) and \( k_v \) are the number of intervals each station, \( u \) and \( v \), has to stay awake in every \( T \) intervals, and \( m \) is the number of overlaps of the schedules of \( u \) and \( v \) will have over every \( T \) intervals. To offset the low power efficiency of asynchronous schemes, stations will track the schedule and clock their neighbors passively. The authors also introduce on-demand power management which causes stations to stay in active mode, i.e. no sleep state at all, when the data packet backlog reaches a certain threshold. Ye et al. groups the stations in a network based on their clock values [120]. Stations stay awake for the whole length of a beacon interval from time to time just to listen for other synchronization groups’ beacon announcements. Once a station discovers a different clock value among its neighbors, it starts to follow both its own ATIM window and the ATIM window of the newly discovered group. Eventually, the different groups could be interconnected and the whole system will become fully connected. Other researchers focus on other aspects of power management. Chen et al. study the relation between power management and routing [18]. With stations in the doze state, the routing table could change dramatically. The authors propose a group of nodes to form a core backbone to keep the connectivity throughout the system. Jung et al. take a look at contentions in the IEEE 802.11 power management [58]. The authors
try to fine tune the length of the ATIM window relative to the beacon interval to reduce contention in the data part of the interval.

The other major approach to power-saving is power control. In this approach, energy is saved by reducing transmission power to the minimum level needed to deliver the packet to its destination. The IEEE 802.11 standards support this by accepting as many as eight different transmission power levels that can be defined by manufacturers. However, the standards themselves do not elaborate how to adapt these different power levels into operations. This is understandable since the current IEEE 802.11 standards are built for a single hop network. The power control in it is straightforward. However, in a MANET, power control is closely related with routing algorithms, and is much harder. Most researchers take transmission power and remaining energy into consideration when picking routes [111, 80, 113, 75, 103]. Ryu et al. divide stations into battery-powered terminals and outlet-powered terminals and treat them differently in the routing participation [37]. In [62], the authors combine not only the remaining power of each station but also the rate that the battery is drained into the routing algorithm. On the other hand, when tracking the battery is too expensive, thresholds are set to determine the routing behavior of stations [64]. By combining battery information in routing calculations, and based on Minimum Total Transmission Power Routing and Minimum Battery Cost Routing, Toh et al. propose Min-Max Battery Cost Routing and Conditional Min-Max Battery Routing [110]. In [17], the authors split a flow onto different paths by a flow augmentation algorithm to maximize system lifetime. Cho et al. introduce a delay which is inversely proportional to the residual energy of the station to the flooding routing request message in on-demand routing [21].
Another group of power-aware routing schemes use the cluster-based approach [107, 18]. These algorithms create a hierarchical structure in the system, and use only a subset of the stations to relay the traffic. They work well in a network where some nodes are more resourceful than others in terms of computing and power. However, since we consider a general wireless ad hoc network in this paper, even though stations could be heterogeneous, the randomness usually inherited in this kind of network still makes us take a non-clustered, pure ad hoc approach.

In the MAC layer, Jung et al. introduce Power Control MAC, which uses different power levels for control packets and real data packets [59]. Also, many researchers argue that lower transmission power can allow more transmissions co-exist without interference. Thus, several joint scheduling and power control schemes are proposed in [27, 82] to reduce contention and maximize throughput.

Other than turning off the transceiver and reducing transmission, researchers also study the relation between contention and power consumption [20]. In [28], Fang et al. take a cross-layer approach, which uses the MAC layer information in routing selection to reduce medium contention. Bononi et al. propose a Power-Save Distributed Contention Control in [10]. Transmitting stations keep track of slot utilization in the backoff windows, and use that and the number of transmission attempts already performed to calculate the transmission probability. Designing electronic devices which consume less energy is always an active front in energy conservation research. However, we will leave those issues to electrical engineers, and will not discuss physical layer power-saving in this work.
2.3 Multi-code CDMA and H.263

2.3.1 Multi-code CDMA

CDMA was first introduced in the 1940s. Because of its special anti-jamming feature and the complexity of its transceiver, it was originally only used by the military. In the 1980s, with the improvements in both computing power and chip making technologies, in addition to TDMA and Frequency Division Multiple Access (FDMA), CDMA became commercially available as another cellular network multiple access scheme.

CDMA has many desirable features like high spectrum efficiency, soft-capacity, soft-handoff, securer transmission, etc. So it has become the most promising multiple access scheme for the next generation (3G) of wireless networks. In a CDMA system, every user makes use of the whole band assigned to the service provider. All the transmissions in the same cell are actually “jammed” together. In order to make out the signals from the sender at the receiving end, different codes are used. A code is a special bit-sequence. The bits of a code are called chips, to distinguish them from the information bits being transferred. If a user wants to transfer some data, it needs to get a code, usually from the BS. Once it is assigned a code, it uses this chip-sequence to encode every bit it sends. The receivers apply the same code to the jammed signal it got, which contains all the senders’ signals, and only the information bits from the sender which it is interested in will be decoded. A system with a 64-chip code length can generate as many as 64 different orthogonal codes. And in such an orthogonal-code system, the only noise is background noise. If we want to increase system capacity, we can use longer codes or generate non-orthogonal codes. Longer codes will reduce the bandwidth that can be used by each user, since it requires
more chips to represent each information bit. However, the non-orthogonal code introduces interference among users. Other users’ transmissions will contribute to the total interference each user experiences. The more users transferring at the same time, the higher the interference. And if that interference goes beyond an acceptable level, the decoder will generate erroneous bits. So there is a trade-off between the number of users that a system can accommodate at a time and the interference each user will get. With the advance of today’s receiver design technology, CDMA systems use non-orthogonal codes for larger capacity.

Because both the chip length and bandwidth are fixed, the information bandwidth linked to each code is, to a certain extent, also fixed. It is sufficient for voice users, but not good enough for multimedia traffic, especially in the case of video transmission, which usually has a large variance in video bandwidth requirement. This can be solved by allocating more than one code to a user when needed. This scheme is called multi-code CDMA. It is the most common way for multimedia transmissions in the CDMA environment to provide users multi-rate services.

Also in this paper, we use two different kinds of receivers, the Matched Filter (MF) receiver and the Minimum Mean-Squared Error (MMSE) receiver [84]. The MF receiver uses a linear filter to maximize the signal to noise ratio from the background “white noise” at the output. By making some assumptions (the wave shape of the signal is known, the input (of the filter) noise is addictive, and its power spectrum density is known) the received signal is decomposed as

\[ r(t) = s(t) + n(t) \]  

(2.1)

After this, a transformation which can maximize the signal to noise ratio at the sampling time \( t_0 \) is done. This filter is commonly used in signal processing. From
the characteristic of CDMA, we can see that if all codes in the system are truly orthogonal, other users’ transmission is truly “white” noise to the intended receiver. However, in the real world where non-orthogonal codes are used, that is no longer the case. Multiple Access Interference (MAI) from other users causes the result of the MF receiver to deteriorate quickly as the number of users in the system increases. The MMSE receiver takes the advantage of knowing of other users’ existence in the system. It denotes the output signal of the receiver bank in a vector form as

\[ y = RA_d + z \]  

(2.2)

Matrix \( R \) is the correlation matrix of codes used in the system, and matrix \( A \) is a diagonal matrix of received amplitudes. Other vectors in the formula are the output of receiver bank \( y \), data being transferred \( d \), and noise at the receiving end \( z \). By applying transformation \( L_{\text{MMSE}} \) to conventional receiver output \( y \), we try to minimize the mean-squared error between the actual data and the output of the MMSE receiver, \( E[|d - L_{\text{MMSE}}y|^2] \), and the we can find the MMSE detector transformation

\[ L_{\text{MMSE}} = [R + (N_0/2)A^{-2}]^{-1} \]  

(2.3)

### 2.3.2 H.263 Video Encoding

H.263 and H.263+ are recommendations from the Geneva-based ITU [47]. They are based on the previous recommendation H.261. H.261 was made to transmit the packet video stream over the Integrated Service Business Network (ISBN). Due to their bandwidth efficiency and versatile controls, H.261/263 recommendations are among the most popular schemes in low bandwidth video transmission.

H.261 supports frame resolutions of 176 by 144, called the Quarter Common Intermediate Format (QCIF). It also optionally supports the resolution of 352 by 288
H.261 uses a compression algorithm based on the Moving Picture Experts Group (MPEG). Just as MPEG does, it divides frames into intraframe (I-frame), and interframe (P-frame). Only I-frames have all the information, which can be decoded without referring any other frames. In order to take advantage of static redundant information to save bandwidth, P-frames contain only the information which differs from the previous frame. When decoding a P-frame, a decoder needs information from previous frames. The motion detecting algorithm is used to generate P-frames. Discrete Cosine Transform (DCT) and Differential Pulse Code Modulation (DPCM) coding are used, such as the standard MPEG algorithm.

H.263 inherits all the features of H.261 and adds many new ones to give users more control over the video streams created, and to incorporate new techniques for better performance. H.263 extends its supported resolution range from 128 by 96 to 1408 by 1152 to fit different available bandwidths. In addition, H.263 also allows users to insert synchronization points inside a frame for fast recovery from erroneous transmission. It adds options for a syntax-based arithmetic coding, a new motion predicting mode, an advanced intraframe coding algorithm, etc. And all these options give the user flexibility to make a better use of bandwidth while transferring the video across the network.
CHAPTER 3

COMBINING PACKET SCHEDULING AND POWER MANAGEMENT IN IEEE 802.11 AD HOC NETWORKS

In this chapter, based on the observation that the power management function defined in the original standards becomes less efficient under today’s ever faster transmission rate, we present a protocol with only a small modification to the power management in the standards, which adapts to the higher rate and also incorporates a contention-free scheduling function for data transmission within the beacon interval. We show, by simulations, that this enhanced algorithm results in better efficiency on both system throughput and energy preservation.

The rest of the chapter is organized as follows. We analyse the issues and observations in Section 3.1, and then present our proposed improvements in Section 3.2. In Section 3.3, the simulation results are presented and discussed. Finally, we summarize the whole chapter in Section 3.4.

3.1 Issues

As shown in Section 2.2.1, with the development of technologies in the physical layer, the transmission speed of IEEE 802.11 gets faster and faster. The increase of the transmission speed reduces the transmission time of each packet. Based on
the Physical Layer Management Entity (PLME) TXTIME calculation indicated in
the standards, we show the transmission time of a 1KB data packet at different
transmission rates in Table 3.1. The decrease of transmission time lessens the time
spent on the actual transmission by the stations who have only a few packets to
transfer. However, according to the standard, those stations still have to stay in AM
for the whole beacon interval. That makes the power-saving algorithm described in
the standard inefficient.
<table>
<thead>
<tr>
<th>Rate</th>
<th>FH</th>
<th>DS</th>
<th>OFDM</th>
<th>802.11</th>
<th>802.11a</th>
<th>802.11b</th>
<th>802.11g</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CCK</td>
<td>PBCC</td>
<td>ERP-OFDM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>48</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>54</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.1: Transmission time for a 1KB data packet at different data rates in IEEE 802.11 standards (in µs)
A shorter transmission time of a single packet also makes the CSMA/CA multiple access costlier, especially to the stations which do not have many packets to transfer in a beacon interval. Such a short transmission time means more stations could be accommodated in a beacon interval. Simple simulations show that in the CSMA/CA multiple access scheme used in IEEE 802.11, the collision rate for the same transmission window increases fast as the number of stations increases (Figure 3.1). Contentions could also get worse because of the synchronization effect caused by the power saving procedure mentioned in the previous section. Although the authors of the standard noticed this problem, their solution becomes less effective when there are more stations transferring in a beacon interval as the packet transmission time reduces.

From Figure 3.1 we can see that the number of successful stations in the same size window changes along with the number of stations participating in contention. The fixed lengths of both the ATIM window and the beacon interval, and the two-contention transmission (one for ATIM and one for data packet) of the IEEE 802.11 ad hoc mode power-saving operation, often cause the mismatch of the number stations in each contention period. When the ATIM window is too short, the contention in the ATIM is high and only a few stations can send their ATIMs successfully. Then the following data window of the same beacon interval is under-utilized. If the ATIM window is too long with many transmitting stations in the system, many stations can enter the data transmission. This will cause not only higher contention in the data window, but will be a waste of energy for those who could not get a chance to transfer during the current beacon interval, since all stations who succeed transferring their ATIMs are required to stay in the awake state throughout the beacon interval. The
Figure 3.1: Contention of CSMA/CA algorithm at different loads and window length
relationship between the lengths of the ATIM window and the corresponding beacon interval is discussed in [116]. The authors indicate that the ATIM window size should be approximately \( \frac{1}{4} \) of the beacon interval, but do not focus on the effects of load variety. We can see here that it affects this optimal ratio between the ATIM window and the beacon interval quite significantly. In [58], the authors notice the inefficiency of the fixed ATIM window size under different loads, so a dynamic adjustment of the ATIM window size is presented. However, the increase and decrease of the ATIM window can only ameliorate the contention in ATIMs, and the mismatch between the ATIM window and the data window is not fully addressed.

### 3.2 Proposed Algorithm

More than once we have mentioned that in an IBSS, all stations are required to be able to hear from each other directly. This requirement put a limit on the formation of the IBSS, but also could be taken advantage of for operations of the IBSS. Because all stations are one-hop neighbors of each other, and stay awake during the ATIM window, all the successful ATIM packets are received by every station in the IBSS. The proposed algorithm uses these overheard ATIMs to generate a contention-free schedule for data transmission in the rest of the beacon interval, rather than let those stations who have succeeded in the ATIM window contend again for the data transmission. With all the information received at each station during the ATIM window, a deterministic scheduling can be generated. This not only eliminates extra contention in the data transmission but also increases the efficiency of power saving. And all of these proposals could be accomplished with only minimal extra information exchange.
As the transmission time of each packet reduces, it is neither necessary nor efficient to require all stations who have succeeded in ATIM transmissions to stay in the awake state throughout the whole beacon interval. With a typical beacon interval of 100 ms and a transfer rate of 11Mbps in IEEE 802.11b, a station which has only one packet to transfer only needs to be in the awake state for around 1% of the data window (assuming a 1KB packet). It is easy to see that this time will become even smaller as the transmission rate of the physical layer keeps increasing. A contention-free schedule can allow the stations finishing their transmissions early in the data window to switch to the doze state in the middle of the data window. To further improve this scheme, we let stations with less packets to transfer transmit earlier than those that have many packets to exchange. This makes more stations go to the doze state in the middle of the beacon interval and stay in it longer.

Since we eliminate the contention for data transmissions, we can know the transmission time of each station if we know the number of packets to be transferred by each station. We modify the standard a little bit to include that information in the ATIM. Because the ATIM defined in the standard has only a MAC header, Frame Check Sequence (FCS) and an empty frame body, it is easy to add a frame body of one or two octets to deliver that information that will not affect other operations defined in the standard. We also make the length of beacon interval variable, so that it can accommodate all the packets from all the stations that have succeeded in the ATIM window. This flexible window length can avoid a situation in which a station has succeeded in the ATIM window but could not transfer during the data transmission part, i.e. the mismatch of the ATIM window and the data window. The variable length of the beacon interval can also increase the system efficiency by adapting the
beacon interval length according to the system load. Because switching between the awake and the doze states takes some amount of energy too [104, 58], for a light load situation, we set a lower bound for the length of the beacon interval to prevent the algorithm from adopting a very short one. That is also the reason we set a threshold for when stations shall stop going to the doze state in the middle of a beacon interval.

If the power saved by switching to the doze state for the remaining beacon interval cannot justify the power for turning the transceiver off and then back on at the beginning of the next beacon interval, then the station should just stay in the awake state.

To summarize, we present our modified power-saving algorithm in Figure 3.2.

3.3 Simulations and Results

We have simulated both the original IEEE 802.11 power-saving operation and the proposed algorithm at different transfer rates. Simplifications we have made in the simulations include a simple traffic pattern which the same number of stations contend for the ATIM in every beacon interval, a single sink as the destination of all data traffic, a uniform distribution between one and MAX_PKT (MAX_PKT is defined as 4, 10 and 20 for IEEE 802.11, 802.11b and 802.11a respectively) packets of the number of packets to be transferred in a beacon interval for each station, the use of no control packet in the simulation except RTS, CTS and ACK, and an error-free medium for all transmissions. Other configurations are that RTS/CTS is used to hold the channel, and priority is given to the continuous transmission. With this kind of medium-holding mechanism, when a station has multiple packets to transfer, after it acquires the medium by a successful exchange of RTS and CTS packets, it
Modified power-saving algorithm

for each beacon interval
    Contend to send beacon as defined in IEEE 802.11
Upon receiving a beacon, enter ATIM window:
    if have packet to transmit
        then
            Contend to send ATIM as defined in IEEE 802.11
        endif
    endif
    Listen to every ATIM and gather scheduling info
    Sort stations who successfully send ATIM based on the number of packets they will transfer and their ID (as tie-break)
End of ATIM:
    Calculate the length of current beacon interval based on all ATIM received during ATIM window
    if successfully transfer ATIM during ATIM window
        then
            Transfer data packet according to the sequence decided in ATIM window
        endif
    else
        Go to PS until next beacon interval
    endif
else
    Go to PS until next beacon interval
endif
endfor

Figure 3.2: New power-saving algorithm
will not transmit RTS/CTS for any of the following packets. Other stations will defer their transmissions according to either the Network Allocation Vector (NAV) or the information about the number of packets to be sent carried in the ATIM of the currently-transmitting station. A very similar mechanism is defined in the standard called the control of channel for transmitting multiple-fragment Multiple Access Control Service Data Unit (MSDU).

3.3.1 IEEE 802.11 and its enhancement algorithm results

In IEEE 802.11, the slot length is 20\(\mu\)s. It takes 221 slots to transfer a 1KB packet. Using a typical value of 100\(ms\) for aBeaconPeriod and an ATIM window of 50 to 500 slots, the upper limit of the number of packets that can be transferred per beacon interval is from 21 to 23. The number of collision slots mainly remains under 20 in a beacon interval except for more than 40 stations and a 500-slot ATIM window (Figure 3.4(b)). Because the time spent on each data packet transmission is so long, collision slots and idle slots are not very significant. We can see that the change of parameters does not affect the number of packets transferred, i.e. the system throughput, much in Figure 3.4(a). And these numbers are very close to the upper limit. This is not true only when there are too many or too few active stations. From Figure 3.3(a), we know that both of those cases lead to very few stations entering data transmission, either because the contention in the ATIM window is too high, or because the traffic is too low. So the power-saving algorithm works well in the original standard, and the mismatch between the ATIM window and the data window is not significant in most cases. However, in the next section, we will see that these claims are no longer true at a higher transmission rate.
Figure 3.3: IEEE 802.11 results(I) (a) the number of successful stations in ATIM window and (b) the number of successful stations in data window
Figure 3.4: IEEE 802.11 results(II): (a) the number of data packets transferred and (b) the number of collision slots.
In our enhanced algorithm, because of the contention-free data transmission and variable beacon interval length, the mismatch problem no longer exists. This can be seen by the similarity in the patterns of different results in Figure 3.5 and Figure 3.6. The more stations enter data transmission, the more packets are transferred, while the system throughput remains at a steady level. At the same time, stations can benefit more from entering the doze state in the middle of a beacon interval. In our simulation results, no matter how many active stations are in the system and what the ATIM success rate is, our algorithm can adapt accordingly by changing the beacon interval length from almost 200ms (10,000 slots) to less than 100ms (5,000 slots) and keep a high and steady system throughput. The only concern is that a generally optimal ATIM window size is needed so that there cannot be too many stations in data window causing a very long beacon interval (as in 500-slot ATIM window, when there are 80 active stations, they all enter the data window transferring and create a data window of more than 500ms). Extra long beacon intervals can introduce intolerable delay to stations which have a certain QoS requirement. From the results, the 100-slot ATIM window can produce a reasonable length for the beacon intervals at all loads we have simulated.

Since the transmission time of a single packet is longer than our threshold for switching to the doze state in the middle of a beacon interval, all stations except the one that transmits the last can benefit from the switching. This yields an average energy saving of around 60%, based on the number of slots a station has to be kept in the awake state on average. Due to the similar pattern of the results, in the next section, we will only show the number of successful stations in ATIM window and data packets transferred for the enhanced algorithm.
Figure 3.5: Results of enhanced IEEE 802.11 (I): (a) the number of successful stations in ATIM window and (b) the number of packets transferred
Figure 3.6: Results of enhanced IEEE 802.11 (II): (a) the data window size and (b) the average number of PS slots per PS station
### 3.3.2 IEEE 802.11b and IEEE 802.11a Results

In IEEE 802.11b, as the transmission speed increases, the number of slots needed to transmit a single data packet becomes smaller and smaller. More transmitting stations can be accommodated in a beacon interval. The contention in both the ATIM window and the data window increases. The system also becomes more sensitive to the slots wasted in collision and idleness in data window. In IEEE 802.11a, the most packets transferred is at 80 or 100 active stations with a data window of 9500 slots (Figure 3.8(b)). But in both of those configurations, only about half of the stations entering the data window are able to transmit. In Figure 3.9, we show the number of data packets transferred in IEEE 802.11b and IEEE 802.11a. These graphs can be used to compare to the corresponding results of the enhanced algorithms next.

Our enhanced algorithm have better efficiency and throughput. While the IEEE 802.11a standard can only transfer about 375 packets in a 10000-slot beacon interval, the enhanced algorithm can reach almost 500 packets with a similar length beacon interval (Figure 3.11(b)). As to power saving, although the single packet transmission time of both IEEE 802.11a and 802.11b are below our switch threshold (The threshold of switching to the doze state in the middle of a beacon interval is set at 80 slots, while the transmission times for single packet in IEEE 802.11a and 802.11b are 20 and 44 slots respectively), because we make the station with the most packets to transfer transmit the last, we still can achieve very good results on this extra energy-saving scheme.
Figure 3.7: IEEE 802.11b results: (a) the number of successful stations in ATIM window and (b) the number of successful stations in data window
Figure 3.8: IEEE 802.11a results: (a) the number of successful stations in ATIM window and (b) the number of successful stations in data window
Figure 3.9: The number of packets transferred: (a) IEEE 802.11b and (b) IEEE 802.11a
Figure 3.10: Results of enhanced IEEE 802.11b: (a) the number of successful stations in ATIM window and (b) the number of packets transferred
Figure 3.11: Results of enhanced IEEE 802.11a: (a) the number of successful stations in ATIM window and (b) the number of packets transferred
3.4 Summary

Focusing on the ad hoc mode of the IEEE 802.11 standard, we have proposed an enhanced algorithm for power-saving operation in IBSS. By taking the advantage of a fully connected IBSS and information exchanged in the ATIM, our algorithm creates a variable length beacon interval which can accommodate all the stations that have succeeded in the ATIM transfer, and introduces a contention-free schedule for data packet transfer. Also, the schedule lets short packet sequence transfer first so that stations can benefit from going to PS in the middle of a beacon interval. The simulation results show that this enhanced algorithm improves throughput, reduces delay in the system, and saves energy.
CHAPTER 4

ACHIEVING CLOCK SYNCHRONIZATION FOR POWER MANAGEMENT IN MULTI-HOP AD HOC NETWORKS

One of the common underlying assumptions in IEEE 802.11 standards’ ad hoc mode operations is that all stations in the same IBSS can hear each other directly. In other words, the ad hoc mode of 802.11 supports only single-hop ad hoc networks. When extended to multi-hop environments, IEEE 802.11 protocols show a poor performance, if they are able to work at all. This has been acknowledged by researchers of wireless ad hoc networks [76, 118, 119]. However, because 802.11 already has a world-wide customer base — millions of devices around the world are “802.11-capable,” and that number is quickly increasing — much research has been done on IEEE 802.11-based, MANET-oriented protocols. In this chapter, we try to alleviate the IEEE 802.11 MANET problem by considering two fundamental issues in wireless networks: time synchronization and power management. We ponder the relationship between them, and compare different power management schemes. Eventually, we establish that clock synchronization plays an very important role in efficient power management algorithms, and present an integrated protocol that merges synchronized subnetworks to achieve power-saving in a global synchronized environment. We then
show the correctness of the proposed protocol, and finally present simulation results that demonstrate the merits of the proposed protocol.

The rest of this chapter is organized as follows. Section 4.1 analyzes the efficiency of different schemes under various circumstances, Section 4.2 describes the proposed protocol and shows its correctness, and Section 4.3 presents performance studies and simulation results. Concluding remarks are given in Section 4.4.

4.1 The Problem

As we discussed in Section 2.2.4, power management is an important approach towards energy conservation in MANET, and is adopted by IEEE 802.11. We also pointed out that due to a station’s periodically turning off its transceiver, paging becomes an essential part of the power management protocol. It is important for the stations to wake up at the same time to receive the paging information. For this reason, time synchronization plays an important role in power management in IEEE 802.11-based ad hoc networks\(^7\). However, clock synchronization itself is an issue yet to be solved. For example, TSF in IEEE 802.11 (Section 2.2.1) has two main characteristics:

- There is only one station sending the beacon packet in each beacon interval.
- Clocks are adjusted forward only.

Because of these characteristics, Huang et al. point out the IEEE 802.11 TSF suffers scalability problems [40]. Huang et al. shows that due to inefficient clock adjustment

\(^7\)Time synchronization is also important for networks that employ frequency hopping, where stations need to hop from frequency to frequency simultaneously.
and multiple access collision, the IEEE 802.11 TSF’s time synchronizing ability deteriorates quickly when more stations are in the system. To solve this problem, the Adaptive Timing Synchronization Procedure is proposed. In this procedure, instead of all stations participating the beacon transmission in every beacon interval, each station maintains a different frequency for beacon transmissions. This frequency is related to the relative speed of the station’s clock. The faster a station’s clock, the more frequently it will participate in the beacon transmission at the beginning of beacon intervals. This will reduce the contention of beacon transmission while increasing the convergence speed of clock synchronization. Other synchronization research is underway in multi-hop networks. Since clock synchronization is difficult and unreliable in MANET, some researchers propose power management protocols without using any clock information at all [112, 121]. We have already reviewed these approaches in Section 2.2.4. Hence, we come to the question which is the center of this chapter, How important is clock synchronization, especially when considering power management? To answer this question, we will have to compare the advantages and limitations the two power management schemes on some important issues.

4.1.1 Power Efficiency

A synchronized system gives the most simple and efficient support to power management, and has the best performance on power-saving and the shortest delay, since stations wake up at the same time and only when it is necessary. It also makes the beacon and ATIM window possible, which can separate short but vital beacon and ATIM packets from data packets. When overall traffic is very low, each station only has to stay awake for the beacon and ATIM windows, which usually counts for less
than 20% of each beacon interval. However, synchronous power management does rely on the performance of TSF, which not only is difficult to maintain a good scalability to the size of the multi-hop network, but also has to be able to discover and merge subnetworks, because dynamic topology is one of the main characteristics of MANET.

The schemes based on no clock synchronization give up on maintaining the synchronization in a dynamic distributed system. When transmission is needed, the station has to try to discover the station it wants to talk to using the schemes devised in the papers. To guarantee such a discovery in the random environment of an asynchronous system, the station has to stay awake for several beacon intervals. According to the lower bound proof in [121], stations need \( \sqrt{T} \) fully awake intervals for one overlap in every \( T \) intervals. A larger \( T \) value will bring down the overall percentage of time the station is awake, but it also introduces unacceptable delay. For example, at very low traffic situations, to reach the performance of the synchronous system of a 20% awake ratio (the percentage time stations have to stay awake), \( T \) has to be at least 25. That means that in the worst case, the receiver will not get paging information until the 25th beacon interval after the sender starts the contact initiation. A reasonable delay with \( T = 7 \) used in [121] equals a 43.86% awake ratio. The authors of asynchronous systems are well aware of this problem. In [121], they propose that the station passively tracks the clock and the schedules of the neighboring stations once it gets them, and uses them to guide future packet exchanges. This is very close to a synchronization mechanism, although it does not actively adjust the clock.
4.1.2 Broadcast and Multicast

Broadcasting and multicasting is important for fully distributed systems like wireless ad hoc networks. Most maintenance and control packets have to be sent as broadcast messages, and we also can imagine that in some scenarios for the use of wireless ad hoc networks, such battlefields, the aftershocks of disasters or emergencies, and sensor networks, there might be many data packets that are broadcast or multicast traffic. This presents a big challenge for asynchronous systems.

In a synchronous system as described in the IEEE 802.11 standard, because of the synchronized clock in each station which can make all stations stay awake for the same period of time, and the existence of the ATIM window used to notify all stations of incoming broadcast/multicast packets, all the receivers will be ready for the packets when the sender sends them. Even in the case of multicast, the non-participating stations can be made to yield to the multicast traffic since they were also notified during the ATIM window. In the IEEE standard for Wireless LAN, there is no ACK packet for the broadcast data packet.

However, in an asynchronous system, because the offsets among a group of stations could be any amount from zero to the length of the beacon interval, the broadcast/multicast packets have to be sent in the intersection of all the receiving stations’ data windows. The more receiving stations for a broadcast/multicast session, the shorter the valid sending window for the broadcast/multicast packets will become. To overcome this, the authors of [112] propose that both MTIM packets (their version of ATIM packets) for the broadcast/multicast packets and the real broadcast/multicast
packets are sent to a group of receivers who have some overlap for their ATIM windows. Will this solve the problem? Let us take a look at the broadcast/multicast procedure in the asynchronous system.

For the MTIM packet, because the ATIM window is only a small part of a beacon interval (a typical ATIM window is about 10% of the length of a beacon interval), it is harder to find overlap for multiple ATIM packets. We calculate the expected number of overlapped ATIM windows over the time of one ATIM window with a uniform distribution of beacon interval starting times.

\[
E(overlapATIM) = \sum_{i=1}^{N_r} i \times C_{N_r}^i t_w (1 - t_w)^{N_r-i}
\]  

(4.1)

where \(N_r\) is the number of receivers of the broadcasting packets and \(t_w\) is ratio of the ATIM window to the whole beacon interval. This shows that about one ninth of the broadcast receiving stations can find an intersection within one ATIM window when an ATIM window occupies 10% of a beacon interval. When the ATIM window expands to 20%, this expected overlap stations number increases to one fourth of the receiving stations. There are two more factors we have to consider when estimating the overlap stations number in a real life scenario. The first is the length of overlap. In equation 4.1, any amount of overlap qualifies, but in the real world, as small as an ATIM packet is, it still needs time to transmit. Also, the protocol has to take the collision of ATIM packets into consideration. From the standard, we can tell that sending a management packet like an ATIM could take from 40 \(\mu s\) to 82 \(\mu s\), depending on the data rate used. In a synchronous system with a dedicated ATIM window at the beginning of each beacon interval, the sender’s broadcasting ATIM has to compete with other stations’ ATIM packets, or the protocol can have a special broadcast ATIM which has priority over unicast TIMs, like the Delivery
Traffic Indication Message (DTIM) in IEEE 802.11 for AP in the infrastructure mode. However, in an asynchronous system, the ATIM packets have to contend for the medium with the data and control packets from other stations. Since each ATIM packet represents at least one data packet to be transferred in the beacon interval, the number of ATIM packets should always be less than or equal to the number of data packets. Besides this, the longer data packet also makes it difficult for the contending ATIM packet to gain access to the medium. So even though the ATIM packets have a higher priority than the data packets (ATIM packets use SIFS to decide whether the medium is free or not, see Section 4.1.3), they still have to wait for the current packet to finish transmission.

In [121], the authors did not specifically talk about broadcast/multicast, but in this kind of asynchronous system, broadcast/multicast packets have to be sent more than once to groups of receivers during the overlap of their asynchronous data windows. So the problems we discussed above are still present. The scheme presented in [121] no longer uses the ATIM packet to pre-announce the incoming data packets. Although this eliminates the extra steps of finding the overlap of the receiving stations’ ATIM windows and competing to send the ATIM packets, this brings in new issues. In a system with special ATIM packets, one more round of contention for ATIM packets does allow for some control over the number of sending stations that can participate in the contention in the data window (those who failed the ATIM contention will not contend to send data packets in the following data window). By limiting the length of the ATIM window (so the number of stations passing the ATIM contention is also limited), the protocol can limit the contention in the data window. In [116], the authors study this relationship between the length of the ATIM window and data
window contention. Controlling the contention in the data window is specially desirable for broadcast/multicast transmissions because the broadcast/multicast packets are not acknowledged. Hence, eliminating the ATIM might not always bring desirable results.

### 4.1.3 Beacon and Data Contention

In a synchronous system like IEEE 802.11, beacon and ATIM packets are protected from data traffic by being assigned to transfer in a special period of time when other packets are not allowed to transmit. We cannot implement this kind of protection in an asynchronous system. However, giving control and management packets higher priority over data by using different inter-frame spacings will solve this problem. Even when a beacon packet starts in the middle of an on-going data transmission, the delay of waiting for data packet to finish will become less significant as the transfer rate gets faster in the physical layer. Spreading beacon and ATIM packets throughout the whole beacon interval can reduce the contention of those messages with each other. In an asynchronous system, beacons play a less important role as TSF, so this benefit is reduced. However, this can help us in the scalability problem of TSF.

Other than the issues we discuss above, physical layer operations might also require a synchronous system, which we will not detail here. Thus, a global clock synchronization scheme in a multi-hop environment would be a great benefit, if not essential, for an efficient power management protocol. How to achieve and maintain such a synchronization is the main challenge, and that becomes the problem we try to solve in the next section.
4.2 The Protocol

In a system where a central station exists and all other stations synchronize with it, like cellular networks, it is very easy to achieve such global synchronization, but in a distributed mobile system like MANET, both scalability and mobility bring serious challenges to synchronizing clocks across the whole system. As we mentioned before, the TSF in the IEEE 802.11 standard is based on a single-hop environment, for a system with a dynamic topology in a multi-hop MANET. The dramatic increase in the number of stations that need to be synchronized when going from single-hop to multi-hop necessitates that something be done to the TSF. In [40], the proposed Adaptive Timing Synchronization Procedure has already partly solved the scalability issue for synchronizing a system of a reasonable size. Recently, Lai and Zhou [66] have proposed a clock synchronization protocol for multi-hop networks with a synchronization accuracy of less than 100 microseconds. However, this protocol hinges on two undesired assumptions:

- The MANET is initiated by a single station. Because of the lack of merging groups of unsynchronized stations, the protocol handled one station a time. It results in the perception that the MANET is “grown” from a single point.

- The MANET, as a graph, is always connected. If there exist two disconnected subgraphs, they must be considered as two different MANETs and it is assumed that there is no communication between them.
With all these previous works and unsolved issues, we set up our research on the assumption that a scheme already exists to keep a connected sub-network of a reasonable size\(^8\) in sync within a certain limit, within which the software can ignore the clock difference. This assumption allows our protocol to adopt any advancement in clock synchronization research. These synchronous sub-networks are not aware of each other due to either being out of range, or being “grown” from different nodes\(^9\).

Our goal is to present a protocol that can discover these sub-networks and merge them into a bigger system as they come into communication range with each other (due to mobility), so that a synchronous power management protocol can be used to for energy conservation.

### 4.2.1 Proposed Protocol

Achieving global synchronization as described above is done in three steps. First, stations try to discover unsynchronized stations periodically throughout their lifetime. Second, after unsynchronized stations are discovered in the first step, a merge process will synchronize the stations in two groups, i.e. merging two sub-networks. Third, during other times, the regular clock synchronization scheme (based on our assumption) is used for maintaining synchronization. If, for any reason, like moving out of range, or being turned off for a long period, stations loose synchronization, they can always be re-discovered and re-join the system via steps one and two. Before we present our algorithm, we will visit some design issues.

\(^8\)We will not elaborate on how big a “reasonable size” is. Since with the development of the scalability research on the clock synchronization, it will accommodate a larger and larger number of nodes in a system. Here, we assume the synchronization scheme can handle the synchronization the merged network.

\(^9\)A third possibility is that when the network outgrows the management of the synchronization algorithm, it breaks down into multiple groups. These groups are synchronous inside, but asynchronous among the groups.
• The discovery process cannot be run too often since it consumes much more energy. Its frequency depends on the trade-off between power efficiency and the delay of discovery. We always want some kind of "hint" to improve the efficiency of discovery process, i.e. running it as few times as possible, but still making the discovery in a timely manner. In our protocol, an adaptive approach based on the change of neighbors is used as the trigger of the discovery process. We think the topology change would be a good time to run the discovery process, and use the change of neighbors as an indicator of possible topology change around the station. However, neighbor changes do not necessarily mean new nodes in the neighborhood, and power management makes them difficult to track, a station makes the decision with a probability based on the changes of its neighbors it has noticed (neighbor change threshold $Th_n$), and backs it up with a fixed interval (discovery threshold $Th_d$), during which the discovery must be run at least once. Both of these thresholds should be system parameters and adjusted to vary the discovery delay.

• During the merging process, we let the sub-network with a slower clock adopt the clock value of a sub-network with a faster one. This complies with the standard.

• During the merging step, stations that have just acquired a new clock value will maintain two different beacon windows so that they can populate the new clock value to their neighbors who still use the old clock, by sending their beacons with the new clock in its original window. This beacon will be specially
marked so that stations receiving it can distinguish it from beacons in the regular synchronization process, and can change their clock immediately.

- Since beacon packets are not acknowledged, stations that adopt the new clock value have to decide when all its neighbors have come to the new clock so that it can exit the merging step. To achieve this, these stations stop sending beacons in the original beacon window after $N$ beacon transmissions, and start to listen for the beacon that still carries original clock value. This listening period should last for $\Delta/2r_{d_{\text{min}}}$ (see Section 4.2.2 for definition and details about this time), and the value $N$ is a system parameter which will be fine-tuned in the real world.

Putting these ideas that we have discussed so far together, we get three procedures in our protocol. One is the discovery procedure, which runs repeatedly to search for unsynchronized sub-networks. In this procedure, like in those asynchronous power management schemes, a station stays in awake state for the full beacon interval to listen to the beacons outside its beacon window. If such beacons are received, it means stations with different schedules are within transmission range. Then the clock values carried in those beacons are stored in the clock table, and the station will enter the merging procedure in the next interval. In the merging procedure, the station tracks two schedules at the same time. One is the schedule associated with the fastest clock in the clock table, and the other is its original schedule. During the beacon window of its original schedule, it sends a special beacon packet mentioned above to change its neighbors’ schedule. During the new schedule, it follows the regular synchronization algorithm. While a station is not in the discovery or the merging procedures, it runs the regular procedure for handling regular operations, including
the maintenance of clock synchronization and power management. In this regular procedure, it also checks for neighbor changes to decide whether to run the discovery procedure. The protocol proposed here can adopt any regular procedure, but we use the TSF and ATIM operations in IEEE 802.11 as an example. The pseudo-code of the these procedures are in Figure 4.1, Figure 4.2, and Figure 4.3.

4.2.2 Correctness of the Protocol

In this section, we discuss the correctness of the above protocol. We start with rephrasing the assumption presented at the beginning of Section 4.2.
Merging Procedure

After discovery procedure, there should be more than one clock values in the clock table. Find the one with the fastest clock to adopt, discard the rest except the original one.

Adjust clock to adopt the faster clock.

Reset original beacon counter.

if In the beacon window corresponding to the original clock
then
if Beacon counter < N
then
Send special beacon with new clock value using the same beacon transmission procedure as in Regular Procedure
if Transmission is successful
then
Beacon counter++
if Beacon counter = N
then
Reset and start listen timer
endif
endif
else
endif
else
if Listen timer < $\Delta/2d_{\min}$
then
Listen for regular beacons
if Found regular beacon
then
Reset and stop listen timer
Reset beacon counter
endif
else
Enter Regular Procedure
endif
endif
else
Follow the same actions as in Regular Procedure according to the newly adopted clock
endif
endif

Stay in Merging Procedure for the next interval

Figure 4.2: Merging procedure of the new power-saving algorithm
Regular Procedure
Running existing protocols for clock synchronization, power management and other operations

if Received special beacon
then
Enter Merging Procedure at the end of interval
else
Check for neighbor change at the end of interval
Calculate based on neighbor change
if Detection probability > Thn OR discovery counter = Thd
then
Enter Discovery Procedure in the next beacon interval
else
Discovery counter++
endif
endif

Figure 4.3: Regular procedure of the new power-saving algorithm

Assumption (Partial Synchronization Assumption). There is a distributed clock synchronization algorithm based on beacon broadcasting, which can limit the clock difference between any two stations no more than $\Delta$ in a connected system.

From this assumption, we can see the difference between any pair of stations’ clock values, $|t_A - t_B|$, is bounded by the maximal time between two consecutive successful beacon exchanges, $\Delta t_A$ and $\Delta t_B$ as shown in Figure 4.4, and the clock drift rate, $r_d$.

$$Max\{|t_A - t_B|\} = Max\{\Delta t_A\} \times r_{dA} + Max\{\Delta t_B\} \times r_{dB} \quad (4.2)$$

Because the propagation delay of RF signal can be ignored, other transmission delay, like IFS and transceiver preparation time, has already been taken into consideration in timestamp calculations, and there is no queuing delay for beacon packets (since they are transmitted in a special window in which only beacon packets can be transferred
and each station only transfers one beacon at a time), we can unify \(\Delta t_A\) and \(\Delta t_B\) as \(\Delta t\). So Equation 4.2 can be reduced to

\[
Max\{|t_A - t_B|\} = Max\{\Delta t\} \times (r_{dA} + r_{dB})
\] (4.3)

Assume the station’s clock drift rate, \(r_d\), is greater than 0 (if \(r_d = 0\), the clock is a perfect clock, then synchronization will be achieved without any special protocol. We do not discuss this rare case in this work), and using \(\Delta\) in the Partial Synchronization Assumption to replace \(Max\{|t_A - t_B|\}\), we can get the maximal time between any consecutive successful beacon exchanges

\[
\Delta t_{\text{max}} = \Delta / (r_{dA} + r_{dB})
\] (4.4)

When we consider more than just two stations in the synchronized system, the maximal time between consecutive successful beacon exchanges at any station is bounded by

\[
\Delta t_{\text{max}} \leq \Delta / 2r_{d_{\text{min}}}
\] (4.5)

Thus, we have got the relationship between the maximal time between consecutive successful beacon exchanges, \(\Delta t_{\text{max}}\), the maximal clock difference, \(\Delta\), and the minimal clock drift rate, \(r_{d_{\text{min}}}\).

Now, we will show that a station running the protocol described in Section 4.2.1 will discover stations with different clock values inside its range in finite time. \(A\) and \(B\) denote the two stations with different clock values and inside each other’s range. Without loss of generality, let us assume that \(A\) is trying to discover \(B\). From the Partial Synchronization Assumption above, station \(B\) will be able to successfully send at least one beacon in \(\Delta / 2r_{d_{\text{min}}}\). We also know, from the protocol, that \(A\) will stay
awake for at least a full beacon interval in every $Th_d$ intervals. So $B$’s beacon will be heard by $A$ eventually.

Next, we take a look at the correctness of the merging procedure. Assume two subnetworks have discovered each other. Stations in the sub-network with the faster clock, $A$, do not need to do anything special. Let $o$ denote the station that discovered the faster clock of $A$, and belongs to the sub-network $B$. We try to show that $o$ can populate all stations in $B$ with the faster clock value. For any station, $i$, in the sub-network $B$ that does not know the new clock, there exists at least one path from $o$ to $i$ due to the connectivity of the sub-networks. $o, i_1, i_2, \ldots, i_n, i$ denotes the stations on this path. The stations next to each other in this list are in each other’s transmission range. According to the protocol in Section 4.2.1, starting from $o$, stations adopting
the faster clock will broadcast the special beacon in the original beacon window. The Partial Synchronization Assumption guarantees that any station in a sub-network will get at least one chance to successfully broadcast its own beacon in every $\Delta/2r_{d_{\min}}$. Therefore, in a finite amount of time (no more than $\Delta/2r_{d_{\min}}$), the special beacon from $o$ will be heard by $i_1$. Upon receiving the special beacon, $i_1$ will change its clock and broadcast special beacons to its neighbors including $i_2$. So unless this process stops prematurely, the faster clock value carried in the special beacons will eventually propagate along the path and reach $i$.

Next, we show the above process will not stop prematurely. Stations broadcasting the special beacons like $o$ will not exit the merging procedure until they hear no more regular beacon in the original beacon window during a continuous period of $\Delta/2r_{d_{\min}}$. From the Partial Synchronization Assumption, we know that in such a period, if there is a station that still runs on the original clock in the neighborhood of $o$, $o$ will hear at least one beacon from it. And if this is the case, according to the protocol, $o$ will not exit the merging procedure.

Last, if the synchronization algorithm in the Partial Synchronization Assumption is not a distributed algorithm. For example, it uses a hierarchical structure to select only some stations as leaders to broadcast beacons, then we shall modify the election process so that station $o$ can force itself to become a leader during the merging procedure.

### 4.3 Simulations and Results

To confirm our analysis of the relation between clock synchronization and power management, we compare the results on power-saving of different power management
algorithms in a simulated environment. We implement our protocol as an example of synchronous power management, and also pick the Periodically-Fully-Awake-Interval protocol from [112] as a representative of asynchronous power management protocols. The Periodically-Fully-Awake-Interval protocol requires a station to stay awake for a full beacon interval in every $T$ intervals, where $T$ is an important parameter of the protocol. To further clarify that the trends shown in the results are caused by clock synchronization, we also include the power-saving scheme described in [120], which uses a scheme running between totally asynchronous algorithms and those seeking global synchronization throughout a system like the one that we propose in this chapter, and mark it "local synchronization" in the figures. In our simulation, data traffic is presented as single packet with a fixed length. We describe it using a Bernoulli process with access probability $p_{traffic}$. We do not have any particular requirement for the network topology. Nodes connect with each other randomly, with 20% to 30% of total nodes being neighbors. Node movement is described as another Bernoulli process with moving probability $p_{move}$. However, since we did not implement any routing procedure, all traffic happens between directly connected neighbors, and the node will not move when it has packets to send. Other simulation parameters are shown in Table 4.1.

Before comparing different protocols with each other, we first check the behaviors of each protocol under different configurations. We show the effects of the main parameters, $T$ and the number of nodes in the system, of the Periodical-Fully-Awake-Interval protocol under different traffic loads and node mobility in Figure 4.5. Among the graphs in Figure 4.5, (a) shows the effect of $p_{move}$ and the distance between the Fully-Awake-Interval, $T$, with 100 nodes in the system; (b) shows the effect of node
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beacon interval length</td>
<td>2000 slots</td>
</tr>
<tr>
<td>Beacon window length</td>
<td>64 slots</td>
</tr>
<tr>
<td>ATIM window length</td>
<td>200 slots</td>
</tr>
<tr>
<td>Beacon length</td>
<td>2 slots</td>
</tr>
<tr>
<td>ATIM packet length</td>
<td>2 slots</td>
</tr>
<tr>
<td>Data packet length</td>
<td>40 slots</td>
</tr>
<tr>
<td>Min. contention window size, $CW_{min}$</td>
<td>31 slots</td>
</tr>
<tr>
<td>Max. contention window size, $CW_{max}$</td>
<td>255 slots</td>
</tr>
<tr>
<td>Simulation time</td>
<td>1000 beacon intervals</td>
</tr>
</tbody>
</table>

Table 4.1: Simulation configuration

number, $n$, on the node’s power efficiency with $p_{move} = 0.2$ and $T = 4$; (c) shows the average packet delay of the same configurations. Not surprisingly, we can see that $T$ has a major effect on the awake ratio in (a). The difference among different $T$ values is clear. On the contrary, the effect of node mobility on the PFAI protocol is almost negligible. That is because in this protocol, a node’s wake-sleep schedule is totally unrelated to other nodes. From (b) and (c), we can see that the PFAI’s behavior changes under different traffic loads as the node density of the network changes. When network node density is high (more nodes inside the communication range of a node), the power consumption starts to level off as the traffic load keeps increasing. However, when there are less nodes in the system, the awake ratio (the average percentage of time a node stays awake) increases almost linearly with traffic load. The reason is contention. When node density is high, a heavy traffic load will cause a lot of contentions during ATIM transferring. Those stations that could not send the ATIM successfully will go to sleep until the next beacon interval. That translates to more sleep beacon intervals for a node on average even when a station
has a data packet to send. This can be verified from the delay graph in (c). A dense system, $n=100$, has a much higher delay than a system that is sparse, $n=20$. And the delay keeps increasing as traffic load increases. When $n=20$, the average packet delay does not change much with the traffic load.

In Figures 4.6 and 4.7, we can see the effects of different parameters on the performance of stations with various traffic loads in the power-saving scheme that is based on local clock synchronization. From graph (a) of Figure 4.6, which shows the effects of mobility, $p_{\text{move}}$, and the percentage of nodes which have multiple schedules, $\text{Overlap}$, on the protocol with $n = 100$ nodes, $s = 20$ different schedules in the system, and each node scanning for new neighbors every $T = 10$ beacon intervals, we can see the algorithm is not sensitive to node mobility. This is because in this protocol, multiple schedules are observed. When some neighbors with schedule A are replaced by neighbors with schedule B, as long as the total number of schedules a node follows is not changing, the node does not have much more work. $\text{Overlap}$ has an apparent effect. This is quite straightforward since it controls the percentage of nodes with multiple schedules in the system. These are the nodes that stay awake much longer than those that have only one schedule. Graph (b) shows the effect of the number of different schedules, $s$. Other parameters are: $n = 100$, $T = 10$, $p_{\text{move}} = 0.5$, $\text{Overlap} = 40\%$. In it, we can see that the average number of schedules of those multi-schedule nodes following, $s$, has the most significant effect on the awake ratio. With that number increasing from 2 to 20, when there is no traffic, the average awake ratio of each node climbs from 25\% to over 50\% although less than half of the nodes are multi-schedule nodes. The proposers of this scheme argue that there would be only a few different schedules in the system. However, since no schedule change is
Figure 4.5: Simulation results of Periodical-Fully-Awake-Interval protocol
devised, and considering the complex situation of a mobile ad hoc network, we think that making the protocol able to handle a number of different schedules is important. When there are only a few schedules in the system and traffic load is heavy, due to contention, nodes spend less time awake. But the delay per packet increases in these situations. This can be seen in our delay data (it is not shown here, but is similar to the PFAI behavior when the number of nodes in the system increases). The effect of how often a node has to stay awake for the whole beacon interval to scan for different schedules is shown in graph (a) of Figure 4.7. The graphs in Figure 4.7: (a) show the effect of the distance between scan intervals, $T$, with other parameters: $n = 100$, $s = 20$ if not specified, $p_{\text{move}} = 0.5$ and $Overlap = 15\%$, and (b) show the effect of the number of nodes in the system, $n$, with $s = 4$, $T = 20$, $p_{\text{move}} = 0.2$ and $Overlap = 15\%$. The difference is clear, but not as significant as the effect of $s$. And in Figure 4.7 (b), the number of nodes in the system changes the behavior of the protocol when the traffic is heavy, but it does not have much effect on the awake ratio when the traffic is light. We had the same observations for the PFAI protocol.

Both Figure 4.8 and Figure 4.9 show our protocol’s performance and the effects of the parameters. Graph (a) in Figure 4.8 shows the effects of mobility, $p_{\text{move}}$, and the percentage of nodes that have multiple schedules, $Overlap$, on the protocol with $n = 100$ nodes, $s = 20$ different schedules in the system, and each node scans for new neighbors every $T = 10$ beacon intervals. Graph (b) shows the effect of the number of different schedules, $s$. Other parameters are: $n = 100$, $T = 10$, $p_{\text{move}} = 0.2$, $Overlap = 15\%$. In Figure 4.9, (a) shows the effect of the distance between scan intervals, $T$, with other parameters: $n = 100$, $s = 20$ if not specified, $p_{\text{move}} = 0.2$ and $Overlap = 40\%$; (b) shows the effect of the number of nodes in the system, $n$,
Figure 4.6: Simulation results of power-saving scheme based on local synchronization (part I)
Figure 4.7: Simulation results of power-saving scheme based on local synchronization (part II)
with \( s = 4 \), \( T = 10 \), \( p_{move} = 0.2 \) and \( Overlap = 40\% \). The results in Figure 4.8 (a) indicate that the protocol is more sensitive to movement than \( Overlap \). This is because clock synchronization and the power-saving schedule merging try to unify all nodes' schedules. Those nodes in the overlap area (with multiple schedules) eventually reduce to one schedule only. However, when node mobility is high, new schedules are generated due to node movements. It takes more effort to keep them all synchronized. This schedule merging effect can also be seen clearly in graph (b) of Figure 4.8, which shows that the number of different schedules has a much smaller impact on the results.

For Figure 4.9, in (a), \( T \) becomes one of the main parameters that will define the power performance in this protocol. Although the awake ratio changes a lot from \( T = 4 \) to \( T = 10 \), it stays very close whether the nodes search for new neighbors at least every 10 beacon intervals, or every 20 beacon intervals. This is expected since periodical detection is not the only way our protocol decides when to execute the detection procedure. The protocol also monitors neighbor change and starts the detection procedure based on this change. The simulation results show that after a certain point, the estimation used in the protocol for neighbor discovery is pretty effective. Again in (b), it is shown that the effect of the number of nodes in the system on power-savings is the same as in the other two protocols.

When we put all three protocols together for comparison, we unveiled some interesting results. First, we compare them in a highly mobile system with \( p_{move} = 0.5 \), i.e. around half of the nodes change their position (and clock due to loss of synchronization), in this group of figures. In Figure 4.10, we can see a saturated system (100 nodes) in which the awake ratio goes up and then goes back down, and the
Figure 4.8: Simulation results of our protocol (part I)
Figure 4.9: Simulation results of our protocol (part II)
delay keeps increasing, as traffic load increases. When there are only 20 nodes in Figure 4.11, the system has not reached its saturation point. The awake ratio becomes higher and higher to accommodate the increasing traffic, while delay does not change much. Among the three protocols, the Periodically-Fully-Awake-Interval protocol shows some stability. It responds to most of the parameters slowly, except for the distance between the Fully-Awake-Intervals, $T$. It has the highest awake ratio when the traffic is very light, because of the short period between the Fully-Awake-Intervals. With the increase of traffic load, both awake ratio and packet delay increase, but at a relatively low rate. Although the scheme based on local clock synchronization has a lower awake ratio and delay value in a light traffic situation, $p_{traffic} < 0.2$ in this case, it surpasses PFAI with $T = 4$ when traffic picks up. And the delay increment is even more dramatic. This can be contributed to the high contentions coming from issues with multiple schedules that we pointed out in previous sections. Our protocol shows the best result overall. Because of the clock synchronization and the merged schedule, its performance is close to the underlying CSMA/CA MAC protocol. After the system reaches its saturation point, the power-saving scheme saves energy by preventing data transmissions from increasing, since it can only introduce more contention rather than getting more packets through. And the packet delay does not make a sudden jump, as it does in the scheme based on local synchronization.

In Figures 4.12, 4.13, 4.14, and 4.15, we show the comparison in a low mobility ($p_{move} = 0.2$) and stable environment ($p_{move} = 0$). In a low mobility and stable environment, the protocols perform similarly. With less changes happening in the system, the saturation point is pushed to a heavier traffic load. This leads to a little
Figure 4.10: Comparison of protocols at high mobility with 100 nodes
Figure 4.11: Comparison of protocols at high mobility with 20 nodes
improvement of the scheme based on local clock synchronization when comparing to the PFAI protocol.

One interesting point we want to make here is the packet delay of the PFAI in a dense network, i.e. \( n = 100 \). We notice that only at a very light traffic load, \( p_{traffic} \leq 0.1 \) or even less depending on node mobility, a smaller \( T \) value shows a lower packet delay than a larger \( T \) value. For most of the configurations, the \( T = 4 \) delay curve is below, i.e. delay is shorter than, the \( T = 3 \) curve. And for the awake ratio, \( T = 4 \) is still significantly less than \( T = 3 \). After tracing different configurations of traffic loads and \( T \) values, we found that the longer delay is due to higher contention with a smaller \( T \). A smaller \( T \) value means a packet is more likely being transferred. But when the system is saturated, putting more packets into transmission will not do any good. So from this we can see that the parameter \( T \) is very important in PFAI.

### 4.4 Summary

From studying the relationship between clock synchronization and power-saving schemes, we showed that clock information is very important for the efficiency of power-saving protocols which are based on the IEEE 802.11 power management procedure. Based on this observation, we presented an integrated protocol for both clock synchronization and power-saving for mobile ad hoc networks. In this protocol, we addressed the problems that rise from topology change introduced by node mobility. Our protocol can handle the discovery and merging of new sub-networks with different clock values. From the simulation results, we can see the protocol performs more efficient power-saving than other protocols of its type.
Figure 4.12: Comparison of protocols at low mobility with 100 nodes
Figure 4.13: Comparison of protocols at low mobility with 20 nodes
Figure 4.14: Comparison of protocols at no mobility with 100 nodes
Figure 4.15: Comparison of protocols at no mobility with 50 nodes
CHAPTER 5

COMBINING POWER MANAGEMENT AND POWER CONTROL FOR MULTI-HOP WIRELESS AD HOC NETWORKS

5.1 The Problem

As reviewed in our survey in Section 2.2.4, energy conservation research can be divided into two categories. The power management schemes shut down the transceiver and put the node into sleep mode for some period of time when it appears to be idle, while the power control schemes adjust transmission power dynamically so that only the desired level of power is used by the RF amplifier, and the algorithm tries to find the optimal power level.

It is well-known that the transmission power needed to deliver a packet from node $A$ to $B$, $E_{AB}$, is an exponential function of the distance between two nodes, $dist_{AB}$, i.e.

\[ E_{AB} = \beta \times dist_{AB}^\gamma \]  \hspace{1cm} (5.1)

with $\gamma > 1$ as the path-loss exponent which depends on the RF environment and is generally between 2 and 4 for an indoor situation [93]. This usually results a power control algorithm that picks multiple short hops over a long transmission for better
Specifications | Measurement
--- | ---
Agere WLAN | Cisco 350 | Lucent WaveLAN
Sleep mode | 9mA | 15mA | 10mA
Idle | n/a | n/a | 156mA
Receive | 185mA | 270mA | 190mA
Transmit | 285mA | 450mA | 284mA
Input Voltage | 5V | 5V | 4.74V

Table 5.1: WLAN card power consumption comparison

energy conservation. However, both lab tests [30] and manufacturer’s specifications [1, 23] show that receiving and idle states consume a considerable amount of energy as well (Table 5.1). Thus, it is desirable to combine power control schemes with power management to take the power consumed by replaying stations in receiving and waiting into consideration when choosing optimal transmit power.

A routing algorithm with a goal of energy conservation using power control is given in [34]. For each pair of neighbors, the transmission power required for a successful delivery is tracked based on the distance between them and a two-ray propagation model. This power is then used as the weight of the link to participate in path selection and redirection. Stations overhear ongoing transmissions and calculate whether it is better to redirect some transmission through themselves. If it is better to redirect, this station sends a redirecting message, and in the next slot, the path going through it will replace the old one. However, the authors have not considered any power management procedure. The overhearing and redirecting cause that a path with multiple short hops, in which more intermediate stations are involved in relaying packets, is preferred over a path with long hops.
In [83], the authors study the effects of transmission power/range on both the energy per bit successfully transmitted and the network capacity. The simulations show energy conservation improving as the transmission range is reduced. However, those simulations are done without any power management implemented. The authors also find that reducing transmission power has a reverse effect on network capacity, even though a smaller transmission range means less multiple access collision, which should introduce a capacity gain. It turns out that this capacity reduction is caused by the increased number of hops/transmissions per packet.

After studying these power control schemes, we realize that power management and power control deal with two different aspects of power-saving, and in most of the situations, the optimal conditions which they are seeking conflict with each other. However, if we can combine them, we can take advantage of both schemes to achieve maximum power-saving. So in this chapter, we propose an algorithm that combines and balancing the two different approaches. From the simulation results, we can see the proposed algorithm successfully achieves that goal.

In the next section, we present the proposed algorithm. After it, in Section 5.3, simulation results are presented. Finally, we conclude this chapter with a short summary and some discussions in Section 5.4.

5.2 Our Protocol

In this section, we are going to propose a power-saving scheme which combines both IEEE 802.11 power management and dynamic transmission power control.

In the IEEE 802.11 standard, at the beginning of each slot, all stations must stay awake to exchange control packets announcing pending data packets. Those control
packets work as a hand-shaking mechanism for the senders and receivers. Those sta-
tions which succeed in their hand-shaking process remain awake for data transmission,
while the rest go to the doze state until the next slot. From Table 5.1, we can see that
PS mode saves power much more significantly than reducing transmission power. So
in the proposed algorithm, the priority is given to the power management procedure,
that is, only those stations which stay active according to the power management
procedure will try to further conserve energy using power control. For example, in
the situation shown in Figure 5.1, point O is the source of a data packet, and E is
the destination outside the maximum transmission range of O. Point O, along with
points A, B and C, is active in this slot, and point D is in the doze state. Because D
is in PS according to the power management procedure, it will not enter the power
control procedure for relaying packets from O to E. However, in previous power aware
routing schemes, D will be picked as the next hop node of the transmission. This
example shows the main difference of the proposed scheme from other power control
algorithms.

Also in the example shown in Figure 5.1, the transmission from O to C will be a
direct one-hop transfer. Although the fact that $E_{OA} + E_{AC} < E_{OC}$ makes point A a
good relaying candidate, to avoid delay and congestion inspired by relaying, a direct
link is preferred over multi-hop link when available.

When a direct link is unavailable, i.e. a multi-hop transmission is inevitable, the
source node will try to find the next-hop node according to the local information that
it has at that time. We do not use the overhearing-redirecting scheme presented in
[34] for several reasons:

102
The overhearing-redirecting scheme is based on a system without power management. However, in a system with power management, the active neighbor nodes are different and unpredictable from slot to slot. So the overhearing node might not participate in transmission at all (in PS) in the next slot.

The overhearing-redirecting scheme uses many packet exchanges for updating routing information. As pointed out in [83, 30], these extra control packet exchanges are not good for conserving energy. So in the proposed scheme, it is the sender’s responsibility to pick the next-hop nodes among its neighbors.

It takes multiple rounds to converge in an energy efficient route in [34]. This not only takes time but also requires each node to maintain a routing table. In
an environment where node mobility is high or packet exchanges are short and
sporadic, this convergence time may become too large a price to pay.

Using the example in Figure 5.1 again, for any active neighbor $N$ of sender $O$, the
one that can reach destination $E$ with the most minimal transmission power shall be
selected as the next-hop node from $O$ to $E$, i.e. it will find $N$ which satisfies

$$
\min \{ E_{ON} + E_{NN_1} + \sum_{i=1}^{h-1} E_{N_iN_{i+1}} + E_{NhE} | N \in N^*_A(O) \} 
$$

(5.2)

where $E_{ON}$ is the transmission power needed for transferring a packet from $O$ to
$N$, $N_{1,2,3,\ldots,h}$ represent intermediate nodes in the path from the next-hop node $N$ to
final destination $E$, and $N^*_A(O)$ denotes the active neighbor set of node $O$ in slot $s$.

However, as the active neighbor set of a node changes in each slot, it is impossible to
find a pre-defined route from $O$ to $E$. A route has to be decided hop by hop on the fly.

This has two implications for route selection. First, the route is not strictly optimal
based on energy consumption, since it is impossible to find the shortest path in a
graph that keeps changing. Second, the selection of next-hop $N$ has to be based on
estimation. The power needed for a direct transmission from $N$ to $E$, $E_{NE}$, is used in
place of the transmission energy of a path from $N$ to $E$ ($E_{NN_1} + \sum_{i=1}^{h-1} E_{N_iN_{i+1}} + E_{NhE}$).

This is a tradeoff to reduce the complexity of the algorithm and make it rely only on
local information.

To summarize the points we have talked above, the proposed algorithm is de-
scribed in Figure 5.2.

5.3 Simulations and Results

This section presents the simulation made to demonstrate the proposed algo-

rithm’s efficiency and its results. The simulation shows the algorithm works well and
for each slot
    for each node $O$ which has pending traffic
        Mark current node as active node
        if the destination $D$ is one of the neighbors
            then
                Choose destination as next-hop node, mark it active and send TIM packet
            else
                for each active node $N$ in the neighbor
                    if $d^2_{ON} + d^2_{ND} < d^2_{OD}$
                        then
                            Find the $N$ which gives the smallest $d^2_{ON} + d^2_{ND}$ and pick it as next-hop node, mark active, send TIM packet
                        endif
                    endfor
                endif
            endif
        endif
    endfor
endfor

Figure 5.2: Proposed power-saving algorithm
takes the advantages of both power management and power control. Let us take a
detailed look of the simulation and discuss about the results.

In the simulation, 200 stations with a transmitting range of 30 are evenly dis-
tributed in a square area of 100 by 100. Node movement is not considered in this
paper, as the scheme that handles the mobility relies on the routing scheme rather
than the power-saving functions. Random traffic composed of one packet or a burst
of packets that are enough to be filled in a slot per transmission is used, and all three
algorithms use the same traffic in order to make comparisons more precise. Also, to
highlight the effect of different power-saving schemes and make the analysis of results
easier, the simulation does not include MAC functions like contention avoidance and
error control. It is assumed that all packets stored in each station will be delivered
successfully to the next-hop stations by the end of each slot. For the power man-
agement algorithm, a globally optimized hop-count-based Shortest Path First (SPF)
routing is used. Power control routes with a globally optimized SPF routing, using
the square of the distance of the link as the link weight. The combined algorithm
follows the description in Section 5.2.

Because the power control scheme prefers multiple short hops over a single long
one, it will have more stations involved in transmitting/forwarding than the power
management, and we can see this clearly in Figure 5.3, which counts the number of
nodes involved in traffic for different schemes. The numbers shown in this figure are
the sum of the active station count in each slot throughout our simulation interval.
Since the combined algorithm always tries to route packets to an already active node,
it performs even a little bit better than the power management algorithm, especially
when the traffic is heavy. However, since the power management scheme uses a
Another advantage of the power management scheme over power control is that there are less hops for each packet to get to the destination. Figure 5.4 shows the average hops per packet. We can see that the proposed scheme shows a performance close to that of the power management scheme, which represents the lower bound on the hop count. Less hops means a shorter delay, which could be very important for some applications.

The goal of power control algorithms is to reduce the transmission range. As we can see in Figure 5.5, it has a very low numbers, and the graph of the power management algorithm stays in a straight line since it always uses the maximum power.
Figure 5.4: Comparison of average number of hops needed for a packet to be delivered to its final destination

(Please note that in this figure, the $y$ axis represents the square of transmission distance, since the transmission power is a linear function of the square of that distance. So though maximum power in our simulation is 30, it is depicted in the figure as 900). Since all three algorithms handle the exact same traffic, the average distance of each transmission is closely related to the number of hops it takes to reach the final destination, so the combined algorithm runs between the power management and the power control.

Based on the relationship between transmission power consumed and the corresponding transmission range and WLAN card power consumption specifications given in Section 5.1, we devise this simplified function for the total power consumption of
Figure 5.5: Comparison of the average of the square of transmitting distance of each transmission

\[ P_{\text{card}} = 156 \times N_{\text{sending}} + 0.14222 \times d_{\text{tran}}^2 + 190 \times N_{\text{receiving}} \] (5.3)

Here, we plug in the measured number of Lucent’s WaveLAN in Table 5.1 to get the coefficient \( \beta \). Using this function as a guideline, the power consumed by the LAN card is shown in Figure 5.6 for different schemes based on simulation results. Because the power control can only control 45% of maximum power, power management delivers a better performance by putting as many stations into sleep mode as possible, especially when the traffic is low. However, the higher the traffic level, the more likely the relaying traffic can tag along the transmitting stations’ own transmissions when the power management looses its advantage, although it still bests the power control scheme. For the combined algorithm, just like in Figure 5.3, the heavier the traffic,
the more clear it becomes that it is the best among all three algorithms. Although the graph does not show the situation for different node densities, we can expect that with a higher node density, the combined algorithm will also show a bigger improvement over the power management scheme.

5.4 Summary

In this chapter, a power-saving scheme which combines the advantages from both power management and power control is presented. It integrates the two main approaches for power-saving, and shows a better performance in a more realistic environment. Some researchers propose power-aware routing algorithms that also take advantage of the information about a station’s battery capacity and/or remaining power [111, 80, 62]. In this paper, we do not take this into account. Although we
do not include that information in the proposed algorithm, it can be added easily provided that such information is made available by the system’s physical hardware. Just like power control, this consideration should be taken after power management has been done.
CHAPTER 6

MANAGING CODE ASSIGNMENT FOR H.263 VIDEO IN MULTI-CODE CDMA NETWORKS

In this chapter, we present the second part of our research focusing on cellular networks. As a centralized system, many critical resources like power, clock and medium can be controlled by the BS in the polling mechanism in cellular networks. As we discussed in Chapter 1, multimedia traffic is increasing fast in cellular networks. The high bandwidth, real-time requirements, and other special characteristics of video traffic present many challenges to the cellular networks. Research has been done to meet these challenges. Due to the diligent work of scientists and engineers, today’s networks are much more flexible, and able to handle various applications. Among the innovations, multi-code CDMA is specifically aimed at meeting the requirement of dynamic traffic such as video streams. In this chapter, we show the problems associated with transferring H.263 or H.263+ packet videos over multi-code CDMA wireless networks. Then we describe schemes for video transmission across wireless networks, and compare their results and discuss the effects of different configurations.

In Section 6.1, we discuss the problem that we try to solve in this paper. Then, Section 6.2 presents our approaches. Section 6.3 has simulation results and discussions. And finally, we conclude this paper in Section 6.4.
6.1 The Problem

As reviewed in Section 2.3.2, in order to reduce the size of the data transmitted across the network, H.263 (and most compressed video encoding schemes) uses I-frames and P-frames, and does not encode all the information in every frame. Because the intercoded frames contain only partial information, a transmission error in an I-frame might spread into all the consecutive P-frames until the next I-frame comes. So we can say that I-frames are more important than P-frames. However, the size of an I-frame is much larger than that of a P-frame. Our sample stream has 600 frames encoded with one I-frame per 133 frames interval. The I-frame sizes range from 2752 bytes to 2828 bytes, with an average of 2800.8 bytes. While the sizes of P-frames fall into a range between 22 bytes and 471 bytes, and the average is only 191.4 bytes. The ratio of average size of I-frames to P-frames is as high as 14.63 in our sample stream. But there is no difference on the playback times for an I-frame and a P-frame, so to get a smooth playback at the receiving end, an I-frame has to be transferred in the same amount of time as a P-frame. In the scenario of multi-code CDMA, that means more codes will be used in the system when an I-frame is being transmitted even though the number of users in the system does not change. We know that CDMA system performance degrades as the number of codes being used increases. This raises the problem of that system has a higher BER while transferring important information.

We try to adjust both the video encoder and wireless system itself to make them work harmoniously. However, both the video stream and the CDMA system are highly flexible. H.263, or any kind of digital video coding scheme, has many control
parameters so that you can balance among measurements like video quality, bandwidth requirement, delay, and complexity. The CDMA system allows you to trade the number of users that can be supported in the system for a larger bandwidth and a lower error rate. Due to this complexity, a lot of research has been done to find an adaptive scheme for this problem. Some researchers have presented adaptive coding schemes, like in [39, 72, 95]. They control video encoders according to the wireless channel condition; when there is more bandwidth available, the encoder generates a video stream with better quality. Otherwise, it decreases video quality to reduce the bandwidth requirement. Adaptive resource allocation algorithms are proposed in [14, 50], where wireless channels are tuned according to incoming multimedia traffic. More bandwidth is assigned to the video user when large frames are being transferred. Also, there are proposals to change the levels of error protection for the same user based on the importance of the information in the video stream to combat the high error rate in the wireless communication [71].

6.2 Our Approaches

In this chapter, we put multiple video users into the system shown in Figure 6.1. The resource allocator applies the same code assignment algorithm to each user, and a simulator is used to simulate CDMA using either the MMSE or the MF receiver. Each user sends a video clip over the simulated channels and output is decoded at the receiving end. We compare the decoded frames with the original ones and measure Peak Signal to Noise Ratio (PSNR) for playback quality. We try to show the impact of different transmission schemes on the playback quality in multi-user CDMA wireless networks.
Of course, the basic way to transfer a video stream is to request as many codes from the system as you need to transfer an encoded frame every 33ms, regardless whether it is an I-frame or a P-frame. However, this approach suffers from the problem mentioned in last section, and large I-frames make the traffic flow in the system very uneven.

A natural remedy to this problem is to limit the number of codes being used to transfer those I-frames. A limit is set on the number of codes a user can request at a time. Usually, this limit is reached only when transferring I-frames. In this case, the user has to push back the transmission of the next frame until the current frame is completed. For example, if we call 33ms a slot, the non-interrupted video stream needs a frame to be transferred in each slot. However, if we set the limit on the number of codes a user can request per slot to two. Under this algorithm, six time slots are needed to transfer an I-frame which needs 12 codes to transfer in one
time slot. And all the frames after this I-frame will be pushed off by five slots. This scheme puts an upper-bound on the unevenness of the traffic, and protects I-frames by reducing the number of codes in the system. However, delay is introduced, which might build up to an intolerable level at the end of a long stream.

The third approach can both protect the I-frame from the higher system BER and keep timely playback at the receiving end. In this approach, we still set a limit on the number of codes a user can request in one time slot, like in the second scheme, to protect the I-frames from a high BER. However, since P-frames are rather small and less important than I-frames, we only apply that limitation during I-frame transmissions. When P-frames are being transferred, we go back to the need-based code request, like in the first scheme. Furthermore, to reduce the delay introduced by transferring I-frames with a limited number of codes, we let the system recover the delay by transferring more than one P-frame per slot when needed. Let us use our last example again, where a 12-code I-frame takes six slots to transfer. So a delay of five slots has been introduced into the system. Assuming there are 100 P-frames before the next I-frame, then our approach will recover 5 slots during the transmission of those 100 P-frames, which means 100 P-frames will be transferred in 95 slots. In some slots, more than one P-frame is transferred. By doing this, we can prevent delay from accumulating during playback.

Not only does this approach avoids delay build-up, it can also achieve smooth playback in common situations. If all the I-frames in a video clip are about the same size, and thus take the same number of codes to transfer, as in the case of our sample video, only a certain amount of initial delay is introduced at the playback. Afterwards, a smooth play will be maintained until the end of the video. Again,
let us use our last example once more. If the very first I-frame causes a delay of five slots, ignoring transmission delay, that frame will arrive at the receiving end at the sixth slot. During our catch up procedure, we transfer the next 100 P-frames in the following 95 slots. So at the 101st slot, the receiving end is playing the 96th frame, but it has already received all the frames until the 101st. Between the 102nd slot and the 106th slot, although the 102nd frame, another I-frame, is still being transferred, and no new frame is received, the receiving end can still play the 97th to the 101st frame received by the 101st slot. In the 107th slot, when we have no more buffered frames to play, the 102nd frame will complete transferring in 6 slots, since we assume each I-frame takes about the same amount of bandwidth. So the 102nd frame will be ready to be played back in the 107th slot. This pattern repeats itself continuously. Therefore, we maintained a smooth playback with only an initial delay at the receiving end. We can see an illustration of the three approaches in Figure 6.2.

### 6.3 Simulations and Results

To compare our approaches, we set up a simulation environment to transfer video streams across the wireless CDMA network. In this environment, there are multiple users transferring video streams simultaneously. Each of them is transferring the same video clip but with a random starting time so that the I-frames from different users are randomly distributed. The video clip is 600 frames long at 176 by 144 QCIF resolution. The quantization parameter is 13 for both I-frames and P-frames. We do not apply any of the extended features of H.263/H.263+, like advanced prediction mode, or BP mode. At the receiving end, data from only one user is decoded, and the PSNR is calculated for each frame after decoding. The users’ data is protected.
(a) Straight-forward approach

(b) Second approach

(c) Third approach

Figure 6.2: Frame transmission approaches
by the same (255, 191, 8) BCH (Bose-Chaudhuri-Hocquenghem) code [69]. In the physical layer, a simulation of the CDMA system is created with a spreading gain of 64 and noise variance of 0.1. Both MF and MMSE receivers are simulated. Packet length is set at 2040 bits, which is eight blocks of BCH encoded data.

We simulate three algorithms with different configurations. Each of them is run with a different number of users, from one to 33, in the system. For the second and third algorithm, different limits on the number of codes a user can request in one slot are also run under each configuration. The rage of that limit is from one to four. And finally, we simulate those configurations on both MF and MMSE receivers. Since there is a picture header at the beginning of each H.263 frame which contains some information that tells the decoder which special functions are used, an error that occurs there might cause a big degradation in the result, or even halt the decoding process. In order to reduce the effect of such exceptions in the comparison, we run each configuration five times.

First, let us see the overall comparison of three algorithms at different configurations. In Figure 6.3 to Figure 6.6, we show the performance of the three algorithms mentioned in Section 6.2 under various number of users in the system, and with different limits on the number of codes a user can have at a time (only for the second and third algorithms since it does not apply to the first algorithm). Each point in the graph represents a different configuration. And as we have said earlier, these points are the average of five runs of the same configuration. And we also average the BER of each slot throughout the whole simulation time, and the PSNR of each frame. In Figure 6.3(a), we show the BER of the first two algorithms under the configurations from only one user in the system to 32 more users in the background (plus the one
whose output stream will be decoded at the receiving end) using the MF receiver. It also shows the effect of different limits on the number of codes each user can get, from 1 to 4, in the second algorithm. The result of the first algorithm is shown by the line marked “no limit” on the “limit on # of codes” axis. Figure 6.3(b) shows the PSNR results of the same simulations of graph (a). Figure 6.4 shows the BER and PSNR of the third algorithm under the same set of configurations. Figure 6.5 and Figure 6.6 are the results from MMSE receiver. From these figures, we can divide the PSNR output into three sections when the number of codes being used in the system changes. For example, with the MF receiver, when there are a few codes being used in the system, like the configurations of only one or two video users, each using no more than four codes a time, or up to four users with the limit on the number of codes used set at two, or no more than 16 users with only one code per user, the average PSNRs fall in the range of 13 to 14 dB. This is the region of the PSNR if we just decode the H.263 encoded sample video stream without introducing any error (the difference between the result and the original frames are due to coding losses). After that, if we keep increasing the number of codes being used in the system by either increasing either the number of video users in the system, or the number of codes used by each user, the PSNR drops, and the BER increases sharply. And if we keep increasing users or codes used by each user, the PSNR becomes steady at around the 7dB range, despite the different BERs. But the higher the BER, the more likely an error will occur in the frame header, which might cause the decoding process to stop after just a few frames. And also, the spreading gain of the CDMA system set another limit on the number of codes that can be used in the system — there cannot be more codes used at the same time than the Spreading Gain of the system.
MMSE receiver follows the same trend, only it performs much better than the MF receiver — the PSNR does not start to drop until there are 32 users with algorithm one. So there is only one region where the system is sensitive to parameters like the number of users and the limit of codes each user can use. Before or beyond that region, changing those parameters will not generate much effect on the result.

Figures 6.7, 6.8, and 6.9 show the effect of the number of users in the system in detail. As mentioned before, as the number of codes being used increases, the degradation of the PSNR can be divided into three sections. Here, each graph shows some examples of each section. Figure 6.7 is from one to nine users, while using the first algorithm and the MF receiver. In graph (a), the BER of nine users is the highest. The spikes in the BER correspond to the slots when the I-frames are transferred. Since there is no limit on the number of codes that can be used in this algorithm, the codes being used by a user can jump to 12 to 14 codes during the I-frame slots, while a P-frame uses one or two codes at most. Three-user configuration is plotted in the dashed line. It is much lower than the nine-user configuration, but still has spikes reaching 5%. The error rate in the system with two users are even lower than one with three users. It has the range of [0.001:0.006]. The error rate with only one user is very close to zero. When we look at the PSNR graph of the same configurations, graph (b), we can not see much difference. All the PSNR curves are in the range of 7 to 8 dB. The upper one is the PSNR curve when there is no background users (only one user in the system), and the lower one is the configuration with one background user. The other two curves fall on the lower curve, and the result from the errors in the frame header which made the decoding process stop before reaching the end can be seen here. For example, the curve that shows the configuration of
Figure 6.3: Results of algorithms 1 and 2 with MF receiver

(a) BER results

(b) PSNR results
Figure 6.4: Results of algorithm 3 with MF receiver
Figure 6.5: Results of algorithms 1 and 2 with MMSE receiver
Figure 6.6: Results of algorithm 3 with MMSE receiver
eight background users stops at around 80 frames. Figure 6.8 shows the results of the second algorithm with 0, 4, 8, and 16 background users. In the BER graph, graph (a), the spikes in the BER with 16 background users reach only about the half height of those with 8 background users. We also can see that, due to the restriction that is set at the number of codes each user can use (for the case plotted, that limit is two), the variation of the BER reduces a lot. Graph (b) are the PNSR curves. Again, both one background user’s curve and 4 background users’ curve are very close. They are the upper ones at 13 to 14 dB range. The curve with eight background users is at 7.5 dB at the beginning, then jump to almost 12 dB at about the 400th frame. That shows the situation that an I-frame corrects some errors. The 16 background users’ curve is close to the one with 8 background users. It stops just past 100th frames due to a fatal decoding error. From graph (b) of Figure 6.3, we know that with a limit of one code per user, a system with 16 background users runs pretty well even with the MF receiver. However, if we change to the MMSE receiver, the results improve a lot. Figure 6.9 shows the second algorithm with the limit of one code per user on the MMSE receivers. The graph shows very low and almost invariant BER, even with 32 background users in the system. The PSNR curves are as high as 13 to 14 dB, and that does not change much with the number of background users.

Figure 6.10 and Figure 6.11 show the effect of different limits on the number of codes that can be used by each user in the second and the third algorithm. In these figures, we chose an MF receiver with four background users, and tested limitations of 1, 2 and 4 codes per user. We include the first algorithm (“no limit” curve) as a point of comparison. Figure 6.10 shows the first and second algorithm, and Figure 6.11 shows the third algorithm with the same configurations. Still, let us look at the BER
Figure 6.7: Effects of background users: algorithm 1 with MF receiver
Figure 6.8: Effects of background users: algorithm 2 with MF receiver
Figure 6.9: Effects of background users: algorithm 2 with MMSE receiver
graphs first. Except in the “no limit” curve in Figure 6.10(a), which has spikes as high as 0.06, others are usually below 0.02, with only a few slots increasing to above 0.025 in algorithm three with one code per user configuration (they are covered by the solid line of the first algorithm). The second algorithm’s BER is mainly in the 0.003 to 0.01 range. In graph (a) of Figure 6.10, the second algorithm with four code per user configuration has the highest BER except for that of the first algorithm. It is evident that one code per user configuration makes the least variation and lowest overall BER. However, in graph (b) of the same figure, not only is algorithm three worse than algorithm two in general, it also has the worst performance when the limit is set to one code per user. That is because in algorithm three, we have to catch up the delay caused by the I-frames by transferring multiple P-frames in a slot. The lower the limit goes, the more slots it takes to transfer an I-frame. Since it has to be re-synchronized by the time the next I-frame arrives, there are less slots left for the remaining 132 P-frames. That leads to more slots in which two P-frames are sent. Those slots are where we locate a higher BER, instead of the slots when the I-frames are being transferred in other algorithms or configurations. Even though the lowest limit on the number of code that can be used to transfer I-frames causes the worst BER, it does not mean that we will have the best BER with the highest limit. Actually, in graph (b), configuration with four codes per user has a higher BER than that of two codes per user. This is because a limit of four codes per user will make the I-frame slots trouble-makers again. Since most of the P-frames can be transferred by one code in a slot, even if it has to send two P-frames at a time, two codes are enough. While compared to the four codes in consecutive three or four slots to transfer an I-frame, the system still has less interference while transferring P-frames. So to get the
lowest BER, we want the number of codes requested evenly distributed in all slots. When looking at the PSNR graph in Figure 6.10(b) and Figure 6.11(b), we find that the lowest BER does not necessarily link to the best PSNR performance. We can also see this situation in Figure 6.4(b), although the BER graph in Figure 6.11 is not very clear, due to the small difference between the BERs of different configurations and the average process. This observation justifies our goal of giving more protection to I-frames than P-frames.

Even though we end up with a higher BER overall with the limit of one code per user compared to the configuration of four codes per user, since I-frames are transmitted in the low BER slots, the PSNR of one code per user is better than that of 4 codes per user. Although the configuration of one code per user has a better PSNR performance than four codes in the simulation results, a higher BER of one-code-per-user configuration still leave its mark on other aspect which is not shown in these figures. Three of the five runs of the one-code-per-user configuration ended before the 600th frame, because of fatal header errors. The one in Figure 6.11(b) does not reach the end either. It stops at the 421st frame. On the contrary, even a four-code-per-user configuration has a worse PSNR at a range of 7 to 8 dB (while one-code-per-user is at 13 to 14 dB), and every run finished 600 frames.

There is something else not shown in these figures. From the overall comparison in Figures 6.3 to 6.6, we can see that algorithm two is the best algorithm, but we also said before that it will introduce delay into the playback. In the case of limiting four codes each user, every I-frame needs four slots instead of one to finish transmission. That is a three-slot delay for each I-frame. There are 5 I-frames in our 600-frame testing clip. No delay is introduced by P-frames. So at the end of the clip, the delay
Figure 6.10: Effects of code limit in algorithms 1 and 2
Figure 6.11: Effects of code limit in algorithm 3
accumulated to 15 slots. If the limit is set to one code per slot, which gives us the best PSNR performance, the I-frames will take 14 slots each. And neither can every P-frame be transferred in one slot. Among 600 frames of our testing sequence, there are 231 P-frames that need one more slot to transfer, and 19 that need two more. All these accumulate to as many as 339 slots of delay at the end of the sequence. However, there is no delay introduced in algorithm one. The third algorithm, as we said in the last section, will not introduce any delay during the playback other than the initial delay, since the I-frames are about the same size.

6.4 Summary

In this chapter, we present the problems that arise in transferring video streams across the CDMA wireless network. We point out that due to the flexibility of both the video coding scheme and the CDMA system, there are many trade-offs that need to be balanced in order to achieve optimal performance for multimedia wireless applications. We show three different schemes for video transmission in wireless networks, and compare them in a multi-user multi-code CDMA environment. We discover the effect of different configuration parameters on the PSNR of decoded video streams at the receiving end, and also show that in H.263 video encoding, I-frames have a more important role on the playback quality than P-frames. So it is better to transfer them while the system enjoys a good transmission environment. Due to the similar methodology behind the H.263 Recommendation and other predictive encoding schemes like MPEG, our observations apply to transferring video streams encoded by those schemes, too.
CHAPTER 7

CONCLUSIONS AND FUTURE WORK

We conclude this dissertation with a summary of our contributions and discussions of future research in power and code management for wireless networks.

7.1 Summary

With the extension of wireless networks into every aspect of our lives, both wireless data networks like WLAN and cellular networks will continue to play an important role in global computer networks. In the world of mobile computing, there are many critical resources which put limits on the system. The efficient management of these resources has a deep impact on system performance. Since there are fundamental differences between those two networks, we focus on different resources in our research in WLAN and CDMA networks. We first set out to shed light on the power-saving issues in wireless ad hoc networks. Concentrating on the power management algorithm adopted by the IEEE 802.11 standard for its ad hoc mode, this dissertation presents a thorough study of the power management issues in multi-hop wireless ad hoc networks. In the second part, we study the code assignment in multi-code CDMA networks to facilitate video traffic.
Starting from the power management framework of the standard, we first improve it to adapt to today’s ever-faster transmission rates brought by the advancement of physical layer technologies. Taking advantage of the broadcasting nature of power management ATIM packets, we tag on traffic information so that stations can use it to deduct a contention-free schedule for the rest of the beacon interval. We also change the rules for going to doze mode. After these improvements, power-saving performance increases significantly with very little modification of the standard and almost no newly introduced overhead.

Then, we go beyond the major limitation of IEEE’s power management — single-hop assumption. To solve the complex issue of power management in multi-hop environments, we explore the relationship between power management and clock synchronization. Finally, based on existing synchronization schemes, we present a protocol to reach global synchronization in multi-hop ad hoc networks to facilitate efficient synchronous power management schemes.

Last, we further extend our power management study by incorporating another major energy consumption approach, power control. Proposing a scheme that combines both power management and power control, we take advantage of both approaches and maximize the power-saving performance.

In cellular networks, we work on the code assignment for multi-code CDMA networks. Multimedia traffic has increased significantly in the last couple of years in the world of cellular networks. We take a detailed look at the code assignments for H.263 video streams in multi-code CDMA networks. We study the effect on reducing jitter and the playback quality of different assignment algorithms, especially when there are multiple video streams present in the system.
7.2 Future Work

The popularity of wireless networks stimulate new applications and new services. Some of them bring along different requirements on many network operations, and some are built on quite different frameworks. The advance of technologies also renders old protocols invalid or requires them to be re-examined. We have only covered energy conservation in wireless ad hoc networks and code management in CDMA networks. There are still other problems, not to say other emerging areas, of wireless network research. Here, we touch on other related topics we could not cover in this dissertation. Hopefully, these could serve as inspirations to future research.

Power-Saving and Clock Synchronization Realizing the importance of clock synchronization in the multi-hop ad hoc network operations and the difficulty of the problem, we would like to continue working on it. Scalability and multi-hop remain the main issues of clock synchronization. Designing a simple and efficient algorithm with less overhead can almost never end.

Power-Saving in Sensor Networks Sensor network has so many special characteristics that it always stands out from general wireless ad hoc networks. The nodes in sensor networks are much simpler in both computing power and functions performed. It also calls for a higher degree of autonomy in all the operations, and the lifetime of the network could be much longer. Data traffic could be very scarce, compared to regular ad hoc networks composed of laptop computers, PDAs and other portable computing devices. Solving power-saving problems in sensor networks will definitely require a completely different mindset. Since the power issue is as important in sensor networks as in any other
wireless networks, power-saving in sensor networks will become another important aspect of energy conservation research. Sporadic traffic patterns will make power management more suitable for sensor networks. However, simplicity and synchronization will become more important in sensor network power management.

Power-Saving with Real-time Traffic Following the footsteps of wireline networks, real-time multimedia traffic is coming into wireless networks and will soon take larger and larger percentages of total traffic. One of the tradeoffs made by power management is to gain power-saving by increasing packet delay. Real-time traffic will dramatically change this picture. Power management has to be studied closely with delay constraints. Brand new ways of achieving power-saving might be needed in such an environment.

Integration of Cellular Networks and WLAN Cellular networks and WLAN both have advantages and shortcomings. The convergence on digital applications makes those two networks get closer and closer. A seamless integration and inter-operability between cellular networks and WLAN “hot spots” will be welcomed by the consumers. This will enable end users to roam in an environment composed of several different wireless access networks. Those networks might be overlapped at some areas. Users would be connected to the network that best fits their applications. This requires a smart handoff mechanism between networks using different access technologies. It involves many aspects of networking operations, including protocol translation, service discovery and negotiation, packets buffering, rearranging and rerouting, etc. There will be many detailed
protocol designs to make them happen efficiently and transparently. Different kinds of data need different treatments according to their service requirements and levels of sensitivity to the change of physical channel specifications, such as transmission speed and error rate. We can learn from handoff protocol designs in the cellular networks and zone routing protocols in ad hoc networks.

**The Next Generation (4G) Cellular Networks** 3G cellular networks are being deployed around the world. Research and development of the next generation (4G) cellular networks are already well underway, although there are still many questions to be answered [65, 85]. 4G networks could be aeronautical based to provide larger cover area for customers traveling at a faster speed [51], or could be service oriented [4], or could adopt newer accessing technologies like Ultra-Wideband Radio (UWB) [91]. But we can be sure that new technologies will bring not only better services and applications, but new issues and challenges as well.
### APPENDIX A

#### ACRONYMS AND ABBREVIATIONS

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2GFSK</td>
<td>Two-level Gaussian Frequency Shift Key</td>
</tr>
<tr>
<td>4GFSK</td>
<td>Four-level Gaussian Frequency Shift Key</td>
</tr>
<tr>
<td>ACK</td>
<td>Acknowledgement</td>
</tr>
<tr>
<td>ACL</td>
<td>Asynchronous Connection-less</td>
</tr>
<tr>
<td>AM</td>
<td>Active Mode</td>
</tr>
<tr>
<td>AMPS</td>
<td>Advance Mobile Phone Service</td>
</tr>
<tr>
<td>AP</td>
<td>Access Point</td>
</tr>
<tr>
<td>ATIM</td>
<td>Ad hoc Traffic Indication Message</td>
</tr>
<tr>
<td>BCH</td>
<td>Bose-Chaudhuri-Hocquenghem</td>
</tr>
<tr>
<td>BD_ADDR</td>
<td>Bluetooth Device Address</td>
</tr>
<tr>
<td>BER</td>
<td>Bit Error Rate</td>
</tr>
<tr>
<td>Bluetooth SIG</td>
<td>Bluetooth Special Interest Group</td>
</tr>
</tbody>
</table>

140
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>BRAN</td>
<td>Broadband Radio Access Network</td>
</tr>
<tr>
<td>BS</td>
<td>Base Station</td>
</tr>
<tr>
<td>BSS</td>
<td>Basic Service Set</td>
</tr>
<tr>
<td>CCK</td>
<td>Complementary Code Keying</td>
</tr>
<tr>
<td>CDMA</td>
<td>Code Division Multiple Access</td>
</tr>
<tr>
<td>CIF</td>
<td>Common Intermediate Format</td>
</tr>
<tr>
<td>CRC</td>
<td>Cyclic Redundancy Code</td>
</tr>
<tr>
<td>CSMA/CA</td>
<td>Carrier Sense Multiple Access with Collision Avoidance</td>
</tr>
<tr>
<td>CTS</td>
<td>Clear To Send</td>
</tr>
<tr>
<td>DBPSK</td>
<td>Differential Binary Phase Shift Keying</td>
</tr>
<tr>
<td>DCF</td>
<td>Distributed Coordination Function</td>
</tr>
<tr>
<td>DCT</td>
<td>Discrete Cosine Transform</td>
</tr>
<tr>
<td>DFWMAC</td>
<td>Distributed Foundation Wireless Medium Access Control</td>
</tr>
<tr>
<td>DHCP</td>
<td>Dynamic Host Configuration Protocol</td>
</tr>
<tr>
<td>DIFS</td>
<td>Distributed Coordination Function Inter-Frame Spacing</td>
</tr>
<tr>
<td>DNS</td>
<td>Domain Name Service</td>
</tr>
<tr>
<td>DPCM</td>
<td>Differential Pulse Code Modulation</td>
</tr>
<tr>
<td>DQPSK</td>
<td>Differential Quadrature Phase Shift Keying</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>DSDV</td>
<td>Destination-Sequenced Distance-Vector Routing Protocol</td>
</tr>
<tr>
<td>DSR</td>
<td>Dynamic Source Routing Protocol</td>
</tr>
<tr>
<td>DSSS</td>
<td>Direct Sequence Spread Spectrum</td>
</tr>
<tr>
<td>DTIM</td>
<td>Delivery Traffic Indication Message</td>
</tr>
<tr>
<td>ENIAC</td>
<td>Electronic Numerical Integrator and Computer</td>
</tr>
<tr>
<td>EPROM</td>
<td>Erasable and Programmable Read-Only Memory</td>
</tr>
<tr>
<td>ERP</td>
<td>Extended Rate Physical Layer</td>
</tr>
<tr>
<td>ESS</td>
<td>Extended Service Set</td>
</tr>
<tr>
<td>ETSI</td>
<td>European Telecommunications Standards Institute</td>
</tr>
<tr>
<td>FCS</td>
<td>Frame Check Sequence</td>
</tr>
<tr>
<td>FDMA</td>
<td>Frequency Division Multiple Access</td>
</tr>
<tr>
<td>FEC</td>
<td>Forward Error Correction</td>
</tr>
<tr>
<td>FH</td>
<td>Frequency Hopping</td>
</tr>
<tr>
<td>FHSS</td>
<td>Frequency Hopping Spread Spectrum</td>
</tr>
<tr>
<td>GSM</td>
<td>Global System for Mobile Communications</td>
</tr>
<tr>
<td>HYPERLAN</td>
<td>High Performance Radio Local Area Network</td>
</tr>
<tr>
<td>IBSS</td>
<td>Independent Basic Service Set</td>
</tr>
<tr>
<td>ID</td>
<td>Identification</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>------------------------------------------------</td>
</tr>
<tr>
<td>IEEE</td>
<td>Institute of Electrical and Electronics Engineers</td>
</tr>
<tr>
<td>IFS</td>
<td>Inter-Frame Spacing</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>IR</td>
<td>Infrared</td>
</tr>
<tr>
<td>ISBN</td>
<td>Integrated Service Business Network</td>
</tr>
<tr>
<td>ISM</td>
<td>Instrument, Scientific, and Medical</td>
</tr>
<tr>
<td>ITU</td>
<td>International Telecommunications Union</td>
</tr>
<tr>
<td>L2CAP</td>
<td>Logical Link Control and Adaptation Protocol</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>MAC</td>
<td>Multiple Access Control</td>
</tr>
<tr>
<td>MACA</td>
<td>Multiple Access with Collision Avoidance</td>
</tr>
<tr>
<td>MAI</td>
<td>Multiple Access Interference</td>
</tr>
<tr>
<td>MAN</td>
<td>Metropolitan Area Network</td>
</tr>
<tr>
<td>MANET</td>
<td>Mobile Multi-hop Ad Hoc Network</td>
</tr>
<tr>
<td>MEMS</td>
<td>Micro-Electro-Mechanical System</td>
</tr>
<tr>
<td>MF</td>
<td>Matched Filter</td>
</tr>
<tr>
<td>MMSE</td>
<td>Minimum Mean-Squared Error</td>
</tr>
<tr>
<td>MPEG</td>
<td>Moving Picture Experts Group</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>MSDU</td>
<td>Multiple Access Control Service Data Unit</td>
</tr>
<tr>
<td>MT</td>
<td>Mobile Terminal</td>
</tr>
<tr>
<td>NAT</td>
<td>Network Address Translation</td>
</tr>
<tr>
<td>NAV</td>
<td>Network Allocation Vector</td>
</tr>
<tr>
<td>OFDM</td>
<td>Orthogonal Frequency Division Multiplexing</td>
</tr>
<tr>
<td>PAN</td>
<td>Personal Area Network</td>
</tr>
<tr>
<td>PBCC</td>
<td>Packet Binary Convolutional Coding</td>
</tr>
<tr>
<td>PCF</td>
<td>Point Coordination Function</td>
</tr>
<tr>
<td>PDA</td>
<td>Personal Digital Assistant</td>
</tr>
<tr>
<td>PIFS</td>
<td>Point Coordination Function Inter-Frame Spacing</td>
</tr>
<tr>
<td>PLCP</td>
<td>Physical Layer Convergence Protocol</td>
</tr>
<tr>
<td>PLME</td>
<td>Physical Layer Management Entity</td>
</tr>
<tr>
<td>PPM</td>
<td>Pulse Position Modulation</td>
</tr>
<tr>
<td>PRNET</td>
<td>Packet Radio Network</td>
</tr>
<tr>
<td>PS</td>
<td>Power-Saving Mode</td>
</tr>
<tr>
<td>PSNR</td>
<td>Peak Signal to Noise Ratio</td>
</tr>
<tr>
<td>PSTN</td>
<td>Public Switched Telephone Network</td>
</tr>
<tr>
<td>QCIF</td>
<td>Quarter Common Intermediate Format</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Full Form</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
</tr>
<tr>
<td><strong>QoS</strong></td>
<td>Quality of Service</td>
</tr>
<tr>
<td><strong>RF</strong></td>
<td>Radio Frequency</td>
</tr>
<tr>
<td><strong>RFID</strong></td>
<td>Radio Frequency Identification</td>
</tr>
<tr>
<td><strong>RTS</strong></td>
<td>Ready To Send</td>
</tr>
<tr>
<td><strong>SCO</strong></td>
<td>Synchronous Connection Oriented</td>
</tr>
<tr>
<td><strong>SIFS</strong></td>
<td>Short Inter-Frame Spacing</td>
</tr>
<tr>
<td><strong>SoC</strong></td>
<td>System-on-the-Chip</td>
</tr>
<tr>
<td><strong>SPF</strong></td>
<td>Shortest Path First</td>
</tr>
<tr>
<td><strong>SRI</strong></td>
<td>Stanford Research Institute</td>
</tr>
<tr>
<td><strong>SURAN</strong></td>
<td>Survivable Adaptive Radio Network</td>
</tr>
<tr>
<td><strong>TDM</strong></td>
<td>Time Division Multiplex</td>
</tr>
<tr>
<td><strong>TDMA</strong></td>
<td>Time Division Multiple Access</td>
</tr>
<tr>
<td><strong>TDMA/TDD</strong></td>
<td>Time Division Multiple Access/Time Division Duplex</td>
</tr>
<tr>
<td><strong>TSF</strong></td>
<td>Time Synchronization Function</td>
</tr>
<tr>
<td><strong>UCLA</strong></td>
<td>University of California at Los Angeles</td>
</tr>
<tr>
<td><strong>UCSB</strong></td>
<td>University of California at Santa Barbara</td>
</tr>
<tr>
<td><strong>UWB</strong></td>
<td>Ultra-Wideband Radio</td>
</tr>
<tr>
<td><strong>VoIP</strong></td>
<td>Voice over Internet Protocol</td>
</tr>
<tr>
<td><strong>W-CDMA</strong></td>
<td>Wideband Code Division Multiple Access</td>
</tr>
<tr>
<td><strong>WLAN</strong></td>
<td>Wireless Local Area Network</td>
</tr>
</tbody>
</table>


