ABSTRACT

JANG, BEACKCHEOL. Wireless MAC Protocol Design and Analysis. (Under the direction of Professor Mihail L. Sichitiu).

Wireless networks are becoming very common due to their advantages such as rapid deployment and support for mobility. In this dissertation, we design and analyze the Medium Access Control (MAC) protocol for two popular wireless networks: Wireless Sensor Networks (WSNs) and Wireless Local Area Networks (WLANs). For WSNs, we design and analyze an energy efficient MAC protocols. Energy efficiency is a key design factor of a MAC protocol for WSNs. Existing preamble-sampling based MAC protocols have large overheads due to their preambles and are inefficient at large wakeup intervals. Synchronous scheduling MAC protocols minimize the preamble by combining preamble sampling and scheduling techniques; however, they do not prevent energy loss due to overhearing. In this dissertation, we present an energy efficient MAC protocol for WSNs, called AS-MAC, that avoids overhearing and reduces contention and delay by asynchronously scheduling the wakeup time of neighboring nodes. We also provide a multi-hop energy consumption model for AS-MAC. To validate our design and analysis, we implement the proposed scheme on the MICAz and TELOSb platforms. Experimental results show that AS-MAC considerably reduces energy consumption, packet loss and delay when compared with other energy efficient MAC protocols. For WLANs, we present a saturation throughput model for IEEE 802.11, the standard of WLAN, for a simple infrastructure scenario with hidden stations. Despite the importance of the hidden terminal problem, there have been a relatively small number of studies that consider the effect of hidden terminals on IEEE 802.11 throughput. Moreover, existing models are not accurate for scenarios with the short-term unfairness. In this dissertation, we present a new analytical saturation throughput model for IEEE 802.11 for a simple but typical infrastructure scenario with small number of hidden stations. Simulation results are used to validate the model and show that our model is extremely accurate. Lastly, we provide a saturation throughput model for IEEE 802.11 for the general infrastructure scenario with hidden stations. Simulation results show that this generalized model is reasonably accurate.
Wireless MAC Protocol Design and Analysis

by

Beakcheol Jang

A dissertation submitted to the Graduate Faculty of North Carolina State University in partial fulfillment of the requirements for the Degree of Doctor of Philosophy

Computer Science

Raleigh, North Carolina

2009

APPROVED BY:

Dr. Rudra Dutta
Dr. Khaled Harfoush

Dr. Mihail L. Sichitiu
Chair of Advisory Committee

Dr. David Thuente
DEDICATION

For my parents, my parents-in-law, my dear wife Wooae Ki, and my daughter Sarah Alyce Jang.
BIOGRAPHY

Beakcheol Jang received the Bachelor of Science degree in Department of Computer Science from Yonsei University in 2001 and the Master of Science degree in Computer Science from the Korea Advanced Institute of Science and Technology in 2002. In 2005, he started his work as a doctoral student under the guidance of Dr. Sichitiu. Beakcheol Jang' research interest includes wireless networking including ad-hoc and wireless sensor networks and wireless local area networks.
ACKNOWLEDGMENTS

This dissertation would not have been possible without the support of many people. First of all, I would like to thank my advisor Dr. Mihail L. Sichitiu for his thoughtful directions and affectionate encouragements. I would like to acknowledge my advisory committee members Dr. David Thuente, Dr. Khaled Harfoush and Dr. Rudra Dutta. And finally, I would like to thank my parents, wife, daughter, and numerous friends who endured this long process with me, always offering support and love.
# TABLE OF CONTENTS

## LIST OF TABLES

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>vii</td>
</tr>
</tbody>
</table>

## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>viii</td>
</tr>
</tbody>
</table>

## 1 Introduction

1.1 Wireless Sensor Network ................................. 1
1.2 Wireless Local Area Network ............................. 2
1.3 Contributions ............................................. 3


2.1 Introduction ................................................ 5
2.2 Related Work ................................................ 6
2.3 AS-MAC Design ............................................. 9
   2.3.1 Initialization Phase ................................... 9
   2.3.2 Periodic Listening and Sleep Phase .................... 10
2.4 Energy Consumption Analysis .............................. 14
   2.4.1 Energy Model .......................................... 14
   2.4.2 Numerical Results ..................................... 18
2.5 Experimental Evaluation .................................. 20
   2.5.1 MICAz Platform ....................................... 20
   2.5.2 TELOSB Platform ...................................... 29

## 3 IEEE 802.11 Saturation Throughput Analysis in the Presence of Hidden Terminals for Simple Infrastructure Scenario

3.1 Introduction ................................................ 35
3.2 Background .................................................. 38
   3.2.1 CSMA/CA ............................................. 38
   3.2.2 RTS/CTS ............................................. 40
   3.2.3 The Hidden Terminal Problem .......................... 41
   3.2.4 The Short-term Unfairness Problem .................... 42
3.3 Problem Definition and Related Work ...................... 43
   3.3.1 Problem Definition ................................... 43
   3.3.2 Related Work ......................................... 46
3.4 Analytical Model .......................................... 48
   3.4.1 Simple Model for the IEEE 802.11 Network with Two Hidden Terminals 49
   3.4.2 An Enhanced Model for the IEEE 802.11 Network with Two Hidden Terminals 55
   3.4.3 IEEE 802.11a Physical Layer Model .................... 60
3.5 Model Validation .......................................................... 62  
  3.5.1 Simulation Setup .................................................. 62  
  3.5.2 Simulation Results and Analysis .............................. 66  

4 IEEE 802.11 Saturation Throughput Analysis in the Presence of Hidden 
Terminals for the General Infrastructure Scenario ..................... 74  
  4.1 Analytical Model ...................................................... 75  
  4.1.1 The Hidden Terminal Scenario with Two Groups of Stations in Region 
A and Region B .......................................................... 75  
  4.1.2 General Scenario .................................................. 77  
  4.2 Model Validation ...................................................... 77  
  4.2.1 Simulation Setup .................................................. 77  
  4.2.2 Simulation Results and Analysis .............................. 78  

5 Conclusion ........................................................................ 87  

Bibliography ........................................................................ 89
LIST OF TABLES

Table 2.1 Notations and default values.................................................. 15
Table 2.2 RAM and ROM sizes on the MICAz platform............................... 29
Table 2.3 RAM and ROM sizes on the TELOS-B platform.............................. 33
Table 3.1 Fixed parameters of IEEE 802.11a........................................... 62
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Example of wireless sensor network</td>
<td>2</td>
</tr>
<tr>
<td>1.2</td>
<td>Example of wireless LAN configuration</td>
<td>3</td>
</tr>
<tr>
<td>2.1</td>
<td>Initialization phase showing how the new node (C) is finding its offset</td>
<td>9</td>
</tr>
<tr>
<td>2.2</td>
<td>Finite state machine for the periodic listening and sleep phase</td>
<td>11</td>
</tr>
<tr>
<td>2.3</td>
<td>Communication at Hello time</td>
<td>13</td>
</tr>
<tr>
<td>2.4</td>
<td>Communication at wakeup time</td>
<td>13</td>
</tr>
<tr>
<td>2.5</td>
<td>A 10 by 10 grid WSN topology for the MICAz platform</td>
<td>19</td>
</tr>
<tr>
<td>2.6</td>
<td>Energy consumption of Node I for the MICAz platform</td>
<td>20</td>
</tr>
<tr>
<td>2.7</td>
<td>Sources of energy consumption for Node I for the MICAz platform</td>
<td>21</td>
</tr>
<tr>
<td>2.8</td>
<td>Energy consumption of Node II for the MICAz platform</td>
<td>22</td>
</tr>
<tr>
<td>2.9</td>
<td>Sources of energy consumption for Node II for the MICAz platform</td>
<td>23</td>
</tr>
<tr>
<td>2.10</td>
<td>Energy consumption as a function of the number of senders on the MICAz platform</td>
<td>24</td>
</tr>
<tr>
<td>2.11</td>
<td>Sources of energy of senders in the star topology with two senders on the MICAz platform</td>
<td>25</td>
</tr>
<tr>
<td>2.12</td>
<td>Sources of energy of senders in the star topology with five senders on the MICAz platform</td>
<td>26</td>
</tr>
<tr>
<td>2.13</td>
<td>Packet loss at the sink for the multihop chain topology on the MICAz platform</td>
<td>27</td>
</tr>
<tr>
<td>2.14</td>
<td>Per-hop delay as a function of the wakeup interval on the MICAz platform</td>
<td>28</td>
</tr>
<tr>
<td>2.15</td>
<td>Energy consumption as a function of the wakeup interval on the TELOS B platform</td>
<td>30</td>
</tr>
<tr>
<td>2.16</td>
<td>Energy consumption as a function of the data generation interval on the TELOS B platform</td>
<td>31</td>
</tr>
</tbody>
</table>
Figure 2.17 Throughput of AS-MAC and CC2420 CSMA/CA as a function of the data generation rate on the TELOS B platform. ........................................ 32

Figure 2.18 Per-hop delay of AS-MAC and CC2420 CSMA/CA as a function of the wakeup interval on the TELOS B platform. ........................................ 33

Figure 3.1 Undesired effects of CSMA/CA in the presence of hidden terminals; decreasing throughput and (b) deteriorating short-term fairness. ................. 37

Figure 3.2 CSMA/CA exchange when the access point and station A and B are in communication range of each other. ........................................... 39

Figure 3.3 RTS/CTS exchange when the access point and station A and B are in communication range of each other. ........................................... 40

Figure 3.4 (a) A scenario with an access point and two clients hidden from each other and (b) an example of a collision in CSMA/CA. ............................ 41

Figure 3.5 An example packet trace showing the short-term unfairness problem for the scenario depicted in Fig. 3.4 (a) ........................................... 42

Figure 3.6 (a) A scenario with an access point and a client hidden from the other client; (b) Normalized saturation throughput for scenario (a); (c) A scenario with an access point and five clients hidden from the other five clients; (d) Normalized saturation throughput for scenario (c). ........................................... 45

Figure 3.7 Collision probability as a function of retransmission stage for the scenario in Fig. 4 (a) and Fig. 4 (c) when the frame size is 250 bytes. ................. 46

Figure 3.8 Basic infrastructure scenario with two hidden terminals. ................. 48

Figure 3.9 Markov chain model for the system state $b_{ij}$ corresponding to station $A$ being in retransmission stage $i$ and station $B$ being in retransmission stage $j$. .... 51

Figure 3.10 Time for the transition (successful transmission or collision) ............. 54

Figure 3.11 Example of a successful transmission occurring in the previous transition. 56

Figure 3.12 Example with a collision occurring in the previous transition. .......... 58

Figure 3.13 Normalized saturation throughput for CSMA/CA as a function of the frame size at the payload data rate of 54Mbps for two exposed station scenario. .... 63

Figure 3.14 Normalized saturation throughput for CSMA/CA as a function of the frame size at the payload data rate of 54Mbps for ten exposed station scenario. .... 64
Figure 3.15 Normalized saturation throughput for RTS/CTS as a function of the frame size at the payload data rate of 54Mbps for two exposed station scenario........ 64

Figure 3.16 Normalized saturation throughput for RTS/CTS as a function of the frame size at the payload data rate of 54Mbps for ten exposed station scenario. ........ 65

Figure 3.17 Normalized saturation throughput for CSMA/CA as a function of the frame size at the payload data rate of 6Mbps.............................. 66

Figure 3.18 Normalized saturation throughput for CSMA/CA as a function of the frame size at the payload data rate of 54Mbps............................. 67

Figure 3.19 Normalized saturation throughput for RTS/CTS as a function of the frame size at the payload data rate of 6Mbps.............................. 67

Figure 3.20 Normalized saturation throughput for RTS/CTS as a function of the frame size at the payload data rate of 54Mbps............................. 68

Figure 3.21 Normalized saturation throughput for CSMA/CA as a function of the payload data rate at the frame size of 250 bytes............................. 69

Figure 3.22 Normalized saturation throughput for CSMA/CA as a function of the payload data rate at the frame size of 1500 bytes....................... 70

Figure 3.23 Normalized saturation throughput for RTS/CTS as a function of the payload data rate at the frame size of 250 bytes............................. 70

Figure 3.24 Normalized saturation throughput for RTS/CTS as a function of the payload data rate at the frame size of 1500 bytes....................... 71

Figure 3.25 Normalized saturation throughput for CSMA/CA as a function of the retransmission stage limit at the frame size of 500 bytes and the payload data rate of 48Mbps.............................................. 72

Figure 3.26 Normalized saturation throughput for CSMA/CA as a function of the retransmission stage limit at the frame size of 1500 bytes and the payload data rate of 48Mbps.............................................. 72

Figure 3.27 Normalized saturation throughput for RTS/CTS as a function of the retransmission stage limit at the frame size of 500 bytes and the payload data rate of 48Mbps.............................................. 73

Figure 3.28 Normalized saturation throughput for RTS/CTS as a function of the retransmission stage limit at the frame size of 1500 bytes and the payload data rate of 48Mbps.............................................. 73
Figure 4.1 Example of $n_a$-to-$n_b$ infrastructure scenario with hidden terminals, where there are six exposed stations in the region $A$ and four exposed stations in the region $B$. Stations in different regions are hidden from each other. 75

Figure 4.2 Example of general infrastructure scenario, where there are 5 stations in the communication range of the access point, and the stations may irregularly be exposed or hidden from each other. 76

Figure 4.3 Normalized saturation throughput for $n_a=1$ as a function of $n_b$ for CSMA/CA, the number of stations in region $B$, at the payload data rate of 54Mbps and the frame size of 1500 bytes. 78

Figure 4.4 Normalized saturation throughput as a function of $n_a$ and $n_b$ for CSMA/CA, the number of stations in both regions, at the payload data rate of 54Mbps and the frame size of 1500 bytes. 79

Figure 4.5 Normalized saturation throughput for $n_a=1$ as a function of $n_b$ for RTS/CTS, the number of stations in region $B$, at the payload data rate of 54Mbps and the frame size of 1500 bytes. 79

Figure 4.6 Normalized saturation throughput as a function of $n_a$ and $n_b$ for RTS/CTS, the number of stations in both regions, at the payload data rate of 54Mbps and the frame size of 1500 bytes. 80

Figure 4.7 Normalized saturation throughput as a function of frame size for CSMA/CA for 4-to-1 scenario. 81

Figure 4.8 Normalized saturation throughput as a function of frame size for CSMA/CA for 3-to-2 scenario. 81

Figure 4.9 Normalized saturation throughput as a function of frame size for RTS/CTS for 4-to-1 scenario. 82

Figure 4.10 Normalized saturation throughput as a function of frame size for RTS/CTS for 3-to-2 scenario. 82

Figure 4.11 Normalized saturation throughput as a function of frame size for CSMA/CA for the general scenario in Fig. 4.2. 83

Figure 4.12 Normalized saturation throughput as a function of frame size for RTS/CTS for the general scenario in Fig. 4.2. 84

Figure 4.13 Example of general infrastructure scenario, where there are 10 stations in the communication range of the access point, and the stations may irregularly be exposed or hidden from each other. 84
Figure 4.14 Normalized saturation throughput as a function of frame size for CSMA/CA for the general scenario in Fig. 4.13 ................................................................. 85

Figure 4.15 Normalized saturation throughput as a function of frame size for RTS/CTS for the general scenario in Fig. 4.13 ................................................................. 85
Chapter 1

Introduction

Wireless networks are becoming very common and popular due to several advantages they have over their wired counterparts, primarily mobility and ease and speed of installation. In this dissertation, we design and analyze the Medium Access Control (MAC) protocol for two types of popular wireless networks: Wireless Sensor Networks (WSNs) and Wireless Local Area Networks (WLANs). We begin with an overview of WSNs and WLANs and then present our contributions in these areas.

1.1 Wireless Sensor Network

WSNs comprise a number of autonomous sensors and one or more sinks to cooperatively monitor physical or environmental conditions. Wireless Sensor Networks (WSNs) hold significant potential for distributed sensing of large geographical areas. Figure 1.1 presents a typical example of a wireless sensor network configuration. Many sensor nodes are scattered into the sensing area; each sensor node is equipped with wireless transceiver, a small microcontroller and an energy source. The network, through its sensors, samples the physical environment and transmits this information to the sink from node to node (i.e., each sensor supports a multi-hop routing algorithm). The sink gathers the data from the sensors, and transmits it to the monitoring stations through a backhaul infrastructure such as Internet.

WSNs have been used in various applications that monitor plants and animals, natural phenomena (e.g., weather pollution and earthquakes), military battlefield surveil-
lance, etc. Energy is often the scarcest resource of WSN nodes, and it determines the lifetime of WSNs. WSNs are often deployed in large numbers in various environments, including remote and hostile regions. For this reason, algorithms and protocols need to address lifetime maximization, robustness, fault tolerance and self-configuration.

1.2 Wireless Local Area Network

In the past few years, wireless LANs have come to occupy a significant niche in the LAN landscape. The main advantage of WLAN are the support for mobile and nomadic network connectivity. In some situations (e.g., large open areas, historical buildings, small offices and houses), WLANs are used as an alternative to a wired LAN.

Figure 1.2 presents a simple wireless LAN configuration that is typical of small deployments (e.g., hot spot or home). The figure depicts a backbone wired LAN (often Ethernet) that supports servers and workstations; the access point acts as an interface to the wireless LAN. Many computers such as laptops, desktops and cellular phones are connected to the backbone through the access point. The access point act as a bridge between the WLAN and the backbone.

A WLAN must meet typical requirements for LANs such as high capacity, ability to cover short distances, full connectivity among attached stations and broadcast and multicast
capability. There are also a number of requirements specific to the WLANs such as coverage, power consumption, reliability and security, license free operation, handoff/roaming and dynamic configuration. IEEE 802.11 is an evolving standard for WLANs developed by the IEEE LAN/MAN Standards Committee. It defines standards for the physical (PHY) and Media Access Control (MAC) layers for WLAN.

1.3 Contributions

Energy efficiency of the MAC protocol is a key design factor for wireless sensor networks (WSNs). Due to the importance of the problem, a number of energy efficient MAC protocols have been developed for WSNs. Existing preamble-sampling based MAC protocols (e.g., B-MAC and X-MAC) have overheads due to their preambles, and are inefficient at large wakeup intervals. SCP-MAC, a very energy efficient scheduling MAC protocol, minimizes the preamble by combining preamble sampling and scheduling techniques; however, it does not prevent energy loss due to overhearing; in addition, due to its synchronization procedure, it results in increased contention and delay. In this dissertation, we present an energy efficient MAC protocol for WSNs that avoids overhearing and reduces
contention and delay by asynchronously scheduling the wakeup time of neighboring nodes. We provide an energy consumption analysis for multi-hop networks. To validate our design and analysis, we implement the proposed scheme on the MICAz and TELOS B platforms. Experimental results show that AS-MAC considerably reduces energy consumption, packet loss and delay when compared with other energy efficient MAC protocols.

Due to its wide deployment, IEEE 802.11 has been the subject of numerous analytic studies but still lacks a complete analytical model. Despite the importance of the hidden terminal problem, there have been a relatively small number of studies that consider the effect of hidden terminals on IEEE 802.11 throughput. Moreover, most of existing work is focused on the ad-hoc or multi-hop cases rather than the far more common, basic infrastructure case. Furthermore, existing models are not accurate for scenarios with a small number of users when a short-term unfairness occurs. In this dissertation, we present a new analytical saturation throughput model for IEEE 802.11 for CSMA/CA and RTS/CTS for a simple but very typical infrastructure scenario with a small number of hidden stations. Simulation results are used to validate the model. Simulation results are used to validate the model and show that our model is very accurate. Lastly, we provide a saturation throughput model for IEEE 802.11 for the general infrastructure scenario with hidden stations. Simulation results show that this general model is reasonably accurate.
Chapter 2


2.1 Introduction

Wireless Sensor Networks (WSNs) hold significant potential for distributed sensing of large geographical areas. Energy efficiency is one of the most important requirements in designing a WSN, and the radio is recognized as a major source of the energy consumption in sensor nodes [1]. The Medium Access Control (MAC) protocol, in addition to controlling medium access, can be designed to reduce the energy consumption of the radio in WSNs. Idle listening is often the largest source of energy waste [1–5], and duty cycling mechanism (i.e., periodically putting the radio in a sleep state) is considered as one of the best techniques to reduce energy consumption in WSN MAC protocols.

Several MAC protocols using duty cycling have been developed for WSNs [1–5]; however, some of them (e.g., S-MAC [2] and T-MAC [3]) are not energy efficient because they have long uptimes. Others, like B-MAC [4] and X-MAC [5], minimize the uptime by using Low-Power Listening (LPL); however, the use of a long preamble limits the possible energy savings at long wake-up intervals. SCP-MAC [1] uses LPL to minimize the uptime and synchronizes the wake-up time of sensor nodes to reduce the need for a long preamble. However, the requirement on synchronizing the wake-up times results in high overhearing,
contention and delay penalties.

In this chapter, we present a simple but very energy efficient *Asynchronous Scheduled* MAC protocol (AS-MAC). AS-MAC employs duty cycling like the previous schemes to avoid idle listening and uses Low-Power-Listening (LPL) to minimize the periodic wakeup time. The nodes store the wakeup schedules of their neighbors; therefore they know when their neighbor wake up and they do not need to add long preambles at the beginning of transmission. AS-MAC also asynchronously coordinates the wakeup times of neighboring nodes to reduce overhearing, contention and delay unavoidable in synchronous scheduled MAC protocols such as S-MAC, T-MAC, and SCP-MAC.

An important advantage of AS-MAC is that it can build on existing underlying MAC protocols inheriting their properties (e.g., hidden terminal avoidance through RTS/CTS, reliability via ACKs and collision avoidance) while adding power efficiency. In what follows we assume that AS-MAC builds on a CSMA with collision avoidance and without RTS/CTS and ACKs (primarily for a fair comparison with SCP-MAC and for consistency with our implementation that builds on the CC2420 MAC available in MICAz and TELOSB motes).

To determine the performance of AS-MAC, we present a theoretical analysis, in which we compare AS-MAC with a very energy-efficient previous approach, SCP-MAC. We implement our proposed scheme on MICAz [6] and TELOSB [7] platforms and evaluate our protocol design and theoretical analysis in single- and multi-hop scenarios. The theoretical analysis and experimental results show that AS-MAC considerably reduces energy consumption while providing low latency and packet loss in comparison with SCP-MAC.

### 2.2 Related Work

Existing work fits into the *preamble-sampling* approaches [4, 5, 8], *scheduling* approaches [2, 3] and *hybrid* approaches [1].

The *preamble-sampling* MAC protocols (i.e., B-MAC [4], X-MAC [5] and WiseMAC [8]) exploit Low-Power-Listening (LPL) for sampling the preambles of the packets. LPL minimizes the duty cycle when there are no packet exchanges, but during transmissions, the preamble needs to be longer than the wakeup interval to guarantee that the receiver detects the channel activity. Thus, the overhead of preambles becomes large as the wakeup
interval increases. X-MAC reduces the overhead of receiving long preambles by using short and strobed preambles allowing unintended receivers to sleep after receiving only one short preamble and the intended receiver to interrupt the long preamble by sending an ACK packet after receiving only one strobed preamble. However, even in X-MAC, the overhead of transmitting the preamble still increases with the wakeup interval, limiting the efficiency of the protocol at very low duty cycles. WiseMAC (Wireless Sensor MAC) minimizes the length of the preamble, by using knowledge of the sensor nodes’ sampling schedules. WiseMAC is developed for the downlink of infrastructure-based WSNs; on the other hand, B-MAC and X-MAC are designed for fully distributed WSNs.

S-MAC [2] and T-MAC [3] are scheduling MAC protocols that synchronize the wakeup schedules of sensor nodes in a neighborhood. To synchronize the wakeup schedule, nodes periodically exchange SYNC packets. S-MAC was the first duty cycling WSN MAC protocol where all nodes in a neighborhood simultaneously wake up and listen to the channel. A drawback of this scheme is the need for a long periodic wakeup time that has to include the collision avoidance backoff, RTS-CTS exchange and compensation for clock drift as well as waiting for eventual transmissions from the neighbors. T-MAC reduces the long wakeup time of S-MAC by using a timer that shortens the wakeup time if the channel is idle; however its wakeup time is also much longer than LPL time because the timeout should be longer than the summation of the length of the contention interval, the length of an RTS packet and the turn-around time.

SCP-MAC [1] combines preamble sampling with scheduling techniques. It synchronizes the wakeup time of neighboring nodes, which minimizes the length of preamble. It also minimize the periodic wakeup time by LPL. In comparison with other MAC protocols it is very energy efficient, especially at very low duty cycles; however the synchronous mechanism of SCP-MAC has several drawbacks.

The first problem is that since all nodes in a neighborhood wake up at the same time, nodes cannot avoid overhearing the packets from and for each of their neighbors. S-MAC and T-MAC prevents overhearing by using RTS/CTS; however an RTS/CTS exchange has relatively high overhead (especially since data packets in WSNs are short). SCP-MAC also provides a mechanism to avoid overhearing by sleeping upon the receipt of a header for a different destination; however this approach only avoids overhearing the payload of the packet and also requires sufficiently low level access to hardware (such as for Mica2 using
a bit-streaming radio, Chipcon CC1000), and cannot be implemented on sensor platforms with a packet level radio (such as MicaZ and TelosB that use Chipcon CC2420).

The second problem is that SCP-MAC results in increased contention. At each synchronized wakeup time, every sender in a neighborhood has to contend to acquire the channel. This high contention increases packet loss and degrades the energy efficiency and throughput due to the resulting collisions. It can also result in congestion, which, in turn, deteriorates application-level reliability.

Finally, SCP-MAC incurs relatively large delays in multi-hop scenarios. The per-hop delay is at least equal to the wakeup interval. It can be much longer because all losing nodes in the contention have to postpone their transmissions to the next synchronized wakeup time. SCP-MAC provides an adaptive channel polling mechanism that reduces the average delay by adding \( n \) high-frequency LPL in the same frame, immediately following its regular LPL, when receiving a packet. It decreases the average delay when bursty traffic occurs (the next packet is then transmitted within the additional \( n \) LPLs). However, adaptive polling is not effective in reducing delay (and increases energy consumption) when traffic is light.

Our proposed scheme aims to solve all the above-mentioned problems of SCP-MAC while inheriting many of its advantages.

Several MAC protocols for IEEE 802.11-based ad-hoc networks [9, 10] have used an asynchronous wakeup mechanism similar to our proposed mechanism. Their purpose is not to decrease the energy consumption but rather to increase the robustness of network: they trade energy for network reliability. Tseng at al. propose three asynchronous wakeup protocols [9]. The protocols overlap periodic wakeup times of neighboring nodes; therefore, they can increase the reliability of the network, but the wakeup times of nodes are very long, which consume large energy. In [10], nodes store the wakeup schedules of neighbors like our proposed approach. The approach in [10] uses the scheduling information to allow the detection of neighbor departures. That is, maintaining neighbor schedules in the approach does not decrease energy consumption.

SPARE MAC [11] proposes a distributed scheduling solution which assigns to each sensor its own time slots for reception. It shares the basic principle of the asynchronous wakeup with our proposed protocol, which makes it reduce energy consumption. However, it is a collision free protocol that relies on a dynamic TDMA slotted structure.
2.3 AS-MAC Design

In this section, we describe the proposed *Asynchronous Scheduled* WSN MAC protocol (AS-MAC). The basic idea in AS-MAC is that nodes wake up periodically (but asynchronously from their neighbors) to receive packets. Nodes intending to transmit wake up at the scheduled wakeup time of the intended target node. We first describe the initialization phase and then introduce the periodic listening and sleep phase.

2.3.1 Initialization Phase

When a new node joins a WSN, it performs the initialization phase, in which it builds the neighbor table that stores its neighbors’ scheduling information, and chooses and announces its own unique offset of the periodic wakeup. Existing nodes may be in the initialization phase or the periodic listening and sleep phase. Nodes in the periodic listening and sleep phase perform LPL every wakeup interval, $I_{\text{wakeup}}$, and send Hello packet every Hello interval, $I_{\text{hello}}$. Hello packets are used to publish scheduling information: $I_{\text{wakeup}}$, $I_{\text{hello}}$ and offset of the periodic wakeup, $O_W$.

First, the new node receives the Hello packets from its neighbors, listening to the channel for a fixed amount of time, which should be longer than $I_{\text{hello}}$ to guarantee that the new node receives the Hello packets from its all neighbors. Each time the node receives a Hello packet from a neighbor, it determines the starting time of the reception of the Hello packet to the neighbor’s $O_W$ and stores $I_{\text{hello}}$, $I_{\text{wakeup}}$ and $O_W$ with the neighbor ID in its neighbor table.
After building the neighbor table of its neighbors, the new node will set its $O_W$ uniquely. One possible approach is the node sets its $O_W$ to a random value different from its neighbors’ $O_W$. However, it is better to distribute the $O_W$s of neighboring nodes as evenly as possible. To achieve this distribution, one possible approach is that the node sets its $O_W$s at the half point of the longest interval among the neighbors’ $O_W$s. Figure 2.1 presents an example of setting the unique $O_W$. Node C is the new participant. The interval from B’s $O_W$ to A’s $O_W$ is longer than the interval from A’s $O_W$ to B’s $O_W$. In result, node C sets the half point of the former interval as its $O_W$. If the new node does not receive any Hello packet from neighbors and its neighbor table is empty, it sets its $O_W$ at a random time.

Once the new node sets its unique $O_W$, it enters the periodic listening and sleep phase. To publish its schedule to its neighbors, it sends its scheduling information to its known neighbors (discovered during the initialization phase) at their upcoming $O_W$s separately.

We assume that $I_{\text{wakeup}}$ is large enough to asynchronously accommodate $O_W$s of all nodes in communication range of each other. To prevent that any two neighboring nodes miss each other forever, nodes perform periodic neighbor discovery like existing scheduled MAC protocols [1–3]. For neighbor discovery nodes listen to the channel for a whole $I_{\text{hello}}$ interval. This allows the discovery of missed or mobile nodes. The frequency of the neighbor discovery depends on the rate of change of the network topology and can be done adaptively.

### 2.3.2 Periodic Listening and Sleep Phase

After the initialization phase, a node enters the periodic listening and sleep phase. Figure 2.2 presents the simple finite state machine for the periodic listening and sleep phase. The node starts the periodic listening and sleep phase setting wakeup interval, $I_{\text{wakeup}}$. A node performs LPL every $I_{\text{wakeup}}$ timeout to receive an incoming packet. If the channel is busy, the node receives the incoming packet. If the wakeup time of the node is also Hello time, the node receives the packet after sending a Hello packet. When a node has a packet to send, it waits in sleep state until the receiver is scheduled to wake up, and it wakes up when the receiver does. If the wakeup time of the receiver is Hello time, it receives the Hello packet and then sends the packet. If not, it directly sends the packet with the preamble compensating clock drift. We first describe the operation of a node acting as a receiver,
Figure 2.2: Finite state machine for the periodic listening and sleep phase

and then the operation as a sender.

As a receiver, if the wakeup time is also a Hello time, if the channel is clear, the node broadcasts a Hello packet. After that, the node waits to receive a packet until a timeout, $t_O$. If a packet is sent before $t_O$, the node receives it (e.g., Receiver in Figure 2.3). The value of $t_O$ should be only slightly longer than the maximum backoff time of the senders. After receiving a packet, the receiver goes back to sleep. When the wakeup time is not a Hello time, if the channel is silent, the node immediately goes back to sleep. If the channel is busy, the node stays in the listen state and receives the incoming packet. After reception, the node returns to sleep (e.g., Receiver in Figure 2.4).

As a sender, the node does not wake up at neighbors’ wakeup time if it has nothing to send. If a node has a packet to send, it waits in sleep state until the receiver’s wakeup time. Every node stores its neighbors’ scheduling information in its neighbor table; therefore it can predict the remaining time, $t_{remain}$, from the current time to the upcoming wakeup time of the receiver:

$$t_{remain} = I_{wakeup}(i) - (t_C - O_W(i)) \mod I_{wakeup}(i),$$  \hspace{1cm} (2.1)
where \( t_C \) is the current time, and \( i \) is the ID of the receiver. When the receiver’s wakeup time is also a Hello time, as shown in Figure 2.3, to compensate for the potential clock drift between the sender and the receiver, the sender wakes up earlier than \( t_{\text{remain}} \) by a guard time, \( t_{G1} \):

\[
t_{G1} = 2C_{\text{drift}}(t_C - O_W),
\]

where \( C_{\text{drift}} \) is the maximum clock drift rate. The sender waits for the receiver’s Hello packet with a timeout \( t_{G2} \):

\[
t_{G2} = 4C_{\text{drift}}(t_C - O_W) + t_{\text{LPL}},
\]

where \( t_{\text{LPL}} \) is the time for LPL. If the sender does not receive the Hello packet before \( t_{G2} \) seconds elapse, it postpones the transmission to the next wakeup time of the receiver. If the sender does receive the Hello packet from the intended receiver, it updates the receiver’s \( O_W \) in its neighbor table to the start time of the reception of the Hello packet (e.g., Sender I and II in Figure 2.3). To avoid the collision with the potential senders, the sender performs collision avoidance backoff and carrier sensing by randomly selecting a slot within the fixed contention window. If it loses the contention, the sender postpones the transmission to the receiver’s next wakeup time (e.g., Sender I in Figure 2.3). If it wins the contention, it sends the packet, after that it goes back to sleep (e.g., Sender II in Figure 2.3).

When the receiver’s wakeup time is not a Hello time, as shown in Figure 2.4,
sender should perform the collision avoidance backoff to avoid collision with other potential senders; therefore the guard time $t_{G3}$ is longer than $t_{G1}$ by the maximum contention window time. If it loses the contention, the sender postpones the transmission to the receiver’s next wake-up time. If it wins the contention, the sender sends the data packet with the preamble of $t_{G4}$ long that is longer than $t_{G2}$ by the remaining contention time (e.g., Sender in Figure 2.4).

The disadvantage of the asynchronous wake-up interval in AS-MAC is the inefficiency of broadcast. To broadcast a packet, AS-MAC has to transmit the packet once for each neighbor. Alternatively, it could broadcast a packet with long preamble like B-MAC. In AS-MAC, nodes do not overhear any packets because each receiver has its own unique $O_W$. Moreover, only senders trying to send to the same receiver contend to acquire the channel at the receiver’s wake-up time. Therefore, AS-MAC results in fewer collisions than SCP-MAC. In terms of delay, AS-MAC is faster on the average than SCP-MAC by a factor of two. In SCP-MAC, the per-hop delay of a packet is equal to wake-up interval, because the packet can be propagated at only the synchronized wake-up time. In AS-MAC, the per-hop delay of a packet is on the average half the wake-up interval. In AS-MAC, each node stores only its neighbors’ schedules.

### 2.4 Energy Consumption Analysis

In this section, we present a brief analysis of the performance of AS-MAC and compare it with SCP-MAC. We first provide energy consumption models, and then the numerical results based on this model.
Table 2.1: Notations and default values

<table>
<thead>
<tr>
<th>Notation</th>
<th>Meaning</th>
<th>Value (MICAz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L_{data}$</td>
<td>Data packet length</td>
<td>50 bytes</td>
</tr>
<tr>
<td>$L_{sync}$</td>
<td>SYNC packet length</td>
<td>18 bytes</td>
</tr>
<tr>
<td>$L_{hello}$</td>
<td>Hello packet length</td>
<td>18 bytes</td>
</tr>
<tr>
<td>$P_{tx}$</td>
<td>Power in transmission mode</td>
<td>52.2 mW</td>
</tr>
<tr>
<td>$P_{rx}$</td>
<td>Power in reception mode</td>
<td>56.4 mW</td>
</tr>
<tr>
<td>$P_{listen}$</td>
<td>Power in listen mode</td>
<td>56.4 mW</td>
</tr>
<tr>
<td>$P_s$</td>
<td>Power in sleep mode</td>
<td>0.003 mW</td>
</tr>
<tr>
<td>$P_{lpl}$</td>
<td>Power in LPL mode</td>
<td>12.3 mW</td>
</tr>
<tr>
<td>$t_{lpl}$</td>
<td>Time for LPL</td>
<td>2.5 ms</td>
</tr>
<tr>
<td>$t_B$</td>
<td>Time to TX/RX a byte</td>
<td>0.032 ms</td>
</tr>
<tr>
<td>$t_{slot}$</td>
<td>Time for contention slot</td>
<td>0.04 ms</td>
</tr>
<tr>
<td>$S_{cw}$</td>
<td>Contention window size</td>
<td>16</td>
</tr>
</tbody>
</table>

2.4.1 Energy Model

We design the energy model for AS-MAC in a multi-hop network. We assume that the network is organized as a tree rooted at the sink, and that every node periodically generates and sends a data packet to its parent; we also assume that there are not collisions, and that all wake-up intervals are the same. In SCP-MAC, we consider its basic mechanism without the collision avoidance, the two-phase contention and the adaptive channel polling. We assume that SYNC schedules of all nodes are evenly distributed such that SCP-MAC incurs the smallest overhead for transmitting and receiving preambles. We assume that the offsets of wake-up times of all nodes in a neighborhood are different such that neighboring nodes do not overhear each other.

Table 2.1 presents the notations, and default values of variables used in our analysis. We use the values of MICAz platform for the variables. The total energy consumption per second, $E$ includes transmission, reception, listen, sleep, and LPL, denoted as $E_{tx}$, $E_{rx}$, $E_{lx}$, $E_s$, and $E_{lpl}$, respectively:

$$E = E_{tx} + E_{rx} + E_{lx} + E_{lpl} + E_s,$$  \hspace{1cm} (2.4)

$$E = T_{tx} P_{tx} + T_{rx} P_{rx} + T_{lx} P_{lx} + T_{lpl} P_{lpl} + T_s P_s,$$  \hspace{1cm} (2.5)

where $T_{tx}$, $T_{rx}$, $T_{lx}$, $T_{lpl}$, $T_s$, and $P_{tx}$, $P_{rx}$, $P_{lx}$, $P_{lpl}$, $P_s$ are percentage times and powers for
transmission, reception, listen, LPL, and sleep, respectively. We assume that a node sleeps when it is not doing anything else:

\[ T_s = 1 - T_{tx} - T_{rx} - T_{lx} - T_{lpl}. \]  
(2.6)

For SCP-MAC, we follow the energy analysis in [1] (while trivially adapting it to the multi-hop network scenario we consider). In this section, we determine the values of \( T_{tx}, T_{rx}, T_{lx}, T_{lpl}, \) and \( T_s \) for AS-MAC.

In AS-MAC, nodes have the Hello interval, \( I_{hello} \) and the wake-up interval, \( I_{wakeup} \). We assume that \( I_{hello} \) is \( m \) times larger than \( I_{wakeup} \), i.e., \( I_{hello} = mI_{wakeup} \), where \( m \) is a positive integer. We assume that when a node sends \( m \) data packets, on the average it receives the parent’s Hello packet once.

The expected transmission time is:

\[ T_{tx} = T_{txhello} + T_{txdata} + T_{txpre}, \]  
(2.7)

where \( T_{txhello}, T_{txdata}, \) and \( T_{txpre} \) are the percentage times for transmitting Hello packets, data packets, and preambles respectively. Every hello interval, a node sends a hello packet:

\[ T_{txhello} = \frac{1}{I_{hello}} L_{hello} t_B. \]  
(2.8)

Nodes generate and send a packet every data generation interval, \( I_{data} \), and receive and forward packets generated by each of their children.

\[ T_{txdata} = \frac{n_{children} + 1}{I_{data}} L_{data} t_B, \]  
(2.9)

where \( n_{children} \) is the number of children of a node.

When a node needs to send data packets, it undergoes \( (m - 1) \) wake-up times without Hello packet of its parent for \( m \) transmissions; i.e., it sends \( (m - 1) \) preambles for \( m \) transmissions:

\[ T_{txpre} = \frac{m - 1}{m} \frac{n_{children} + 1}{I_{data}} T_{pretx}, \]  
(2.10)

where \( T_{pretx} \) is the transmission time for a preamble:

\[ T_{pretx} = 4r_{clk} m \frac{I_{data}}{n_{children} + 1} + t_{iple}. \]  
(2.11)
The expected reception time is:

\[ T_{rx} = T_{rx\text{hello}} + T_{rx\text{data}} + T_{rx\text{pre}}. \tag{2.12} \]

When a node needs to send data packets, it receives a Hello packet for each \( m \) data transmissions:

\[ T_{rx\text{hello}} = \frac{1}{m} \frac{n_{\text{children}}}{I_{\text{data}}} + \frac{1}{L_{\text{hello}} t_B}. \tag{2.13} \]

Each node receives a data packet generated by every child every data generation interval. The time spent receiving data packet is:

\[ T_{rx\text{data}} = \frac{n_{\text{children}}}{I_{\text{data}}} L_{\text{data}} t_B. \tag{2.14} \]

When a node receives data packets, it receives \((m - 1)\) preambles for \( m \) receptions because its adjacent child undergoes \((m - 1)\) wake-up times without Hello packet among \( m \) transmissions:

\[ T_{rx\text{pre}} = \frac{m - 1}{m} \frac{n_{\text{children}}}{I_{\text{data}}} T_{\text{prerx}}, \tag{2.15} \]

where \( T_{\text{prerx}} \) is the time for one preamble:

\[ T_{\text{prerx}} = 2 m r_{\text{clk}} \frac{I_{\text{data}}}{n_{\text{children}} + 1}. \tag{2.16} \]

The expected listening time is:

\[ T_{lx} = T_{lx\text{gt}} + T_{lx\text{cw}} + T_{lx\text{to}}, \tag{2.17} \]

where \( T_{lx\text{gt}}, T_{lx\text{cw}} \) and \( T_{lx\text{to}} \) are the percentage listening time while compensating for clock drift, while in contention and waiting for the time-out respectively.

To compensate for the clock drift, nodes listen the channel during:

\[ T_{lx\text{gt}} = 4 * r_{\text{clk}} + \frac{1}{m} \frac{n_{\text{children}} + 1}{I_{\text{data}}} t_{\text{ple}}. \tag{2.18} \]

When a node sends a data packet, it performs one collision avoidance backoff:

\[ T_{lx\text{cw}} = \frac{n_{\text{children}} + 1}{I_{\text{data}}} S_{\text{cw}} t_{\text{slot}}, \tag{2.19} \]
where $S_{cw}$ is the contention window size.

When nodes wake up, they listen to the channel on average for half of $t_O$ after sending Hello packet to receive an incoming packet:

$$T_{txto} = \frac{1}{2I_{hello}} t_O, \quad (2.20)$$

where the timeout $t_O$ is longer than the maximum contention time by one slot time:

$$t_O = (S_{mcwas} + 1) t_{slot}. \quad (2.21)$$

When a node wakes up, they perform LPL.

$$T_{ipl} = \frac{1}{I_{wakeup}} t_{tple}. \quad (2.22)$$

To calculate the optimal hello interval that minimizes the total energy of AS-MAC, we substitute $mI_{wakeup}$ for $I_{hello}$ of each equation and differentiate the total energy with respect to $m$:

$$\frac{dE}{dm} = 0, \quad (2.23)$$

by solving (2.23), we can calculate $Opt_m$, (rounded as it is an integer). Thus, the optimal $I_{hello}$ is:

$$Opt_{I_{hello}} = Opt_m I_{wakeup}, \quad (2.24)$$

where,

$$Opt_m = \sqrt{\frac{A_{as} + B_{as} + C_{as} + D_{as} + E_{as} + F_{as}}{G_{as} + H_{as}}}, \quad (2.25)$$

$$A_{as} = \frac{1}{I_{wakeup}} L_{hello} t_B(P_{tx} - P_s),$$
$$B_{as} = \frac{n_{children} + 1}{I_{data}} L_{hello} t_B(P_{rx} - P_s),$$
$$C_{as} = \frac{n_{children} + 1}{I_{data}} t_{tple}(P_{tx} - P_s),$$
$$D_{as} = \frac{n_{children} + 1}{I_{data}} S_{mcwas} t_{slot}(P_{tx} - P_s),$$
$$E_{as} = \frac{1}{2I_{wakeup}} t_O(P_{tx} - P_s),$$
$$F_{as} = \frac{n_{children} + 1}{I_{data}} t_{tple}(P_s - P_{tx}),$$
$$G_{as} = 4r_{ck}(P_{tx} - P_s),$$
$$H_{as} = 2r_{ck}(P_{rx} - P_s).$$
In AS-MAC, there is an optimal $I_{\text{hello}}$ with respect to $I_{\text{wakeup}}$, and the energy consumption monotonically decreases as $I_{\text{wakeup}}$ increases.

2.4.2 Numerical Results

In this section we present numerical results based on the energy models in Section 2.4.1 in a scenario with one hundred nodes in the ten by ten uniform rectangular grid with a sink in a corner, as shown in Figure 2.5. We assume that all nodes have the same wake-up interval and nodes only communicate with their immediate neighbors and forward data based on the fixed routing presented in Figure 2.5. We consider the energy consumption of two nodes. Node I receives and forwards data for 6 nodes and hears 126 transmissions in each sampling period; Node II receives and forwards data for 63 nodes and hears 143 packets during each sampling period. Because the choice of the wake-up interval effectively allows a trade-off between energy and delay, for fairness, we compare AS-MAC and SCP-MAC at similar per-hop delays (assuming that the per-hop delay for SCP-MAC is equal to the wake-up interval and for AS-MAC is half of the wake-up interval). We apply the optimal values calculated by our energy model for Hello interval and SYNC interval. The data generation interval is fixed at 100 seconds.

Figure 2.6 shows the energy consumption of Node I as a function of per-hop delay.
AS-MAC clearly outperforms SCP-MAC at a similar delay penalty. For the per-hop delay equal to 1.2 seconds, AS-MAC has a energy consumption of only 17.3% of that of SCP-MAC. Figure 2.7 depicts the sources of energy consumptions of each protocol for Node I. It is readily apparent that the main advantage of AS-MAC results from the elimination of overhearing, resulting in a considerable reduction of the energy spent in receiving mode. At higher wake-up intervals, AS-MAC can reduce most of the sources of power consumption. In contrast, the savings of SCP-MAC are negatively affected by the consumption due to overhearing that remains unchanged (and continues to dominate the overall energy budget).

Figures 2.8 and 2.9 show the energy consumptions and the sources of energy consumptions of Node II. Because Node II forwards far more packets than Node I, the benefit of AS-MAC in terms of energy consumption is smaller in Node II than in Node I. Qualitatively, the results for Node II are similar to the ones for Node I, with AS-MAC obviously outperforming SCP-MAC at all delays.

In summary, we can expect that at a similar latency, AS-MAC will outperform SCP-MAC primarily due to the overhearing avoidance. The higher the network size and
density, the larger the performance gap between AS-MAC and SCP-MAC.

2.5 Experimental Evaluation

To validate the design of AS-MAC and the energy consumption model in Section 2.4, we implemented AS-MAC in TinyOS [12] on two platforms equipped with the Chipcon CC2420 [13] radio: MICAz [6] and TELOSB [7]. In Section 2.5.1, we compare the performance of AS-MAC with the existing implementation of SCP-MAC on the MICAz platform. In Section 2.5.2, we compare AS-MAC with the default CC2420 CSMA/CA MAC protocol in the TELOSB platform, as there are no other implementation of duty cycling MAC protocols in TinyOS on the TelosB platform (X-MAC is implemented in MantisOS).

2.5.1 MICAz Platform

We consider three traditional evaluation metrics: energy consumption, packet loss, and delay. We evaluate AS-MAC and SCP-MAC by varying three parameters: the number
of neighbors, data generation interval and wakeup interval, or, in other words, network density, traffic load and duty cycle respectively.

**Energy Consumption**

To measure the power consumption, we monitor changes in the state of the radio. We used counters that accumulate time in each state of the radio (e.g., transmit, receive, listen, sleep and wakeup). At the end of the experiment, considering the energy consumption in each state in Table 2.1, we compute the total energy consumption.

We compare the energy consumption on the periodic listen and sleep phase of AS-MAC to that of SCP-MAC. The energy consumptions on the initialization phase and the neighbor discovery phase of AS-MAC are similar to those of SCP-MAC. Especially, the energy consumption in the initialization phase can be ignored because the lifetime of WSN is assumed to be very long, tens or hundreds of days, and the initialization phase is relatively very short.

![Figure 2.8: Energy consumption of Node II for the MICAz platform](image)
We use a star topology consisting of one receiver and up to five senders - all nodes are in communication range with all others; for the multi-hop experiments, we set up a chain network with six nodes, with sixth node as the sink at one end of the network. In the topology, each sender transmits a packet to the receiver every 10 seconds 20 times. The wakeup interval is set to one second in both MAC protocols, and the Hello interval of AS-MAC and SYNC interval of SCP-MAC is 60 seconds. The contention window size is 16 for both MAC protocols. We consider the power consumption as a function of the number of senders. Ideally, the energy consumption of MAC protocol should not increase with the increase in the number of neighbors. To validate our energy model in Section 2.4.1, we also show the energy estimations from the model. Before starting the data generation we use a random jitter to avoid periodic collisions.

Figure 2.10 presents average energy consumption of the senders as a function of the number of senders in MICAz platform, and Figures 2.11 and 2.12 show the sources of energy consumptions with two and five senders respectively in MICAz platform. The energy consumption of AS-MAC is almost constant regardless of the number of senders,
while the energy consumption of SCP-MAC increases with number of senders. Figures 2.11 and 2.12 show that the difference is due to the fact that AS-MAC can avoid overhearing while SCP-MAC cannot. The energy consumption in receive mode for AS-MAC is almost similar in Figures 2.11 and 2.12; however, that of SCP-MAC in Figure 2.12 is much larger than that of SCP-MAC in Figure 2.11. There are significant differences in the absolute values between the theoretic and the experimental results although qualitatively the models capture the behavior of the two MAC protocols. The root cause of the difference between the experimental result and the theoretical analysis is a discrepancy between the theoretical and real listen time of the protocol. In theory, we assume that the radio can perform LPL perfectly, but in reality, the radio sometimes recognizes a busy channel even when the channel is idle (this may be due to noise in the 2.4GHz ISM band). In this case, both AS-MAC and SCP-MAC will wait for the packet that does not exit. To prevent this error, both AS-MAC and SCP-MAC set up a timer. This additional listening time is not taken into account in the theoretical analysis of either protocols. Furthermore, we also neglected processing time in the theoretical analysis. We can expect that in a deployment scenario
Figure 2.11: Sources of energy of senders in the star topology with two senders on the MICAz platform.

with less noise the energy consumption will *closely* match the results of the analysis.

**Packet Loss**

To evaluate the packet loss, we use the chain multi-hop network. We set up a chain network with six nodes, with sixth node as the sink at one end of the network. For the multi-hop network, we reduced the RF output power of the node to its minimum value and placed the nodes about twenty centimeters apart; this results in good communication links between neighboring nodes and intermittent connectivity between nodes two hops apart. In this experiment the first five nodes periodically send packets to their parent (we used static routing), which in turn forwards it toward the sink. Each experiment lasted until each node generated and sent 10 packets (not counting the forwarded packets). The contention window size is four for both AS-MAC and SCP-MAC. Neither the SCP-MAC, nor the AS-MAC use buffers for retransmissions or queue the packets. We calculate the packet loss at the sink.
Figure 2.12: Sources of energy of senders in the star topology with five senders on the MICAz platform.

Figure 2.13: Packet loss at the sink for the multihop chain topology on the MICAz platform.
Figure 2.13 shows the packet loss in the sink node in MICAz platform. The packet loss of SCP-MAC is much larger than that of AS-MAC. The major cause of the large packet loss in SCP-MAC is the increased contention (as all nodes wake up at the same time to send packets) - the lack of buffers may amplify this effect.

Delay

Duty cycling MAC protocols trade-off delay for energy saving. To quantify the delay of AS-MAC and SCP-MAC, we use the same chain network of six nodes (with the sink being the sixth node and data being generated at the first node). We vary the wakeup interval and data generation rate from 1s to 8s.

Figure 2.18 shows the average per-hop delay of the five hops in MICAz platform (we measure the end-to-end delay and divide it by the hop count - five). As expected, the delay of AS-MAC is very close to half of the wakeup interval (the variations are due to the random choices of the offsets of the wakeup times during the initialization phase). On the
Table 2.2: RAM and ROM sizes on the MICAz platform

<table>
<thead>
<tr>
<th></th>
<th>RAM</th>
<th>ROM</th>
</tr>
</thead>
<tbody>
<tr>
<td>MICAz</td>
<td>4,000 bytes</td>
<td>128,000 bytes</td>
</tr>
<tr>
<td>SCP-MAC</td>
<td>898 bytes</td>
<td>15,638 bytes</td>
</tr>
<tr>
<td>AS-MAC</td>
<td>944 bytes</td>
<td>17,288 bytes</td>
</tr>
</tbody>
</table>

other hand, the per-hop delay of SCP-MAC is close to (just a little longer than) the wakeup interval, i.e., double that of AS-MAC. The adaptive channel polling of SCP-MAC cannot reduce the delay in this light traffic scenario.

**Memory Footprint**

The main overhead of AS-MAC in terms of memory usage is for maintaining the neighbor table. In our implementation of AS-MAC in the MICAz platform, the neighbor table consumes 11 bytes per neighbor, specifically, 2 bytes for neighbor ID, 4 bytes for wakeup interval, 4 bytes for the offset of wakeup interval and 1 byte for the Hello interval. The Hello interval is stored as a multiple of the wakeup interval.

Table 2.3 provides RAM and ROM sizes of AS-MAC and SCP-MAC in MICAz platform, in which AS-MAC has the neighbor table with five neighbors and its RAM and ROM footprints are slightly larger than those of SCP-MAC. RAM and ROM sizes of AS-MAC without the neighbor table are 889 bytes and 16,004 bytes respectively, which are very similar to those of SCP-MAC.

2.5.2 TELOSB Platform

We evaluate the performance of AS-MAC in terms of energy consumption, throughput and delay.

**Energy Consumption**

For the experiment we use a chain with four nodes with the fourth node as the sink at one end of the chain. The other three nodes periodically generate and send packets with same data generation interval. Each experiment lasted for 300s (5 minutes). We consider the energy consumption as a function of the wakeup interval and the data generation interval.
Figure 2.15 presents the measured energy consumption of the second node for AS-MAC, CC2420 CSMA/CA as a function of the wakeup interval in TELOS B platform. The data generation interval to one sample every 10 seconds. Not surprisingly, the energy consumption of CC2420 CSMA/CA is constant. The energy consumption of AS-MAC decreases with the increase in the wakeup interval. With a wakeup interval of 100s, the energy consumption of AS-MAC is only 0.37% of that of CC2420 CSMA/CA.

Figure 2.16 presents the energy consumption of AS-MAC, CC2420 CSMA/CA as a function of the data generation interval for the second node in TELOS B platform. In this experiment we fix the wakeup interval to 10 seconds. The energy consumption of CC2420 CSMA/CA is again practically constant.

The energy consumption of AS-MAC decreases as the data generation interval increases because the sleep time increases, and energy consumption in sleep mode is much smaller than in any other mode. At data generation intervals of 100s (i.e., one packet every 100 seconds) the energy consumption of AS-MAC is approximately 0.51% of that of CC2420 CSMA/CA.
Throughput

We measure the throughput varying data generation rate on a single-hop network with one sender and one receiver. We measured the throughput as:

$$T = \frac{N_{rp} S_p}{T_e},$$

where $T$ is the throughput, $T_e$ is the duration of the experiment, $N_{rp}$ is the number of received packets, and $S_p$ is the packet size. In our the experiment, the packet size is 38 bytes, and the duration is 100 seconds. The wakeup interval of the sender and receiver is fixed to one second.

Figure 2.17 shows the throughput of AS-MAC and CC2420 CSMA/CA as a function of the data generation rates in TELOSB platform. The maximum throughput of AS-MAC is about 70% of that of CC2420 CSMA/CA. The root cause of this reduction in throughput in our experiments is that sometimes, the receiver does not receive the data packets of the sender (due to transmissions errors) and goes back to sleep although the
transmitter still has packets in its queue. While the receiver is sleeping the transmitter’s queue overflows, thus resulting in packet loss.

Delay

To quantify the delay of AS-MAC, we use the same chain of four nodes (with the sink being the fourth node and data being generated at the first node). We vary the wakeup interval and data generation rate from 1s to 40s.

Figure 2.18 shows the average delay of the three hops in TELOS B platform. As in the experiment in MICAz platform, the delay is very close to half of the wakeup interval. The per-hop delay of the default CSMA/CA protocol is about 12ms.

Memory Footprint

Table 2.3 provides RAM and ROM sizes of AS-MAC and CC2420 CSMA/CA in TELOS B platform, in which AS-MAC has the neighbor table with five neighbors. RAM and
Figure 2.18: Per-hop delay of AS-MAC and CC2420 CSMA/CA as a function of the wakeup interval on the TELOSB platform.

Table 2.3: RAM and ROM sizes on the TELOSB platform

<table>
<thead>
<tr>
<th></th>
<th>RAM</th>
<th>ROM</th>
</tr>
</thead>
<tbody>
<tr>
<td>TELOSB</td>
<td>16,000 bytes</td>
<td>48,000 bytes</td>
</tr>
<tr>
<td>CC2420 CSMA/CA</td>
<td>546 bytes</td>
<td>9,268 bytes</td>
</tr>
<tr>
<td>AS-MAC</td>
<td>784 bytes</td>
<td>11,906 bytes</td>
</tr>
</tbody>
</table>
ROM footprints of AS-MAC are larger than those of CC2420 CSMA/CA because AS-MAC was implemented on top of CC2420 CSMA/CA. The neighbor table consumes 12 bytes per neighbor, specifically, 2 bytes for neighbor ID, 4 bytes for wakeup interval, 4 bytes for the offset of wakeup interval and 2 byte for the Hello interval. The Hello interval is stored as a multiple of the wakeup interval. RAM and ROM sizes of AS-MAC without the neighbor table are 724 bytes and 11,238 bytes respectively.
Chapter 3

IEEE 802.11 Saturation Throughput Analysis in the Presence of Hidden Terminals for Simple Infrastructure Scenario

3.1 Introduction

IEEE 802.11 is one of the most widely adopted standards since its development for Wireless Local Area Network computer communication by the IEEE LAN/MAC Standards Committee in 1996. Many amendments have been standardized and widely deployed in a number of devices such as personal computers, laptops, mobile phones, home networks and other electronic devices that benefit from wireless networking due to their advantages such as low cost, high throughput and convenience. IEEE 802.11 provides infrastructure mode and ad-hoc mode. The infrastructure mode using access points is mainly used for permanent installations. The ad-hoc mode is also implemented in most of IEEE 802.11 systems but rarely used. In the infrastructure case, the clients can access the Internet through the access point. It also presents three MAC algorithms: CSMA/CA, RTS/CTS and PCF. This study is limited to CSMA/CA and RTS/CTS. Because communication ranges of the access point and clients vary significantly due to obstacles, interference, transmission power, antenna
gain, and location, the hidden terminal problem is common in IEEE 802.11. Many service providers and enterprises deploy small, low-cost access points serving a small number of active users. While there may be a large number of associated stations, the number of active stations is small in most cases. Thus, the infrastructure case with a small number of active stations in the presence of hidden stations is a very common scenario for IEEE 802.11 networks.

The hidden terminal problem effectively disables the carrier sense capability of the protocol and negatively affects the performance of system. Figure 3.1 presents simulation results for CSMA/CA of IEEE 802.11a at the data rate of 12Mbps: normalized saturation throughput and sequence traces of frames successfully received by the access point for two scenarios with an access point and two stations that are either exposed or hidden from each other. In the scenarios, stations are in the communication range of an access point and always have packets to send to the access point. Figure 3.1 (a) shows that the saturation throughput severely deteriorates in the hidden station scenario as the frame size increases. Because the collision probability increases with the frame size in the hidden station scenario the resulting saturation throughput decreases. Figure 3.1 (b) shows that the hidden terminal scenario is not fair on the short-term time scale. This short-term unfairness negatively affects upper layer protocols such as TCP timeouts and high jitter for real-time audio and video streams [14]. The short-term unfairness problem occurs when the number of stations is small and reduces as the number of stations increases.

Due to its wide deployment, IEEE 802.11 has been the subject of numerous analytic studies [14–26], but it still lacks a comprehensive analytical model. Despite the importance of the hidden terminal problem, there are a relatively small number of studies [19,21–24] that consider the effect of hidden terminals on the IEEE 802.11 throughput. Moreover, much of existing work is centered on the ad-hoc or multi-hop cases rather than the far more common, basic infrastructure case [20, 24, 25]. Furthermore, existing models [19, 20] are not accurate especially for scenarios with a small number of stations, when the short-term fairness deteriorates.

In this chapter, we provide a new analytical saturation throughput model for both distributed MAC protocols (i.e., CSMA/CA and RTS/CTS) for a simple but typical infrastructure scenario with hidden stations that the short-term unfairness occurs. We then evaluate the accuracy of the model through extensive simulations. The simulation result
Figure 3.1: Undesired effects of CSMA/CA in the presence of hidden terminals; decreasing throughput and (b) deteriorating short-term fairness.
shows that our model is extremely accurate for a wide range of conditions.

The rest of this chapter is organized as follows. Section 3.2 briefly reviews CSMA/CA, RTS/CTS, the hidden terminal problem and the short-term unfairness problem. In Section 3.3, we describe the problem tackled in this chapter and summarize the related work. Section 4.1 provides our analytical models for the saturation throughput for CSMA/CA and RTS/CTS of IEEE 802.11 in the presence of hidden terminals. Section 3.5 validates the accuracy of our model through simulation.

3.2 Background

IEEE 802.11 provides three types of MAC algorithms: CSMA/CA, RTS/CTS and the Point Coordination Function (PCF). CSMA/CA and RTS/CTS distribute the transmission decision to the stations using a carrier sense mechanism similar to Ethernet. PCF is an alternative access method and relies on polling from the access point. CSMA/CA and RTS/CTS are contention-based algorithms, while PCF is a contention-free algorithm. This study is limited to CSMA/CA and RTS/CTS. In this section, we briefly summarize the contention-based MAC algorithms (i.e., CSMA/CA and RTS/CTS) and then reviews the hidden terminal problem and the short-term unfairness problem.

3.2.1 CSMA/CA

CSMA/CA stands for carrier sense multiple access with collision avoidance. When a station has a frame to transmit, it senses the medium. If the medium is idle, it waits to see if the medium remains idle during a period equal to a Distributed Coordination Function Inter-frame Space (IFS) (DIFS). IEEE 802.11 provides three IFS: DIFS, PIFS and Short IFS (SIFS). DIFS is used as a minimum delay for contending frames. SIFS is used for immediate response actions such as ACK and CTS. If the medium is busy, the station defers transmission and continues to monitor the medium until the current transmission is over. Once the transmission is over, the station waits for another DIFS. If the medium remains idle for this period, the station waits for a random back-off time while sensing the medium. If the medium remains idle during this interval, the station transmits the frame. If the medium becomes busy during the back-off time, the back-off timer is halted and resumes when the medium becomes idle again. If the transmission is unsuccessful, as determined by
the absence of an acknowledgment, then it is assumed that a collision has occurred and the station retransmits the frame. The number of retransmission is limited by the maximum number of retransmissions, \( m \). CSMA/CA uses a discrete-time back-off scale. The random back-off time is slotted and the station can transmit only at the beginning of each time slot. To ensure the stability of the back-off process, CSMA/CA employs a technique known as binary exponential back-off. Stations choose a random number, \( r \), in the range of \((0, w-1)\) where \( w \) is called the contention window. The resulting back-off time is the product of \( r \) and the slot time, \( \sigma \). At the first transmission attempt, \( w \) is set to the minimum contention window size, \( CW_{\text{min}} \). After each unsuccessful transmission, \( w \) is doubled, up to the maximum value \( CW_{\text{max}} = 2^n CW_{\text{min}} \). Once the contention window size reaches \( CW_{\text{max}} \), it stays at \( CW_{\text{max}} \) from the \((n + 1)^{th}\) to the \( m^{th}\) retransmission stages.

CSMA/CA consists of a DATA transmission followed by an ACK transmission. Figure 3.2 provides an example of CSMA/CA for the scenario, where station A, station B and an Access Point (AP) are in same communication range, and A and B always have a packet to send. In the first attempt of transmission, A chooses 3 and B chooses 10 for the random back-off times. Station A wins the contention because its back-off time is smaller than B’s; therefore it transmits a data frame without collision. During the time, B senses that the medium is busy (as it listens the data frame transmitted by A), it halts its back-off time at 7. The AP receives the data frame from A, and transmits ACK to A after a SIFS delay. In the second transmission attempt, A chooses 7 for the back-off time, and it transmits a data frame after the back-off time. At that time, B also transmits a data
frame because remaining back-off time of B was also 7. As back-off times of A and B reach 0 at the same time, the nodes transmit simultaneously leading to a collision. Thus, when competing nodes can sense each other’s carriers, collisions occur only when back-offs of senders (A and B) countdown to 0 at the same time. During the collision, the AP does not receive either data frame correctly, and therefore it does not send an ACK.

3.2.2 RTS/CTS

RTS/CTS consists of a RTS transmission followed by a CTS transmission followed by the DATA/ACK exchange. Before a station transmits a data frame, it transmits a special short frame, request to send (RTS). The station receiving RTS responds with a clear to send (CTS) after SIFS. RTS and CTS carry the information of the length of the packet to transmitted. Any other stations overhearing a RTS or CTS update their values of the network allocation vector (NAV) that contains the information of the period of time in which the channel will remain busy; therefore, even when a station is hidden from either the transmitting or the receiving station, by detecting just one RTS or CTS frame, it can delay further transmission, thus avoid collisions. RTS/CTS exchange is also effective because it can reduce the wasted channel time that occurs during collisions, as the RTS frame size is much smaller than that of regular data frames.

Figure 3.3 provides an example of an RTS/CTS exchange for the same scenario as the Fig. 3.2. In the first attempt of transmission, station A wins, and transmits RTS.
Figure 3.4: (a) A scenario with an access point and two clients hidden from each other and (b) an example of a collision in CSMA/CA.

Station $B$ hears the RTS and updates its NAV and can compute the period of time during which the channel will remain busy. The $AP$ receives the RTS and response with CTS after SIFS. Station $A$ transmits the data frame. The second attempt of transmission in Fig. 3.3 shows a situation when a collision occurs.

### 3.2.3 The Hidden Terminal Problem

The hidden terminal problem occurs when two stations transmitting to a third one are out of carrier sense range to each other. Figure 3.4 (a) depicts a general hidden terminal scenario, in which station $A$ is considered to be hidden from the station $B$, and vice versa. Such situation occurs commonly when stations $A$ and $B$ are separated by an obstacle, but have clear line of sight to the $AP$. Assume that stations attempt transmissions to $AP$ using the basic CSMA/CA method; $A$ will begin its frame transmission once the link is detected to be idle. Since station $B$ is out of range of station $A$, station $B$ may sense the link idle and also begin a transmission. Since the $AP$ is in the range of stations $A$ and $B$, the $AP$ will not receive either transmission correctly. Thus, in the hidden terminal scenario, collisions not only occurs when the back-off times of two senders reach zero simultaneously, but possibly also when they reach zero at different times. Figure. 3.4 (b) provides an example, in which the back-off times of $A$ and $B$ reach zero at the different times, but a collision occurs. To avoid the collision, the difference between the back-off times of $A$ and $B$ has to be longer than the sum of transmission time for the data frame, an SIFS and the propagation time.

If the RTS method is used, station $B$ will not hear the RTS from station $A$ but will
Figure 3.5: An example packet trace showing the short-term unfairness problem for the scenario depicted in Fig. 3.4 (a)

hear the CTS response from AP and will know to be silent during the entire transmission period. In this scenario, it is still possible for station B to collide with the RTS from station A, but the lost channel time is limited to the RTS frame, whereas when using CSMA/CA the lost channel time includes the entire length of the data frame. In the absence of collisions, the RTS/CTS is less efficient than CSMA/CA due to the lengthier handshake. In the presence of collisions, RTS/CTS can be more efficient by reducing the number of unsuccessful transmissions.

3.2.4 The Short-term Unfairness Problem

IEEE 802.11 employs the binary exponential back-off to ensure the stability of the back-off process. Stations choose a random number, r, in the range of (0, w-1), where w is called the contention window. At the first transmission attempt, w is set to the minimum contention window size, CW_{min}. After each collision, w is doubled, up to the maximum value \( CW_{max} = 2^n CW_{min} \). Once the contention window size reaches CW_{max}, it stays at CW_{max} from the \((n + 1)^{th}\) to \((m - 1)^{th}\) retransmission stages, where m is the maximum retransmission stage. The exponential back-off guarantees fairness for both long-term and short-term time scales in the absence of hidden stations, but it is not fair on the short-term time scale in the presence of hidden stations as shown in Fig. 3.1 (b).

Figure 3.5 presents an example of the short-term unfairness problem for the basic
hidden terminal scenario in Fig. 3.4 (a). In the first and second transmission attempts, stations try to transmit frames, but the contention window sizes are too small, and collisions occur. In the third transmission attempt, the contention window sizes of stations are large enough and the differences of resulting back-off times of the stations are large enough to avoid a collision. Let’s assume A’s back-off was smaller than B’s back-off at that time. Station A successfully transmits a frame, and resets its retransmission stage to zero for the transmission of the next frame. The resulting back-off time of A is very small, and it may be able to transmit a frame successfully several times (4th and 5th transmission attempt in Fig. 3.5) before B’s back-off time reaches zero. At the sixth transmission attempt, B’s back-off time reaches zero and a collision occurs, and the retransmission stages of both stations increase by one. Now, B’s contention window size is four times of A’s contention window size in the seventh transmission attempt; therefore the back-off time of A is very likely smaller than that of B. In result, A transmits a frame successfully, and A’s retransmission stage is reset to zero again. A’s back-off time is small and A transmits frames successfully several times (8th to 11th transmission attempt in Fig. 3.5). The monopoly of station A is likely to continue until B’s retransmission stage reaches the limit.

3.3 Problem Definition and Related Work

In this section, we define the problem tackled in this chapter and briefly summarize the related work.

3.3.1 Problem Definition

Bianchi [15] presents a novel Markov chain model for saturation throughput for IEEE 802.11 in the absence of hidden terminals. This study has been considered as one of the seminal papers for the throughput model of IEEE 802.11 and has been extended in many subsequent studies [14–26]. A few of them, [19–21] extend Bianchi’s model to consider hidden terminals. In this dissertation, we call the similar models in [19–21] extended Bianchi’s models (although other papers extended Bianchi’s model in other directions, we are only interested in hidden terminal extensions). However, Bianchi’s model cannot accurately model hidden terminals as we will shortly explain.

In Bianchi’s Markov chain model [15], the MAC state is represented by two vari-
ables: the current retransmission stage, and the remaining back-off time. Through the
model, Bianchi obtains the transmission probability, $\tau$, that the station transmits a packet
in a randomly chosen slot time using the collision probability of a transmitted packet, $p$.
Finally, it calculates the saturation throughput using $\tau$ and $p$. The key approximation that
enables this model is the assumption that the probability of collision, $p$, is constant and
independent of the retransmission stage. Bianchi assumes that $p$ is the probability that at
least one of the $n - 1$ remaining stations transmits in a time slot, where $n$ is the number
of stations. This assumption is reasonable in the exposed terminal scenario because the
collision can occur only during the contention phase in the absence of hidden terminals.
The resulting probability $p$ is [15]:

$$p = 1 - (1 - \tau)^{n-1}. \quad (3.1)$$

Extended Bianchi’s models [19–21] use the same original Markov chain model of [15], but
it obtains $p$ while taking into account a number of hidden terminals. When a frame is
transmitted, it can collide with the frames transmitted by hidden terminals at any time
during a vulnerable period, $T_v$, while it can collide with the frames transmitted by exposed
terminals only during the contention. The resulting probability of collision is:

$$p = 1 - [(1 - \tau)^{N_E}]\left(\frac{(1 - \tau)^{N_H}}{\tau} \right), \quad (3.2)$$

where $N_E$ is the number of exposed stations, $N_H$ is the number of hidden stations, and $\sigma$
is slot time. The vulnerable period, $T_v$ is the time that it takes from the time a station
transmits a frame before the station receives the corresponding ACK:

$$T_v = t_{frame} + SIFS + \delta, \quad (3.3)$$

where $t_{frame}$ is the transmission time for the frame, and $\delta$ is the propagation delay. These
models also assume that $p$ is constant regardless of the retransmission stage.

Figure 3.6 presents the normalized saturation throughput of CSMA/CA as a func-
tion of the frame size derived by the simulation and the extended Bianchi’s model when
numbers of stations are two and ten respectively, where IEEE802.11a is considered and
the channel rate is 24Mbps. The figure shows that the extended Bianchi’s model is inaccu-
rate especially when the number of stations is small. The key reason for the inaccuracy is
that the extended Bianchi’s models inaccurately assume that $p$ is constant regardless of the
Figure 3.6: (a) A scenario with an access point and a client hidden from the other client; (b) Normalized saturation throughput for scenario (a); (c) A scenario with an access point and five clients hidden from the other five clients; (d) Normalized saturation throughput for scenario (c).
retransmission stage. Figure 3.7 shows that the collision probability is not constant but rather increases in scenarios of Fig. 3.6 as the retransmission stage increases. The variation of the collision probability is especially severe when the number of stations is small. This variation is due to the short-term unfairness problem presented in Sec. 3.2.4. When there are hidden stations, stations have a high collision probability at large retransmission stages. Thus, in the extended Bianchi’s models, the assumption that $p$ is constant is not accurate, and the resulting throughput is also inaccurate.

In this chapter, we present a new Markov chain model reflecting the variation of the collision probability as a function of the retransmission stage for the saturation throughput for IEEE 802.11 in the presence of hidden terminals for the infrastructure case with a small number of stations.

### 3.3.2 Related Work

Existing work for throughput models of IEEE 802.11 can be classified into work that considers hidden terminals and work that does not.

Work in [15–18] presents throughput models for IEEE 802.11 in the absence of hid-
den terminals. Bianchi [15] presents a novel saturation throughput model for IEEE 802.11 under the assumption of ideal channel conditions (no hidden terminal and no noise) and a finite number of terminals. The key approximation that enables the model is the assumption of constant and independent collision probability of a frame transmitted by each station, regardless of the back-off stage. Simulation results show that the model is accurate as the number of stations increases. Vishnevsky et al. [16] provide a saturation throughput model for the IEEE 802.11 in the presence of noise. This method allows estimating a probability of a frame rejection occurring when the number of frame transmission retries attains its limit. Work in [17] presents a throughput model for IEEE 802.11a that incorporates non-saturated traffic and the SNR. The model shows the need for an admission control mechanism and suggests a mechanism for maximizing the throughput while maintaining a fair allocation. Malone et al. [18] analyzes 802.11 throughput in non-saturated heterogeneous conditions. The authors of [18] argue that typical network conditions are non-saturated and heterogeneous. The model captures interesting features of non-saturated operation. In particular, the model predicts that the peak throughput occurs prior to saturation. The model allows stations to have different traffic arrival rates, thus allowing the model to address the question of fairness between competing flows.

Work in [19,20,24,25] provides throughput models for IEEE 802.11 in the presence of hidden terminals. Work in [19] consider the infrastructure case, while [20,24,25] focus on the ad-hoc case. Ekici et al. [19] extend Malone’s model [18] to analyze IEEE 802.11 unsaturation throughput for the infrastructure case with hidden nodes. This model assumes that the collision probability is constant regardless of the retransmission stage. Although the simulation result shows that the model is accurate in a scenario with two hidden stations, the model is accurate only when the offered load is small. As the load increases, the results of the model become inaccurate as shown in Fig. 3.6.

Work in [20,24,25] presents analytical models for multi-hop and ad-hoc networks. Hou et al. [20] analyze the throughput of the IEEE802.11 DCF scheme using the RTS/CTS access mechanism in multi-hop ad-hoc networks. The simulation results show that the model is accurate; however, if this model is applied to CSMA/CA, it becomes inaccurate, as it does not consider the retransmission stage for obtaining the collision probability. Work in [24] presents an analytical model for deriving saturation throughput in multi-hop ad hoc networks with nodes randomly placed according to a two dimensional Poisson distribution.
In [25] the authors argue that stations may send more traffic than can be supported by the network, which results in high packet-loss rate, routing instability and unfairness problems. The authors shows that controlling the offered load at the sources can eliminate these problems. They also provide an analysis to estimate the optimal offered load that maximize the throughput of a multi-hop traffic flow. The paper provides both simulation results and experimental results with a real six-node multi-hop network.

3.4 Analytical Model

In this section we present a saturation throughput model using a Markov chain model for IEEE 802.11 DCF in the presence of hidden terminals for the infrastructure case with two hidden stations. We consider the basic infrastructure scenario with two hidden terminals shown in Fig. 3.8, where stations $A$ and $B$ are in the communication range of the access point but are hidden from each other. The reason why we consider only two stations is that this scenario is one of the most basic and serious situations when the short-term unfairness occurs. We assume that stations always have frames to transmit to the access point, as we are interested in the saturation throughput. The access point does not transmit any data frame, only receives the frames from the stations and transmits ACKs to the
stations. In the infrastructure case with hidden stations, the results of the previous transmissions always affect on those of the next transmissions. Specifically, when a successful transmission occurs, the hidden station that did not transmit overhears the ACK, and it halts its countdown, then resumes it with remaining back-off time at the next transmission attempt. When a collision occurs, the starting times of the next transmission attempts of the stations that collided (when they are hidden from each other) become different, and the resulting collision probability at the next transmission attempt is affected by the difference. Markov chain models are memoryless, in other words, given the present state, future states are independent of the past states; therefore our model using a Markov chain cannot exactly model IEEE 802.11; however, simulation results in Section refsec:validation show that our model is reasonably accurate. Section 3.4.1 presents a simple model. Section 3.4.2 presents an enhanced model to approximate the dependency property.

3.4.1 Simple Model for the IEEE 802.11 Network with Two Hidden Terminals

In this simple model, we ignore the fact that the previous transitions can affect the current transitions. We make two assumptions:

- we assume that when a successful transmission occurs, all stations choose new random back-off times instead of halting and restarting the previous back-off time in the next transmission attempt;

- we assume that when a collision occurs, all stations start their next transmission attempts at the same time, i.e., after the station that later transmits the lost frame ends its ACK timeout.

System State Model

Our system model is based on a two dimensional Markov chain as shown in Fig. 3.9. Although at the first glance this may look similar to Bianchi’s model [15], the two are fundamentally different: while Bianchi’s model represents the state of a node, with retransmission stage and back-off counters defining the state, our model represents the state of the entire system, by using the retransmission stage of each station as system state.
In this section, we define and then compute the system state probability matrix $B$. The elements of the probability matrix $B$, $b_{ij}$ represent the probability that station $A$ is in the retransmission stage $i$ and station $B$ is in the retransmission stage $j$ for $i, j \in \{0, m - 1\}$, where $m$ is the maximum retransmission limit. The retransmission stages, $i$ and $j$, start from 0 at the first transmission and are increased by one every time a transmission results in a collision, up to $m - 1$. We construct the Markov chain model of the system state by using system states and transition probabilities between the states. By imposing the normalization condition to the system state model, we find the stationary system state probability $B$.

Whenever a transmission occurs, the system can move from state $(i, j)$ to one of the following three states:

$$
(i, j) \rightarrow \begin{cases} 
(0, j), & i, j \in \{0, m - 1\}, \\
(i, 0), & i, j \in \{0, m - 1\}, \\
(i', j'), & i' = (i + 1) \text{ mod } m, \\
& j' = (j + 1) \text{ mod } m.
\end{cases}
$$

(3.4)

The first and second transitions represent the cases when successful transmissions occur. The first transition represents the case when station $A$ transmits a frame successfully. Once a station (e.g., $A$) transmits a frame successfully, the retransmission stage of the station goes back to zero for the next frame. In this case, the other station (e.g., $B$) participates in the contention but chooses a back-off larger than the sum of the other station’s (e.g., $A$’s) back-off and the vulnerable period, $T_v$. Thus, $B$ does not transmit a frame during the time $A$ transmits a frame and $B$’s retransmission stage does not change.

The vulnerable period, $T_v$, is:

$$
T_v = \begin{cases} 
\frac{t_{frame} + SIFS + \delta}{\sigma}, & \text{if CSMA/CA is used} \\
\frac{t_{rts} + SIFS + \delta}{\sigma}, & \text{if RTS/CTS is used,}
\end{cases}
$$

(3.5)

where $t_{frame}$ is the transmission time for data frame, $t_{rts}$ is the transmission time for RTS, $\delta$ is the propagation delay and $\sigma$ is the slot time. The second transition in (3.4) accounts for the case when station $B$ transmits a frame successfully. The third transition in (3.4) represents the case when a collision occurs. A collision occurs when the difference between the back-offs of the hidden stations is smaller than $T_v$. When a collision occurs,
the retransmission stages of both stations are incremented by one up to $m - 1$. Once the retransmission stage reaches $m - 1$ and a collision occurs, and the retransmission stage is reset to zero for the transmission of the next frame. The Markov chain model for the system state with these transitions is shown in Fig. 3.9, in which the contention window size of the station increases exponentially from $CW_{\text{min}}$ to $CW_{\text{max}}$ as the retransmission stage increases. Therefore, when the system is in the state $(i, j)$, the contention window sizes of stations are:

$$
\begin{align*}
CW_a &= \min\{CW_{\text{max}}, (CW_{\text{min}} + 1)2^i - 1\}, \\
CW_b &= \min\{CW_{\text{max}}, (CW_{\text{min}} + 1)2^j - 1\},
\end{align*}
$$

(3.6)

where $CW_a$ is the contention window size of station $A$, and $CW_b$ is the contention window size of station $B$.

Based on this model, we can find the transition probabilities for the system states. Let $T$ denotes the transition probability matrix. Denote with $t_{ijkl}$, the transition probability that the system state moves from $(i, j)$ to $(k, l)$. Each state has three effective transition probabilities (i.e., $t_{ij0j}$, $t_{ij0i}$ and $t_{ij'j'}$); all other transition probabilities are zero. We first
find the transition probability for a successful transmission (i.e., $t_{ij0j}$ and $t_{iji0}$) and then that for a collision (i.e., $t_{ij'j'}$).

The transition probability for a successful transmission will be computed first. First, let us calculate $t_{ij0j}$, the transition probability when station $A$ transmits a frame successfully. The size of sample space is the product of $CW_a$ and $CW_b$. Station $A$ can win the contention only when the back-off of station $B$ is larger than the back-off of $A$ by at least $T_v$; thus $t_{ij0j}$ is:

$$t_{ij0j} = \frac{\sum_{b_a=0}^{L_a} \sum_{b_b=b_a+T_v}^{CW_b} 1}{CW_a CW_b}$$

where, because if the back-off of station $A$ is larger than $CW_b - T_v$, $A$ cannot win. Hence,

$$L_a = \min\{CW_a, CW_b - T_v\}.$$  \hspace{1cm} (3.7)

In a similar way, $t_{iji0}$, the transition probability when station $B$ transmits a frame successfully is:

$$t_{iji0} = \frac{\sum_{b_b=0}^{L_b} \sum_{b_a=b_b+T_v}^{CW_a} 1}{CW_a CW_b}$$

where,

$$L_b = \min\{CW_b, CW_a - T_v\}.$$  \hspace{1cm} (3.8)

Due to the normalization condition of the transition probability, the transition probability of the collision, $t_{ij'j'}$ is:

$$t_{ij'j'} = 1 - t_{ij0j} - t_{iji0}.$$  \hspace{1cm} (3.11)

By imposing the normalization condition of $B$, $\sum_{i=0}^{m-1} \sum_{j=0}^{m-1} b_{ij} = 1$, and using $B = TB$, we can compute the system state probability matrix $B$.

**Time Model**

We need to determine the time spent in each transition in Fig. 3.9: the average back-off time matrix, $O$, the time for transition matrix, $E$, and the time for payload transmission matrix, $D$. When $t_{ijkl}$ is zero, the corresponding elements of the time matrices should also be zero; therefore we need to only take care of the elements of the time matrixes corresponding to non-zero transitions in (3.4).

Let $O$ denote the average back-off time matrix spent in each transition in the matrix $T$. Denote with $o_{ijkl}$, the average back-off time when the system moves from $(i,j)$
to \((k,l)\). First, let us calculate the average back-off time of the successful transmissions, \(o_{ij0j}\) and \(o_{iji0}\). The total number of cases of back-off times for each transition is the same as the denominator of (3.8). The back-off time in each case is that of the station that transmits a frame successfully. This yields:

\[
\begin{align*}
    o_{ij0j} &= \sum_{L_a=0}^{La} \frac{b_a \sum_{b_b=0}^{CW_b} \sum_{b_b=b_a+Tv}^{b_a} 1}{\sum_{b_b=0}^{CW_b} 1}, \\
    o_{iji0} &= \sum_{L_b=0}^{L_b} \frac{b_b \sum_{b_a=0}^{CW_a} \sum_{b_a=b_b+Tv}^{b_b} 1}{\sum_{b_a=0}^{CW_a} 1}.
\end{align*}
\]  

(3.12)

We have to calculate the average back-off time corresponding to a collision, \(o_{ijij'}\). The total number of collisions is the difference between total number of cases and the number of cases of successful transmissions. A collision occurs when the difference of back-off times of stations is smaller than \(Tv\). In each case, the maximum value between the back-off times of the stations should be considered when considering collisions by the second assumption (when a collision occurs, all stations start their next transmission attempts at the same time right after the station transmitting the lost frame later ends its timeout for ACK.) Thus:

\[
o_{ijij'} = \frac{(\sum_{b_a=0}^{CW_a} \sum_{b_b=b_a}^{b_a+Tv-1} b_b + \sum_{b_a=0}^{CW_a} \sum_{b_b=b_a}^{b_b+Tv-1} b_a)\sigma}{CW_a CW_b - \sum_{b_a=0}^{L_a} \sum_{b_b=0}^{CW_b} 1 - \sum_{b_a=0}^{L_b} \sum_{b_b=0}^{CW_b} 1}.
\]  

(3.13)

Let \(E\) denote the time matrix representing time spent in each transition in the matrix \(T\). Denote with \(e_{ijkl}\) the time for transition, \(t_{ijkl}\). The time for successful transmissions, \(e_{ij0j}\) is as shown in Fig. 3.10:

\[
e_{ij0j} = \begin{cases} 
    DIFS + o_{ij0j} + t_{frame} + SIFS + \delta + t_{ack} + \delta, \\
    \text{(if CSMA/CA is used)} \\
    DIFS + o_{ij0j} + t_{rts} + SIFS + \delta + t_{cts} + SIFS + \delta + t_{frame} + SIFS + \delta + t_{ack} + \delta, \\
    \text{(if RTS/CTS is used)}
\end{cases}
\]  

(3.14)

where \(t_{frame}, t_{ack}, t_{rts}\) and \(t_{cts}\) are transmission times for a frame, \(ACK, RTS\) and \(CTS\) respectively. We obtain \(e_{ijij0}\) by replacing \(o_{ij0j}\) with \(o_{iji0}\).

When a collision occurs, the station waits for the timeout instead of receiving the
The time for a collision, $e_{iji'j'}$ is as shown in Fig. 3.10:

$$ e_{iji'j'} = \begin{cases} 
DIFS + o_{iji'j'} + t_{frame} + \delta + t_{timeout}, & \text{(if CSMA/CA is used)} \\
DIFS + o_{iji'j'} + t_{rts} + \delta + t_{timeout}, & \text{(if RTS/CTS is used)} 
\end{cases} $$

(3.15)

where $t_{timeout}$ is the expected time for receiving ACK or CTS packet. This yields:

$$ t_{timeout} = \begin{cases} 
SIFS + t_{ack} + \delta, & \text{(if CSMA/CA is used)} \\
SIFS + t_{cts} + \delta. & \text{(if RTS/CTS is used)} 
\end{cases} $$

(3.16)

Let $D$ denote the payload transmission time matrix during the transition. Denote with $d_{ijkl}$, the payload time during the transition $t_{ijkl}$. The payload is the data portion of the frame. When a collision occurs, $d_{iji'j'}$ is zero. The payload time for successful transmissions, $d_{ij0j}$, $d_{iji0}$, is the payload transmission time as it is. This yields:

$$ d_{ij0j} = d_{iji0} = t_{payload}. $$

(3.17)
Saturation Throughput Model

Let $S$ denote the normalized saturation system throughput, defined as the fraction of time the channel is used to successfully transmit payload bits. $B$ represents normalized time for each state and $T$ is the transition probability. The normalized saturation system throughput is equal to the total time spent in the payload transmission the payload transferred divided by the total time spent by the system. Thus:

$$S = \frac{\sum_{i=0}^{m-1} \sum_{j=0}^{m-1} \sum_{k=0}^{m-1} \sum_{l=0}^{m-1} b_{ij} d_{ijkl} t_{ijkl}^s}{\sum_{i=0}^{m-1} \sum_{j=0}^{m-1} \sum_{k=0}^{m-1} \sum_{l=0}^{m-1} b_{ij} e_{ijkl} t_{ijkl}^c} + \sum_{i=0}^{m-1} \sum_{j=0}^{m-1} \sum_{k=0}^{m-1} \sum_{l=0}^{m-1} b_{ij} e_{ijkl} t_{ijkl}^c},$$  

(3.18)

where $t_{ijkl}^s$ is the transition probability for a successful transmission, $e_{ijkl}^s$ is the time spent in successful transmission, $t_{ijkl}^c$ is the transition probability for a collision and $t_{ijkl}^c$ is the time spent in a collision.

3.4.2 An Enhanced Model for the IEEE 802.11 Network with Two Hidden Terminals

This model enhances the simple model in Sec. 3.4.1 to better match the behavior of IEEE 802.11. It is still not 100% accurate but it largely accounts for the fact that in IEEE 802.11 the results of the previous transmission affect those of the next transmission. Specifically:

- when a successful transmission occurs, the hidden station that did not transmit, overhears the ACK frame, it halts its countdown and then resumes it with the remaining back-off time in the next transmission attempt. That is, the back-off time of the hidden station reduces as much as sum of $T_v$ and the winner’s back-off time after a successful transmission.

- when a collision occurs, the starting time instances of the next transmission attempts of the stations that collided (when they are hidden from each other) are different.

System State Model

We define an enhanced transition probability matrix, $T'$. Denote with $t'_{ijkl}$, the enhanced transition probability that the system state moves from $(i, j)$ to $(k, l)$. We assume
Figure 3.11: Example of a successful transmission occurring in the previous transition.

that state \((i, j)\) is the current state, \((k, l)\) is the next state, and \((p, q)\) is the previous state with \(i, j, k, l, p, q \in \{0, m - 1\}\). In \(T'\), the next transition probabilities (i.e., \(t'_{ijkl}\)) depend on the result of previous transition. To obtain \(t'_{ijkl}\), we consider two things. First, we consider the probability that the system state \((i, j)\) occurred from a transition from the system state \((p, q)\). We denote this probability with \(p_{pqij}\). Thus:

\[
p_{pqij} = \frac{t_{pqij}}{\sum_{m=0}^{m-1} \sum_{q=0}^{m-1} t_{pqij}}.
\]  

(3.19)

Second, we consider the adjustment of the next transition probability by taking into account the previous transition (either a successful transmission or a collision). We denote this adjustment with \(t^{pqij}_{ijkl}\). These yield:

\[
t'_{ijkl} = \sum_{p=0}^{m-1} \sum_{q=0}^{m-1} p_{pqij} t^{pqij}_{ijkl}.
\]  

(3.20)

Thus we can obtain the transition probability for successful transmissions (i.e., \(t'_{ij0j}\) and \(t'_{ijj0}\)). The transition probability for collision, \(t'_{ijj'j'}\), is simply obtained by the normalization condition:

\[
t'_{ijj'j'} = 1 - t'_{ij0j} - t'_{ijj0}.
\]  

(3.21)

The adjustment in the transition probability, \(t^{pqij}_{ijkl}\), depends on whether the previous transition was a successful transition or a collision. We first focus on how the next transition probability (i.e., \(t^{pqij}_{ijkl}\)) should be adjusted when the previous transition was a...
successful transmission. Figure 3.11 shows the case when a successful transmission occurred during the previous transition, where station $B$ transmitted a frame successfully. Station $A$ performs the back-off countdown, overhears the ACK frame and it halts its back-off countdown; it then resumes the remaining back-off countdown in the next transmission attempt; therefore the back-off time of the station that did not transmit a frame (i.e., $A$) reduces on average by sum of $T_v$ from (3.5) and the average back-off time of the station that did transmit a frame (i.e., $B$). This yields:

\[
 t^{pqij}_{ijkl} = \begin{cases} 
 t_{ijkl}(CW_a, CW_b - n_r CW_r), & \text{if } A \text{ transmitted a frame successfully} \\
 t_{ijkl}(CW_a - n_r CW_r, CW_b), & \text{if } B \text{ transmitted a frame successfully} 
\end{cases}
\]  

(3.22)

where the expression $t_{ijkl}(a, b)$ represents the value of $t_{ijkl}$ as in (3.8), (3.9) and (3.11) with the contention window size of station $A$ being $a$ and that of station $B$ being $b$, $n_r$ is the average number of times of the contention window size to be reduced and $CW_r$ is the contention window size to be reduced each time. $CW_r$ is, as shown in Fig. 3.11:

\[
 CW_r = \frac{o_{pqij}}{\sigma} + T_v,
\]  

(3.23)

We now show, how to compute $n_r$. The example in Fig. 3.11 shows the case when a single successive successful transmission occurred in the previous transmission. If two successive successful transmissions occurred, the contention window size will be reduced by two times of the value of $CW_r$ in the next transmission attempt, and if three successive successful transmissions occurred, the contention window size will be reduced by three times of the value of $CW_r$ in the next transmission attempt, etc. We assume that probabilities of having one, two or more successive successful transmissions are the same. This assumption is fairly reasonable, because once a station transmits a frame successfully, its retransmission stage is reset to 0, and the station very likely transmits a frame successfully again as shown in Fig.3.5. Thus, the average number of reduction of contention window size is:

\[
 n_r = \sum_{i=1}^{l} \frac{i}{2} = \frac{l + 1}{2},
\]  

(3.24)

where $l$ is:

\[
 l = \begin{cases} 
 CW_b & \text{if } A \text{ transmitted a frame successfully}, \\
 CW_a & \text{if } B \text{ transmitted a frame successfully}. 
\end{cases}
\]  

(3.25)
The previous transition was a successful transmission if $i \neq ((p+1) \mod m)$ or $j \neq ((q+1) \mod m)$. The retransmission stage of the station that transmits a frame successfully is reset to 0, thus $i=0$ means that $A$ transmitted a frame successfully, while $j=0$ means that $B$ transmitted a frame successfully. Thus:

$$\begin{cases} 
\text{if}((i \neq ((p+1) \mod m)) \text{ or } (j \neq ((q+1) \mod m))) \text{ and if}(i=0), & A \text{ transmitted a frame successfully,} \\
\text{if}((i \neq ((p+1) \mod m)) \text{ or } (j \neq ((q+1) \mod m))) \text{ and if}(j=0), & B \text{ transmitted a frame successfully.} 
\end{cases}$$

(3.26)

We now show how the next transition probability (i.e., $t_{pqij}$) should be adjusted when the previous transition was a collision. Figure 3.12 shows the case when a collision occurred in the previous transition. The previous transition was a collision in the following cases:

$$\text{if}(i = ((p+1) \mod m)) \text{ and } (j = ((q+1) \mod m)).$$

(3.27)

Let’s consider the transition probability (i.e., $t_{pqij}$) that station $A$ transmits a frame successfully in the next transmission attempt. First, let us consider the case when station $A$ transmitted a frame faster than station $B$ in the previous transmission as shown in Fig. 3.12. If station $A$ chooses a back-off time between $0 \leq b_b < T_c$ in the next transmission, $A$ cannot avoid a collision with the previous transmission of station $B$. The average difference between the back-off times of two contending stations when a collision occurs is
the half of $T_v$ because the collision occurs only when the difference between the back-off times is smaller than $T_v$. On the other hand, $T_c$ should not be a negative number, and when $\frac{T_v}{2} < \frac{t_{\text{timeout}} + DIFS}{\sigma}$, the collision with the previous transmission of station $B$ does not occur. Therefore we consider the maximum value between zero and $\frac{T_v}{2} - \frac{t_{\text{timeout}} + DIFS}{\sigma}$. Thus, the value of $T_c$, the reduction in the back-off time that can lead to successful transmissions is:

$$T_c = \max\{0, \frac{T_v}{2} - \frac{t_{\text{timeout}} + DIFS}{\sigma}\}. \quad (3.28)$$

Therefore, $0 < b_b < T_c$ should be excluded in the calculation of the successful transition probability.

If station $A$ chooses a back-off time among $T_c \leq b_a \leq CW_a$, station $B$ should choose a larger back-off time than station $A$ by $\frac{T_v}{2}$ in order for station $A$ to transmit a frame successfully because of the assumption that the back-off of station $A$ starts earlier than the back-off of station $B$ by $\frac{T_v}{2}$ on average; therefore the number of events when station $A$ transmits a frame successfully is:

$$E_{AA} = \sum_{b_a = 0}^{CW_a} \sum_{b_b = T_c}^{CW_b} 1. \quad (3.29)$$

Second, let us consider the case when station $B$ transmitted a frame faster than station $A$. Station $B$ should choose a larger back-off time than station $A$ by at least $\frac{3T_v}{2}$ in order for station $A$ to transmit a frame successfully because of the assumption that the back-off of station $B$ starts earlier than the back-off of station $A$ by $\frac{T_v}{2}$ on average; therefore the number of events leading to a successful transmission by station $B$ is:

$$E_{AB} = \sum_{b_a = 0}^{CW_a} \sum_{b_b = b_a + \frac{3T_v}{2}}^{CW_b} 1. \quad (3.30)$$

We assume that the probability that $A$ transmitted a frame faster than $B$ in the previous transmission is equal to the probability that $B$ transmitted faster than $A$; therefore the sample space of all the cases leading to a successful transmission by $A$ is $2CW_aCW_b$. Thus $t^{pqij}_{ij0j}$, the adjusted transition probability that station $A$ transmits a frame successfully in the next transition, is:

$$t^{pqij}_{ij0j} = \frac{E_{AA} + E_{AB}}{2CW_aCW_b}, \text{ where } i = ((p+1) \mod m) \text{ and } j = ((q+1) \mod m). \quad (3.31)$$
In a similar way, \( t'_{ij0} \), the adjusted transition probability that station \( B \) transmits a frame successfully in the next transition, is:

\[
 t'_{ij0} = \frac{E_{BA} + E_{BB}}{2CW_aCW_b},
\]

where \( E_{BA} \) and \( E_{BB} \), are:

\[
\begin{align*}
E_{BA} &= \sum_{b_a=T_v}^{CW_a} \sum_{b_b=b_a+1}^{CW_b} 1, \\
E_{BB} &= \sum_{b_b=0}^{CW_b} \sum_{b_a=b_b+1}^{CW_a} 1.
\end{align*}
\] (3.33)

We can calculate \( t'_{ij0} \) and \( t'_{ij'0} \) using (3.20), (3.22), (3.31) and (3.32) and \( t'_{ij'j'} \) using (3.21). In result, we obtain the adjusted transition probability matrix, \( T' \), and by imposing the normalization condition, \( \sum_{i=0}^{m-1} \sum_{j=0}^{m-1} b'_{ij} = 1 \), and using \( B' = T'B' \), we can obtain an enhanced stationary probability matrix \( B' \).

**Time Model**

The time for transition matrix, \( E \), and the time for payload matrix, \( D \), are similar to those of Sec. 3.4.1. The average back-off time matrix, \( O \) is also similar to that of Sec. 3.4.1 except the average back-off time corresponding to a collision, \( o_{ij'j'} \). In this model, we consider that the next transmission starts at the time when the station that transmitted a frame faster in the previous transmission ends (e.g., station A in Fig. 3.12) the time-out for ACK in the previous transmission as in Fig. 3.12; therefore the back-off time corresponding to a collision is the minimum back-off time of the two back-off times of the contending stations. Thus \( o_{ij'j'} \) is:

\[
o_{ij'j'} = \frac{(\sum_{b_a=0}^{CW_a} b_a \sum_{b_b=b_a+T_v-1}^{CW_b} 1 + \sum_{b_b=0}^{CW_b} b_b \sum_{b_a=b_b}^{b_b+T_v-1} 1)\sigma}{CW_aCW_b - \sum_{b_a=0}^{CW_a} \sum_{b_b=b_a+T_v}^{CW_b} 1 - \sum_{b_b=0}^{CW_b} \sum_{b_a=b_b+T_v}^{CW_a} 1}.
\] (3.34)

**3.4.3 IEEE 802.11a Physical Layer Model**

The 802.11a standard operates in 5 GHz band and uses Orthogonal Frequency-Division Multiplexing (OFDM) modulation. It provides eight payload data rates (i.e., 6, 9, 12, 18, 24, 36, 48, 54Mbps) with different modulation schemes (i.e., BPSK, QPSK, 16-QAM and 64-QAM) and coding rates. In this section, we show how to obtain the transmission
times of frame, payload, RTS, CTS and ACK for IEEE 802.11a. Refer to the IEEE 802.11a standard [27] for a more complete presentation.

In IEEE 802.11a standard, the MAC data frame (i.e., MAC Protocol Data Unit (MPDU)) consists of the MAC Header, payload (0 to 2312 bytes) and Frame Check Sequence (FCS). The MAC header and FCS together are 28 bytes, the RTS MPDU is 20 bytes, and the CTS and ACK MPDUs are 14 bytes long. During transmission, a PLCP preamble and a PLCP header are added to the MAC frame to create the physical layer frame (e.g., PLCP Protocol Data Unit (PPDU)). The PLCP preamble field, is composed of 10 repetitions of a short training sequence (0.8 $\mu$s) and two repetitions of a long training sequence (4 $\mu$s). The PLCP header except the SERVICE field constitutes a single OFDM symbol (4 $\mu$s). The 16-bit SERVICE field of the PLCP header and the MAC frame (along with six tail bits and pad bits) are transmitted at the data rate specified in the RATE field. Therefore the transmission times of data frame, RTS, CTS and ACK is:

$$
\begin{align*}
  t_{\text{data}} &= t_{\text{PLCP preamble}} + t_{\text{PLCP header}} + t_{\text{SERVICE field}} + t_{\text{tail}} + t_{\text{MAC header}} + t_{\text{FCS}} + t_{\text{payload}}, \\
  &= 16\mu s + 4\mu s + \frac{(16+6)+28+8+\text{payload bit}}{\text{dataRate}(m)}, \\
  t_{\text{rts}} &= t_{\text{PLCP preamble}} + t_{\text{PLCP header}} + t_{\text{SERVICE field}} + t_{\text{tail}} + t_{\text{rts MPDU}}, \\
  &= 16\mu s + 4\mu s + \frac{(16+6)+20+8}{\text{dataRate}(m)}, \\
  t_{\text{cts}} &= t_{\text{PLCP preamble}} + t_{\text{PLCP header}} + t_{\text{SERVICE field}} + t_{\text{tail}} + t_{\text{cts MPDU}}, \\
  &= 16\mu s + 4\mu s + \frac{(16+6)+14+8}{\text{dataRate}(m)}, \\
  t_{\text{ack}} &= t_{\text{PLCP preamble}} + t_{\text{PLCP header}} + t_{\text{SERVICE field}} + t_{\text{tail}} + t_{\text{ack MPDU}}, \\
  &= 16\mu s + 4\mu s + \frac{(16+6)+14+8}{\text{dataRate}(m)}.
\end{align*}
$$

(3.35)

The frame formats of IEEE 802.11g [28] are the same as those of IEEE 802.11a; therefore the transmission times of data frame, RTS, CTS and ACK of IEEE 802.11g are identical to those of IEEE 802.11a.
Table 3.1: Fixed parameters of IEEE 802.11a

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAC header size</td>
<td>34 bytes</td>
</tr>
<tr>
<td>ACK size</td>
<td>14 bytes</td>
</tr>
<tr>
<td>RTS size</td>
<td>20 bytes</td>
</tr>
<tr>
<td>CTS size</td>
<td>14 bytes</td>
</tr>
<tr>
<td>$CW_{\text{min}}$</td>
<td>15</td>
</tr>
<tr>
<td>$CW_{\text{max}}$</td>
<td>1023</td>
</tr>
<tr>
<td>slot time</td>
<td>9 $\mu$s</td>
</tr>
<tr>
<td>SIFS</td>
<td>16 $\mu$s</td>
</tr>
<tr>
<td>DIFS</td>
<td>34 $\mu$s</td>
</tr>
</tbody>
</table>

3.5 Model Validation

To validate our saturation throughput model in Sec. 4.1, we perform detailed simulations. We compare the result of our model with that of extended Bianchi’s model and the simulation result. We first describe the simulation setup and then provide the results.

3.5.1 Simulation Setup

We use the discrete event simulation environment, OMNET++ [29], as the simulator and its mobility framework [30] as the simulation model for IEEE 802.11. The mobility framework considers the signal-to-noise ratio and the bit-error to determine whether a frame is transmitted correctly or not. Because we consider neither the signal-to-noise ratio nor bit-errors in our model, we disable these features. Thus, the receiver considers collisions the cases when it receives any other frames during the reception of a frame. OMNET++ mobility framework considers IEEE 802.11b. Because we are interested in IEEE 802.11a, we modify the parameters of the physical layer of IEEE 802.11 implementation of the mobility framework as shown in Sec. 3.4.3. The fixed parameters of IEEE 802.11a are shown in Table 3.1. We validated the accuracy of the IEEE 802.11 model in OMNET++ by comparing the results of two exposed station scenarios (two exposed station and ten exposed station scenarios) with Bianchi’s model, which is widely accepted as an accurate model for exposed terminal scenarios and has been in turn validated with other network simulators (e.g., ns-2 and OPNET) and hardware-based experiments. Figures 3.13 to 3.16 show the results of
For the validation, we consider the simple infrastructure scenario with two hidden stations shown in Fig. 3.8. Station A and B are hidden from each other, and they are in the communication range of the access point. We are interested in the saturation throughput, therefore the clients always have packets to send to the access point. The access point does not transmit any data frame and only receives the data frames from the stations and replies with ACK frames. To obtain the normalized system throughput, we observe the number of frames per second received by the access point, $F_n$, and divide the product of $F_n$ and the size of payload, $P_s$, by the payload data rate, $D_r$. Therefore the normalized system throughput $S$ (also known as channel utilization) is:

$$ S = \frac{F_n P_s}{D_r}. \quad (3.36) $$

We consider following three traditional evaluation parameters: frame size, payload...
Figure 3.14: Normalized saturation throughput for CSMA/CA as a function of the frame size at the payload data rate of 54Mbps for ten exposed station scenario.

Figure 3.15: Normalized saturation throughput for RTS/CTS as a function of the frame size at the payload data rate of 54Mbps for two exposed station scenario.
Figure 3.16: Normalized saturation throughput for RTS/CTS as a function of the frame size at the payload data rate of 54Mbps for ten exposed station scenario.

data rate and retransmission stage limit. In each simulation, the system runs for 300 seconds.

- **Frame size**: As frame size increases, the collision probability also increases, and in result, the throughput decreases. In IEEE 802.11a, the maximum frame size at the MAC level is 2346 bytes, but we only consider the range of 0 to 1500 bytes, because, in practice, the access point is almost always connected to Ethernet in the infrastructure case, and the maximum frame size of Ethernet is 1500 bytes.

- **Payload data rate**: IEEE 802.11 provides eight payload data rates (6, 9, 12, 18, 24, 36, 48, and 54Mbps). We consider all of the data rates in our evaluations.

- **Retransmission Stage Limit**: The throughput increases with the retransmission stage limit. More importantly, as the retransmission state limit increases, the short-term unfairness problem becomes more severe. $CW_{min}$ is set to 15 and $CW_{max}$ is set to 1023.
Figure 3.17: Normalized saturation throughput for CSMA/CA as a function of the frame size at the payload data rate of 6Mbps.

3.5.2 Simulation Results and Analysis

In this section, we present simulation result as a function of frame size, data rate and retransmission stage limit, and evaluate the accuracy of our model.

Frame Size

Figures 3.17 to 3.20 show the normalized saturation throughput as a function of the frame size for the data rates, 6 and 54Mbps for CSMA/CA and RTS/CTS respectively, where "Our Model" is the enhanced model presented in Sec. 3.4.2 and "Extended Bianchi’s Model" is the one in [19–21].

For the CSMA/CA (Figs. 3.17 and 3.18), for small frames, as the frame size increases, the normalized throughput also increases because the header induced overhead of the frame decreases. After the normalized throughput reaches its maximum, it decreases as the frame size increases because the vulnerable period that causes collisions also increases. Note that the vulnerable period increases with the frame size as shown in (3.3). The normalized throughput of CSMA/CA may appear high in comparison with a pure ALOHA system in Figs. 3.17 and 3.18 (with a maximum value of about 0.35 to 0.4). This relatively
Figure 3.18: Normalized saturation throughput for CSMA/CA as a function of the frame size at the payload data rate of 54Mbps.

Figure 3.19: Normalized saturation throughput for RTS/CTS as a function of the frame size at the payload data rate of 6Mbps.
Figure 3.20: Normalized saturation throughput for RTS/CTS as a function of the frame size at the payload data rate of 54Mbps.

High channel utilization is due to the short-term unfairness problem caused by the binary exponential back-off used in IEEE 802.11, in which if a station transmits a frame successfully, the station resets its contention window size to the minimum value. As a result, the winning station has a smaller back-off time than the other stations, and easily monopolizes the channel (resulting in a decreased collision probability) until another station wins the contention. This effect deteriorates the short-term fairness but increases the throughput.

Our model shows a good accuracy in Fig. 3.17 and 3.18. The results of our model are very similar those of the simulation, while the results of extended Bianchi’s model are far from those of the simulation. The main difference between our model and extended Bianchi’s model is that our model considers the variation of the collision probability with the retransmission stage, while extended Bianchi’s model assumes a constant value for the collision probability regardless of the retransmission stage. Due to the memoryless property of the Markov chain model, our model cannot exactly reflect the behavior of the real system; However, as shown in these figures, our enhanced model is very accurate.

For RTS/CTS (Figs. 3.19 and 3.20), the throughput increases with the frame size because the size of the RTS frame is constant regardless of the frame size, resulting
in a constant collision probability and resulting in increasing throughput. For RTS/CTS, both the results of extended Bianchi’s model and those of our model match with those of the simulation because the size of RTS frame is small enough to reduce the short-term unfairness problem.

**Data Rate**

Figures 3.21 to 3.24 present the normalized saturation throughput as a function of the payload data rates for frame sizes of 250 and 1500 bytes for CSMA/CA and RTS/CTS respectively. For CSMAC/CA (Figs. 3.21 and 3.22), the normalized throughput first increases with the data rate because the vulnerable period decreases with an increase in data rate. After it peaks, the normalized throughput decreases because the physical layer header overhead of IEEE 802.11a is constant regardless of the data rate resulting a reduced channel utilization. The results of our model are exactly same as those of the simulation.

For RTS/CTS (Figs. 3.23 and 3.24), the throughput decreases as the data rate increases because the physical layer header overhead of IEEE 802.11a is constant regardless of the data rate. For RTS/CTS, both the results of extended Bianchi’s model and those of
Figure 3.22: Normalized saturation throughput for CSMA/CA as a function of the payload data rate at the frame size of 1500 bytes.

Figure 3.23: Normalized saturation throughput for RTS/CTS as a function of the payload data rate at the frame size of 250 bytes.
Figure 3.24: Normalized saturation throughput for RTS/CTS as a function of the payload data rate at the frame size of 1500 bytes.

our model match with those of the simulation.

Retransmission Stage Limit

Figures 3.25 to 3.28 present the normalized saturation throughput as a function of the retransmission stage limit for frame sizes of 500 and 1500 bytes for the payload data rate of 48 Mbps for CSMA/CA and RTS/CTS respectively. For CSMA/CA (Figs. 3.25 and 3.26), the throughput increases with the retransmission stage limit because the collision probability decreases as the retransmission stage limit increases. However, the increase of the retransmission stage limit incurs the large back-off time, and resulting overhead increases. The results of our model are very accurate, but those of extended Bianchi’s model are not.

For RTS/CTS (Figs. 3.27 and 3.28), the normalized throughput reaches the maximum with smaller retransmission stage limit at RTS/CTS than at CSMA/CA because the frame size of RTS is too small. Both the results of extended Bianchi’s model and those of our models match with those of the simulation.
Figure 3.25: Normalized saturation throughput for CSMA/CA as a function of the retransmission stage limit at the frame size of 500 bytes and the payload data rate of 48Mbps.

Figure 3.26: Normalized saturation throughput for CSMA/CA as a function of the retransmission stage limit at the frame size of 1500 bytes and the payload data rate of 48Mbps.
Figure 3.27: Normalized saturation throughput for RTS/CTS as a function of the retransmission stage limit at the frame size of 500 bytes and the payload data rate of 48Mbps.

Figure 3.28: Normalized saturation throughput for RTS/CTS as a function of the retransmission stage limit at the frame size of 1500 bytes and the payload data rate of 48Mbps.
Chapter 4

IEEE 802.11 Saturation Throughput Analysis in the Presence of Hidden Terminals for the General Infrastructure Scenario

In Chapter 3 we presented an IEEE 802.11 saturation throughput model for a simple infrastructure scenario with two hidden stations. The model is valuable because it accurately models a hidden terminal scenario not previously modeled successfully; In this chapter, we will consider a more general scenario. Stations are always exposed to the access point, as the stations are connected to Internet through the access point in the infrastructure case. However, in general there may be more than two stations. Moreover, the stations can be irregularly exposed or hidden from each other due to obstacles, interference, transmission power, antenna gain, and location. In this chapter, we extend our enhanced model in Sec. 3.4.2 to accommodate the general infrastructure scenario with hidden stations, and then we evaluate this general model with the extensive simulation.
4.1 Analytical Model

In this section, we present a saturation throughput model for the general infrastructure scenario with hidden stations. We assume that stations always have frames to transmit to the access point, as we are interested in the saturation throughput. The access point does not transmit any data frame, only receives the frames from the stations and transmits ACKs to the stations. First, we present a saturation throughput model for the infrastructure scenario having many stations neatly separated into two groups in Sec. 4.1.1. In Section 4.1.2, we model the saturation throughput of the general infrastructure scenarios with a mixture of hidden and exposed stations using the model from Sec. 4.1.1.

4.1.1 The Hidden Terminal Scenario with Two Groups of Stations in Region A and Region B

In this section, we consider the infrastructure scenario with hidden stations shown in Fig. 4.1. We assume that all stations are in the communication range of the access point and that there are $n_a$ stations in region $A$, and $n_b$ stations in region $B$. Stations in the same region are exposed to each other, and stations are in different regions are hidden from each other. We call this scenario as the $n_a$-to-$n_b$ scenario.

We adjust our enhanced model in Sec. 3.4.2 twice to model the $n_a$-to-$n_b$ scenario. First, to account for the variable number of stations in each region, we adjust the contention
windows sizes in the calculation of the transition probability (i.e., $T$ and $T'$). We divide the size of the contention windows by the number of stations in each region because the probability of success for the $n_a$-to-$n_b$ scenario are reduced directly proportionally with the number of competing stations. We can simply account for the fact that there are $n_a$ stations in the region of $A$ by dividing $CW_{min}$ and $CW_{max}$ of station $A$ by $n_a$. Similarly, for the region of $B$ we divide $CW_{min}$ and $CW_{max}$ of station $B$ by $n_b$. These yield:

$$
\begin{align*}
CW'_a & = \min\left\{ \frac{CW_{max}}{n_a}, (\frac{CW_{min}}{n_a} + 1)2^i - 1 \right\}, \\
CW'_b & = \min\left\{ \frac{CW_{max}}{n_b}, (\frac{CW_{min}}{n_b} + 1)2^j - 1 \right\}.
\end{align*}
$$

(4.1)

Second, we use the minimum value instead of the maximum value in (3.28).

$$
T_c = \min\{0, \frac{T_v}{2} - \frac{t_{timeout} + DIFS}{\sigma}\}. 
$$

(4.2)

The enhanced model in Sec. 3.4.2 captured the dynamics of the one-to-one hidden station scenario well. However, the one-to-one scenario is fundamentally different from the $n_a$-to-$n_b$ scenarios ($n_a > 1$ or $n_b > 1$). The short-term unfairness problem occurs in one-to-one scenario, but it is reduced in the $n_a$-to-$n_b$ scenarios ($n_a > 1$ or $n_b > 1$). Moreover, we may need to consider the retransmission stages of all stations not only two when we calculate the transition probability of the $n_a$-to-$n_b$ scenarios. The use of the minimum value in (3.28) accounts for these differences. This is seemingly incredibly crude way to
account for multiple stations in each region, but as we will show in Sec. 4.2, it is reasonably accurate.

4.1.2 General Scenario

In this section, we consider the general infrastructure scenario with hidden stations shown in Fig. 4.2. In this scenario, \( n \) stations (e.g., 5 stations in Fig. 4.2) are randomly distributed in the communication range of the access point, and they are arbitrarily exposed or hidden from each other. We model this general scenario by using the results of the \( n_a \)-to-\( n_b \) scenario applied to each station. For example, station \( A \) in Fig. 4.2 has three exposed stations (\( n_a = 3 \) including itself) and two hidden stations (\( n_b = 2 \)), such that we can say that station \( A \) is in the 3-to-2 scenario. Similarly, station \( B \) is in a 4-to-1 scenario. Thus, we can model this general scenario from the perspective of each station using the results of the \( n_a \)-to-\( n_b \) scenario. For example, the scenario in Fig. 4.2 consists of two stations in 4-to-1 scenario (B and C) and three stations in 3-to-2 scenario (A, D and E). We can calculate the normalized saturation throughput of the general hidden terminal scenario by calculating the weighted average of the normalized throughput of each \( n_a \)-to-\( n_b \) scenario using our model in Sec. 4.1.1. Thus:

\[
S_g = \frac{\sum_{k=1}^{l} n_k S(n_{ak}, n_{bk})}{n},
\]

(4.3)

where \( S_g \) is the normalized saturation throughput of the general infrastructure scenario with hidden stations, \( n \) is the number of stations in the scenario, \( l \) is the number of different \( n_a \)-to-\( n_b \) scenarios in the general scenario, \( n_k \) is the number of stations in the \( k^{th} \) \( n_a \)-to-\( n_b \) scenario, and \( S(n_{ak}, n_{bk}) \) is the normalized saturation throughput of the \( k^{th} \) \( n_a \)-to-\( n_b \) scenario. For example, for the situation in Fig. 4.2: \( S_g = \frac{3S(4,1) + 2S(3,2)}{5} \).

4.2 Model Validation

4.2.1 Simulation Setup

We use the same IEEE 802.11 simulation model and mobility framework of OMNET++ as in Sec. 3.5. We consider the \( n_a \)-to-\( n_b \) scenario shown in Fig. 4.1 and the general scenario shown in Fig. 4.13.
4.2.2 Simulation Results and Analysis

The $n_a$-to-$n_b$ Scenario

Figure 4.3 shows the normalized saturation throughput for CSMA/CA as a function of the number of station for the hidden terminal scenario in Fig. 4.1 fixing $n_a=1$ and varying $n_b$ from one to ten. The normalized throughput is almost constant regardless of $n_b$, the number of stations in the region $B$. The increased number of exposed stations in the region $B$ increases the contention in region $B$, but as the number of stations in the region $B$ increases, the frame the station in the region $A$ transmits is easy to collide with the frame stations in the region $B$ transmits; in result, the station in the region $A$ stays in the large retransmission stage. In the simulation, the number of frames the station in the region $A$ transmits successfully decreases as the number of stations in the region $B$ increases. On the other word, the number of frames stations in the region $B$ transmit successfully increases as the number of stations in the region $B$ does. In result, the normalized throughput is almost constant regardless of $n_b$, the number of stations in the region $B$. Figure 4.4 shows the normalized saturation throughput for CSMA/CA as a function of the number of stations
Figure 4.4: Normalized saturation throughput as a function of $n_a$ and $n_b$ for CSMA/CA, the number of stations in both regions, at the payload data rate of 54Mbps and the frame size of 1500 bytes.

Figure 4.5: Normalized saturation throughput for $n_a=1$ as a function of $n_b$ for RTS/CTS, the number of stations in region $B$, at the payload data rate of 54Mbps and the frame size of 1500 bytes.
for the scenario in Fig. 4.1 when simultaneously varying both \( n_a \) and \( n_b \) from one to ten. In this scenario, the normalized throughput decreases as the number of stations increases because the increased number of exposed stations in both regions increases the contention, and the resulting collision probability increases. As shown in these figures, our model is very accurate.

For RTS/CTS (Figs. 4.5 and 4.6), the normalized system throughput is almost constant regardless of number of stations, and both our model and extended Bianchi’s model are accurate.

**General Scenario**

We consider two general scenarios for the validation of our generalized model. First, the scenario in Fig. 4.2 consists of two stations in 4-to-1 scenario (B and C), three stations in 3-to-2 scenario (A, D and E). Figures 4.7 through 4.10 shows the normalized saturation throughput as a function of the frame size CSMA/CA and RTS/CTS respectively for each \( n_a \)-to-\( n_b \) scenario in the general scenario in Figs. 4.2. Our model is very accurate.
Figure 4.7: Normalized saturation throughput as a function of frame size for CSMA/CA for 4-to-1 scenario.

Figure 4.8: Normalized saturation throughput as a function of frame size for CSMA/CA for 3-to-2 scenario.
Figure 4.9: Normalized saturation throughput as a function of frame size for RTS/CTS for 4-to-1 scenario.

Figure 4.10: Normalized saturation throughput as a function of frame size for RTS/CTS for 3-to-2 scenario.
for the various $n_a$-to-$n_b$ scenario for both CSMA/CA and RTS/CTS.

Figures 4.11 and 4.12 show the normalized saturation throughput for CSMA/CA and RTS/CTS as a function of the frame size for the general infrastructure scenario in Fig 4.2, where "Simulation mixed" is obtained by substituting simulation results not the model results for each $S(n_{ak}, n_{bk})$ of (4.3). The result of our model is very similar to the result of simulation mixed. There are some differences between the result of our model and that of the simulation because weighted average of the $n_a$-to-$n_b$ scenario cannot exactly capture the complex dynamics of the general scenario. However, as shown in these figures, our model is reasonably accurate for both CSMA/CA and RTS/CTS, while extended Bianchi’s model is accurate only for RTS/CTS.

Second, we consider the scenario in Fig. 4.13, which consists of two stations in 5-to-5 scenario (C and J), one station in 7-to-3 scenario (F), two stations in 8-to-2 scenario (B and H) and five stations in 9-to-1 scenario (A, D, E, G, and I).

Figures 4.14 and 4.15 show the normalized saturation throughput for CSMA/CA and RTS/CTS as a function of the frame size for the general infrastructure scenario in Fig 4.13. As shown in these figures, our model is reasonably accurate for both CSMA/CA and
Figure 4.12: Normalized saturation throughput as a function of frame size for RTS/CTS for the general scenario in Fig. 4.2

Figure 4.13: Example of general infrastructure scenario, where there are 10 stations in the communication range of the access point, and the stations may irregularly be exposed or hidden from each other.
Figure 4.14: Normalized saturation throughput as a function of frame size for CSMA/CA for the general scenario in Fig. 4.13

Figure 4.15: Normalized saturation throughput as a function of frame size for RTS/CTS for the general scenario in Fig. 4.13
RTS/CTS, while extended Bianchi’s model is accurate only for RTS/CTS.
Chapter 5

Conclusion

In this dissertation, we design and analyze the Medium Access Control (MAC) protocol for two popular wireless networks: Wireless Sensor Networks (WSNs) and Wireless Local Area Networks (WLANs).

In chapter 2, we present an asynchronous, scheduled, energy-efficient MAC protocol; in the proposed approach, each node stores the wakeup schedules of their neighbors. The protocol asynchronously coordinates the wakeup times of neighboring nodes to reduce overhearing, contention and delay, unavoidable in synchronous scheduled MAC protocols. We present a multi-hop energy consumption model for the proposed protocol and compare its performance with SCP-MAC. To validate the design and the energy model, we implement AS-MAC in TinyOS on MICAz and TELOSB platforms equipped with Chipcon CC2420 radio. The experimental results shows that AS-MAC considerably reduces energy consumption while providing good delay and packet loss in comparison with other energy efficient MAC protocol and that the energy model is a fairly good approximation of the real energy consumption of the proposed MAC layer.

In chapter 3, we presented a saturation throughput model for IEEE 802.11 in the presence of hidden terminal for the infrastructure case with two hidden stations. In this scenario, a short-term unfairness problem occurs, and as a result the collision probability increases as a function of the retransmission stage. This phenomenon is not captured by existing throughput models. We present a new Markov chain model that accounts for the increase of the collision probability as a function of the retransmission stage. Simulation results show that our model is extremely accurate in a wide variety of cases.
In chapter 4, we consider the general case with a mix of hidden and exposed terminals. In this chapter we extend the results of chapter 3 first to a scenario neatly differentiating the stations into two groups, and then to the more general case. Simulation results show that, while not perfect, our model provides a reasonable approximation for the behavior of an extremely complex system.
Bibliography


