

**Performance Evaluation of a Binary Exponential Code Backoff Algorithm for
IEEE 802.11**

by

Kang Sun

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Auburn, Alabama
May 3, 2014

Keywords: MAC, CDMA, WiFi, Multi-hop

Copyright 2014 by Kang Sun

Approved by

Saad Biaz, Associate Professor of Computer Science and Software Engineering
Xiao Qin, Associate Professor of Computer Science and Software Engineering
David Umphress, Associate Professor of Computer Science and Software Engineering
Weikuan Yu, Associate Professor of Computer Science and Software Engineering

Abstract

IEEE 802.11 is the most widely deployed medium access control (**MAC**) protocol for wireless access. Simple and efficient, this MAC protocol works well as the last wireless hop and is a key enabler of the widespread success of WiFi. Unfortunately, despite over twenty years of advances in routing algorithms for multi-hop networks, there are no widespread commercial multi-hop networks because IEEE 802.11 delivers poor throughput and fairness on multi-hop networks. This poor performance is due to the hidden terminal often presents on multi-hop networks. In order to address this issue, researchers have proposed numerous new approaches. However, none works well enough to develop commercial multi-hop networks. This work proposes a dynamic hybrid MAC protocol dubbed the Binary Exponential Code (**BEC**) that adapts to the traffic: in presence of the hidden terminal problem, the MAC assigns reserved CDMA-like channels, otherwise it works like the a contention based protocol (e.g., IEEE 802.11). The scheme is dynamic because the CDMA-like channels support variable bit rates. Formal modeling and simulations are used to validate and evaluate the proposed BEC protocol. The simulation results and mathematical analyses confirm the throughput and fairness improvements achieved by BEC on a variety of topologies and traffic patterns. The network overall throughput is increased by 20% - 40% for UDP applications.

Acknowledgments

I would like to express my sincere appreciation and thanks to my advisor, Dr. Saad Biaz, for your significant support of my study and research, for your guidance, patience and understanding. Your mentorship was paramount in my master study. You always encouraged my research and helped me to become a research scientist. I would also like to thank you for the priceless advice on both my research and my career.

Except for my advisor, I would like to thank my committee members, Dr. Xiao Qin, Dr. David Umphress and Dr. Weikuan Yu, for serving as my committee members as well as your encouragement and insightful suggestions for my thesis.

I also thank my fellow labmate, Jagadeesh Balasubramani, for the valuable discussions about this project we have made and for the exciting days and nights we were working together before.

At last, a special thank goes to my family members. Without the tremendous support and trust from my parents, Caiyue Sun and Dongju Zhang, I am unable to continue my study and pursue a master degree at Auburn University. Thanks for your spiritual encouragement throughout my life.

Table of Contents

Abstract	ii
Acknowledgments	iii
List of Figures	vi
List of Tables	viii
1 Introduction	1
2 Background	5
2.1 Problem Description	5
2.1.1 A Multi-hop TCP connection	6
2.1.2 Performance Degradation as Bit rates Increase	7
2.2 IEEE 802.11	8
2.3 Related Work	12
2.3.1 Contention-based	13
2.3.2 Combining Allocation-based and Contention-based Approaches	16
3 Proposed Solution	20
3.1 Basic Idea	21
3.1.1 Code Division Multiple Access	21
3.1.2 BEC: a Dynamic CDMA	22
3.2 Features of BEC	23
3.3 Implementation	24
4 Evaluation	27
4.1 Simulation	27
4.2 Analytical Model	29
4.2.1 Markov Chain Model for IEEE 802.11	29

4.2.2	Novel Model for BEC	31
5	Result	35
5.1	Verification for Analytical Model	38
5.2	UDP	39
5.3	TCP Traffic	43
5.4	Discussion and Future Work	46
	Bibliography	47

List of Figures

1.1	Two Senders, One Receiver with a Hidden Terminal Problem	2
2.1	Chain Multi-hop Network with TCP Traffic	6
2.2	Basic Mechanism of DCF	10
2.3	Example for RTS/CTS mode	11
2.4	Virtual Channel Sense	12
3.1	The State Transition of Allocating Bandwidth	26
4.1	Markov Model for IEEE 802.11	29
4.2	State Transition Diagram	32
5.1	Two senders, two receivers Topology	36
5.2	Four senders, one receiver Topology	36
5.3	Chain Topology	36
5.4	Comparisons between Analytical Model and Simulation Result	39
5.5	Two Senders, Two Receivers	40
5.6	Four Senders, One Receiver	41
5.7	Chain	42

5.8	UDP Multi-hop Topology with One Flow	43
5.9	Two Senders, Two Receivers	44
5.10	Four Senders, One Receiver	44
5.11	Two senders, One receiver: Overall Throughput	45
5.12	Chain	45

List of Tables

2.1	802.11 versions	9
5.1	Simulation Configuration	35

Chapter 1

Introduction

Wireless local area networks (**WLAN**) have proliferated almost everywhere from homes, campuses, companies, and airports to restaurants and farms. Simplicity and good performance allowed this expansion in the last decade. Most *WLAN* use the IEEE 802.11 set of networking protocols. These protocols achieve good performance in wireless network while maintaining simplicity. Despite occasional high bit error rates due to unstable links, bit rates as high as 150 Mbps are currently achieved using IEEE 802.11 protocols. However, the IEEE 802.11 *MAC* protocol works well only on the last wireless hop in which only the link between the access point and the user is wireless while the remaining is still wired.

When it comes to **multi-hop** networks, no commercial wireless network has yet been widely deployed. Despite two decades of research on wireless routing and *MAC* protocols, the network performance and channel utilization still remain low in wireless multi-hop networks. Even when nodes are static (e.g., mesh networks), multi-hop networks fail to deliver a performance similar to that enjoyed on last hop wireless networks: the throughput dramatically decreases beyond two wireless hops and greatly fluctuates based on the traffic and the position of the nodes. Although IEEE 802.11 provides the support for an ad hoc mode in multi-hop case, it performs so poorly that this mode is rarely used.

Many studies have researched the performance of IEEE 802.11 in ad hoc mode and pointed out the cause for its bad performance. When all competing nodes hear each other, the performance is remarkable. Otherwise, nodes cannot efficiently coordinate their accesses. The hidden terminal problem discussed in [11] is a major performance killer in IEEE 802.11 *WLAN*. Just as shown in Figure 1.1, Sender 1 cannot hear Sender 2 and would send frames to the same receiver. Since, they cannot hear each other, they cannot coordinate their accesses.

Under such circumstances, the IEEE 802.11 *MAC* protocol works just as the slotted Aloha *MAC* mechanism in which the channel utilization will not exceed 36%. As a result, it leads to many consecutive collisions with a serious performance degradation on *WLAN*.

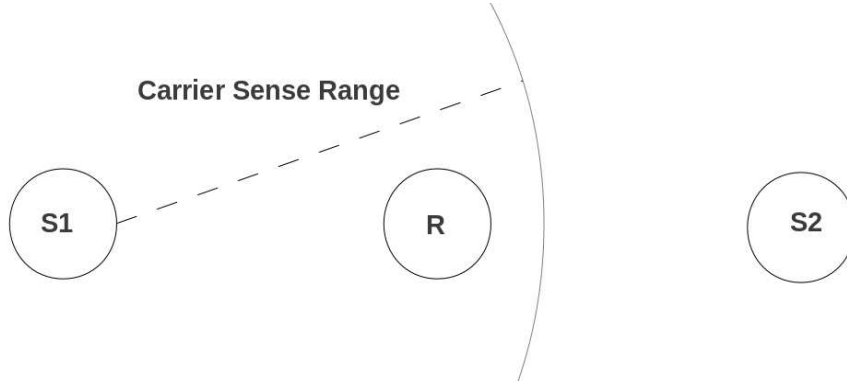


Figure 1.1: Two Senders, One Receiver with a Hidden Terminal Problem

In order to solve the hidden terminal problem, a modified mechanism of Distributed Coordination Function (**DCF**) using Request to Send (**RTS**) and Clear to Send (**CTS**) control messages was proposed and implemented in IEEE 802.11. In this approach, the sender must send a *RTS* message before the data packet transmission. Once the *RTS* message reaches the receiver, the receiver would respond with a *CTS* message (if there is no competing traffic around it). Fortunately, all other hidden terminals within the transmission range of the same receiver could receive this *CTS* message which would set Network Allocation Vector (**NAV**) for themselves on the basis of time information it extracts from *CTS* message. Therefore, all hidden terminals remain silent while the original sender is transmitting packets, which addresses this drawback somewhat in IEEE 802.11. However, this approach suffers a key weakness. In real scenarios, short data packets are more common than long data ones. Compared to the transmission time consumed for each short data packet, the overhead by transmitting *CTS/RTS* messages is too large. As reported in [29], the use of *RTS/CTS* decreases the overall throughput from 17 Mbps to 13.5 Mbps while the bandwidth efficiency crumbles down to 25% from 32%. The throughput is diminished by 20.58% as the overhead

for *CTS/RTS* packets. It even worsens when a packet size smaller than 500 bytes. Due to this large overhead, IEEE 802.11 uses *CTS/RTS* only for frames larger than 2347 bytes.

Since the contention based IEEE 802.11 *MAC* protocol does not work well with hidden terminals, another approach is to use collision free (through reservation/allocation) *MAC*. Several researchers tried a hybrid approach that combines the contention-based approach with allocation-based to address the hidden terminal problem. For example, Z-MAC proposed in [24] combines Time Division Multiple Access(**TDMA**) and Carrier Sense Multiple Access (**CSMA**). Instead of a fixed time slot assignment, Z-MAC is a hybrid dynamic scheme in which each non-owner of a time slot could steal an owned slot if its owner has been silent for a *long* period. Unfortunately, an accurate mechanism to discover active neighbors (competitors) is not provided by Z-MAC. Moreover, *TDMA* relies on synchronization that may reveal hard especially in presence of hidden terminals. Such a mechanism is critical for the success of Z-MAC. [25] proposed a *CDMA-based CSMA* protocol which combines Code Division Multiple Access (**CDMA**) with *RTS/CTS*. The idea is to send outband *RTS/CTS*. Codes for *CTS/RTS* messages and data frames are different. However, the code assignment for *CDMA* is always set in advance before it comes to a specific network traffic pattern. That is to say that it faces a challenge of a predetermined number of individual codes which might not work for all traffic patterns.

This work presents a new hybrid *MAC* scheme that addresses the weaknesses of the two above approaches and combines *CSMA* and a dynamic *CDMA*. When nodes hear each other, *CSMA* is used and when problems arise (due to hidden terminals for example), a dynamic *CDMA* is used. This scheme is dynamic because as collisions persist, the size of the *CDMA* codes increases therefore lowering the data rate for each node. The idea is that instead of using the binary exponential backoff mechanism in time, a Binary Exponential Code (**BEC**) is used. Instead of the backoff timer, the code size increase implemented in *BEC* aims to resolve collisions. As long as the codes size double perfectly and remain pairwise orthogonal with each other, all frames can be transmitted simultaneously and, hence, collisions would

not occur if there are enough codes (allocated channels). As collisions persist, data rates for all nodes decrease. Furthermore, since the number of active stations keeps changing, a fixed and static code division mechanism cannot accommodate a variable traffic. Therefore, in *BEC*, the code division is designed to adapt to dynamic conditions. As far as we know, this is the first research of allocating the orthogonal codes dynamically to avoid collisions on the basis of IEEE 802.11 protocol, which improves the overall performance of *WLAN* ideally, especially for the multi-hop scenario.

In order to evaluate the effectiveness of *BEC*, *BEC* was implemented in the *ns-2* simulator. In addition, a Markov-based analytical model was developed to formally evaluate *BEC* and validate the simulation results. Based on simulation and formal analysis, *BEC* performs better by improving the overall throughput and fairness. The overall throughput is increased by 20% - 40% in UDP applications with the hidden terminal problem. For TCP applications in four senders and one receiver topology, the Jain's Fairness Index in IEEE 802.11 is decreased to 0.71 approximately while, in *BEC*, it is always 1.

This work is organized as follows. First, Chapter 2 1) presents the problem of the IEEE 802.11 *MAC* protocol in the ad hoc mode, 2) introduces the background of IEEE 802.11, and 3) overviews past related work. Chapter 3 discusses the proposed basic idea of *BEC*, its appealing features, and its implementation issues. The simulation and the analytical model used to evaluate *BEC* performance are described in Chapter 4. At last, simulation results obtained by the *ns-2* simulator are presented and compared with the formal model in Chapter 5.

Chapter 2

Background

As mentioned earlier, IEEE 802.11 does not work well because of the hidden terminal problem. This issue will be carefully analyzed here using one TCP connection on a multi-hop network. Subsequently, since our approach is mostly based on the original IEEE 802.11 protocol, this protocol will be briefly described. Later, many papers which have proposed approaches to address the hidden terminal problem are reviewed and discussed.

2.1 Problem Description

Wireless **one-hop** networks are today widely deployed: for most applications, a user experiences a connectivity quality similar to the one on a wired desktop. However, while this one-hop wireless networks such as WiFi proliferated everywhere from homes, companies, airports, factories, restaurants, and even farms, multi-hop or mesh networks did not even take off despite critical and appealing features such as rapid deployment and self organization. After two decades of research in wireless routing and *MAC* protocols, the network performance and channel utilization still remain low in wireless multi-hop networks. The performance in multi-hop networks dramatically drops beyond two hops and wildly fluctuates depending on the position of the node/user, the dynamic channel quality, and the traffic load. Furthermore, the author's research group experienced first hand this poor performance on a 23-node mesh network they deployed at Auburn University and evaluated in the last five years [9]. The research community has been successful in designing efficient transport and routing protocols for wireless networking, but the performance and utilization in fact stumble on under performing *MAC* layer protocols.

The poor performance of IEEE 802.11 over multi-hop networks can be showed through two aspects: 1) One TCP connection over a chain multi-hop network is considered and it will show the contention that arises for only one TCP connection. 2) In addition, the binary exponential back off algorithm is stable and performs quite well as long as nodes hear each other. However, it is not optimal and incurs an overhead that increases as bit rates increase.

2.1.1 A Multi-hop TCP connection

Just **one** TCP connection is considered in the following. All conflicts that arise for one TCP connection on a simple chain topology are enumerated. Furthermore, the hidden terminal problem will likely arise on multiple hops depending on the distances between nodes.

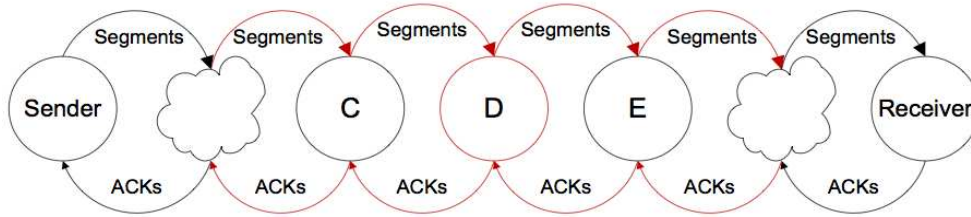


Figure 2.1: Chain Multi-hop Network with TCP Traffic

On wireless networks, sensing is often helpless at the sender: it is at the receiver that interference is relevant and destructive. Consider Nodes C and E on Figure 2.1 trying to communicate with Node D . If Nodes C and E are out of range from each other, they will not hear their transmissions to Node D . Therefore, Nodes C and E cannot sense not only their mutual transmissions, but also their mutual collisions. Nodes C and E will essentially behave like slotted Aloha [4] whereas terminals transmit whenever they have a frame to send, but with longer timeouts due to the binary exponential backoff. Therefore, the hidden terminal problem may often arise in wireless multi-hop networks. Researchers have well documented the impact of the hidden terminal problem on the performance of a chain [2, 3, 5, 6, 7]: consider just one TCP connection T flowing from left to right through intermediary nodes C , D , E . The following will enumerate the competing (conflicting)

frames sent to Node D : 1) TCP segments from C to D , 2) TCP acknowledgements from C to B , 3) TCP acknowledgements from E to D , 4) TCP segments from E to F , 5) TCP segments from D to E , and finally 6) TCP acknowledgments from D to C . These competing frames encompass all nodes between the two clouds on Figure 2.1. The TCP connection T uses Node D with Nodes C and E being out of range of each other, therefore unable to coordinate their transmissions. Since Nodes C and E cannot coordinate their transmissions (TCP segments and acknowledgements) to Node D , the TCP connection T will yield a poor performance. The author's research group observed that over 90% of TCP connections over just three hops to browse popular web sites never complete and reset. Beyond four hops, a user just cannot browse even a server on campus. This poor performance is due to contention similar to what was illustrated at Node D .

Now, the issue with the temporal binary exponential back off is considered. Whatever is the cause of a failure, the resulting temporal binary backoff incurs large delays and jitters, low throughputs, and even service interruptions on a multi-hop networks.

2.1.2 Performance Degradation as Bit rates Increase

A collision usually results from one of two causes: 1) two stations in the same carrier sense zone occasionally pick the same backoff slot to transmit, or 2) two stations do not hear each other and cannot coordinate their transmissions.

In the two phases of the *MAC* protocol, the channel contention constitutes an overhead to the frame transmission. Recent Magistretti's research work [8] revealed that 1) the temporal binary exponential backoff accounts for almost 50% of the overall communication overhead, and 2) the backoff is inefficient when using channel at high bit rates: this overhead reaches 45% at 54 Mbps, 82% at 300 Mbps, and up to 91% at 600 Mbps. This inefficiency can be easily understood from an illustration as follows.

Let us define utilization as $u = \frac{t}{T+t}$ where t represents the time in transmitting a frame and T refers to the time of protocol overhead. In current IEEE 802.11 standard, T consists

mainly of the backoff process, the physical layer preamble, and channel piloting. In T , the backoff process is based on a number of time slots and the physical preamble is always sent at the base rate of 6 Mbps. The duration of a slot is constant regardless of the bit rate. Therefore, T does not vary much because it is dominated by the preamble transmission and the backoff procedure (constant time slot) while t decreases when the frame transmission rate increases. For example, if the bit rate increases from 100 Mbps to 1 Gbps, then the channel utilization at 1 Gbps becomes $u = \frac{0.1t}{T+0.1t} = \frac{t}{10T+t}$, much worse than $\frac{t}{T+t}$ on 100 Mbps networks. The channel utilization is significantly hurt at high bit rates because the time spent on transmitting data frames drastically decreases with high bit rates while the time in protocol overhead remains almost constant (regardless of transmission bit rate).

It is reasonable to expect that physical bit rates will further increase with more efficient physical layer schemes and much larger bandwidths at high frequency spectrum such as 60 GHz. As a result, the inefficiency incurred by the temporal binary exponential backoff will further worsen.

2.2 IEEE 802.11

As a matter of fact, IEEE 802.11 refers to a set of physical and link layers protocols proposed by IEEE LAN/MAN Standards Committee to work over wireless networks. Table 2.1 shows some examples of IEEE 802.11 protocols [22]. These protocols differ in the physical layer. They use different physical technologies, such as the modulation approach and the frequency. They therefore can support different data rates. Notice that the unit for frequency is GHz while, for bandwidth, it is MHz and finally Mbps is unit for data rate in Table 2.1. The protocol standards in all other upper layers including the *MAC* layer are mostly the same among each version of IEEE 802.11.

At first, it is worth knowing the function of the *MAC* layer before delving the details. Typically a channel is always shared by several contending stations. Guaranteeing the equal opportunity of seizing the channel for each station is the objective of a *MAC* layer

Table 2.1: 802.11 versions

Version	Modulation	Freq.	Bw.	Datarate
a	OFDM	5 or 3.7	20	6, 9, 12, 18, 24, 36, 48, 54
b	DSSS	2.4	20	1, 2, 5.5, 11
g	OFDM, DSSS	2.4	20	6, 9, 12, 18, 24, 36, 48, 54
N	OFDM	2.4 or 5	40	15, 30, 45, 60, 90, 120, 135, 150

in networking architecture. Usually *MAC* protocols execute two sequential phases: first, the *channel contention phase* where stations with ready-to-transmit traffic contend for the wireless channel to transmit, and, second, the *frame transmission phase* where the winning station transmits its frame(s) while others remain silent. As a conclusion, the major task of a *MAC* protocol is to manage the distributed channel contention.

In Ethernet (wired network) protocol, a station would start a transmission as soon as the ether becomes idle. Within the first 64 bytes, if a noise burst is not detected, the whole frame is guaranteed to be transmitted successfully. Unfortunately, such a scheme does not apply to a wireless situation. Due to the hidden terminal problem introduced in Chapter 1, a station is unable to know what really happens in the channel. The station may falsely conclude that the channel is idle while, in fact, a hidden station is just transmitting frames at that time. Add to this that in general a wireless station cannot send and receive simultaneously. As a result, the collision detection scheme in Ethernet fails on wireless channel.

To address such an issue, two modes are well designed to be supported in IEEE 802.11: *DCF* and Point Coordination Function (**PCF**). The main difference between these two modes is the determination of the station permitted to transmit: the base station or the distributed stations. *PCF* has a centralized architecture, which means that each station has to gain the allowance of a transmission from the base station. On the contrary, in *DCF*, each station would decide the moment to transmit frames by itself.

CSMA with Collision Avoidance (**CSMA/CA**) is the protocol used in *DCF*. There are actually two versions of *CSMA/CA* protocol. The basic version does not take the hidden/exposed terminal problems into account. It is summarized in Figure 2.2 and works as follows.

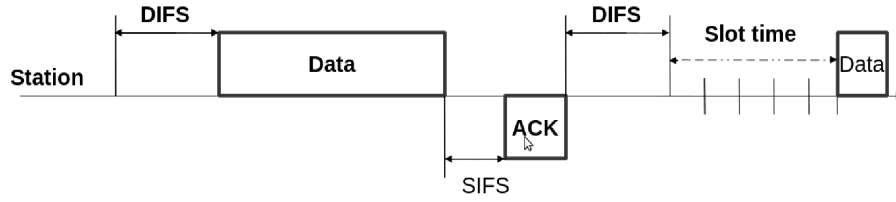


Figure 2.2: Basic Mechanism of DCF

When a station has some frames ready to be sent, it must first sense the channel. Suppose no other signal is detected on the channel, the station would wait for additional Distributed InterFrame Space (**DIFS**) and start transmitting later. Unlike Ethernet, the station does not need to keep monitoring the channel while transmitting frames. Actually the reliability of the transmission in *DCF* cannot be guaranteed by the same mechanism used in Ethernet. The interference at the receiver may not be detectable by the sender. Thus sensing channel while transmitting is not useful for the success of a transmission. A *MAC* acknowledgment (**ACK**) frame will be sent by the receiver after an interval of Short InterFrame Space (**SIFS**) with an idle channel for each successful transmission.

In IEEE 802.11, to adapt to the traffic patterns, an algorithm called Binary Exponential Backoff (**BEB**) is implemented and always triggered by every collision [1]. It works as follows. *BEB* would select a random number of time slots to wait before retransmission after a collision. For instance, after the first collision, a sender *S* picks a random number n_0 within a range (called *contention window*) W_0 . After n_0 time slots are sensed idle, the sender *S* is allowed to retransmit. However, if the retransmission fails again, the contention window would be doubled resulting in $W_1 = 2 \times W_0$ and the sender picks a new random number n_1 within the doubled range W_1 . As collisions persist, the contention window will continuously increase exponentially and the sender *S* will likely draw large random numbers and thus wait longer. As the number of contending senders increases, the contention window increases such that the probability of two senders drawing the same random number decreases. While not optimal, *BEB* is stable and delivers a remarkable performance as long as all nodes hear each other: recall that a sender can retransmit after sensing n_i idle slots.

Therefore, if an *ACK* does not return, a timeout in the sender side implies a collision in the receiver side. The sender would wait for a random period of time based on *BEB algorithm* introduced above. Then the retransmission scheme begins to work. Somewhat like the story in TCP, the feedback from the receiver side is the key factor to maintain the reliability in *DCF* mode. On the other hand, if the channel is sensed busy, the sender would defer itself until an idle channel appears again. However, just like nonpersistent *CSMA*, the sender does not attempt to seize the channel greedily, but waits for a random period of time before the next sensing. Although such a scheme leads to a little longer delay, the high channel utilization obtained pays it back.

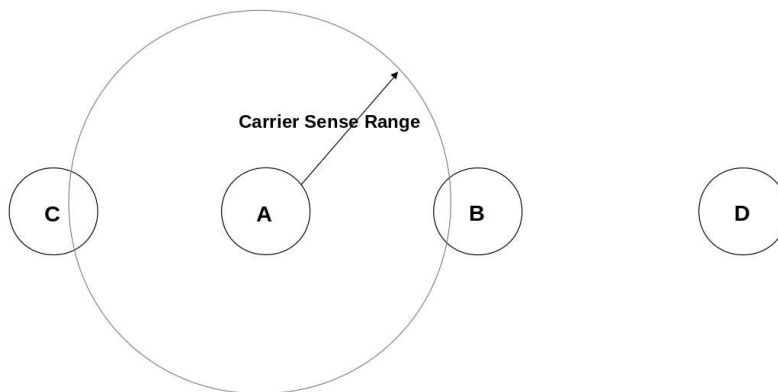


Figure 2.3: Example for RTS/CTS mode

The other version of *CSMA/CA* is based on Multiple Access with Collision Avoidance for Wireless (**MACAW**) which introduces *RTS/CTS* frames. Besides the physical channel sensing mechanism, an additional virtual channel sensing scheme is added in this version. It is achieved through two types of control frames, *RTS* and *CTS*. Here is an example to explain this approach. In Figure 2.3, four stations, *A*, *B*, *C* and *D*, are on the same straight line. In this example, *A* must send frames to *B*. *C* is also within the transmission range of *A* while *D* is not. At the same time, *D* is within the range of *B*. Before *A* sends a data frame, it needs first to send an *RTS* frame to *B* to ask for permission. When *B* receives the *RTS* frame, *C* also gets it. Since the *RTS* frame contains the information of the duration that *A* would use

the channel, *C* right now can set a *NAV* and remain silent during this duration. Then *B* sends a *CTS* frame back to *A*, which informs *A* of starting the transmission. Since *D* is also within the range of *B*, *D* also receives the *CTS* frame and desists from transmitting anything for an amount of time indicated in the *CTS* frame. The hidden terminal *D* now cannot send frames to *B* and interfere the transmission in *B* by such a virtual channel sensing scheme depicted in Figure 2.4. Then *A* starts sending frames to *B* in a collision-free channel. The rest part of process is similar to the basic version of *CSMA/CA*. The shortcoming in this *DCF* mode is a significant amount of time overhead consumed by the *RTS/CTS* control frames. For instance, it takes approximately 60 μ s to finish the transmission of an *ACK* frame which only indicates 1-bit relevant information. It is worth noting that 3240 bits of frame could be transmitted too within 60 μ s in IEEE 802.11g.

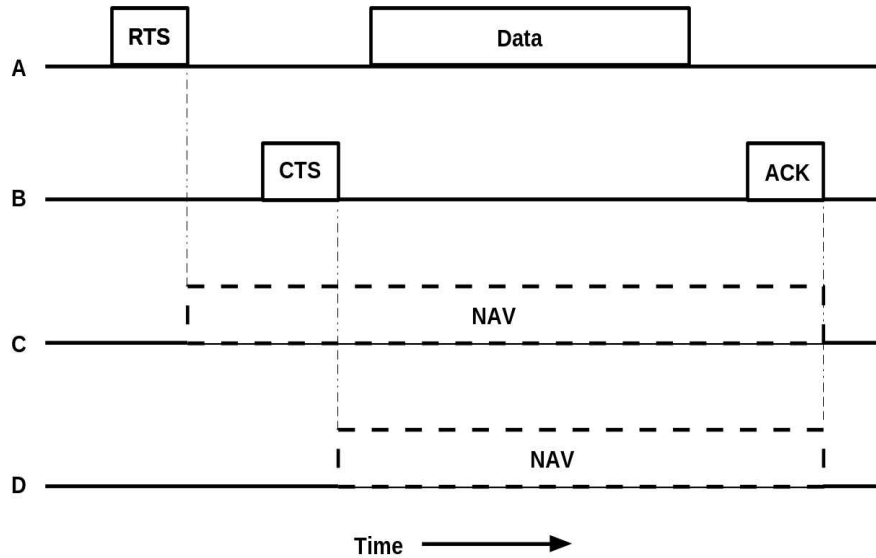


Figure 2.4: Virtual Channel Sense

2.3 Related Work

To overcome the drawbacks the IEEE 802.11 *MAC* protocol, many researchers proposed a variety of approaches. These approaches can be categorized into three different groups:

contention-based, allocation-based and the combination of the above two. In contention-based approaches, the bandwidth cannot be shared by multiple active stations at the same time and hence an efficient algorithm determining the contention mechanism is the key factor to avoid collisions. [12, 13, 19] belongs to contention-based category. The allocation-based methods intend to divide and allocate the resource - namely a bandwidth, a frequency, or a time slot, to each station dynamically, to avoid the contending phase. The stand-alone *TDMA*, Frequency Division Multiple Access (**FDMA**) or *CDMA* belong to this category. Finally, the combination of the above two approaches attempt to exploit the advantages of each one while diffusing their limitations. The schemes introduced in [27, 24, 20, 26, 21, 25] belong belong to this category.

2.3.1 Contention-based

Marek Natkaniec and Andrzej R. Pach pointed out the impact of different values of CW_{min} and CW_{max} on the throughput and packet delay in [12]. As explained earlier, CW stands for contention window. The IEEE 802.11 standards specify protocols for three different physical technologies: Frequency Hopping Spread Spectrum (**FHSS**), Direct Sequence Spread Spectrum (**DSSS**) and Infrared (**IR**). The values of the pair (CW_{min}, CW_{max}) are set to (16, 1024), (32, 1024) and (64, 1024), respectively. CW parameters are fixed and designed to be applied to all different situations. But, such a scheme does not consider the dependencies between CW parameters and the number of active stations. The improper selection of initial CW values leads to both a degradation of the overall throughput by 60% and a considerable increase of the average packet delay. Thus determining a proper value of CW_{min} and CW_{max} is a crucial factor to achieve a good performance in IEEE 802.11.

Several relationships between the networking performance and the value of CW parameters are observed and concluded in [12]. A small value of CW_{min} is proper for a topology with few nodes, whereas a large value of CW_{min} is suitable for one with many nodes. On the other hand, using a too small value of CW_{max} for a topology with heavy workload brings

many collisions. For example, assume that there are 150 nodes in a topology and most of them are busy with transmitting frames while the value of CW_{max} is only 128. Whatever the backoff time slot is selected from this CW_{max} , collisions cannot be avoided. Therefore, setting a reasonable value for CW_{min} and CW_{max} is one potential method to avoid the degradation in a heavy workload case. However, it is hard to know the proper large value for the so-called heavy workload. First, no efficient way to measure the workload is provided in this paper. Second, both 100 nodes and 1000 nodes belong with the category of many. There is a non-linear relationship between the number of nodes and the value of CW_{min} and CW_{max} . It is difficult to establish this relationship. Consequently, the approach of selecting the proper value of CW_{min} and CW_{max} is not a practical solution.

L. Bononi, M. Contih and E. Gregori proposed an approach called Asymptotically Optimal Backoff Algorithm (**AOB**) in [13], which dynamically updates the value of contention window size CW based on the current workload. This approach does not require any modification to the original IEEE 802.11 protocol or hardware. In this approach, two types of statistics are used for measuring network workload: the slot utilization and the average size of transmitted frames. First, it is indicated in [14] that, given the average length of frames, there is an optimal value for the maximum channel utilization. A two-dimension Markov chain model is applied to *DCF* mode of IEEE 802.11 protocol and helps predict the saturation channel utilization. Based on the model, the protocol capacity is found to be independent of network conditions, such as the number of active stations.

In addition, in order to get close to the optimal value, an additional transmission control mechanism is added before a frame is allowed to be sent by the backoff algorithm. It makes sense that, when the current utilization exceeds the maximum theoretical throughput, frames would collide. Thus, as long as the current channel utilization measured by the slot utilization tends to exceed the optimal value, this transmission must be postponed, therefore avoiding a future collision. The simulations results indicate that, under the heavy traffic workload, *AOB* performs significantly better than IEEE 802.11. But, under a light traffic workload,

there is no obvious difference between the performance of *AOB* and that of IEEE 802.11. When the hidden terminal problem arises, the performance of the network still remains fairly low in *AOB*.

Rather than the binary exponential backoff algorithm, Zuyuan Fang, Brahim Bensaou and Yu Wang implemented a measurement-based backoff algorithm in the [19]. A fairness problem exists in *BEB* algorithm as a result of the fact that some stations always have a lower opportunity of accessing the channel than others. Usually *BEB* algorithm tends to grant the permission to the last station that does a successful transmission. It is because CW would always be reset to CW_{min} after a success while CW s of other stations still remain large. It becomes even worse when the density of active stations increases. This fairness issue will be addressed elegantly in this measurement-based backoff approach.

The size of CW is adjusted dynamically. And the level of CW is determined by a factor called the fairness index. The CW of a station can be calculated by another two parameters: the throughput itself and the aggregate throughput of all other nodes it competes with. One major problem is how to calculate the aggregate throughput of others. That is to say that how to determine the overall number of competing nodes and allocate the fairness index fairly. With the existence of the hidden terminal problem, it is not practical to collect this type of information through intercommunication. The solution in this approach is to regard all other competing stations as a single entity. For that case, it can easily arrange 0.5 for itself and another half for that single entity. When each station behaves in this way, it is likely that each station would achieve the same fairness index.

Subsequently, when a node finds out that its estimated throughput is close to its allocated fair share, the value of CW doubles until the CW_{max} . As a consequence, other neighboring nodes have a higher chance to access the channel and hence it keeps the balance of each node's throughput. Otherwise, the value of CW would halve continuously until

CW_{min} . Though the fairness issue is almost solved in this approach, it is obtained by a complicated algorithm to compute fairness index which adds overhead to the processing time in each node.

To solve the hidden terminal problem, F. A. Tobagi and L. Kleinrock designed another interesting approach called Busy-Tone Multiple Access (**BTMA**) in [27]. In *BTMA*, the overall bandwidth is divided into two parts: the message channel and the busy-tone channel. The idea is similar to *CTS/RTS* mechanism. Once a node detects the signal in the message channel which implies an incoming frame, the node transmits another signal into the busy-tone channel continually. For the "hidden" node, it is required to detect the busy-tone channel first before transmitting frames. Because of the busy-tone channel, the "hidden" nodes are disabled to send any frame when the message channel is busy. On the other hand, when a frame is ready to be sent, the sender needs to sense the busy-tone channel for an amount of detection time. After that, if the busy-tone channel still implies a free channel, the sender starts the transmission; otherwise, this frame would be rescheduled to send it after a random amount of time. In case of a collision, a retransmission is triggered later by the absence of an *ACK* without adding much delay overhead.

However, there are several drawbacks in *BTMA* approach. First, in *CSMA*, since the entire bandwidth is used by the data transmission, the detection of the presence of a signal in the channel is not a big problem. Unfortunately, in *BTMA*, the sine wave busy tone signal is transmitted on a narrow-band channel, which makes the detection time non negligible. Second, the paper ignores the issue of fairness in this approach. It is indicated that, if one sender seizes the channel and transmits frames continually, other potential "competitors" which are the actual hidden terminals are less likely to access the channel.

2.3.2 Combining Allocation-based and Contention-based Approaches

In typical *TDMA*, each sender is assigned its own reserved time slot which guarantees that no two frames are sent during the same time slot. Although there is no extra message

overhead as *RTS/CTS* approach, *TDMA* has several non-trivial limitations. First, an efficient time schedule cannot be obtained without a centralized node in this approach. Second, the clock synchronization in *TDMA* consumes a large amount of energy. Third, since channel conditions are updated dynamically as well as interference ranges, it is impossible to develop a fixed time schedule which works well all the time. Last, when it comes to low contention, only a part of channel is used and the packet delay considerably increases. Thus, a stand-alone *TDMA* is not an ideal solution to address the hidden terminal problem issue. Injong Rhee *et al.* in [24] proposed a new hybrid *MAC* scheme called Z-MAC, in which the advantages of *TDMA* and *CSMA* are combined while their drawbacks are eliminated. Under the light workload, Z-MAC behaves like *CSMA*. When the workload increases, it turns into *TDMA*. The time slot assignment is done by an efficient scalable algorithm called Distributed Randomized TDMA Scheduling For Wireless Ad-hoc Networks (**DRAND**). Each node is an owner of a time slot and the others are the non-owners. In Z-MAC, the time slot is not exclusive to the owner. It means that non-owners still have a chance of stealing that time slot. However, the owner is granted with a priority by adjusting the initial *CW* size. As a result, as long as the owners are ready to send frames, non-owners are barred to transmit any frame. On the other hand, non-owners will steal the time slot if the owner remains silent for a long duration. The adaptability of Z-MAC is obtained from the above mechanism.

With increasing the sending rate in Z-MAC, the fairness index decreases rapidly from 0.85 to 0.47 as discussed in this paper. Although the idea of *TMDA* is implemented by setting a shorter *CW* for the owner of a time slot, still the size of *CW* doubles after a collision. Z-MAC cannot guarantee that the owner of a time slot selects a different value from smaller *CW* than that of non-owners each time and hence collisions could still arise. Therefore, for the senders with a larger *CW*, the fairness is badly diminished. On the other hand, the neighbor discovery mechanism in Z-MAC is to send *ping* messages to get the number of neighboring nodes. However, a neighbor node doesn't mean an active neighbor node. It may mislead to a wrong time slot assignment.

Yi Liu *et al.* proposed another approach in [20], which is another way to combine TDMA with CSMA. Each frame mainly consists of *Contention Only Period* (**COP**) and *Transmission Only Period* (**TOP**). As the name indicates, *COP* is used for competing for a transmitting opportunity. Later, the winners in *COP* will share the channel in the coming *TOP* duration. Each winner owns a specific time slot. In addition, a base station works as a central controller in this scheme. It determines whether a node can send frames or not during *COP*. After the base station has the total number of transmissions in the current frame, *TOP* can be correspondingly divided into a reasonable number of time slots. One appealing advantage of this approach is that the number of time slots in *TOP* is dynamically adapted to the current situation (e.g. the number of active stations). It is not a static and fixed time division mechanism. However, the proportion between *COP* and *TOP* should be set properly. For instance, although longer *COP* leads to more senders which can access the channel concurrently, it shortens *TOP* for the transmission. The overall throughput or channel utilization varies on the basis of this proportion. This optimization problem is proven to be a convex programming problem. And the off-the-shelf toolbox is an effective way to address this issue. However, in the proposed scheme, there is still an overhead of transmitting control messages between the base station and access points.

When the workload of the network is at high level, the drawback of *RTS/CTS* mode becomes obvious as a result of collisions of *RTS/CTS* messages themselves. The protocol provided in [21] by Asis Nasipuri, Jun Zhuang and Samir R. Das selects an idle channel from available free channels dynamically. In addition, the channel which has been used for transmitting a frame successfully is given a priority to be selected again. It is called "soft" channel reservation. Compared to traditional *FDMA* or *CDMA*, no central infrastructure to complete the channel assignment is required. It is a distributed protocol. Each node monitors the state of each sub-channels continuously. Once an idle channel is found, this idle channel would be put in a free-list itself. The last-channel records the channel that recently sent successfully a frame. Thus, each station tends to have its own last-channel.

Because each last-channel is more likely to be different, the probability of collisions is reduced. However, if the last-channel is still busy, then the channel is selected randomly from the free-list. Moreover, when no free channel exists, the station has to wait for a backoff period. Unfortunately, the number of sub-channels is predetermined instead of dynamically updated. An optimal algorithm to determine this number is not provided in the paper. The condition of every network traffic pattern changes rapidly. According to results shown in the paper, for the same traffic pattern, the performance of this protocol varies along with the number of available channels. That is to say, determining a suitable value of that number is a key factor to the performance for this protocol.

Gang Qiang designed an approach which combines the advantages of *CDMA* and *RTS/CTS* approach in [25]. The code spectrum in this approach is mainly divided into two types. One is the common code shared by all nodes and used for transmitting *CTS* or *RTS* messages. The other type is a set of individual codes which are used for transmitting data frames. The code assignment is done by the modified *CTS/RTS* approach. When the receiver obtains the *CTS* from the sender on the common code, it replies with a *RTS* frame on its transmission code to the sender. Subsequently, the sender knows the channel selected by the receiver and transmits frame on that specific code. There are two problems in this protocol. *RTS/CTS* frame transmission is a waste of bandwidth resource. In addition, it faces the same challenge of a predetermined number of individual codes.

Chapter 3

Proposed Solution

As indicated above, IEEE 802.11 *MAC* protocol works well only when each active station is within the carrier sense range of other competing stations. When stations cannot sense each other, the contention-based IEEE 802.11 *MAC* algorithm fails to efficiently or fairly share the medium. The key of IEEE 802.11 approach is to sense and detect an idle channel before a transmission. With out of carrier sense, a station will falsely conclude that the channel is idle while another active station is actually sending frames. As a result, many consecutive collisions occur at the receiver. Therefore, a collision-free (allocation) approach is an alternative choice to address the hidden terminal problem. This project proposes the *Binary Exponential Code* backoff *BEC* algorithm which is a an allocation scheme that combines *CSMA* with *CDMA*. Instead of a stand-alone scheme, *BEC* is a hybrid dynamic scheme. If stations can here each other, it works as *CSMA*; otherwise, it shifts to a dynamic *CDMA*.

Just like *TDMA* and *FDMA*, Code Division Multiple Access *CDMA* is another multiple access method. It enables several transmitters to share the same frequency using different code (chip sequences). As long as the chip sequences are pairwise orthogonal with each other, these signals can be transmitted simultaneously and separately extracted by a receiver. More details about this idea are discussed in the first part of this chapter. Later, several features of *BEC* approach will be presented and discussed. At last, to implement *BEC* idea in the simulator, a receiver-based algorithm to collect the neighboring active stations will be detailed.

3.1 Basic Idea

Since *CDMA* is used by *BEC* approach, it will be explained with a specific example in the first part of this section. Then, *BEC* is detailed later in this section.

3.1.1 Code Division Multiple Access

CDMA is a multiple access method which allows several stations to share same frequency channel. It uses the spread-spectrum technology. *CDMA* uses a special coding mechanism to achieve the concurrency attribute. This coding scheme works as described below. In *CDMA*, each bit is encoded with an N -bit chip sequence. Before the start of a transmission, each node is assigned a unique chip sequence. In case of concurrent transmissions, all chip sequences should be pairwise orthogonal: the normalized inner product of any two chip sequences is nul. Let us denote s a chip sequence. A positive code, s , is used for transmitting a 1 bit while a 0 bit is represented by its negative code, $-s$. Considering the case in Figure 1.1, consider two senders S_1 and S_2 that are assigned chip sequences $(-1, -1)$ and $(1, -1)$, respectively. Suppose S_1 must send data $(1, 1, 0, 1)$ to R while S_2 must send data $(0, 1, 1, 0)$ to R at the same time. For convenience, a bipolar notation is used in this example, with 0 being -1 and 1 still being 1. Therefore $(1, 1, -1, 1)$ and $(-1, 1, 1, -1)$ are the data sent by S_1 and S_2 , respectively. To encode the data, the data must be multiplied with its respective chip sequence. $(-1, -1, -1, -1, 1, 1, -1, -1)$ and $(-1, 1, 1, -1, 1, -1, -1, 1)$ are the real data transmitted by the two senders S_1 and S_2 simultaneously. Because R receives the above two signals at the same time, the raw signal produced in R is $(-2, 0, 0, -2, 2, 0, -2, 0)$. Now, to decode signals, it is required to combine the overall signals with the specific chip sequence of each sender. $((-2, 0), (0, -2), (2, 0), (-2, 0))(-1, -1) = ((2 + 0), (0 + 2), (-2 + 0), (2 + 0)) = (2, 2, -2, 2) = (1, 1, -1, 1)$ shows the decoding steps for S_1 . Recall that -1 stands for 0. Then $(1, 1, -1, 1)$ means $(1, 1, 0, 1)$ which is the data sent by S_1 . Decoding is similarly applied to S_2 . As a consequence, multiple signals can be transmitted simultaneously without a collision in *CDMA*. From this example, a key drawback of *CDMA* can be noted: 8 bits are needed to

transmitting only 4 bits. The data rate of the senders is halved: S_1 and S_2 are sharing the channel.

CDMA is appealing because the channel be conveniently shared among an increasing number of competitors by simply increasing the chip code length. When the number of competitors increases, their data rate is decreased to accommodate more senders.

3.1.2 BEC: a Dynamic CDMA

The key idea of *BEC* is to move the channel contention among random access stations from time into a code space. The proposed binary exponential backoff varies the length of the codewords (chip sequence) to manage the channel contention. When contention arises, the length of the chip sequence is doubled to accommodate more simultaneous transmissions without interference. *BEC* backoff starts with a one-bit code per symbol for the first communication between two stations. When contention arises, the size of the code space is adaptively adjusted so that more codes are available to accommodate collision free flows. This concept can also be illustrated with the scenario on Figure 1.1. Since Senders S_1 and S_2 saturate the medium, they will detect a sustained contention and will react by dividing their data rates by using a two-bit orthogonal code (two bits per symbol) so that they can simultaneously transmit without interfering. Periodically, they will double their data rate (i.e. use a one-bit code) and try to use the full channel at full data rate. As long as collisions are experienced, they will back off again to longer chip sequences (lower data rates).

The advantage of *BEC* is its flexibility to switch opportunistically from a contention based *MAC* protocol to a collision free *MAC* protocol when sustained contention arises. For example, when the hidden terminal problem arises such that senders using *CSMA* with Collision Detection (**CSMA/CD**) cannot hear each other and therefore cannot cooperate to share the medium, the idea is to share the medium by allocating separate channels to the senders, i.e., *BEC* switches to a collision free protocol. Sharing the medium through *frequency* (*FDMA*) division multiplexing is not practical because nodes using different frequencies

cannot hear each other: this will limit forwarding capabilities of the nodes on a multi-hop network. Time division multiplexing access (*TDMA*) also is so far impractical because it requires clock synchronization with a fine resolution. The idea is to use *CDMA*. With recent advances, flexible physical layer radios are possible and would support a variable code length to implement the proposed scheme. In most random access *MAC* protocols, competing retransmissions are scheduled at different times to avoid further collisions. The key idea of this approach is to temporarily enable simultaneous competing retransmissions, but at **lower** data rates using *CDMA*. In other words, a somewhat dynamic *CDMA* is proposed. A crucial issue is to design accurate algorithms to determine the appropriate conditions on the network to switch to lower data rates.

However, to simplify the evaluation of *BEC*, the real *CDMA* encoding and decoding the bit stream with their chip sequences were not implemented in the physical layer in the simulator. This work is still in its initial phase. Before the full implementation of *CDMA* in the physical layer is not necessary to prove the feasibility and correctness of *BEC*. We simply emulated *CDMA* on ns-2. This was achieved by allowing simultaneous transmissions if the sum of their bit rates is lower than the channel full bit rate. For *CDMA*, as discussed above, the data rate will be halved with the division of the spectrum. If *CDMA* is implemented in the physical layer successfully, the only effect it would have on the *MAC* layer is the reduction of the data rate theoretically. Such an effect has been implemented and emulated in the simulator.

3.2 Features of BEC

The *BEC* backoff aims to improve network performance and utilization efficiency and may theoretically achieve the maximal channel capacity: if the access protocol can minimize the access overhead, its performance may approach the maximal capacity. Note that the *BEC* backoff does not increase the total access capacity in terms of bit-per-second (bps) because the channel capacity is constant. The *BEC* backoff access simply shares the access

capacity among multiple parallel communications. The portion of the capacity allocated to each communication relies on the size of the code space: longer are the codewords, the more orthogonal codes, and thus more parallel communications. However, each communication gets a smaller capacity share. If the total access capacity in terms of bit rate is R and there are N communications, then each code is N -bit wide and each communication will transmit at symbol rate $\frac{R}{N}$. This relationship can be illustrated from another perspective. Suppose each communication can physically support an R physical bit rate. With an N -bit codeword, its actual data symbol rate is $\frac{R}{N}$ because each data bit is transcoded by an N bit orthogonal code.

The *BEC* backoff random access has three salient features: 1) it enables parallel communications to accommodate multiple simultaneous access requests and to reduce collisions and alleviate the hidden terminal problem by using orthogonal codes, 2) it adaptively and distributively updates the code space with more and longer codes to scale up/down the network for more/less parallel communications, and 3) unlike the temporal binary exponential backoff, it does not occur in the time domain, so, its channel efficiency is independent from future bit rate increases that are inevitable.

3.3 Implementation

To emulate the *CDMA* idea, the data rate in *MAC* layer is halved or doubled when the chip sequence is doubled or halved. The aggregate data rate does not exceed the bandwidth; otherwise, frames will collide with each other. The decision to double or halve the data rate needs to be investigated carefully here: if the sender is greedy and tries to double the data rate too often while the contention perdures, the medium will be wasted due to collisions. On the other hand, if the sender is not greedy and waits too long to double its data rate, the medium will also be wasted if the contending sender is not transmitting. The key factor of a reasonable spectrum allocation is to collect correctly the number of active neighboring nodes around the same receiver. Usually the number of nodes in the same area is likely to

dynamically change. Therefore, since the number of total active nodes is not available in advance, a static allocation (of chip sequences) is not practical. A way to implement *BEC* is to regard the receiver as a temporal central controller within a circle area with current receivers. As suggested in IEEE 802.11, the information of the sender can be extracted from the *MAC* header in the frame. A list of current contending senders is created at the receiver to store the information of each sender that is sending frames to the current receiver. Each time a new frame is received, the list is checked/updated. If the sender of this new frame does not exist in the list, then it is inserted into the list immediately as well as the information of the current time. However, if that sender is already on the list, only its time stamp is updated. If a sender leaves or remains silent for a long time, it is deleted from the list. With the help of time stamp in the list, the most recently received frame from that specific sender is known. Once the difference between that time stamp and current time exceeds a threshold, this sender is deleted from the list accordingly. As introduced, a *MAC ACK* frame is required to be sent after each successful transmission. To take advantage of this point, the overall number of senders got from the list is stored in the *ACK* frame. When the sender receives the *ACK* frame and extracts it, the sender now knows the exact number of active transmissions around the target receiver and hence can reallocate the spectrum accordingly. Figure 3.1 displays the descriptions of the process introduced above.

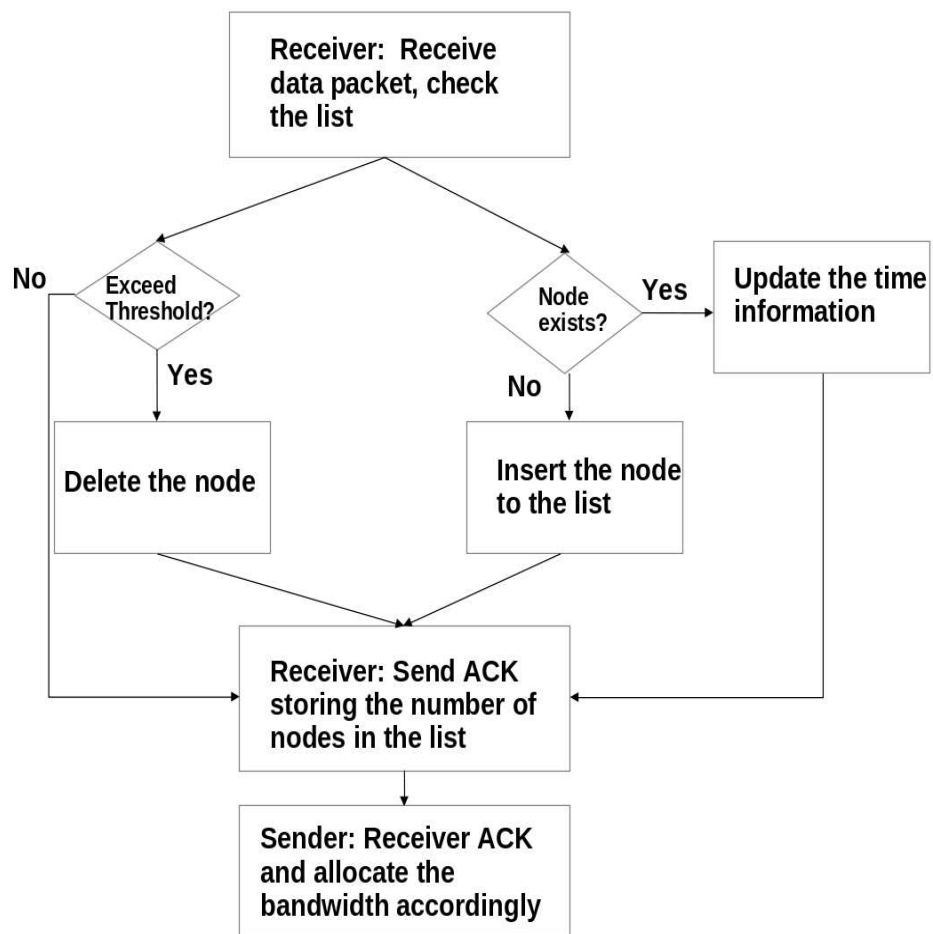


Figure 3.1: The State Transition of Allocating Bandwidth

Chapter 4

Evaluation

Performance evaluation of a novel scheme is typically performed using one or a mix of the following methods: testbed experimentation, formal modeling, and simulation. Although testbed experimentation may be the most convincing, this work uses simulation and formal modeling because 1) experimentation would require specialized hardware and 2) the proposed scheme should be first evaluated to determine whether it is worth and will yield a significantly higher performance than IEEE 802.11.

Compared to testbed experimentation, simulation and formal modeling carry numerous advantages. First, it is usually easier to reconfigure a specific topology: adding nodes, changing their positions, changing their traffic patterns, *etc.* Especially, formal modeling requires only to modify the input parameters and recompute the model in a short time. It allows the performance evaluation under a wide variety of network conditions. Second, the results obtained by simulation are repeatable to allow other researchers to test this work and check the integrity and credibility of the work. Third, the formatted trace files in simulators are well designed to help user to get statistical metrics, such as the throughput and the packet delays. Last, the use of formal modeling and simulation allow to validate the conclusions and accurately predict the behavior of the real system if built.

4.1 Simulation

A network simulator can be implemented using general programming language, simulation package, or pre-built simulators. Considering the efficiency of this project, it is not ideal to design schedulers and handlers scheme and implement all networking layers protocols. Since this project only focuses on *MAC* layer, a pre-built simulator which is extensible

to modify and add new modules is a reasonable choice. It is always true that simulators are easy to use, debugging, tracing, which really facilitates the development process and shortens the duration required by the project. Also, since some simulators, such as *ns-2*, are the event-driven simulators in which simulation time moves in jumps based on time of "next event", it becomes a time-saving approach in some cases. Furthermore, many networking components, such as links, routing, wireless, TCP and UDP, have already been well implemented in the *ns-2* simulator. It is easy to just take advantage of these available components and design the new protocol.

ns-2 and *OPNET* are two main widely used networking simulators. Both of them are effective and excellent simulators. However, since *OPNET* is developed and maintained by OPNET Inc., it is more expensive while *ns-2* is publicly available for free. Although a large amount of wireless modules are available in *OPNET* and more professional and quick technical support are provided by OPNET Inc., *ns-2* still remains a better choice for this project considering its wide acceptance in Academia.

The *ns-2* simulator was designed and developed at UC Berkeley. Many components were later added by a growing world wide community. *ns-2* is popular simulator in networking research area. It is an open source software that makes it easy and convenient to contribute new modules or protocols. Because *ns-2* components are written in C++ and a simulation is built using the object-oriented Tcl that acts as *glue* of the components. Furthermore, as a result of its well-defined modularity, it is quite easy for beginners to become familiar with the structure of the whole software. For example, as each network component is represented by a specific class, it makes it simple to construct the virtual networking layers. The major network protocol stacks are written in C++. In some sense, C++ is responsible for dealing with the "static data" part in the simulator. On the other hand, Otcl is mostly used for configuring the specified scenarios and triggering the expected events. It works like a controller which calls the "static data" part written in C++. Consequently, all simulations for this project are performed using *ns-2* simulator.

4.2 Analytical Model

The original Markov chain model and the analytical model for *BEC* are introduced and discussed in this section.

4.2.1 Markov Chain Model for IEEE 802.11

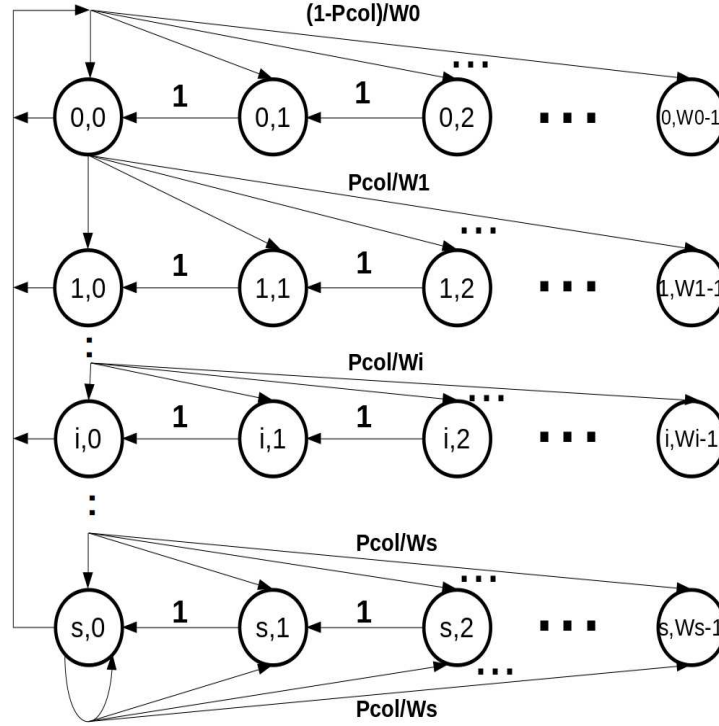


Figure 4.1: Markov Model for IEEE 802.11

A two-dimensional Markov chain model is proposed by Bianchi [10] to predict the saturation throughput of IEEE 802.11 in *DCF* mode. The details about *DCF* mode are explained earlier. Since the theoretical maximum throughput is the target, it can safely assume that, after each successful transmission, another frame will be ready to be transmitted. $S(s, c)$ denoted a state in this two-dimensional Markov chain model where s stands for backoff stage while c refers to the current window size. Recall that s determines the contention window size which is W_s and c varies from 0 to $W_s - 1$. Figure 4.1 depicts the state transition in the model. According to the figure, a frame can only be sent in stage $S(s, 0)$ which means the

window counter reaches 0. Thus the probability of that a station starts to transmit a frame is

$$p = \sum_0^s S(i, 0) \quad (4.1)$$

Furthermore, if a frame collides, there should be at least two transmitting frames. Therefore the probability of a collision can be obtained.

$$P_{col} = 1 - (1 - p)^{n-1} \quad (4.2)$$

If a collision arises, the state goes to $S(s + 1, c)$ with probability of P_{col}/W_{s+1} . On the other hand, if the frame does not collide, the stage goes back to $S(0, c)$ with probability of $(1 - P_{col})/W_0$. It is because the contention window is always reset after each successful transmission.

In the figure, T_{avg} between each nearby stage does not refer to real idle time slot t . In *DCF*, the window counter would not decrease by 1 until an idle channel for t is sensed. To simplify the model, Bianchi replaces it with T_{avg} . In fact, three cases are taken into account to decide T_{avg} [28]. If the channel is idle, T_{avg} is just t . If the channel is busy, it leads to either a successful transmission or a failed transmission. Therefore the T_{avg} can be derived from below equations. As a result, whatever happens in the channel from stage $S(s, c + 1)$ to stage $S(s, c)$, the probability is always 1.

$$T_{avg} = (1 - P_{tr}) * t + P_{tr} * P_s * T_s + P_{tr} * (1 - P_s) * T_c \quad (4.3)$$

where

$$P_{tr} = 1 - (1 - p)^n \quad (4.4)$$

$$P_s = \frac{n * p * (1 - p)^{n-1}}{1 - (1 - p)^n} \quad (4.5)$$

$$T_s = DIFS + T[P] + SIFS + ACK \quad (4.6)$$

$$T_c = DIFS + T[P] \quad (4.7)$$

P_{tr} refers to the situation that there is at least one transmission while P_s is the probability of a successful transmission in which only one transmission occurs in the channel. In addition, T_s indicates the duration required to finish a successful transmission. T_c is the time for a failed transmission which can be implied by the absence of an ACK. $T[P]$ is the duration used for payload transmission.

Three cases for channel situations are introduced above. Among them, only the successful transmission makes a contribution to the saturated throughput. Therefore the saturated throughput can be derived as

$$Th = \frac{P_s * P_{tr} * E[P]}{T_{avg}} \quad (4.8)$$

The $E[P]$ is the average payload length which typically is just the length of a data frame.

4.2.2 Novel Model for BEC

In *BEC*, there is no contention window and all time is used for frame transmission phase. Therefore, since stations would send frames without a contention window and no backoff time slot is selected, the classical Markov chain model which determines the states on the basis of backoff time slots does not apply to *BEC*. On the other hand, as indicated above, the spectrum allocation in *BEC* is determined by the number of concurrent active transmissions. The more transmissions there are, the more sub-spectrum would be set. In addition, the number of active transmissions is updated dynamically. This means that the overall throughput can be affected by the number of active transmissions directly. Consequently, in this new proposed Markov chain model, the states would be set according to the number of concurrent active transmissions. Figure 4.2 shows the state transition diagram in this model.

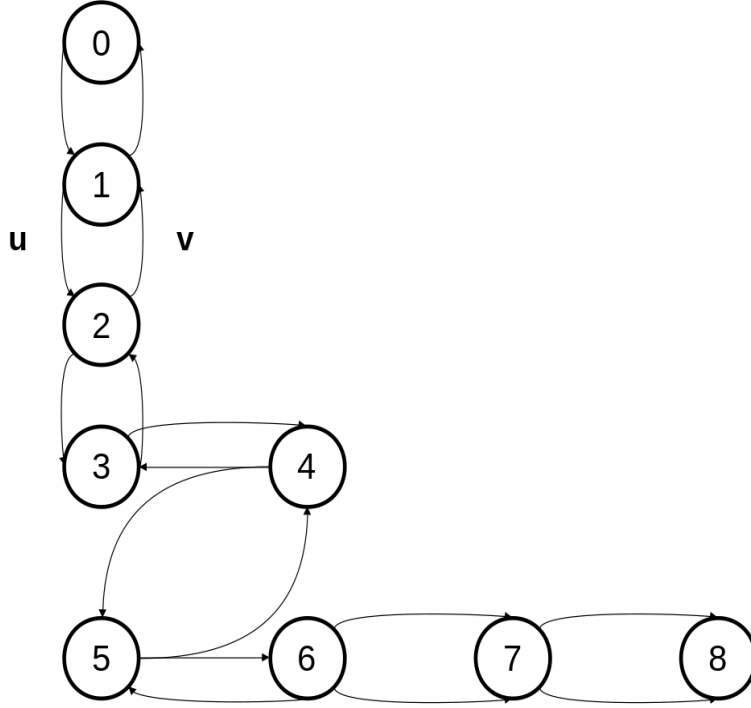


Figure 4.2: State Transition Diagram

u is the arrival rate while v is the departure rate. The "arrival" rate here refers to the number of nodes that arrive per second while the "departure" means that the number of nodes that depart per second. Both of them are applied to the exponential distribution, which approximately reflects the networking load in reality. Thus the difference between each state in the Figure 4.2 is the number of concurrent active senders. Once a node is activated with the probability of u in state k , it goes to the next state $k + 1$. On the other hand, it goes back to state $k - 1$ with a departed active node. At this time, the spectrum would not be divided into more than 8 parts, which means ideally at most 8 active senders can be supported by BEC. This is why there are only 9 states in the Figure 4.2.

As long as the current spectrum allocation provides enough sub-spectrum for current active nodes, the spectrum will not be divided continuously. Therefore, the state 3 and state 4 have the same spectrum allocation, which is applied to all states from 5 to 8. The spectrum allocation would only occur in the transitions from state 1 to 2, 2 to 3 and 4 to 5. It is worth

mentioning that the different spectrum allocation leads to different data rates for each state, which has a huge influence on the overall throughput.

From the basic theories of Markov Process, several equations can be derived below.

$$p_i = \left(\frac{u}{v}\right)^i * \left(\frac{p_0}{i!}\right) \quad (4.9)$$

$$\sum_0^K p_n = 1 \quad (4.10)$$

p_k denotes to the probability of the occurrence of state k . The first equation indicates that all other states can be derived from p_0 . In addition, if we add up the probability of each state in the Figure 4.2, the result is 1 shown in the second equations. Moreover, two equations can be derived as shown below to calculate p_0 and p_i .

$$p_0 = \frac{1}{\sum_0^K \left(\frac{u}{v}\right)^n * \left(\frac{1}{n!}\right)} \quad (4.11)$$

$$p_i = \frac{u^i}{v^i * i! * \sum_0^K \left(\frac{u}{v}\right)^n * \left(\frac{1}{n!}\right)} \quad (4.12)$$

Some details are considered in calculating the overall throughput. As *CSMA* is still implemented in *BEC*, an idle channel should be detected for *DIFS* before each transmission. Then the data frame is being transmitted in a specific data rate which is determined by the state. When the receiver receives the data frame and waits for *SIFS* for an idle channel, a *MAC ACK* will be sent in the basic rate. Once the *MAC ACK* reaches the sender, a successful transmission for one data frame is done. Just as mentioned above, for an ideal *BEC*, the collisions are completely avoided and hence the case for a collision is not taken into account here. For instance, suppose now it comes to the state 4 in Figure 4.2, which means there are four active transmissions in the topology. Firstly, calculate the throughput for one node using the above analysis. Secondly, because each node has the same chance of accessing

the channel, each node would obtain the same throughput approximately. Subsequently, the overall throughput is the result of multiplying the number of total active senders and the throughput of each one. As a whole, now the average overall throughput for a dynamic model can be calculated on the basis of equation shown below.

$$Thr_n = \sum_0^n \frac{i * p_i * E[p]}{DIFS + \frac{E[p]}{r_d^i} + SIFS + \frac{E[ack]}{r_b}} \quad (4.13)$$

Thr_n refers to the average throughput for the situation that the state varies from 0 to n . $E[p]$ is the length of a data frame and $E[ack]$ is the length of a *MAC ACK* frame. r_b stands for the basic rate used for sending control frames including *ACK*. And r_d^i is the data rate for transmitting the data frame. r_d^i is determined by the state. Equation below shows the calculations of the r_d^i in which BW stands for the overall bandwidth.

$$r_d^i = \begin{cases} \frac{BW}{i} & i = 1, 2, 4, 8... \\ \frac{BW}{2^{\lceil \log_2 i \rceil + 1}} & i \neq 1, 2, 4, 8... \end{cases} \quad (4.14)$$

Chapter 5

Result

This chapter will discuss the data collected to evaluate *BEC* using simulation and formal analysis. Simulation is validated using an analytical model developed in this work. The performance of the IEEE 802.11 MAC protocol is compared to *BEC*'s performance for UDP and TCP traffic.

Before delving into the details, some configuration parameters will be briefly introduced first. To simplify simulations, the carrier sense range and the transmission range are set to 550 meters. Some other basic configuration parameters are shown in Table 5.1. As a whole, three topologies are considered in the simulation. They are shown in Figures 5.1, 5.2 and 5.3, respectively.

Table 5.1: Simulation Configuration

Parameter	Value
SlotTime	0.000020 us
SIFS	0.000009 us
Queue Length	50
PLCPDataRate	6 Mbps
MAC Bandwidth	48 Mbps
MAC Datarate	48 Mbps
MAC Basicrate	6 Mbps
Routing Protocol	AODV

- **(Two senders, two receivers)** All four nodes are on a straight line. Each sender sends frames to its own receiver. The distance called **Dsr** between each pair sender-receiver remains set to 275 meters throughout all experiments with this topology while Distance **Dss** between the two senders is dynamically varied. In other words, the distance between

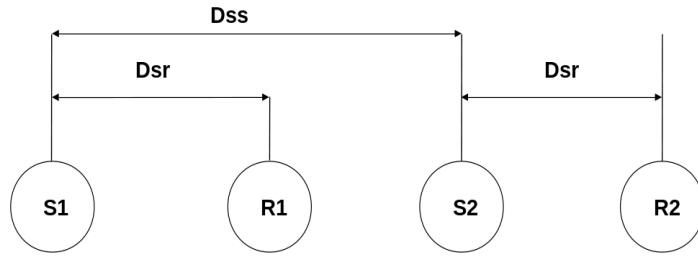


Figure 5.1: Two senders, two receivers Topology

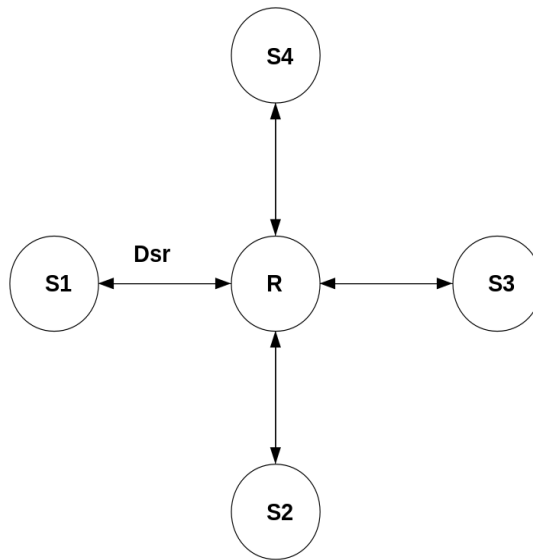


Figure 5.2: Four senders, one receiver Topology



Figure 5.3: Chain Topology

the pairs sender-receiver varies. When Receiver 2 cannot hear Sender 1, the hidden terminal problem arises.

- **(Fours senders, one receiver)** The second topology consists of five nodes: four senders placed on a circle and one common receiver located at the center. The radius D_{sr} varies from 0 to 550 meters. When D_{sr} exceeds 389 meters, senders cannot hear each other, leading to the hidden terminal problem.
- **(Multi-hop chain)** The last topology is a chain with a sender, a receiver, and N intermediary nodes. N varies from 0 to 8. The number of flows between the sender and the receiver is set to 1, 2, 4, and 8. The distance between each node is 300 meters. It indicates that, when there are more than one hop, the sender is unable to access the receiver directly. The intermediary nodes must forward the frames. Since multiple transmissions are competing for the same channel as explained in Chapter 2, the condition of the network can dramatically degrade.

In order to evaluate the *BEC* MAC access approach, two metrics, *overall throughput* and *fairness*, are measured. Since the available bandwidth is the same in each simulation, the higher is the overall throughput, the better the MAC access protocol is. For all three above topologies, the overall throughput will be plotted in the following figures.

Fairness is an important metric in networking performance because it often translates into a better experience for a user. If all senders get a fair share then the performance is smooth as it does not vary much depending on the location of the user or the slight variations of the overall load. There are two types of fairness: temporal and throughput fairness. *Temporal fairness* aims to assign equal long term transmission duration to each communication while *throughput fairness* aims equal throughput for each communication. Temporal fairness is preferred in wireless networking because it is very likely that communications occur at

different bit rates based on channel conditions variations experienced by different communications. If two senders using different bit rates get the same communication duration, they will enjoy different throughputs.

The *BEC* backoff random access inherently offers temporal fairness because communications are not separated in the time domain, but in the code domain. Therefore, the competition for transmission resources among communications does not affect the **temporal** use of the channel of each other. Consider two communications occurring at different bit rates. With the *BEC* backoff, they can both transmit simultaneously and inherently achieve the temporal fairness.

The fairness can be depicted in a variety of ways. For the first topology, as there are only two senders, the fairness is defined by the fraction $\frac{T_i}{T_{avg}}$ where T_i is the throughput of Sender i and T_{avg} is the average $\frac{T_1+T_2}{2}$. If $\frac{T_i}{T_{avg}}$ is close to 1, then the bandwidth is fairly shared; otherwise, the fairness is poor. But, for the second topology, the number of senders is 4. we use Jain's Fairness Index [23] .

5.1 Verification for Analytical Model

To verify the correctness of the analytical model, a group of same parameters is set for both the simulation and the analytical model. The comparison is shown in Figure 5.4.

In the simulation, the maximum number of active senders is set to 2, 4, or 5. The comparison between the simulation results and the analytical model is shown in Figure 5.4a, 5.4b and 5.4c, respectively. On the x-axis, the proportion of the arrival rate to the departure rate varies from 0 to 5. Observe on these figures that the overall trend of the simulation lines are the same as the analytical ones. Especially in Figure 5.4a, the two lines are close to each other. For 4 and 5 active nodes cases, a gap between the two lines increases as the ratio of arrival rate over departure rate increases. This gap comes from the overhead of processing time in the receiver. When more concurrent transmissions occur in the receiver, the overhead increases. On the other hand, when the proportion is small, the model predicts

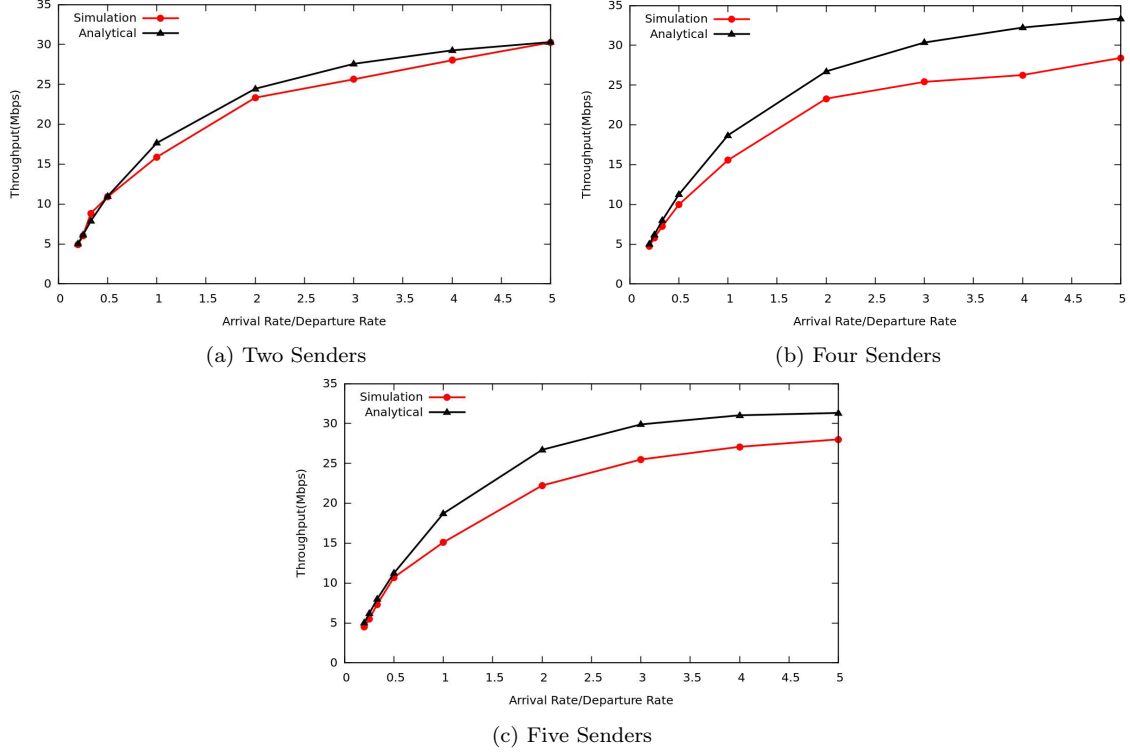


Figure 5.4: Comparisons between Analytical Model and Simulation Result

the behavior of *BEC* accurately for all three cases. If frames depart quicker than they arrive in, few frames request service from the senders. That is to say, all coming frames are fully processed and many other senders remain silent and unused in this case. And when this proportion increases, the gaps slightly increases, which is caused by the same processing overhead mentioned above. Consequently, it confirms the correctness and preciseness of the proposed analytical model.

5.2 UDP

According to the Figure 5.5a, the overall throughput obtained by *BEC* is always larger than that of IEEE 802.11. If a node receives an *ACK* of which the destination is not the current node, the *NAV* would be set and this node is disabled to send any frame in IEEE 802.11. However, in *BEC*, the data rate of each node is set to a proper value and hence it guarantees the success of each transmission without a collision. It indicates that the *NAV*

is not needed here and other nodes are not required to wait for sending frames. As a result, the overall throughput is increased significantly in *BEC*.

When D_{ss} is larger than 825 meters in the first topology, the two transmissions do not interfere anymore with each other. Thus, both of them will achieve the maximum throughput. On the other hand, when D_{ss} is within the range (550, 825), the hidden terminal problem arises. Sender 2 cannot sense the busy state of Receiver 1 as a result of the transmission from Sender 1, although the target of Sender 2 is Receiver 2. Interference arises in Receiver 1. Compared to the performance among the range (0, 550), the overall throughput is decreased by 10.6% and the fairness seriously diminishes in IEEE 802.11 approach shown in Figure 5.5a, 5.5b. On the contrary, *BEC* obtains a higher throughput within the range (550, 825) than that within the range (0, 550), which is caused in Receiver 1 by receiving frames of which the destination is Sender 1 from Sender 2, including the *MAC* header of frames.

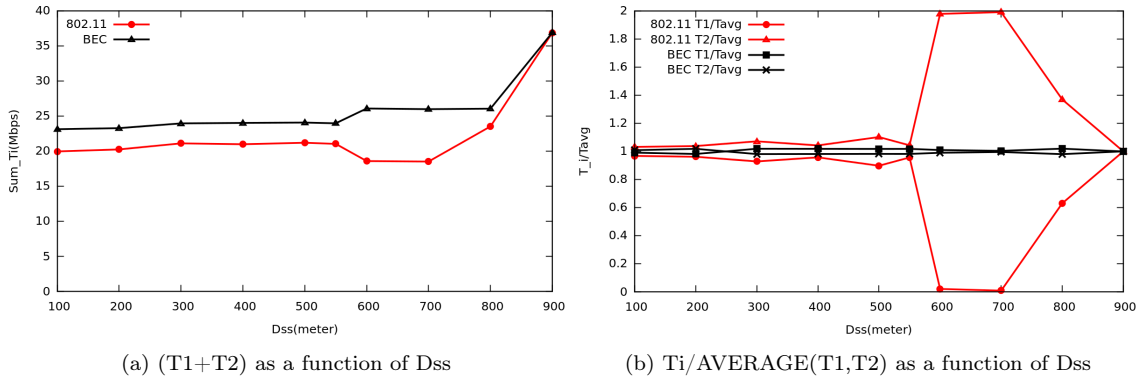


Figure 5.5: Two Senders, Two Receivers

Since the contention window CW is not used by *BEC*, every node can access the channel and send frames fairly. However, in IEEE 802.11, when there is a hidden terminal problem, one of contending nodes would reach a large value for CW as a result of the repeated collisions while the other node maintains a small CW . In IEEE 802.11, the backoff time slot is selected from CW . With a large CW , the channel would keep the channel idle more often. On the contrary, the other node with a small CW would take full advantage of the channel. That

is why in Figure 5.5b, one of the two lines gets close to 2 while the other one reaches almost 0. Recall that the lines in the figure stand for the proportion between the throughput of one sender and the average throughput. For the case of *BEC*, the corresponding two lines always remain 1 approximately, indicating that *BEC* is fairer.

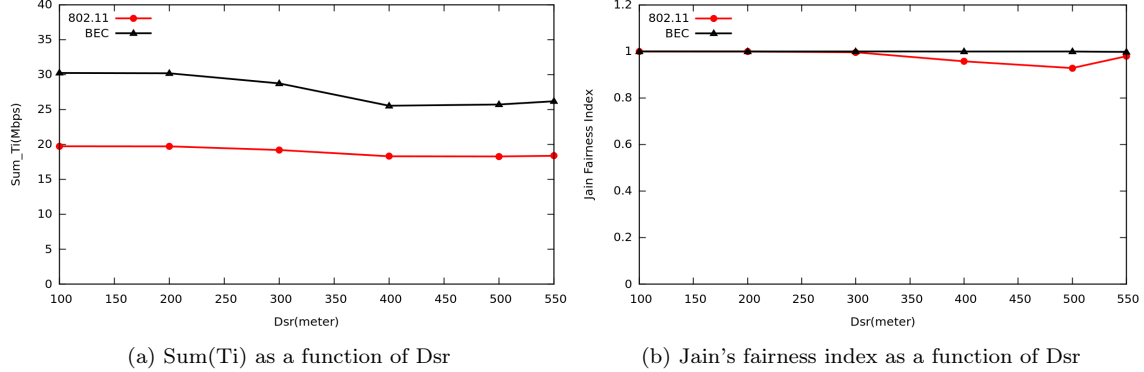


Figure 5.6: Four Senders, One Receiver

The same phenomenon happens with the second topology shown in Figure 5.6a and Figure 5.6b. The performance of *BEC* outreaches IEEE 802.11 in both the overall throughput and the Jain's Fairness Index which is one of metrics used to measure fairness. The Jain Index varies from 0 to 1. If the index is 1, it means that each node shares the channel completely fairly. Figure 5.6a shows that the overall throughput obtained from *BEC* is 46.5% higher than that of IEEE 802.11 on average. When Distance Dsr is larger than 275 meters, some stations begin to experience the hidden terminal problem. It is worth noticing that the overall throughput of *BEC* slightly decreases. A *DIFS* of idle time has to be detected before sending a frame in both *BEC* and IEEE 802.11. Thus, the channel is considered idle all the time with a distance Dsr less than 275 meters while, if Dsr is larger than 275 meters, it is still possible for the station to know whether another station is accessing the channel or not. As a result, more frames would be sent simultaneously. However, only four frames can be transmitted simultaneously in this topology and hence the number of collisions arises, diminishing the performance of the overall throughput in *BEC*. When it comes to the Jain's

Fairness Index, it remains 1 all the time for *BEC*. However, it is slightly diminished for IEEE 802.11 because of the hidden terminal problem.

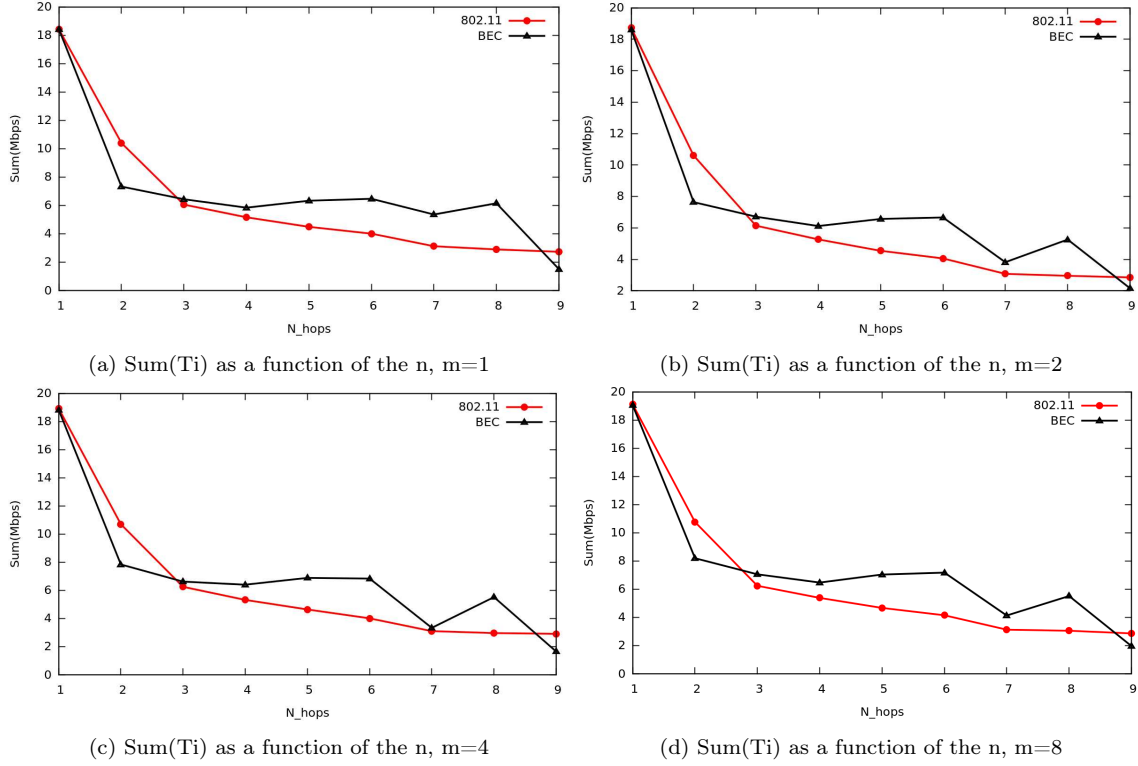


Figure 5.7: Chain

Figure 5.7 shows the relationship between the overall throughput and the number of hops in the third topology. The only difference among the four sub figures is the number of flows from the sender to the receiver, which is set to 1, 2, 4 and 8. Just as shown in these sub figures, no matter what the number of flow is, *BEC* always performs better than IEEE 802.11. The throughput is increased by 21% - 28% on average. In addition, a special case is the topology with only two hops in which Station 1 sends frames to hop 2 through hop 1. As described above, Station 1 cannot communicate with hop 2 directly. Hop 1 works here and saves the frames temporarily from station 1 and sends them to hop 2 later. No hidden terminal problem is here and no two data frames are competing for the same channel. From the perspective of the neighboring nodes discovery algorithm, the code deviation mechanism

doesn't arise in this topology. However, a few of collisions caused by the fact that *MAC ACK* frames still compete with data frames would diminish the performance of *BEC*.

At last, compared to IEEE 802.11, both the average packet delay and the average jitter are reduced obviously in *BEC*, as shown in Figure 5.8. The red lines in the figures, standing for the performance obtained in IEEE 802.11, go up rapidly, when the number of hops exceeds 2. The jitter starts at 0 ms and ends with 3 ms in IEEE 802.11 while, in *BEC*, it is always less than 0.5 ms. As indicated above, more collisions arise as the number of hops increases. Since *CW* increases as a result of collisions, a frame needs to wait for a longer duration for the next retransmission. Thus the packet delay and jitter increase.

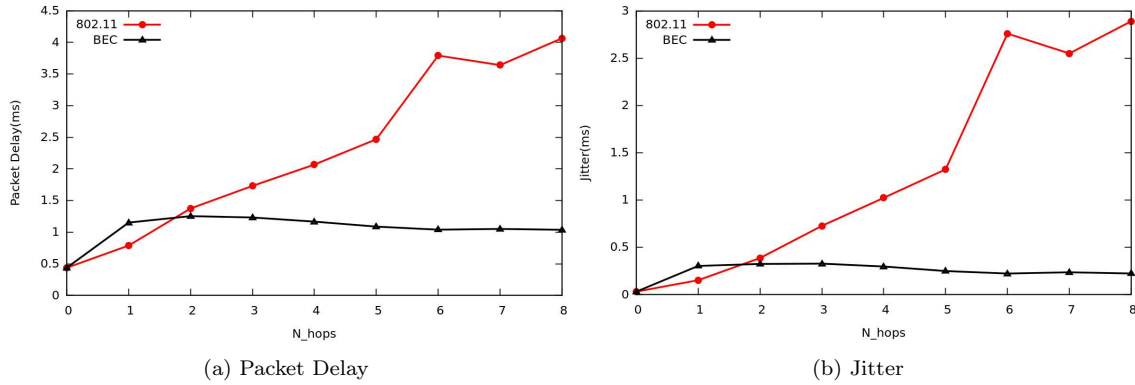


Figure 5.8: UDP Multi-hop Topology with One Flow

5.3 TCP Traffic

In the first two topologies, *BEC* still keeps the fairness of each sender well, which can be concluded from Figure 5.9b and Figure 5.10b. According to Figure 5.10b, the fairness with TCP is not as good as the fairness with UDP traffic, using IEEE 802.11. However, in *BEC*, the fairness in TCP is obtained at a price: the overall throughput decreases. In Figure 5.9a, the overall throughput obtained by *BEC* is the same as that of IEEE 802.11 within the range (0, 550). When D_{ss} exceeds 550, *BEC* begins to achieve a much higher throughput while IEEE 802.11 remains the same with a worse fairness. Another observation made here is the higher overall throughput obtained in IEEE 802.11 instead of *BEC* in Figure 5.10a.

But, in *BEC*, a better fairness performance is achieved. The reason for the decrease of the throughput is the increase of the number of senders. Figure 5.11 shows the comparison, when the number of senders is two instead of four. It is obvious that *BEC* still obtains a higher throughput than IEEE 802.11 for this case. In fact, the more frames are transmitted simultaneously, the more overhead there is with *BEC*.

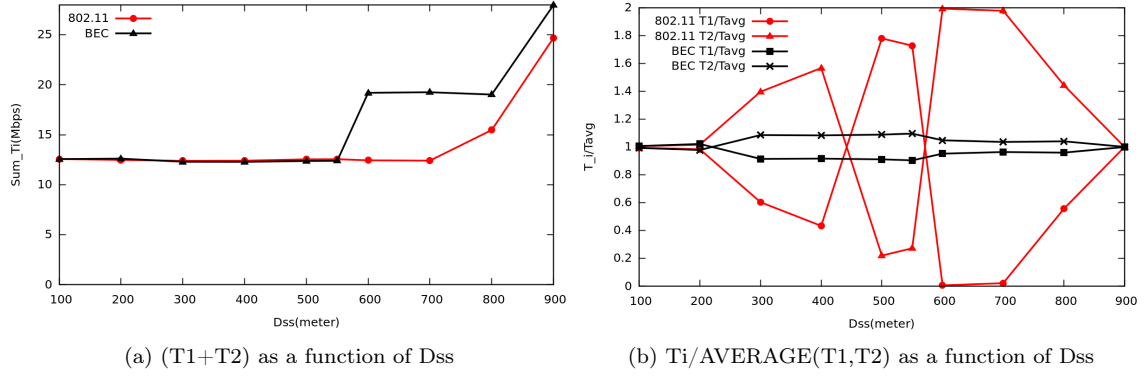


Figure 5.9: Two Senders, Two Receivers

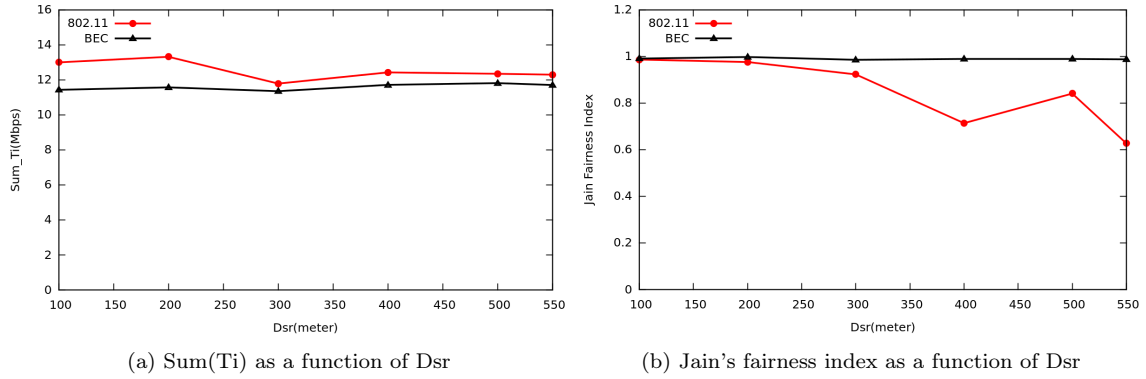


Figure 5.10: Four Senders, One Receiver

Finally, in the chain topology, the throughput with *BEC* does not increase as obviously as that with UDP. It is because of the special characteristics of TCP. Compared to UDP, TCP has its own reliability and congestion control mechanism. First, for each transmission, a TCP ack needs to be sent back to the sender in the transport layer. Therefore, except for the contention among TCP data segments, TCP acks add more contention to the *MAC* layer. In addition, the sender adapts its sending rate based on current channel conditions in TCP with the congestion control mechanism while, in UDP, the sender keeps sending segments

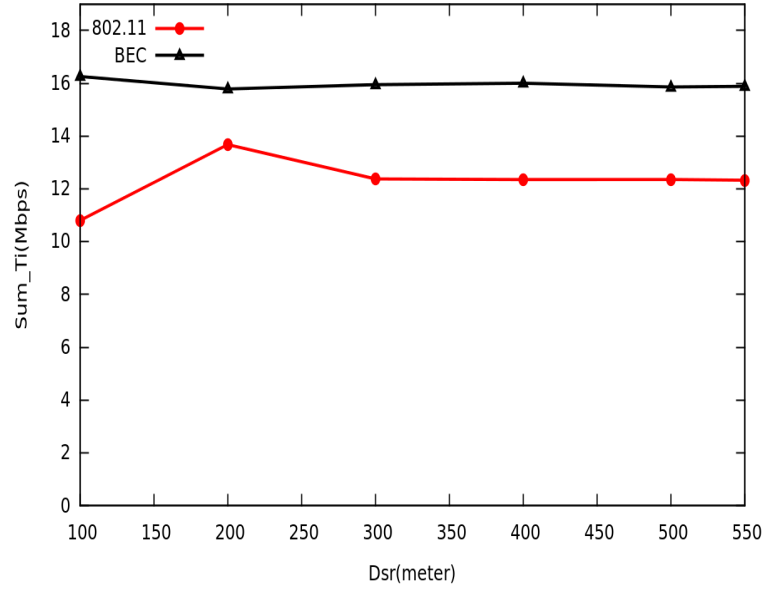


Figure 5.11: Two senders, One receiver: Overall Throughput

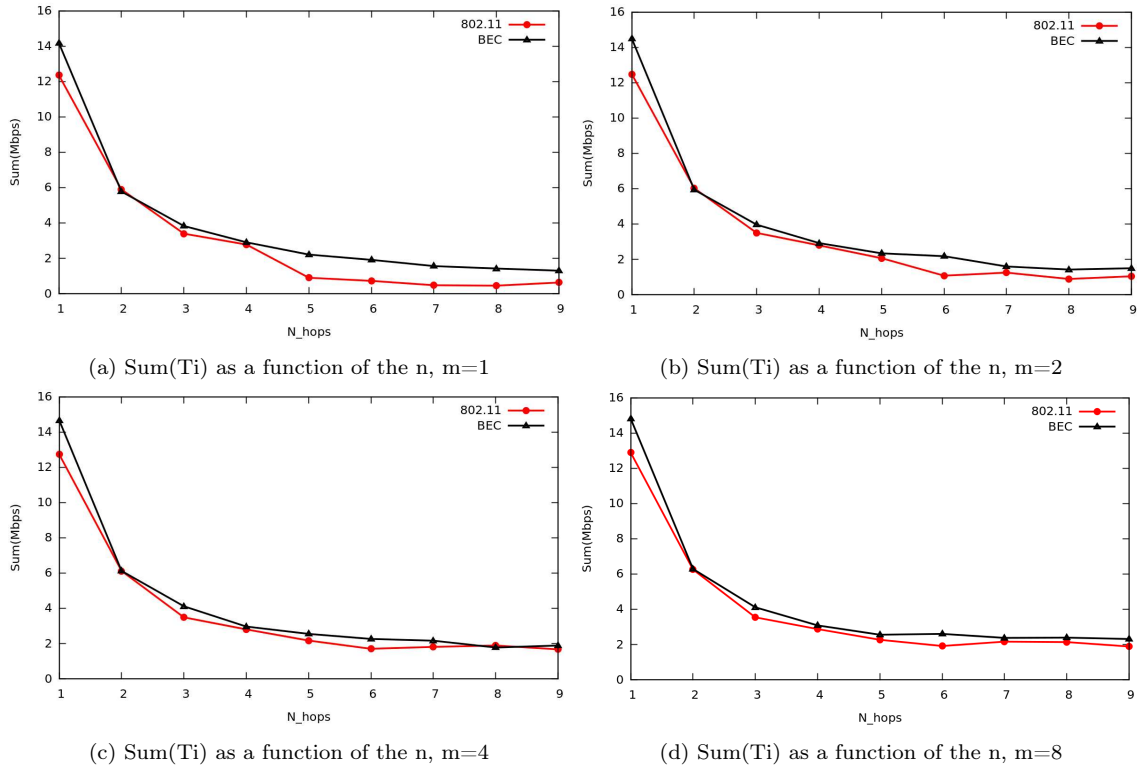


Figure 5.12: Chain

into the channel regardless of congestion. Both of the above two factors lead to slightly low performance of *BEC* for TCP. However, IEEE 802.11 still cannot achieve a throughput as high as *BEC* in TCP, which are depicted in Figure 5.12. In fact, the overall throughput is increased by 15% - 30% approximately in *BEC*.

5.4 Discussion and Future Work

Based on the analytical model and simulation results, it is safe to conclude that the proposed *BEC* is an effective solution to address the issue caused by the hidden terminal problem. For applications using UDP, not only the overall throughput is increased by 20% - 40%, but also the fairness for each node is significantly improved significantly. For applications using TCP, the fairness is fairly kept in most cases and, when it comes to multi-hop network with hidden terminal problems, the overall throughput is increased by 15% - 30%.

On the other hand, there are still several aspects of work that can be done for this project in the future. First, although *BEC* idea has been successfully simulated, a code deviation mechanism is not actually implemented in the simulator this time. The simulation in ns-2 for this work focused on the *MAC* layer and emulated CDMA. Future work should implement CDMA in the physical layer in *ns-2* simulator to study the potential impact it would have on *BEC* approach. Second, because of its complexity and high cost, a real testbed implementation was not possible in this work. Just as explained, the result obtained from real testbed can show exactly how the *BEC* approach behaves in reality and improve the credibility of this project. Finally, the current approach is well designed to deal with the case with a large number of active stations. Scalability is one of the appealing attributes which could be added into the approach in the future.

Bibliography

- [1] T. Razafindralambo and F. Valois, "Performance Evaluation of Backoff Algorithms in 802.11 Ad-Hoc Networks," PE-WASUN conference, 2006.
- [2] Fouad A. Tobagi and Leonard Kleinrock, "Packet Switching In Radio Channels: Part III – Polling and (Dynamic) Split-Channel Reservation Multiple Access," IEEE TRANSACTIONS ON COMMUNICATIONS, 1976.
- [3] Karn and Phil, "MACA a new channel access method for packet radio," Computer Networking Conference, Amateur Radio, 1990.
- [4] Leonard Kleinrock and Fouad A. Tobagi, "Random access techniques for data transmission over packet-switched radio channels," AFIPS National Computer Conference, 1975.
- [5] Jian Ying and Chen Shigang, "Can CSMA/CA networks be made fair?" Proceedings of the 14th ACM international conference on Mobile computing and networking, San Francisco, California, USA, 2008.
- [6] Claude Chaudet, Dominique Dhoutaut and Isabelle Guérin Lassous, "Experiments of Some Performance Issues with IEEE 802.11b in Ad Hoc Networks, " WONS, 2005.
- [7] Chun-cheng Chen, Eunsoo Seo, Hwangnam Kim and Haiyun Luo, "SELECT: Self-Learning Collision Avoidance for Wireless Networks," IEEE Transactions Mobile Computing, 2008.
- [8] Magistretti Eugenio, Chintalapudi Krishna Kant, Radunovic Bozidar and Ramjee Ramachandran, "WiFi-Nano: reclaiming WiFi efficiency through 800 ns slots," Proceedings of the 17th annual international conference on Mobile computing and networking, New York, NY, USA, 2011.
- [9] Riduan M. Abid, Taha Benbrahim and Saad Biaz, "IEEE 802.11s Wireless Mesh Networks for Last-Mile Internet Access: An Open-Source Real-World Indoor Testbed Implementation," Wireless Sensor Network, 2010.
- [10] Giuseppe Bianchi, "IEEE 802.11-Saturation Throughput Analysis," IEEE COMMUNICATIONS LETTERS, 1998.
- [11] Aruna Jayasuriya, Sylvie Perreau, Arek Dadej and Steven Gordon, "Hidden vs. Exposed Terminal Problem in Ad hoc Networks," Australian Telecommunication Networks and Applications Conference, 2004.

- [12] Marek Natkaniec, Andrzej R. Pach, "An Analysis of the Backoff Mechanism used in IEEE 802.11 Networks," Computers and Communications, 2000. Proceedings. ISCC 2000. Fifth IEEE Symposium on.
- [13] L. Bononi, M. Contih and E. Gregori, "Design and Performance Evaluation of an Asymptotically Optimal BackoffAlgorithm for IEEE 802.11 Wireless LANs," System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on.
- [14] Bianchi G., Fratta L. and Olivieri M., "Performance Evaluation and Enhancement of the CSMA/CA MAC protocol for 802.11 Wireless LANs," proceedings of PIMRC, 1996.
- [15] Haitao Wu, Yong Peng, Keping Long, Shiduan Cheng and Jian Ma, "Performance of Reliable Transport Protocol over IEEE 802.11 Wireless LAN: Analysis and Enhancement," IEEE INFOCOM, 2002.
- [16] Nai-Oak Song, ByungJae Kwak, Jabin Song and Leonard E. Miller, "Enhancement of IEEE 802.11 Distributed Coordination Function with Exponential Increase Exponential Decrease Backoff Algorithm," Vehicular Technology Conference, 2003. VTC 2003-Spring. The 57th IEEE Semiannual.
- [17] Jing Deng, Pramod K. Varshney and Zygmunt J. Haas, "A New Backoff Algorithm for the IEEE 802.11 Distributed Coordination Function," Communication Networks and Distributed Systems Modeling and Simulation, 2004.
- [18] Marek Natkaniec and Andrzej R. Pach, "An Analysis of the Modified Backoff Mechanism for IEEE 802.11 Networks," PGTS conference, 2000.
- [19] Zuyuan Fang, Brahim Bensaou and Yu Wang, "Performance Evaluation of a Fair Backoff Algorithm for IEEE 802.11 DFWMAC," MOBIHOC' conference, 2002.
- [20] Yi Liu, Chau Yuen, Jiming Chen and Xianghui Cao, "A Scalable Hybrid MAC Protocol for Massive M2M Networks," IEEE WCNC, 2013.
- [21] Asis Nasipuri, Jun Zhuang and Samir R. Das, "A Multichannel CSMA MAC Protocol for Multihop Wireless Networks," Wireless Communications and Networking Conference, 1999.
- [22] Wikipedia, "IEEE 802.11," <http://en.wikipedia.org/wiki/IEEE802.11>.
- [23] Wikipedia, "Jain's Fairness Index," <http://en.wikipedia.org/wiki/Fairnessmeasure>.
- [24] Injong Rhee, Ajit Warriar, Mahesh Aia, Jeongki Min and Mihail L. Sichitiu, "Z-MAC: A Hybrid MAC for Wireless Sensor Networks," IEEE/ACM TRANSACTIONS ON NETWORKING, 2008.
- [25] Gang Qiang, Zengji Liu, Susumu Ishihara and Tadanori Mizuno, "CDMA-based Carrier Sense Multiple Access Protocol for Wireless LAN," Vehicular Technology Conference, 2001.

- [26] I-Sheng Liu, Fambirai Takawira and Hong-Jun Xu, "A Hybrid Token-CDMA MAC Protocol for Wireless Ad Hoc Networks," IEEE TRANSACTIONS ON MOBILE COMPUTING, 2008.
- [27] F. A. Tobagi and L. Kleinrock, "Packet switching in radio channels: Part 2 the hidden terminal problem in carrier sense multiple-access and the busy-tone solution," IEEE Transactions on Communications, 1975.
- [28] Yawen D. Barowski and Saad Biaz, "Towards the Performance Analysis of IEEE 802.11 in Multi-hop Ad-Hoc Networks," Wireless Communications and Networking Conference, IEEE, 2005.
- [29] Jangeun Jun, Pushkin Peddabachagari and Mihail Sichitiu, "Theoretical Maximum Throughput of IEEE 802.11 and its Applications," Proceedings of the Second IEEE International Symposium on Network Computing and Applications, 2003.