THE COOPER UNION FOR THE ADVANCEMENT OF SCIENCE AND ART

ALBERT NERKEN SCHOOL OF ENGINEERING

Capture-Exploited Fair Rate Adaptation for 802.11 WLANs

by

Seung Hun Kang

A thesis submitted in partial fulfillment

of the requirements for the degree of

Master of Engineering

September 16, 2013

Advisor

Dr. Sam Keene

THE COOPER UNION FOR THE ADVANCEMENT OF SCIENCE AND ART

ALBERT NERKEN SCHOOL OF ENGINEERING

This thesis was prepared under the direction of the Candidate’s Thesis Advisor and has received approval. It was submitted to the Dean of the School of Engineering and the full Faculty, and was approved as partial fulfillment of the requirements for the degree of Master of Engineering.

Dr. Teresa A. Dahlberg
Dean, School of Engineering

Dr. Sam Keene
Candidate’s Thesis Advisor

Acknowledgments

I would like to thank my advisor, Dr. Sam Keene, for his knowledge, teachings, and patience in guiding me through this research. His introduction to the topics and advices helped me to get into this project and develop it into what it is today. I would also like to thank my friends, and other faculties for supporting me through the whole process. I would like to especially thank Christopher Sang for helping me out with ns-3 and giving me advices to make this project better. It would have been very difficult without him. Lastly, I want to thank my family for always being there and supporting me.

Abstract

One of the biggest problems that current IEEE 802.11 based WLANs are facing today is that the wireless networks are getting increasingly congested, but there is no rate adaptation scheme that can properly select rates to provide good performance to everyone in the network. It is difficult as there are two things to consider when selecting rates: channel errors and frame collision errors. As networks become more congested, number of collisions also increase and pose a difficult problem for rate adaptations to mitigate both types of error. Many schemes have been proposed to try to differentiate between these two types of errors and adapt accordingly. However, most of these schemes do not take the physical layer capture effect into account. The capture effect allows stronger frame to be successfully received in a collision. This leads to a problem where nodes that are close to a receiver or Access Point (AP) will have stronger received signal and better chance to be captured in a collision, leading to unfairness in the network. Thus, while the overall throughput might be good, in reality, only select few close to the AP will have good performance while others will have terrible performance. In this paper, the interplay of rate adaptation and the capture effect is closely investigated. This analysis is then used to propose a novel rate adaptation scheme called, Capture-Exploited Fair Rate Adaptation (CEFRA) that will maintain good throughput as well as good fairness regardless of channel conditions. Through simulations, CEFRA is shown to prevent few close nodes from dominating access to a channel while maintaining high overall throughput.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Over the past decade and a half, IEEE 802.11 has been accepted and ushered in as the standard for Wireless Local Area Network (WLAN). It has essentially become the norm in which personal electronic devices communicate wirelessly. With ever increasing portable devices such as laptops, smartphones, and tablets that all rely on WLAN for communication, the use of the standard will only grow in time. Due to this huge increase in personal devices and the convenience of wireless networks, it is fairly easy to see why most homes and many businesses are using and providing WLAN services, more commonly known to the public as Wi-Fi.

However, despite its wide adoption, there are numerous problems with current implementation of WLANs. The increase in number of devices is leading to more dense and congested network. Whereas it was uncommon to see more than couple of devices connected to a wireless network even few years ago, it is common to see more than ten devices connected at a home network and many times more than that in a public places such as a café or library. The problem gets even worse with emerging multimedia applications such as voice- and video-over-wifi. While, some of these effects are mitigated by the current implementation of WLAN with flow-control and dominance of downlink traffic, it is hard to imagine that the technology will sustain performance given the current trend [13]. WLAN is not designed to perform well under a congested and saturated network as it does not detect collisions like wired networks do. It merely tries to avoid collisions. In such environment, a wireless network suffers from both link errors associated with the channel as well as these increasing collision errors. As a result, the performance is severely degraded with increasing number of nodes/devices on a network due to packet collisions.

One way of improving the performance and the throughput of WLAN is by using rate adaptation. IEEE 802.11 specifies several rates that are to be supported. For example, IEEE 802.11a has 8 different data rates that it can send packets at. The rates are meant to change corresponding to the channel condition at the moment as wireless channels are susceptible to many variations. While higher rates would deliver packets faster, they are more prone to channel or link errors. Lower rates are more robust to these channel errors. Thus, a rate adaptation algorithm that can select the optimum rate given the channel condition would improve the system performance. Rate adaptation algorithm is not specified in the 802.11 standards and is thus left to manufacturers to implement.

Due to lack of much information available at each node and the simplicity of its algorithm, Auto Rate Fallback (ARF) has been the most widely implemented rate adaptation. It is a very simple algorithm where consecutive packet failures trigger downshift to the next lower rate and consecutive packet successes trigger upshift to the next higher rate. Its simplicity and conservative nature works fairly well in a non-congested network where most packet errors are from the channel, but it performs poorly in a congested network due to its inability to differentiate collisions from channel errors

[7]. It assumes that all errors are channel errors and changes rates accordingly. Despite all the shortcomings and poor performances, ARF has been the most commonly used rate adaptation since the early days of 802.11 because it is simple to implement and does not have any overhead or require any extensive knowledge of its network.

For years, many different and more sophisticated rate adaptations have been researched and proposed to replace the ancient ARF. In general, most of these new adaptations try to get more information on the channel to better make better rate selections. They do so by using some kind of feedback from the receiver. Most open-loop feedback adaptations try to obtain more information from the lone feedback that it gets back from the receiver, which is a small frame known as Acknowledgement (ACK). The receiver sends ACK back to the transmitter if the data is received successfully and the reception of ACK back at the receiver is considered a successful transmission. Newer rate adaptations such as Adaptive Automatic Rate Fallback (AARF) [10], Onoe [3], and Adaptive Thresholds [5] all use the same basic mechanism as ARF, but extract more information out of the channel and use it more opportunistically to improve throughput.

There are also rate adaptations with loss differentiation or sometimes called closed-loop feedback rate adaptations, that utilize additional packets to observe the channel. They use a special feature in 802.11 known as Request-to-Send (RTS) and Clear-to-Send (CTS) frames. It is a four-way handshake method, in which the transmitter sends out RTS and can only send data packets after receiving CTS back from the receiver. It was originally developed as a way to solve the hidden node problem, which happens when more than one node send packets at the same time because they are out of range from each other and cannot sense that a transmission is taking place already. RTS/CTS is rarely used in a standard transmission since it adds an additional overhead of having to send two extra packets before the data transmission. Rate adaptations such as Collision-Aware Rate Adaptation (CARA) and Robust Rate Adaptation Algorithm (RRAA) use RTS/CTS as a feedback on the channel conditions [12]. Generally, these adaptation schemes provide more information about the channel since it is possible to distinguish between channel and collision errors. As for the overhead involved with RTS/CTS, they try to mitigate it by activating RTS/CTS only when standard transmissions fail and it is desirable to distinguish between the errors. These algorithms tend to perform better in a more congested network.

Additionally, there are other rate adaptations that are categorized as SNR-based and throughput-based. These schemes use other indicators to select rates rather than just using frame errors. Throughput-based adaptation such as the SampleRate [8] algorithm selects the rate that minimizes the mean packet transmission time. SNR-based adaptations use signal-to-noise ratios (SNR) to determine the rate to send packets since in a collision-free channel SNR determines whether a packet will be successfully received or not. While all these new proposed adaptations are a huge improvement over ARF, none of them gained wide adoption as each of these adaptations has disadvantages and limitations that are too troublesome.

Rate adaptation is a crucial component of WLAN that affects performance, but it is only one part of the equation. There is only so much that rate adaptations can do to avoid and distinguish collisions. Frame collision is an inevitable event that occurs when transmitters send frames to same receiver simultaneously. It is especially harder to avoid collisions in an infrastructure 802.11 WLAN networks because in most cases, transmitters are all trying to send it to the same receiver, which is the AP. In the analysis of the IEEE 802.11 Media Access Control (MAC) layer, which is the layer that a rate adaptation is implemented on, it is often assumed that a collision destroys all the frames that are transmitted at the same time. Fortunately, not all of the frames are lost when collisions occur. In practice, at the physical (PHY) layer, a phenomenon known as the capture effect allows strongest signal to be demodulated and be received successfully despite the presence of other interfering signals. Thus, if the difference in their signal strength is sufficiently big enough, one frame will be able to survive a collision. Since the capture effect reduces the number of failures, it has been shown analytically [21] to improve the overall network throughput. This result was further confirmed by experimental studies that examined the capture effect with real deployments [15].

The capture effect has a significant influence in not only the overall network performance, but also in individual performance as well. Since the capture effect tries to capture the strongest signal out of all the simultaneous transmissions, it has tendency to favor stations with highest signal strength, which is often the closest nodes to the receiver due to the signal path or propagation loss [16]. This produces the near-far problem in which closer nodes will perform better and have higher throughput whereas farther away nodes will be at a disadvantage and have lower throughput. It leads to overall network throughput improvement at the expense of favoring certain nodes over others. This imbalance produces what is known as the fairness problem in a capture-enabled network. Other channel variations such as multipath fading and shadowing also have an effect on how the capture effect works [20]. They are shown to allow more captures to occur given homogeneous links. Better understanding of the capture effect will allow better analysis of wireless interference and enable wireless networks to reach higher capacity. The capture effect and its implications are not considered in many facets of WLAN at the moment, but it should be taken into account when designing an algorithm to achieve better and more balanced performance.

One of the reasons why there are not many studies and algorithms that take the capture effect into account is that it is hard to accurately model the PHY layer while developing algorithms for the MAC layer such as the rate adaptation algorithms. The PHY layer controls how each frame is delivered on the most basic level and involves things like encoding and modulation. The MAC layer controls how each node should behave and communicate with each other in a shared medium. Often each layer is simulated or tested separately, independent of each other, thus providing less accuracy and transparency than if both are used in conjunction with one another. For example, popular network simulators such as ns-3 [1] and OPNET all use abstract and probabilistic models to estimate the PHY layer. The PHY layer would typically be simulated and studied with programs such as MATLAB. Lack of such simulator that can simulate both layers leaves hardware-based experimental research as the other sole viable option if an accurate study of real-life network is required. Experimental studies such as [13-15] are very time and resource heavy and cannot easily be done. These difficulties and limitations make it hard to study and implement a cross-layer design.

1.2 Statement of Problem

This paper is intended to address some of the key problems that exist in the wireless networks today. Through this work, comprehensive detail on how the physical layer capture effect and a rate adaptation interact with each other and how they affect the network performance is observed. With this observation, the problem of poor performance in increasingly congested and dense wireless networks provided today is addressed. The poor performance is attributed to the frame collisions that occur in these highly congested networks. High number of frame collisions has a huge impact on the wireless networks due to insufficient rate adaptation algorithms that fail to take collisions and the corresponding capture effect into account. This leads to poor overall throughput as well as unfair distribution of the throughput.

First, to solve this problem, a good simulation setup and environment are needed. As mentioned before, without the proper cross-layer simulation environment, it is not possible to study the interaction between the capture effect and a rate adaptation. There are ways to simulate the MAC layer and there are ways to simulate the PHY layer, but most simulators cannot do both simultaneously. A platform that can fully simulate the PHY layer along with the MAC layer is required to resolve the problem.

Second, while there have been many investigations that proposed new rate adaptations algorithms to improve the throughput of wireless networks, all of them had some kind of limitations and disadvantages that made them hard to be used in real-life deployments. Also, while the capture effect has shown to improve the throughput of wireless networks, it has a tendency to favor nodes that have the strongest signals, which are usually the closest nodes to the AP. This creates a problem, where users further away do not receive a fair share of network bandwidth. It is yet to be addressed and resolved by any rate adaptation algorithm to the best of our knowledge.

Lastly, a new rate adaptation algorithm that can provide proper balance between overall throughput and fairness needs to be devised. This new rate adaptation needs to be simple and be easily deployable in the current WLAN infrastructure. An algorithm that can provide both overall system throughput and fairness in a congested network is ideal for the strenuous conditions that modern wireless networks face.

1.3 Proposed Solution

This work presents a detailed and accurate study of the interplay between the capture effect and a rate adaptation and finds the right model of rate adaptation that takes the capture effect into account. It is tested using a cross-layer simulator that fully simulates the PHY layer on top of the ns-3 simulator to accurately observe all the details of the capture effect and rate adaptations together. With this information, a new novel rate adaptation scheme called CEFRA is proposed. The role of CEFRA is to provide a simple rate adaptation that performs well under various conditions while providing fairness to all the nodes, especially under a heavily congested network. It is easily implementable in the current 802.11 WLAN infrastructure without much change.

CEFRA achieves both good fairness and throughput by exploiting the fact that higher bit-rates lower the chance of capture for stronger nodes. This characteristic is used in setting the rate for RTS frames based on the nodes' relative positions to the AP. This reduces the number of collisions by using small sized frames at faster rates. When collisions do happen, lowered capture probability for closer nodes lead to fairer distribution of access.

The rest of this paper is organized as follows. Chapter 2 explains the underlying background mechanisms of IEEE 802.11 WLAN and the simulation setup. Chapter 3 describes the physical layer capture effect and what its implications are. Chapter 4 classifies different rate adaptation algorithms and their characteristics. Chapter 5 investigates the how these two components affect each other and outline the proposed algorithm, CEFRA. Results are discussed in Chapter 6 and finally, concluding remarks and recommendations are made in Chapter 7.

Chapter 2

IEEE 802.11 Wireless LAN

2.1 Overview

A wireless local area network or WLAN is a link of two or more devices through wireless medium and provides a connection to the internet through an access point. This enables mobility as well as convenience compared to wired networks. Most modern WLANs use IEEE 802.11 as the standard. IEEE 802.11 specifies how the PHY layer and the MAC layer should be implemented on a WLAN. IEEE 802.11 WLAN refers to such wireless network that uses IEEE 802.11 as the standard and is what most people refer to as a Wi-Fi network. Devices that use IEEE 802.11 are simply marketed under the Wi-Fi brand name. WLAN or Wi-Fi has become popular in recent years due to ease of installation and use. Today, it has become the primary choice of connection to the internet in homes and businesses.

The base version of IEEE 802.11 was initially developed in 1997, but has seen numerous amendments made to it over the years to improve it. The first widely used version was the 802.11a, which uses the 5 GHz U-NII band. This was followed by 802.11b and 802.11g, which are more widely used. These are employed on the 2.4 GHz ISM band. 802.11a and 802.11g are essentially the same protocol, both using signaling method known as orthogonal frequency-division multiplexing (OFDM) but in different bands as mentioned. 802.11b uses a different method known as direct-sequence spread spectrum (DSSS). These methods enable these protocols to be more resistant to interference, which is especially problematic in the 2.4 GHz band. 2.4 GHz is shared with other electronic devices such as microwave ovens, cordless phones, and Bluetooth devices. 802.11a and 802.11g have the maximum data rate of 54 Mbps while 802.11b is limited to 11 Mbps.

Most recent versions, 802.11n and 802.11ac are both based on OFDM and they improve on the performance by using multiple streams, bigger bandwidth, and higher order modulations. 802.11n operates on both 2.4 and 5 GHz, while 802.11ac uses the less crowded 5 GHz band. All of these protocols are capable of sending data at different rates and the performance improvement comes from being able to send at higher rates. Currently, 802.11g is the most widely used version and 802.11ac has yet to be fully incorporated into the standard. Differences between 802.11a, b, and g are outlined in Table 2.1. More details on OFDM are provided in the next section.

802.11 Protocol	**Frequency (in GHz)**	**Data Rates**	**Approx. Indoor Range**	**Modulation**
a	5	6, 9, 12, 18, 24, 36, 48, 54	35 m	OFDM
b	2.4	1, 2, 5.5, 11	35 m	DSSS
g	2.4	6, 9, 12, 18, 24, 36, 48, 54	38 m	OFDM

Table 2.1: Difference between 802.11 Protocols

There are two modes that WLANs can use. One is known as the ad-hoc mode and the other is known as the infrastructure mode. The ad-hoc mode is primarily used to communicate between peer to peer, meaning there is no base station and users form their own wireless network to communicate with each other. The infrastructure mode is the mode that everyone is familiar with and employs. Stations or nodes refer to any device or component that can connect to a network. In the infrastructure mode, there is a base station known as the AP that transmits and receives all the communication from the stations. The AP is simply a wireless router in most cases. A typical setup of the infrastructure mode consists of stations communicating through the AP that serves as a bridge to a wired network infrastructure. This is typically seen with the setup of such a network when a wireless router is connected to a modem via Ethernet cable.

The main disadvantage of WLANs over the wired LANs is the unreliable channel. As previously mentioned, the 2.4GHz is heavily congested with many other electronic devices interfering as well as noise that is present. Also, due to the mobility and the openness of the medium, there are other difficulties such as path loss, shadowing, and multipath fading. Lastly, frame collisions, which happen when two or more stations send to the same receiver simultaneously, are a big problem, especially in a congested network. They are prevalent for all a, b, and g versions of the 802.11 networks.

2.2 PHY Layer

The physical layer is the lowest layer in networking. It is the fundamental layer that implements how data is transmitted and received. It defines how raw bits of data are transmitted over a physical link. Everything from frequencies to broadcast on to modulation schemes and coding is all part of the physical layer. In 802.11 WLAN, with the exception of 802.11b, all the other versions of the protocol use OFDM as their method of encoding data and it is the main component behind the WLAN PHY layer. OFDM's main advantage and the reason it is being used in WLANs, is its ability to perform well in unreliable channel conditions. For example, it is able to cope with frequency-selective fading without complex time-domain equalization, which is harder to implement.

Basic principle behind OFDM is that multiple closely spaced orthogonal sub-carrier signals are used to carry data on parallel channels. Each sub-carrier is modulated conventionally at a low symbol rate. The total data rate is similar to conventional single-carrier modulation that uses higher symbol rate. In essence, one rapidly modulated wideband is broken down into multiple slow narrowband signals. This frequency diversity allows OFDM to be more robust to frequency-selective fading. The sub-carriers are chosen to be orthogonal to each other, which allows for high spectral efficiency and efficient implementation of the Fast Fourier Transform (FFT). Lower symbol rate also allows the use of a guard interval between symbols, which helps to mitigate intersymbol interference (ISI).

Another part of the PHY layer that is used to combat the noisy and unreliable channels is forward error correction (FEC) or more specifically, convolution encoding, in 802.11. FEC combined with interleaving in OFDM allows the bit-errors to be corrected. The reason for interleaving is that a burst of errors cannot be resolved using just error correction coding and interleaving attempts to spread out the errors in the bit-stream. In effect, all of these features in OFDM make it a very suitable choice for implementation in 802.11. All of these features help to combat the severe conditions of wireless channels.

In an effort to strike a balance between speed and reliability, all the versions of the 802.11 provide different modulation complexity and different coding rate. Different data rates offered by each protocol are a combination of modulation type and coding rate. A complete list of all eight data rates offered by 802.11a and 802.11g are outlined in Table 2.2 along with their corresponding modulation scheme and coding rate. Lower coding rate and lower order modulation would result in slower bit rate, but it would be more robust to errors, thus more likely to be received correctly. Finding the right rate to use based on the channel condition is the role of a rate adaptation.

Data Rate (in Mbps)	**Modulation Type**	**Error Correction Coding Rate**
6	BPSK	1/2
9	BPSK	3/4
12	QPSK	1/2
18	QPSK	3/4
24	16-QAM	1/2
36	16-QAM	3/4
48	64-QAM	2/3
54	64-QAM	3/4

Table 2.2: Data Rates and the Corresponding Modulation and Coding Rate

2.3 MAC Layer

IEEE 802.11 specifies two types of MAC operation known as the Distributed Coordination Function (DCF) and the Point Coordination Function (PCF). The DCF is the mandatory and default channel access scheme used in almost all modern WLANs and will be the only scheme considered throughout this paper. The DCF mechanism is based on Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) with binary slotted exponential backoff. The operation of the CSMA/CA protocol is illustrated in Figure 2.1.

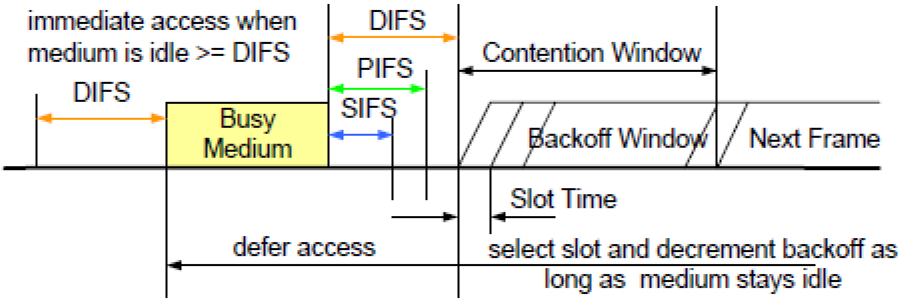


Figure 2.1: Basic Access CSMA/CA Protocol [7]

In a typical CSMA/CA operation, when a station with a packet to send, senses that the channel is idle for certain period of time known as Distributed Inter-Frame Space (DIFS), the transmission proceeds. If the channel is sensed busy, the transmission is deferred and it initializes a backoff counter. It picks a random value uniformly distributed between 0 and the contention window (CW) and starts decrementing the counter by one after each time slot as long as the channel remains idle. The countdown is paused if it senses that the channel has become busy. The countdown resumes after sensing that the channel has been idle again for DIFS.

Once the countdown reaches zero, the station transmits the packet. After successfully receiving the packet, the receiver waits Short Inter-Frame Space (SIFS) and then sends an ACK frame. SIFS is considerably shorter than DIFS and ensures that the ACK has priority over other transmissions. If the sender does not receive an ACK, it assumes that a collision occurred because the backoff counter for other station has reached 0 at the same time and doubles the CW value and re-enter the backoff process to try to retransmit the frame. Upon consecutive unsuccessful transmission attempts, the CW is doubled up to a maximum value $CW_{max} = 2^{m}CW_{min}$, where m is the maximum backoff stage. Upon a successful transmission, the CW is reset to CW_{min}. This mechanism makes DCF a Collision Avoidance (CA) protocol since it tries to reduce the collision probability by using exponentially increasing backoff model. Even with the random backoff, collisions are inevitable when stations end their backoff simultaneously and the problem becomes worse as the number of contending stations increases.

Additionally, the DCF may also employ the RTS/CTS mechanism as shown in Figure 2.2. It follows the same medium access rules as the regular transmission described above, but the transmitter has to send a short RTS frame first, followed by a short CTS frame from the receiver before the data frame can be sent. The RTS/CTS mechanism is developed to prevent collisions from the hidden node problem. The hidden node problem is illustrated in Figure 2.3 and occurs when nodes are out of range from each other, but are in range of the receiver. Thus, they cannot sense each other and send at the same time, causing collisions. By using the RTS/CTS, the channel is reserved within the range of the receiver, mitigating collisions.

The decision to use the RTS is made solely on the transmitter side and is controlled by the RTS threshold. If the frame size is bigger than the threshold, the mechanism is used. Typically, the threshold is set to the largest value and RTS/CTS are rarely used in real life due to the overhead and the latency involved with the exchange. However, it has been proven that the use of RTS/CTS is useful in high contention WLANs and the benefit is amplified with relatively large data frames [4].

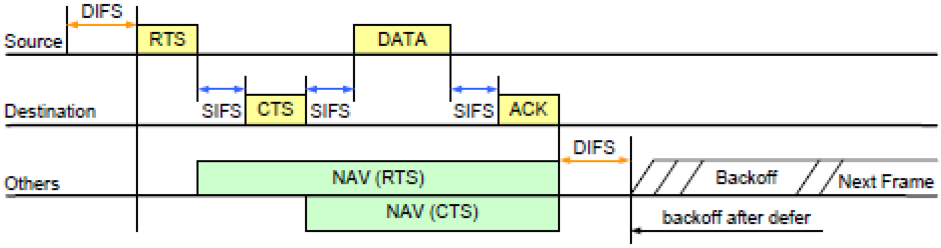


Figure 2.2: RTS/CTS Exchange in the CSMA/CA Protocol [7]

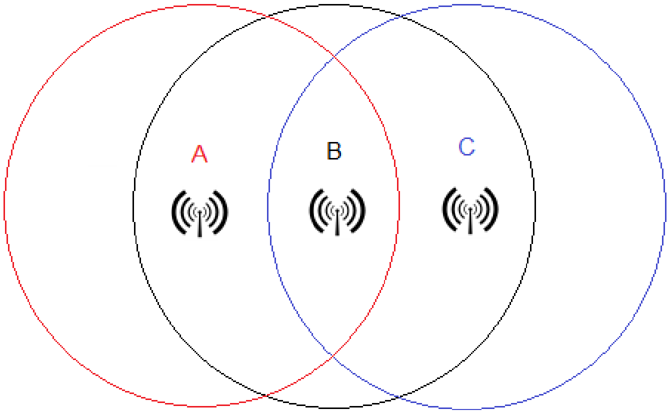


Figure 2.3: Hidden Node Problem

2.4 Testing Environment (ns-3 with PhySim-WiFi)

As described above, the PHY and the MAC layer serves two vastly different purposes in WLAN. As such, it is not an easy task to model these two layers completely working side by side in a network simulator. Most network simulators are discrete-event driven and excel at simulating the MAC layer and higher layer protocols. They provide an easy and quick way of evaluating new proposals for various routing and MAC protocols. However, they employ a rather simple SNR-based bit error rate model to simulate the PHY layer. While this approach will simplify and speed up simulations, it is just not accurate enough for WLANs, where there are so many uncertainties and factors that can affect the performance. This inaccuracy is especially profound with the interference model that most simulators use for a wireless channel. The authors of [23] found that different interference models and their level of detail made a significant difference in the results that they produced. They recommend that complex interference models such as one based on SINR (signal-to-interference-and noise ratio) should be used to correctly simulate WLANs and that the results based on simplified interference models be accepted with caution.

It is fairly easy to see why network simulators opted to use simple error model to estimate the PHY layer as it would be extremely computational heavy to try to fully emulate what a real transceiver would do for the PHY layer. However, to properly study collisions and the capture effect, it is essential to use a full PHY layer simulator. In an effort to do so, a Bi-layer WLAN simulator was designed and implemented on MATLAB. It provided a simple time-based CSMA/CA protocol with the full PHY layer implementation where a frame from the MAC layer would result in a full OFDM-modulated frame being sent and decoded through MATLAB. While it provided the accuracy, the implementation suffered heavily from long run time and numerous bugs. It was extremely slow to try to simulate a network with respect to time and with such small intervals of time (i.e. μs) to accommodate all the features. Due to these drawbacks, this implementation was abandoned.

The simulator that was used for this research was ns-3. While ns-3 is a popular discrete event network simulator used in many research, it still had the same problem of simplifying the PHY layer. To remedy the problem, a separate implementation known as PhySim-WiFi [2] was used to replace the default physical layer implementation in ns-3. The default physical layer in ns-3, YansWifiPhy, implements a packet-level PHY model which abstracts channel effects on individual packet bits by using average bit-error rates. PhySim-WiFi is an accurate implementation of the OFDM-based IEEE 802.11 that simulates the underlying signal processing steps of a transceiver down to the signal level to determine whether or not a packet is received correctly or not. All the lower-layer techniques such as interleaving, forward error correction and OFDM modulation are all applied. Thus, use of this module adds the extra accuracy that was required while being able to access the vast resource of existing simulation code and setups in the ns-3. It is a cross-layer simulator that bridges the gap between the physical layer emulation and network simulation [24]. Figure 2.4 shows how PhySimWiFi connects to the existing WiFi MAC implementation in ns-3 and Figure 2.5 shows the state machine of the physical layer and the transitions between the states. It should be noted that while the accuracy is increased, computational effort is also increased. Using PhySimWifi can be up to 1000 or 10000 times slower than the default YansWifiPhy implementation. Also, it currently only support the 5Ghz band OFDM modulation, 802.11a protocol, but it is sufficient as all other modern WLANs are based on the same principles.

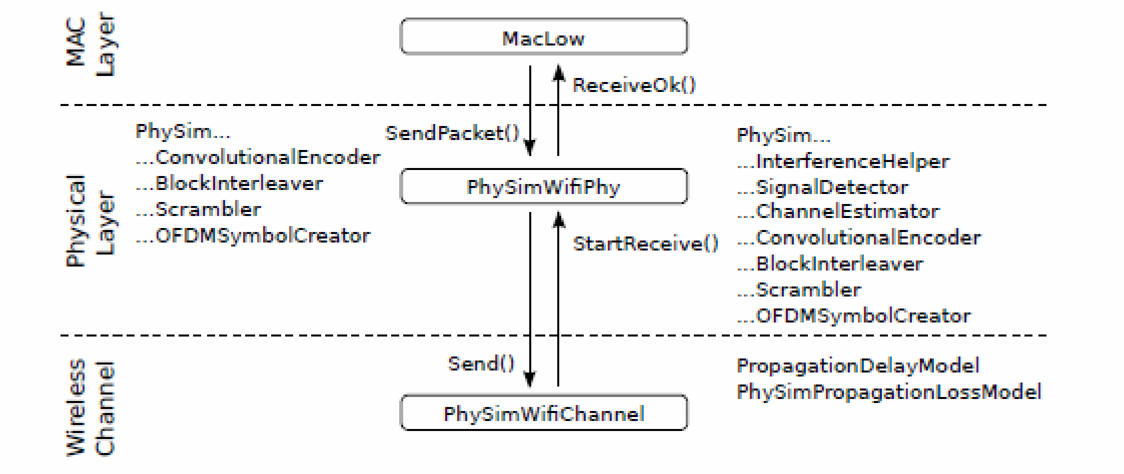


Figure 2.4: PhySim-WiFi Integration with ns-3 [2]

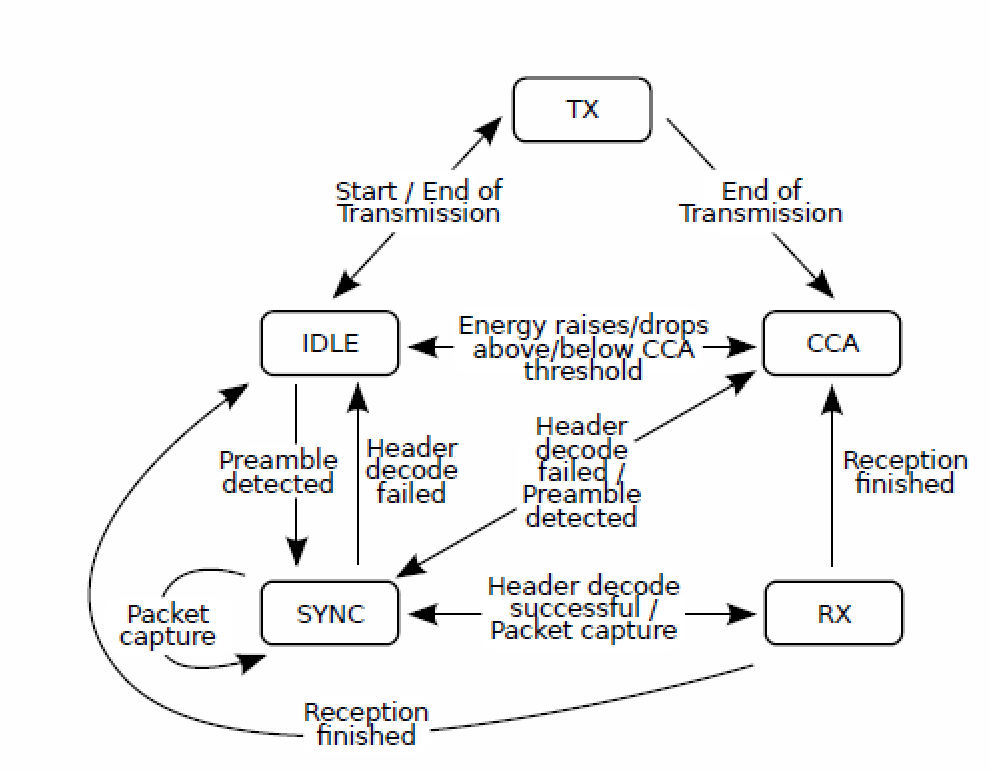


Figure 2.5: State Machine Diagram of PhySim-WiFi [2]

Chapter 3

Physical Layer Capture Effect

3.1 Overview

Unlike wired networks, wireless networks experience a lot of collisions due to its half-duplex and contention nature. It is one of the biggest problems of wireless networks that is yet to be fully addressed. There are many interference cancellation and joint detection studies that are promising and in the works, but none of them are quite feasible yet for wireless networks due to the complexity of these algorithms. Fortunately, wireless networks do not destroy all the frames in a collision, although many studies assume that they do for simplicity. In practice, at the physical layer, a phenomenon known as the capture effect allows the frame with the strongest signal to be demodulated in the presence of other interfering signals [25]. Thus, as long as there is a sufficiently big signal difference, one frame can be successfully received in a collision [17].

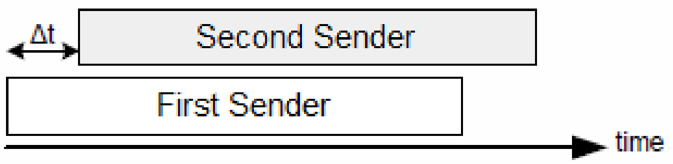


Figure 3.1: Capture Scenario

A typical capture effect scenario can be seen in Figure 3.1. As can be seen in the figure, first frame arrives to the receiver first when the second frame arrives to the same receiver. When the receiver detects the energy increase due to the second frame, the receiver can try to receive and decode one of the two frames. If the energy increase is above a certain threshold known as the capture threshold, the receiver gives up on the first frame and moves on to the second frame. It should be noted that message in message (MIM) mode should be enabled in 802.11 radios for this sequence to happen because it needs to begin the retraining process, which involves synchronizing and demodulating the second frame [15]. If the energy increase is below the threshold, the receiver gives up the second frame and tries to receive the first frame in normal mode. While this scenario only considers two frames, same procedure applies for multiple frame collisions.

The capture effect can thus be broken down into two stages: the preamble detection stage and the frame check sequence (FCS). The retraining of the second frame requires that the preamble of the new frame be successfully be detected. This is the preamble detection stage. Once the preamble detection stage passes, the frame body needs to be successfully decoded in the presence of noise and other interfering signals. At the end, the frame check sequence must succeed, and this is known as FCS check stage. These stages need to be successfully passed in order for a successful capture to occur.

3.2 Analytical Performance

There have been many different studies that tried to accurately model the capture effect analytically to study its behaviors [18, 20-22]. Although these probabilistic models are only an estimate and may not accurately show all the behaviors, they show a general impact of the capture effect such as change in throughput under different loads. One model of the capture effect is the bit decision process under interfering packets and noise model, and properly relates the capture effect to the type of demodulation and the coding scheme [20]. This model is more accurate than a simple model where only the power levels of received packets are considered. In the bit decision process model, the capture effect occurs only if there is no bit error in a packet. Thus, the probability that the strongest packet will receive is given by:

$$P_{N+1}(z) = \left(1 - P_b(z)\right)^L, \tag{3.1}$$

where L is the packet size in bits, $P_b(z)$ is the bit error rate (BER), and z is the capture ratio. The BER accounts for the type of demodulation, the coding scheme, and the signal-to-interference ratio and thus, provide more accurate estimate of the capture effect. Given that the capture effect happens when the power level is the strongest out of all the incoming packets and that the detected bit sequence matches up to the desired bit sequence,

$$P_{N+1}(z) = \sum_{z=0}^{\infty} \left(1 - P_b(z)\right)^L \Pr\left\{w_0 = z \sum_{i=1}^{N} w_i\right\}, \tag{3.2}$$

where w_i is the received power of the i^th^ packet and the w_0 is the power of the strongest packet. According to [20], in this model, the probability that one out of N+1 packets is successfully received is:

$$q_{N+1} = (N+1)P_{N+1}(z). \tag{3.3}$$

Using these capture probabilities and extension of Bianchi's Markov Model [4] to include capture and other channel variations, the authors of [20] found that an increase in packet size decreases the probability of capture as can be seen in Figure 3.2. Due to this increase in capture probability and increased number of collisions, the overall normalized throughput is increased with capture effect at high traffic load. Furthermore, from looking at Figure 3.3, it is clear that the capture effect improves the normalized throughput by increasing the transmission probability and decreasing the collision probability.

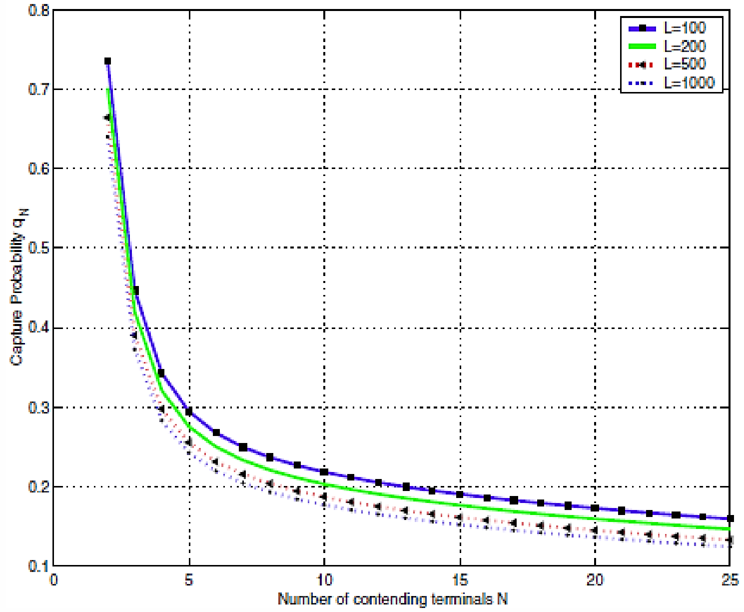
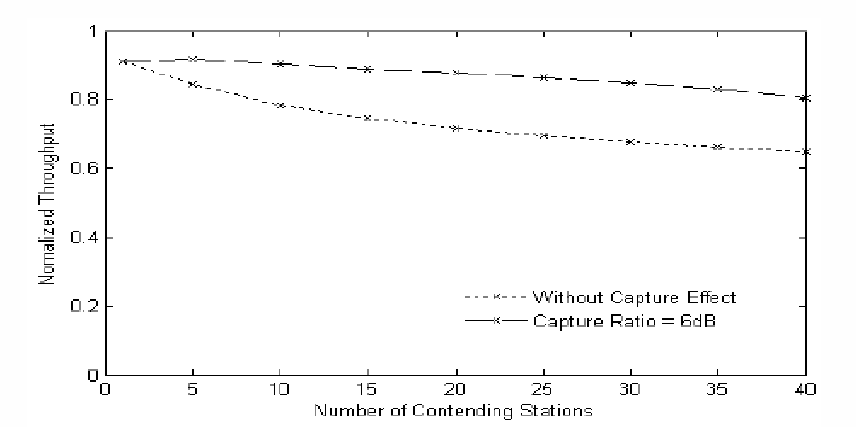
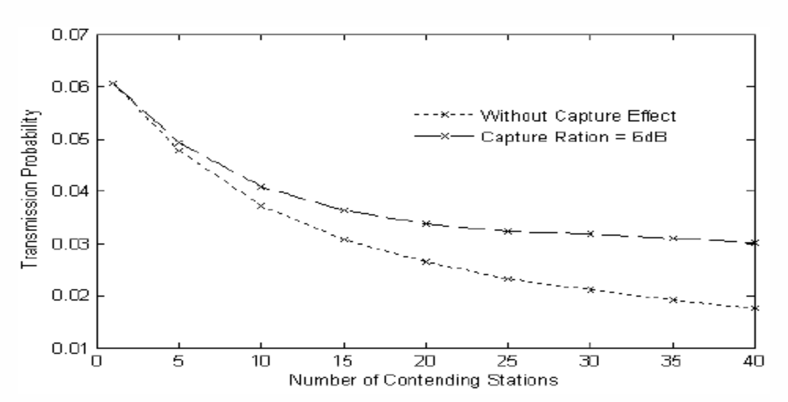


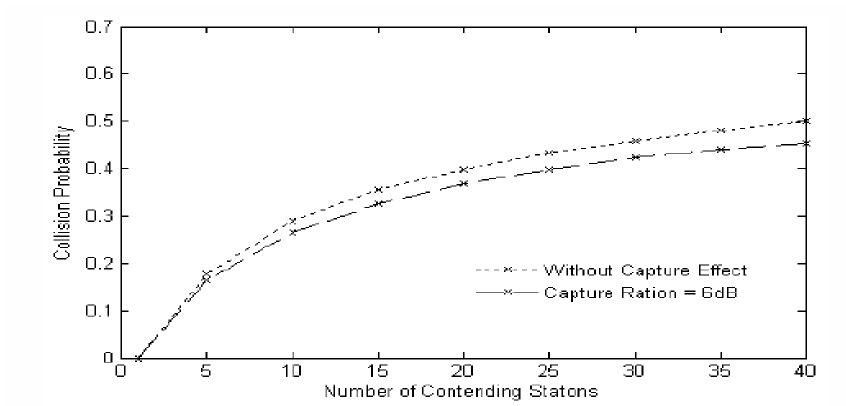
Figure 3.2: Capture Probability of Different Packet Size, L [20]



(a) Impact of the Capture Effect on Throughput



(b) Impact of the Capture Effect on Transmission Probability



(c) Impact of the Capture Effect on Collision Probability

Figure 3.3: Impact of the Capture Effect [18]

3.3 Implication

While the overall throughput of a network might benefit from the capture effect, it also has a side effect that is not desirable in a real wireless network: unfairness. Nodes near the AP have a huge advantage on medium access over the nodes that are more distant due to the capture effect and wireless channel characteristic known as propagation or path loss. Propagation loss is the reduction in power of a signal as it propagates through space. Since all signals are sent at a same power, closer nodes lose less power and have higher received power at the receiver. This causes the capture effect to happen in favor of closer nodes and allow them to capture the channel. This problem is also known as the near-far problem in wireless networking. In an ideal homogenous network environment, with no channel variations, the capture effect is less likely to occur because signals do not have significant strength difference between them.

The consequence of this unfairness problem goes deeper than just losing a packet in a collision. Consider two nodes, i and j, which are unequally spaced apart as shown in Figure 3.4. Assuming that the nodes are distanced such that the ratio of the signal power exceeds the capture threshold, the closer node, i, will be successfully captured over the more distant node, j. After receiving an ACK from the AP, node i will reset its contention window to the minimum, CW_{min}, while node j will double its contention window. After this transmission attempt, the expected backoff counter, b_i, for node i is:

$$E(b_i) = \frac{CW_{min}}{2} \tag{3.4}$$

while the expected backoff counter for node *j* is

$$E(b_j) = \frac{CW_{prev} \times 2}{2} = CW_{prev}, \tag{3.5}$$

where CW_{prev} is the contention window size of node *j* before the transmission. Even assuming $CW_{prev} = CW_{min}$, node *i* will have two more transmission chances before node *j* can get a chance to retransmit the same packet as shown in Figure 3.5. If a collision and the capture effect happen again, same process occurs and the expected backoff counter for node *j* becomes significantly bigger than that of node *i*. Node *i* will send significantly more packets over time and node *j* will have hard time gaining access to the channel.

Figure 3.4: Typical Capture Setup [16]

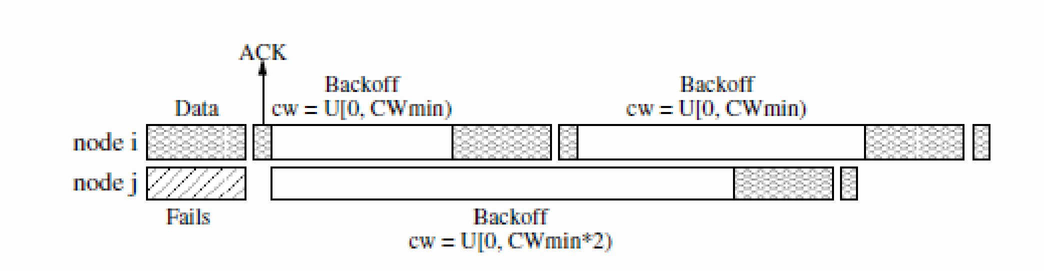


Figure 3.5: Unfair Access Opportunity with the Capture Effect [16]

Bianchi [4] showed that the probability that a node transmits at a given idle slot, τ, and the probability that the transmitted packet collides, p, are:

$$\tau = \frac{2(1-2p)}{(1-2p)(W+1)+pW(1-(2p)^m)} \tag{3.6}$$

$$p = 1-(1-\tau)^{n-1}, \tag{3.7}$$

where W = CW_{min} + 1, m is the maximum backoff stage, and n is the number of nodes. This is assuming that the probability of collision and the probability of transmission in a given slot are identical for all nodes. With distance of each node in mind, authors of [16] estimate the collision probability and the probability that node *i* transmits a packet in an idle time slot are as follows:

$$p_i = 1 - \prod_{\left\{j \mid \frac{P_r(d_i)}{P_r(d_j)} \le z,\ \ j \ne i\right\}} (1-\tau_j) \tag{3.8}$$

$$\tau_i = \frac{2(1-2p_i)}{(1-2p_i)(W+1)+pW(1-(2p_i)^m)}. \tag{3.9}$$

$P_r(d_i)$ and $P_r(d_j)$ are the power of node *i* and node *j* respectively given their distance d_i and d_j. z is the capture threshold. From these equations, it is clear that a node near the AP experience fewer collisions and transmit more packets than a node that is farther away. In effect, node *i* is capturing the channel. This combination of the capture effect and the following binary exponential back-off process is responsible for the unfairness.

3.4 Capture Effect Cases and Characteristics

Many studies assume that the capture effect happens when a collision happens from two or more signals arriving at the same time and the signal of the strongest signal is bigger than the capture threshold, but it is only a simplified estimate of what really happens in real life. Success of the capture effect depends on several different factors. Namely, timing of the collision and the rates of the frames affect the likeliness of the capture effect [15].

In most cases when there is a frame collision, frames that are involved rarely arrive to the receiver at the same time. In general, one frame arrives later than the other, with different rates, and with different frame size, which usually results in interfering over some regions of the frame, but not all. Thus, there are different scenarios in which the capture effect happens. Assuming that there are only two frames colliding, timing relations that characterize distinct capture cases are illustrated in Figure 3.6. In this figure and study done by [15], the sender is the node whose frame is to be captured and the interferer is the node whose frame will be lost in the collision.

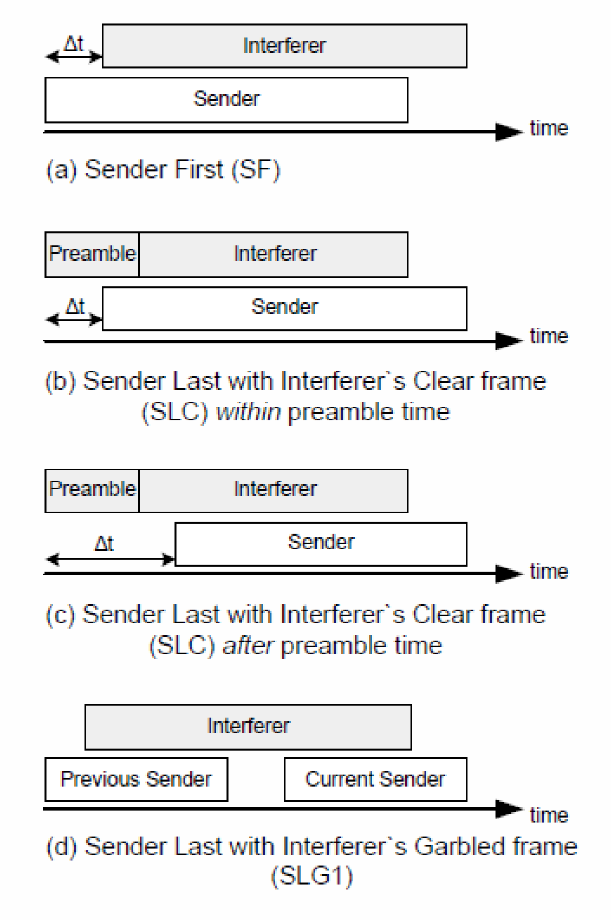


Figure 3.6: Different Capture Scenarios [15]

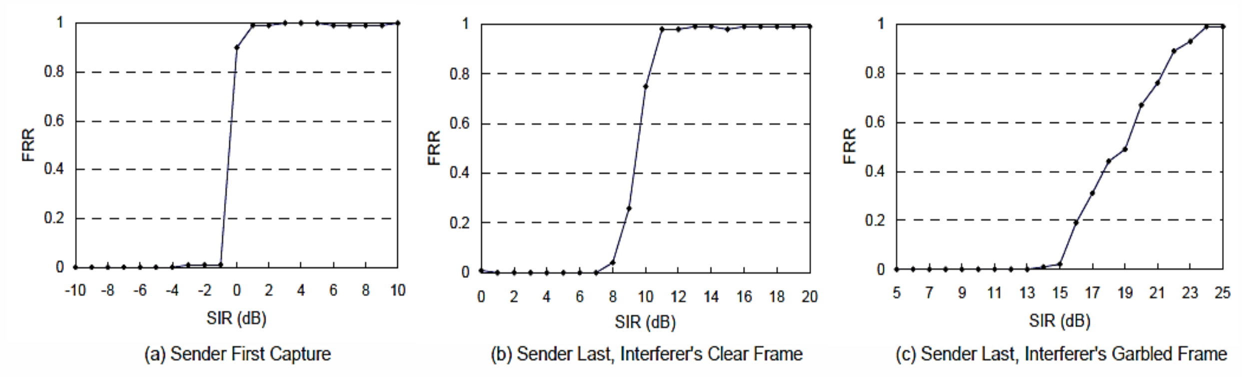


Figure 3.7: FRR vs. SIR for Different Capture Scenarios [15]

In the Sender First capture (SF), the sender's frame arrives to the receiver before the interferer's frame. Frame reception ratio (FRR) of the SF case against varying values of Signal-to-Interference Ratio (SIR) can be seen in Fig 3.7 (a). Data rate is set at 6 Mbps for both the sender and the interferer for the results in Fig 3.7. Given the SF scenario, the sender's frame is received if it arrives before the interferer's and if it is stronger than the interferer's frame. In this case, the preamble detection stage is passed without interference because the sender's frame arrives before the interferer's frame and only the FCS stage suffers from the interference.

In the Sender Last Capture with Interferer's Clear Frame (SLC) case, the sender's frame arrives after the interferer's frame. This can be further broken down into two cases where the sender's frame arrives within and after the preamble time of the interferer's time. It has been shown that despite the fact that the receiver already synchronized to receive the interferer's frame, the receiver can capture the sender's frame with the MIM mode [15]. FRR for the SLC case is shown in Fig 3.7 (b). As can be seen, it takes a lot stronger signal to be captured if it arrives after the first frame.

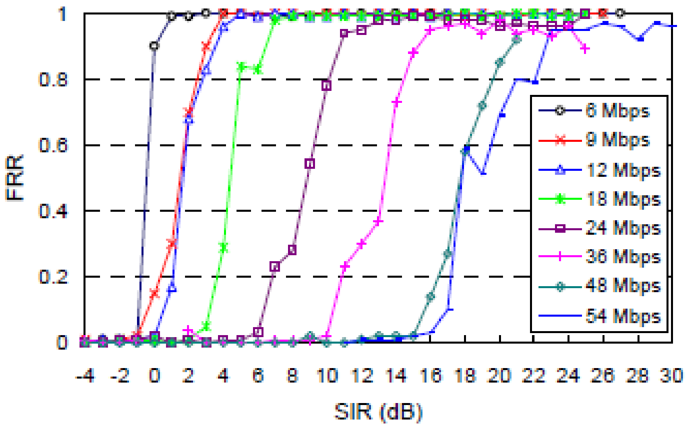
The last case is the Sender Last Capture with Interferer's Garbled Frame (SLG) case. This case is similar to the SLC case, but the first arriving frame is garbled. It can be garbled if the previous arriving frame is being interrupted as can be seen in Fig 3.6 (d) or if it is out of range. If the frame is garbled, the receiver continuously tries to synchronize to it but fails to do so. This hinders the receiver from receiving the new frame that is coming in and makes it harder to capture the new frame. As such, SIR required to capture the frame is even higher than that of the SLC case as seen in Fig 3.7 (c).

Another key factor that influences the capture effect is the PHY bit rate of the sender. Required SIR and corresponding FRR for each rate of 802.11a are shown in Figure 3.8. For the SF case, the required SIR ranges from 0 to more than 20 dB. In this case, only the FCS stage matters since the preamble of the sender's frame is already received and locked onto the frame. The interferer's frame is just considered as white noise and hinders the decoding of the frame body at higher rates. Thus, higher SIR is required to decode the sender's frame at higher rates.

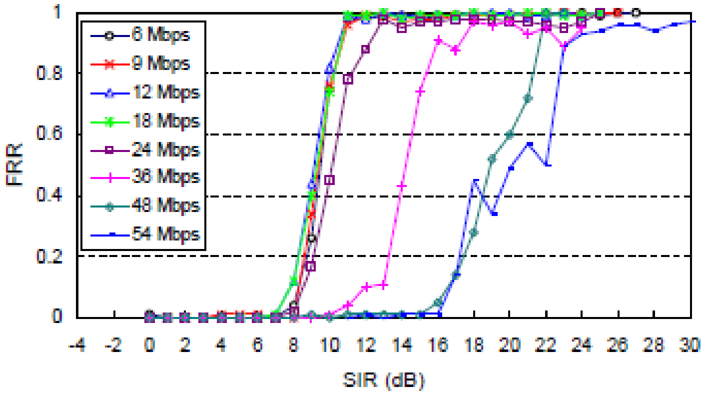
In the SLC case, the SIR threshold tends to stay the same for 6-18 Mbps at around 10 dB. This is thought to be the minimum SIR threshold needed to pass the preamble detection stage. Since a preamble is always sent at the lowest rate regardless of the payload rate, the SIR threshold needed to pass the preamble detection stage is expected to be same for all payload data rates. For rates 6-18 Mbps, the SIR threshold needed for the FCS is equal or smaller than the SIR needed for the preamble detection stage and is determined by the preamble detection stage. However, beyond 18 Mbps, the SIR needed for a successful capture is determined by the FCS stage as evidenced by the similarities of the required SIR in the higher rates between the SF and the SLC case.

Lastly, the SLG case does not seem to show much difference between different rates. The receiver requires high SIR regardless of the data rate since it is already busy trying to lock onto a garbled frame. Due to the difficulty locking onto and synchronizing to the sender's frame, the SIR threshold for the SLG case is determined mostly by the preamble detection stage.

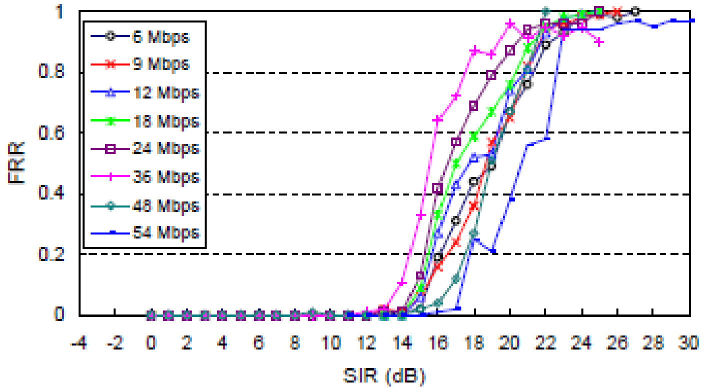
From this study, characteristics of the capture effect can be outlined as follows. First, the two stages of the capture effect have their own SIR threshold to be successfully cleared. Second, the SIR threshold for preamble detection is determined by the state of the interferer’s frame (SLC or SLG), but not by sender’s frame rate. Third, the SIR threshold for the FCS increases as the data rate increases. Fourth, the interferer’s data rate does not seem to affect the capture SIR threshold.



(a) SF capture.



(b) SLC capture.



(c) SLG capture.

Figure 3.8: Required SIR for Different Rates [15]

Chapter 4

Rate Adaptation

4.1 Overview

In recent years, there have been tremendous improvements to the 802.11 protocol, which allows devices to transmit data at high speed. In an ideal world, this would yield great throughput and everyone would be satisfied, but that is not the case. Wireless channels suffer from unstable conditions and users rarely experience the kind of system performance that the new improvements promise. Wireless networks need improvements in rate adaptation to achieve any real significant difference in performance. It is up to rate adaptation algorithms to try to assess the channel conditions and select an optimal rate that will maximize performance.

Over the past decade and a half, rate adaptation for WLAN has been studied extensively. WLAN and the use of it have changed dramatically over the years, and had led to two generations of rate adaptations that differ mostly on the interpretation of frame losses. The first generation rate adaptations implicitly assumed that all losses came from channel degradation with limited congestion losses. This was a valid assumption at the time since WLANs and the devices that connect to them were limited at the time. Channel losses were more significant factor at the time. In many of these rate adaptations, data rate is lowered when a frame is lost. However, this is not a correct response to a collision loss since a lower rate may exacerbate medium congestion because of longer frame transmission duration and wider transmission range (more collision). A lower rate also underutilizes the channel when it can support higher rates [9].

The second generation of rate adaptations brought out loss-differentiation adaptation schemes that diagnosed what caused a frame loss and adapted accordingly. These schemes are often more complex than the non-differentiating ones and use either closed-loop feedback system using the RTS/CTS probes or some other kind of feedback system from the network. Given that each station does not have too much information about the channel, these schemes modify some parts of the protocol to gain more knowledge about the channel. These rate adaptations often outperform the ones without loss differentiation and tend to work better in more congested networks.

Rate adaptations can further be categorized by the type of indicator that they use to change rates. Most commonly used is frame-loss based, which changes rates according to whether frames were successfully received or not. This is indicated by successful reception of ACKs for the corresponding frames. SNR-based schemes use signal strength to gauge the channel and change rates accordingly. Throughput-based schemes select the rate that minimizes mean packet transmission time. For the rest of this chapter, some of the more popular rate adaptations are further explained and discussed.

4.2 Rate Adaptations without Loss Differentiation

Auto Rate Fallback (ARF) [27], originally developed for Lucent Wave-II WLAN devices, is the earliest rate adaptation scheme developed for 802.11 based wireless networks and is still the most widely used rate adaptation scheme to date. Success of ARF comes from its simplicity. It alters the rates by keeping track of only two things: ACKs and timer. If two consecutive ACKs are not received correctly, the next transmission is sent at the next lowest rate and the timer is reset. When the timer reaches zero or number of consecutively received ACKs reaches 10, it uses the next highest rate for the next transmission and the timer is reset. If the first transmission at the new higher rate fails, the sender falls back to the prior rate in what is known as the recovery mode. ARF is the classic frame-loss based scheme in which a lot of the others are based off of. While ARF provides a simple rate adaptation that can adapt to channel conditions, it does not consider collisions and has conservative nature due to its high up-threshold as compared to its down-threshold (i.e. 10 to 2). This conservative nature often underutilizes a channel and yields less than optimal throughput in most cases.

To improve performance in stable environments, authors of [10] proposed Adaptive Auto Rate Fallback (AARF). In AARF, instead of keeping the up-threshold fixed to 10, the threshold is doubled when the first transmission after the rate increase fails. This would reduce rate fluctuations and keep the rates steady for longer period of time. The down-threshold is still kept at 2. AARF is even more conservative adaptation that tries to minimize unnecessary rate up-shift and stay at a rate that is deemed safe for transmission.

ONOE is another early rate adaptation scheme that is also frame-loss based. It is an open source rate adaptation on a Linux driver developed by MadWifi organization for wireless adapters with Atheros chips [3]. It tries to select the highest bit-rate with less than 50% frame loss rate. It keeps a credit score that is incremented if less than 10% of packets required a retransmission and no packets were dropped in the last time period. If the credit score is above a threshold, which is set to 10, the bit-rate is raised. If each data packet required at least one retransmission, bit-rate is lowered and the credit score is reset. The problem with ONOE is that because it adjusts the rate at the end of 1000 ms cycle, it is insensitive to bursty losses and irresponsive to fast changes.

SampleRate [8] algorithm is a throughput-based adaptation that adjusts its rate to the bit-rate that achieves smallest average transmission time in the last sampling period. It initially uses an intermediate bit rate and updates the transmission time afterwards using exponentially weighted moving average (EWMA) based on the number of retransmissions, packet length, and protocol timing overheads. Periodically, it will try a new random rate whose lossless transmission time is lower than the current one. If these sample transmissions yield lower average transmission time, it switches to the new rate. It is meant to achieve the best average throughput in the long term. It suffers when there are only few samples available per node to accurately estimate the transmission time as in the case of highly congested network.

Lastly, Received Signal Strength Link Adaptation (RSSLA) [6] assesses the channel condition based on Received Signal Strength Indicator (RSSI) and uses it to set the rate. A station tries to store RSS from all the stations that it can sense. It adaptively keeps track of all the RSS and updates with each new frame. When a station needs to send a frame, it retrieves RSS of the destination station and maps it to the corresponding rate. While SNR-based such as RSSLA is expected to perform well, it suffers from lack of accurate measurement of RSSI by devices. This is due to the fact that RSSI thresholds for all the rates lie in a small interval of the total RSSI measurement range.

4.3 Rate Adaptations with Loss Differentiation

Collision-Aware Rate Adaptation (CARA) [7] is a frame-loss based closed loop scheme that selectively uses RTS/CTS probes to differentiate losses. It tries to minimize the overhead involved with RTS/CTS by only using it after a frame loss. When a frame gets lost, the retransmission of the frame is preceded by a RTS frame. Since RTS is sent at the lowest rate, it is robust against channel errors; hence, if RTS is lost, it will be from a collision. Subsequent data transmission after RTS should be free of collision and if data is lost, it can be assumed that it is from the channel error. CARA uses the same mechanism as ARF in its treatment of data frames, but prevents unnecessary rate down-shift by distinguishing collision losses from being counted as a frame failure. It performs better than ARF due to its differentiation and does not have the huge overhead of using RTS on every frame, but fails to reach high throughput in congested networks due to its proactive use of RTS after frame losses occur.

Robust Rate Adaptation Algorithm (RRAA) [12] is another rate adaptation scheme that uses RTS/CTS after a frame loss to eliminate further collisions. It consists of three elements: loss estimation, rate change, and Adaptive RTS (A-RTS) filter. In its rate adaptation stage (loss estimation and rate change), a station starts out by transmitting at the maximum rate and measures a loss ratio from recent transmission statistics. Transmissions are sent at one set rate for a cycle, whose period differs for different rates. After the cycle, the loss ratio for this period is compared against two thresholds that determine whether to increase or decrease the rate for the next period. A-RTS filter is used to suppress collision losses. A-RTS maintains a separate variable RTS_{wnd}, which determines how many consecutive frames are to be transmitted after RTS is sent. When a frame without RTS is lost, RTS_{wnd} is incremented by one with the assumption that this frame was lost due to collision. When a frame is lost after RTS or a frame without RTS succeeds, RTS_{wnd} is halved. RRAA also uses RTS to mitigate collisions and uses A-RTS to take advantage of RTS channel reservation and send multiple frames. A disadvantage of RRAA, besides the change of RTS protocol, is that it only adjusts its rate at the end of its transmission window or cycle and does not respond to losses from channel fading within the window.

Adaptive Thresholds [5] is a rate adaptation scheme that differentiates without the use of RTS/CTS. The premise behind the algorithm is that by adjusting the up- and down-thresholds of the ARF, it can be made to adapt to different environments including congested networks. It gauges the channel condition by looking at the retry field of a MAC header from frames sent by neighboring stations. By looking at the ratio between packets that are being retransmitted and packets that are not and comparing it to a predetermined table, new up- and down-thresholds of a typical ARF scheme is determined. Using these thresholds dictates how fast or slow rates should change depending on the channel conditions. Complexity and uncertainty of having to listen and keep track of frames from neighboring stations make this algorithm hard to implement in real life. Characteristics of each rate adaptation are outlined in Table 4.1.

While having a loss differentiation certainly increases throughput in congested networks as can be seen in Figure 4.1 and Figure 4.2, provided by authors of RRAA and Adaptive Thresholds, respectively, it also adds complexity and overhead to the existing system that is more troublesome. As of yet, there is no rate adaptation scheme that is both effective in both channel fading and collision dominated environments while providing both quick and long term performance. In addition, none of these adaptations look at the throughput with the capture effect in mind. They all look to improve the overall throughput, but never accurately consider how each individual node in a network would perform. A simple, easily implementable rate adaptation that improves the overall throughput and be fair to all the nodes in the system is sorely needed in modern WLANs.

Scheme	**Loss Differentiation**	**Condition Indicator**	**RTS/CTS Required?**
ARF	No	Frame Loss	No
AARF	No	Frame Loss	No
ONOE	No	Frame Loss	No
SampleRate	No	Average Transmission Time	No
RSSLA	No	RSSI	No
CARA	Yes	Frame Loss	Yes
RRAA	Yes	Frame Loss	Yes
Adaptive Thresholds	Yes	Frame Loss	No

Table 4.1: Characteristics of Different Rate Adaptation Schemes

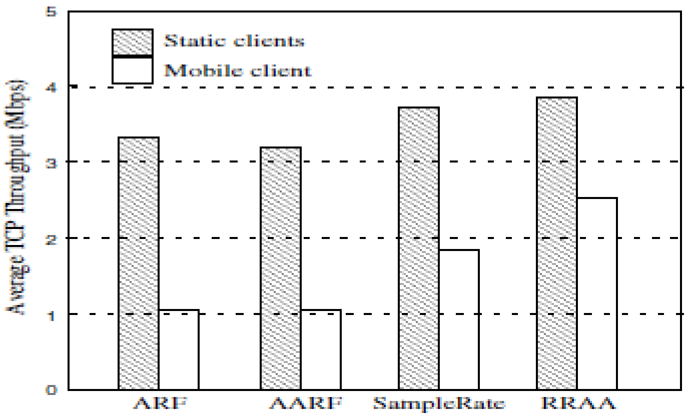


Figure 4.1: TCP Throughput Comparison of RRAA [12]

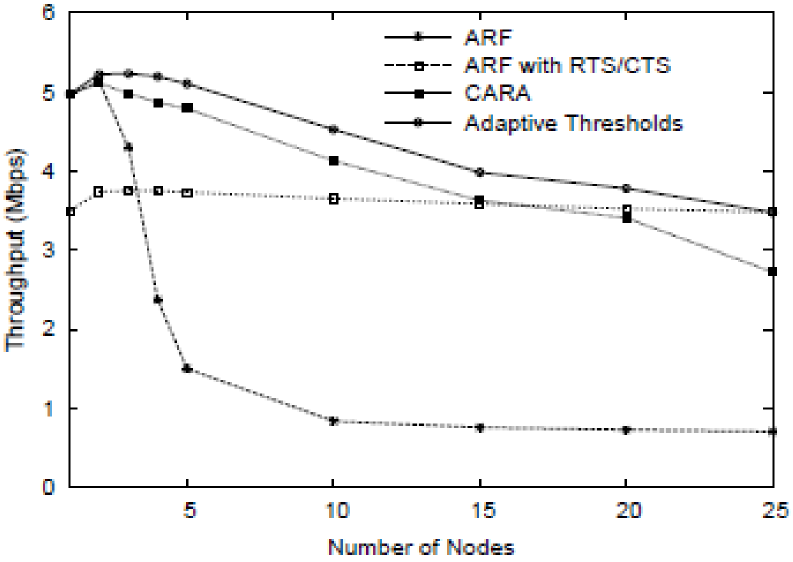


Figure 4.2: Throughput Comparison of Adaptive Thresholds [5]

Chapter 5

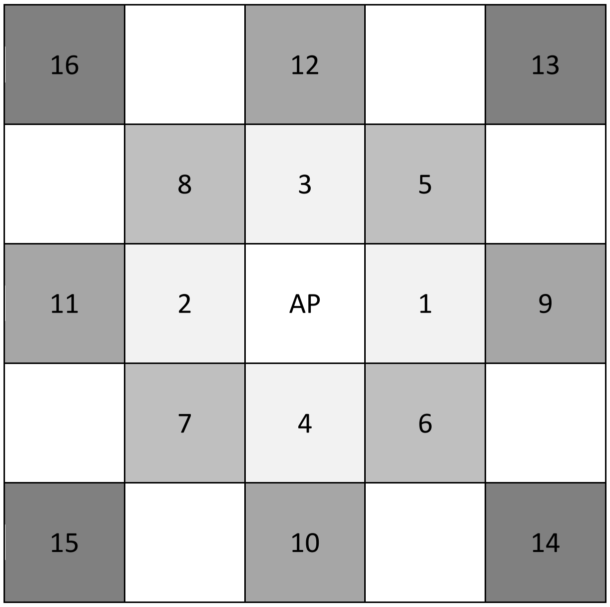
Capture-Exploited Fair Rate Adaptation (CEFRA)

5.1 Fairness of Rate Adaptation Algorithms

Despite the abundance of rate adaptation algorithms, there are very few studies that actually compare the performance of all these algorithms in a realistic environment. Most of the time, authors of each algorithm measure up their algorithm against others in an ideal environment that favors one or the other. A comparison study of how these algorithms perform under realistic scenarios is needed to properly compare these algorithms. Also, as mentioned before, authors of these algorithms never took fairness into account and it is largely unaccounted for. [13] and [14] are two experimental studies that try to fill this gap by testing out these rate adaptations in a testbed and using custom WARP platform, respectively. However, due to complications and the structured setup of the hardware, they are not sufficient enough to represent the various scenarios of real usage. They do not provide the flexibility or scalability of simulation-based studies to thoroughly investigate these algorithms under different environments despite their accuracy.

5.1.1 Simulation Setup

To study this relationship between rate adaptation and fairness caused by the capture effect, a testing environment that reflects a real life environment was set up carefully using ns-3 with the PhySim-WiFi add-on. In order to correctly see the performance of rate adaptations and the capture effect, it was important to set up nodes heterogeneously as well as having path loss to cause difference in RSS. To this effect, 16 nodes were set up in a grid layout with 1 AP stationed in the middle of the grid as shown in Figure 5.1. Nodes 1-4 are same distance away from the AP and will be referred to as Class 1. They are closest to the AP. Nodes 5-8, 9-12, and 13-16 are respectively Class 2, 3, and 4 and are correspondingly further away from the AP. This is illustrated in Figure 5.2. The testing environment is configured this way to easily observe the unfairness between these four classes caused by their distances to the AP. Distance between each grid is called deltaXY and will be varied to simulate real life deployment of different sizes. While this study limited the number of nodes to 16, it can easily be extended to even more nodes, but found it unnecessary to do so as the network is already heavily congested and saturated at this point. Results for simulations beyond 16 nodes converge and produce similar performance. As such, simulations are limited to the maximum of 16 nodes to keep the simulations shorter.



16		12		13
	8	3	5	
11	2	AP	1	9
	7	4	6	
15		10		14

Figure 5.1: Grid Layout Setup

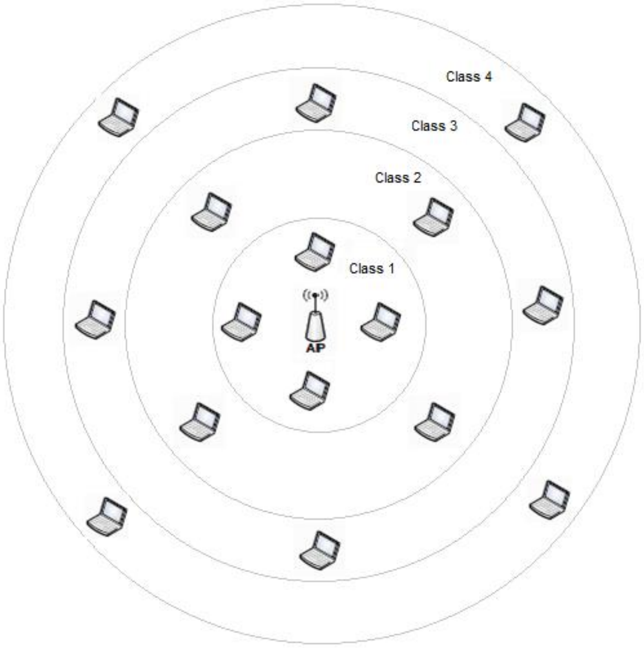


Figure 5.2: Illustration of Different Classes Based on Distance

Other than the physical setup of the nodes, other simulation parameters that best simulate typical real-life environments are used. 802.11a was the choice for the version of WLAN protocol as it was the only fully featured version available through the PhySim-WiFi. Although 802.11g is the most popular version at the moment, 802.11a will work just fine as it is essentially the same protocol on different frequency. The external interferences of the ISM band are not considered in these simulations. User Datagram Protocol (UDP) was used to generate traffic at constant bit-rate of around 9 Mbps and around 35 Mbps for low number of nodes to properly saturate the network. UDP was chosen instead of Transport Control Protocol (TCP) for all the baseline comparisons and study due to its simplicity and transparency. TCP also complicated the study because it had its own TCP ACK that was having collisions of its own. UDP is also a good choice for such a study since media applications like video- and voice-over-wifi all rely on UDP-like transport protocols. It is this kind of usage that tends to saturate a network. Default packet size for UDP was set at 1088 Bytes. MAC control packets, ACK, RTS, and CTS are 14, 20, and 14 bytes, respectively.

As for the wireless channel of the simulation, constant noise floor of -90dBm and log-distance path-loss model was used. Log-distance path-loss model is given by [28]

$$P_r \; dBm = P_t \; dBm + K \; dB - 10\gamma \log_{10} \frac{d}{d_0}. \tag{5.1}$$

P_r and P_t are the received and transmitted power in dBm, respectively. K is known as the reference loss and depends on the antenna characteristics and the average channel attenuation, d_0 is a reference distance from the antenna far field, and γ is the path-loss exponent. In order to simulate an office building or indoor type environment, the path-loss exponent of 3.5 was chosen. 1.0 and 46.6777 were used for the reference distance and reference loss. In this configuration, maximum range supported by the network is around 30 m, which is a reasonable indoor approximation. All the other parameters are default parameters specified by the IEEE 802.11 protocol.

In order to study the performance of the network, trace sources of the PhySim-WiFi in ns-3 were used to see exactly how many packets from each node were transmitted successfully and how many failed. The trace sources allow analysis of how and where in the receiving process, errors occurred. Through the trace sources, number of captures in the MIM mode can also be seen (i.e. the packets that arrive after the first one that gets captured). Lastly, overall throughput and individual node packet flow can also be checked and analyzed through the outputted pcap files.

5.1.2 Cumulative Throughput

For an unbiased comparison of each rate adaptation and fairness, deltaXY in the grid setup was set to 2.3 m. It meant that the nodes in Class 1 are 2.3 m away from the AP and the most distant nodes in Class 4 are 6.5 m away given the set up. This setup will be referred to as the close distance grid layout. Distances of nodes in each class are given in Table 5.1. After a preliminary testing to find the maximum range of each rate, it was found that the highest rate, 54 Mbps, had a maximum supported range of 7 m while maintaining around 5% FER. Rest of the supported ranges for each rate is shown in Table 5.2. Given that there are no collisions, all the nodes should be able to transmit at the highest rate without many errors, but each class is far away from each other that the RSS difference will cause the capture effect. Also, nodes that are farthest apart in this configuration are close enough that they can hear each other, thus mitigating the effect of the hidden node problem. In this infrastructure configuration, all the nodes will be sending their data to the AP and the cumulative throughput is the total amount of data that the AP received divided by the time of transmission. To measure performance more accurately, goodput was used to compare the results. Goodput is data throughput without the control packets such as ACK, RTS, and CTS. It is more accurate measure of performance as rate adaptations that use RTS/CTS will benefit from having those packets included in the throughput measurement when they are actually not part of data that is being sent. For the rest of this paper, the term throughput is used instead of goodput to be consistent, but all measurements are in goodput.

	Close Distance Grid (deltaXY = 2.3 m)	**Medium Distance Grid (deltaXY = 5 m)**	**Far Distance Grid (deltaXY = 10 m)**
Node	*Distance from AP (in m)*	*Distance from AP (in m)*	*Distance from AP (in m)*
1	2.3	5	10
2	2.3	5	10
3	2.3	5	10
4	2.3	5	10
5	3.25	7.07	14.14
6	3.25	7.07	14.14
7	3.25	7.07	14.14
8	3.25	7.07	14.14
9	4.6	10	20
10	4.6	10	20
11	4.6	10	20
12	4.6	10	20
13	6.51	14.14	28.28
14	6.51	14.14	28.28
15	6.51	14.14	28.28
16	6.51	14.14	28.28

Table 5.1: Distances of Each Node with Varying DeltaXY

Data Rate (in Mbps)	**Maximum Distance Supported (in m)**
54	7
48	11.5
36	14
24	16.5
18	21
12	23
9	25.3
6	26

Table 5.2: Maximum Distances Supported for each Rate

Figure 5.3 shows the throughput comparison of different rate adaptations in this network. Constant rates with no adaptation at 6 Mbps and 54 Mbps are also used as a point of reference. As can be seen from the chart, ARF performs the worst in such a congested network with collisions. In fact, most rate adaptation algorithms that do not differentiate losses perform poorly. ARF, ONOE, and constant rate at 6 Mbps all produce similarly poor throughput at around 4.5 Mbps. Assuming that the number of collisions does not vary too much from one adaptation to another, this result shows that due to the collisions and conservative nature of ARF and ONOE, all the nodes seem to be utilizing 6 Mbps most of the time. This is somewhat expected given that these adaptations do not take collisions into account and would quickly shift back down to lower rates when collisions do happen. The severe underutilization of the channel and exacerbation of collisions due to slow rates are the key contributors to their poor performance.

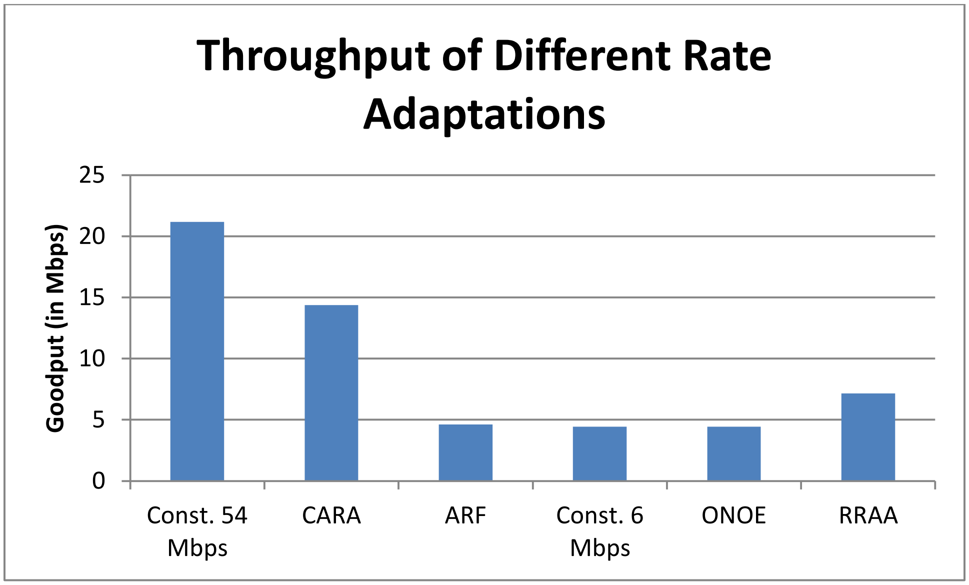


Figure 5.3: Throughput Comparison of Various Rate Adaptations with 16 Nodes

Rate adaptations with loss differentiation perform a lot better. CARA and RRAA both perform respectfully compared to ARF and ONOE. However, CARA's throughput is almost twice as big as RRAA's and seem to be making better decisions regarding collisions than RRAA. CARA's throughput is more than 3 times that of ARF and should definitely be considered in a congested network like this. On the other hand, somewhat surprisingly, constant rate at 54 Mbps has the best performance out of the group. While it is true that since the channel supports the highest rate given that there is no change to the quality and should always be sent at that rate to maximize throughput, it does not try to avoid collisions like CARA or RRAA. It would simply try to retransmit at the highest rate again hoping that there would not be a collision this time. Constant 54 Mbps rate would result in more failed packets, but it would also send more packets in the end compared to CARA. CARA would eventually utilize the channel at the highest rate, but would waste some time and packets turning RTS on only after a collision happens. Thus, while the unnecessary down-shift does not occur, it underutilizes the channel shifting back and forth trying to differentiate losses after they happen. This is evidenced by Table 5.3, which shows that CARA has higher percentage of packets received successfully than constant rate at 54 Mbps.

Rate Adaptation	**Packet Success Percentage**
Constant Rate at 54 Mbps	52.94%
CARA	64.43%
ARF	49.85%
Constant Rate at 6 Mbps	55.90%
ONOE	54.89%
RRAA	67.71%

Table 5.3: Successful Packet Reception Rate

Intuitively, it seems that if constant rate at 54 Mbps can somehow minimize collisions, it would perform even better. To test this theory, RTS/CTS probes were turned on for the constant rate and for ARF. The results comparing with and without RTS/CTS are shown in Figure 5.4. While ARF with RTS/CTS significantly outperforms ARF without RTS/CTS, addition of RTS/CTS slightly degrades the performance for the constant rate at 54 Mbps. In both cases, percentage of successful reception is higher with RTS/CTS due to fewer collisions. For ARF, this prevents nodes from unnecessarily lowering their rates and utilizes the channel more by staying at higher rates. For the constant rate at 54 Mbps, the overhead of RTS/CTS seems to be big enough that it does not improve performance despite fewer collisions. Constant rate at 54 Mbps is the optimum rate or is the rate adaptation that maximizes the throughput in this scenario, but it should be noted this is only true when all nodes can send at 54 Mbps, which is unlikely in real life given that some stations will be located farther away. RTS/CTS is shown to benefit rate adaptations in a congested network by reducing the number of collisions and by reserving the channel for data to be sent at an optimal rate.

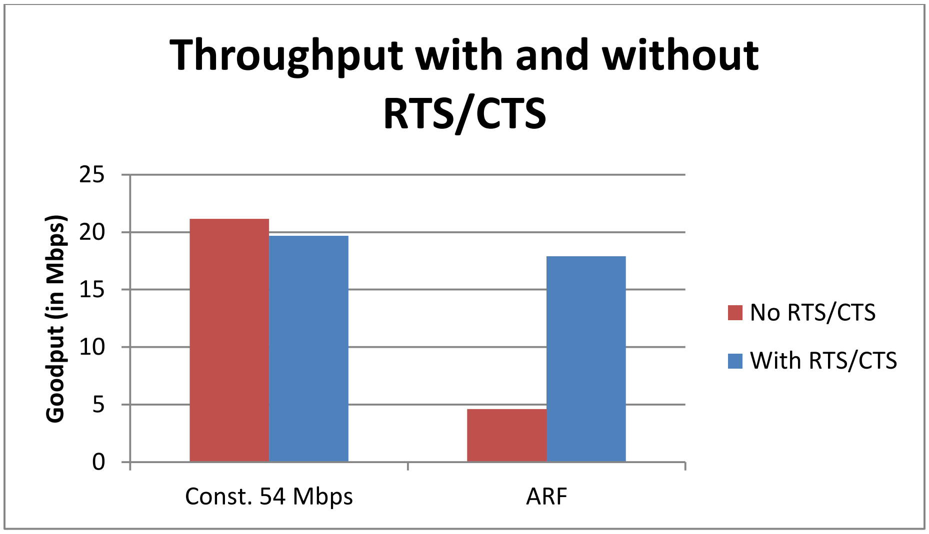
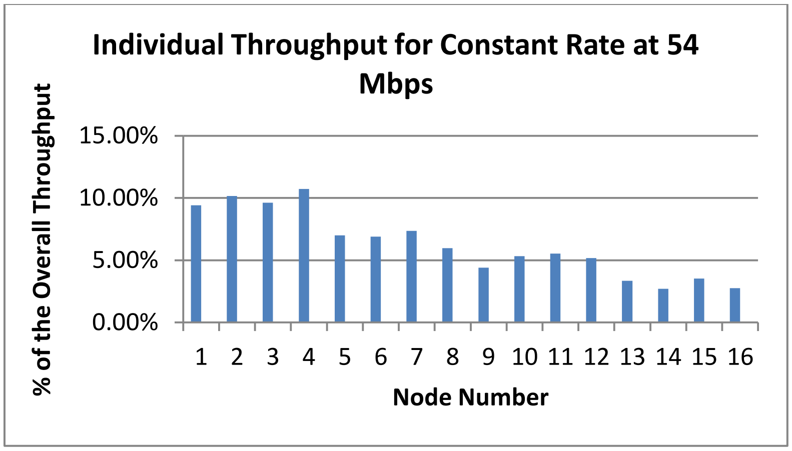


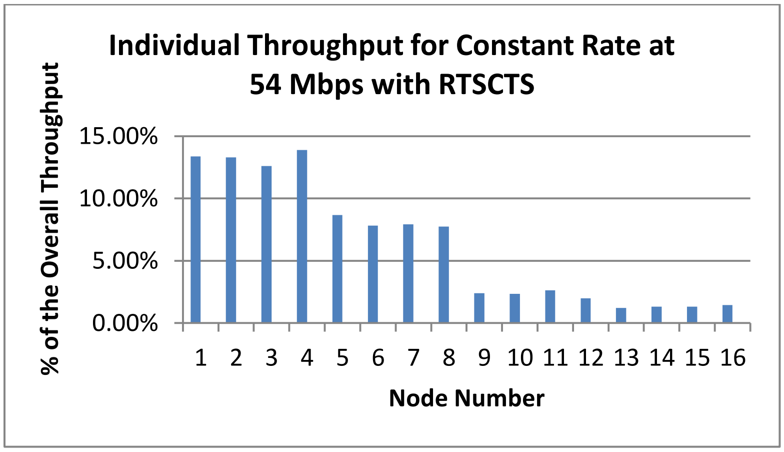
Figure 5.4: Throughput Comparison with and without RTS/CTS

5.1.3 Individual Throughput

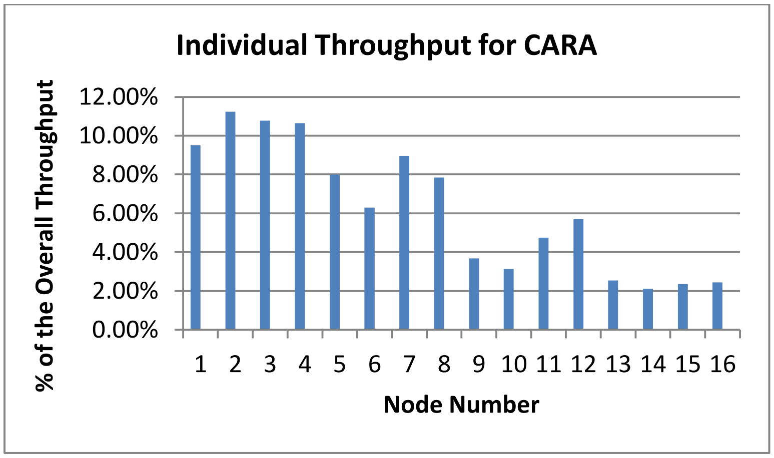
With the cumulative network throughput in mind, throughputs of individual nodes are considered to see how fair each of these algorithms is. Fairness is defined by the share of the network bandwidth of each node. Figure 5.5 shows the individual throughputs of the adaptations as percentages of total packets received. As expected, there is an imbalance in individual throughputs with closer nodes in Class 1 and Class 2 having higher throughput than nodes in Class 3 and Class 4. On the other hand, nodes that are within the same class seem to have almost same individual throughput. It is clear that there is unfairness between nodes depending on their location respect to the AP. Since all the nodes are close enough to the AP that there are only few channel errors, it is evident that this unfairness stems from path-loss and the capture effect. This unfairness would only worsen when the nodes are placed further away since more distant nodes would not be able to send frames at higher rates.



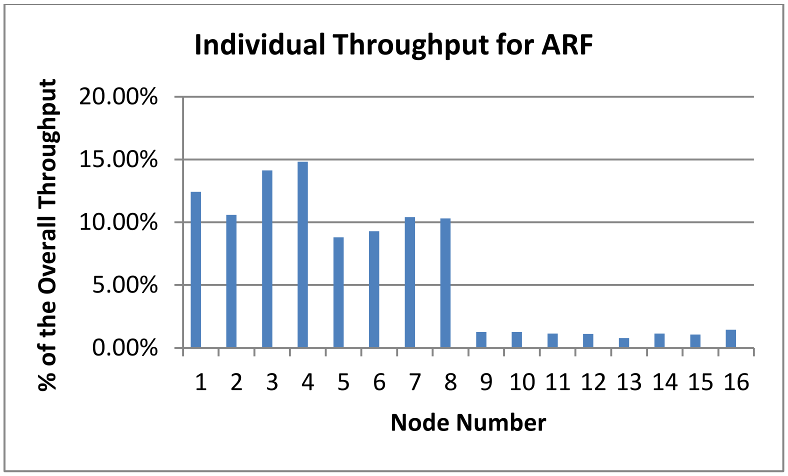
(a) Const. 54 Mbps Individual Throughput



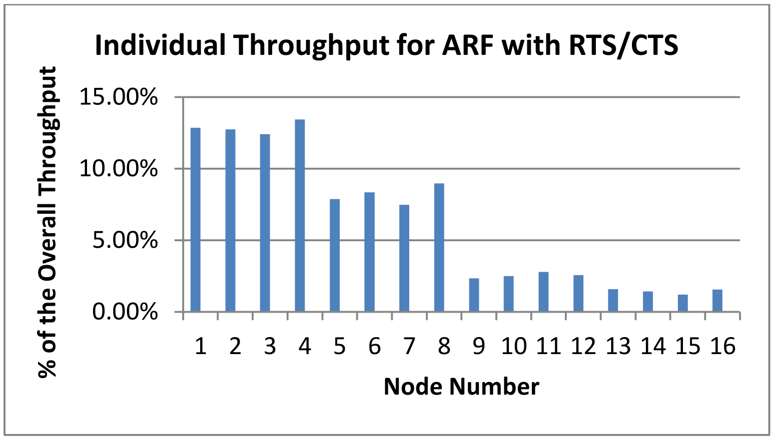
(b) Const. 54 Mbps w/ RTSCTS Individual Throughput



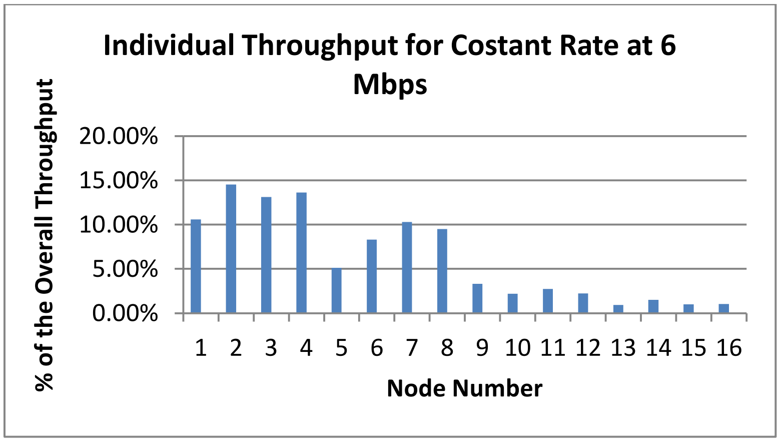
(c) CARA Individual Throughput



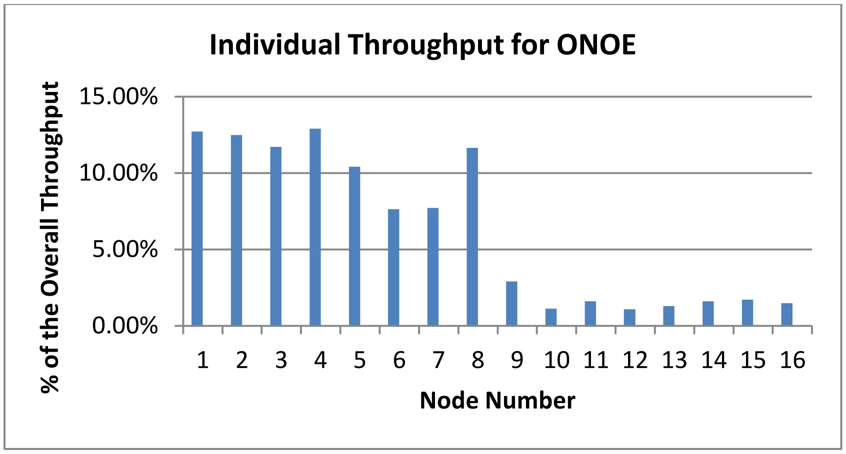
(d) ARF Individual Throughput



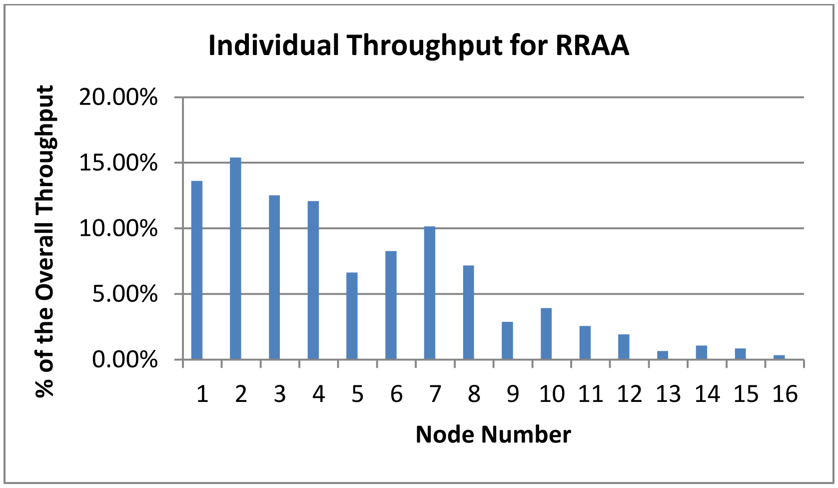
(e) ARF w/ RTSCTS Individual Throughput



(f) Const. 6 Mbps Individual Throughput



(g) ONOE Individual Throughput



(h) RRAA Individual Throughput

Figure 5.5: Individual Throughputs of Different Rate Adaptations in the Close Distance Grid (deltaXY = 2.3 m)

5.1.4 Fairness

Just from a quick glance of Figure 5.5, while all of them are favoring closer nodes, there are few rate adaptations that seem to be biased. To measure how fair these rate adaptations are, Jain's fairness index (JFI) [26] was used to measure the fairness. JFI is calculated as

$$F = \frac{(\sum_i x_i)^2}{n \times \sum_i x_i^2}, \tag{5.2}$$

where x_i is the individual flow throughput and n is the total number of flows. An index value of one is considered to be perfectly fair. JFI of each adaptation is calculated and shown in Table 5.4. For an easy analysis, the cumulative throughput and JFI of each algorithm is shown together in Figure 5.6. Like previously seen with the individual throughputs, most of the rate adaptations do not consider fairness and are unfair given the heterogeneous setup. Only CARA and constant rate at 54 Mbps perform well in terms of fairness.

Rate Adaptation	**JFI**
Constant Rate at 54 Mbps	0.86
CARA	0.79
ARF	0.58
Constant Rate at 6 Mbps	0.62
ONOE	0.62
RRAA	0.61
Constant Rate at 54 Mbps w/ RTSCTS	0.63
ARF w/ RTSCTS	0.65

Table 5.4: JFI of Various Rate Adaptations

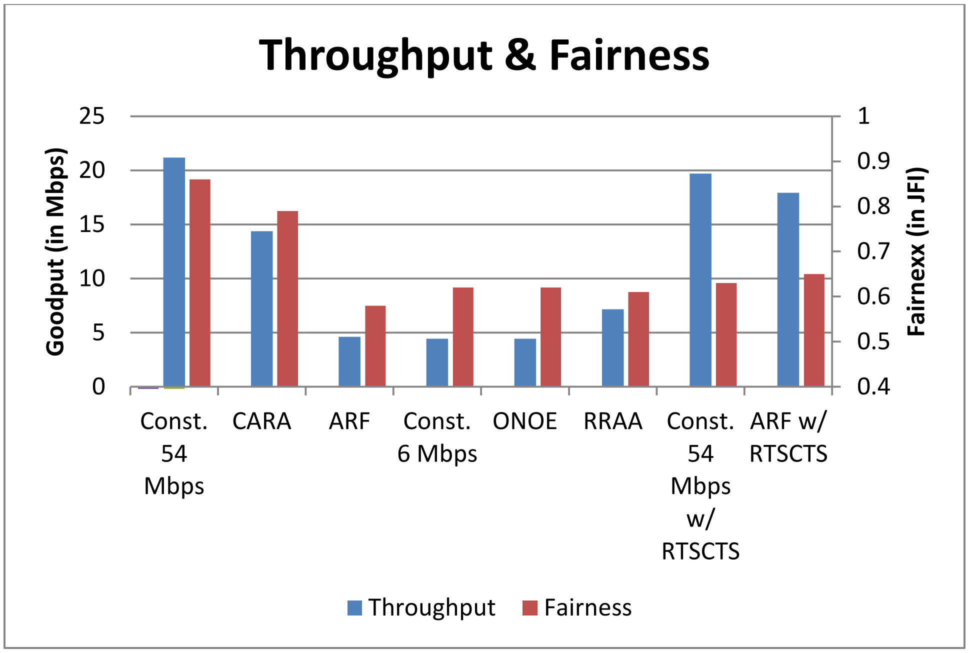


Figure 5.6: Throughput & Fairness of Rate Adaptations in the Close Distance Grid (deltaXY = 2.3m)

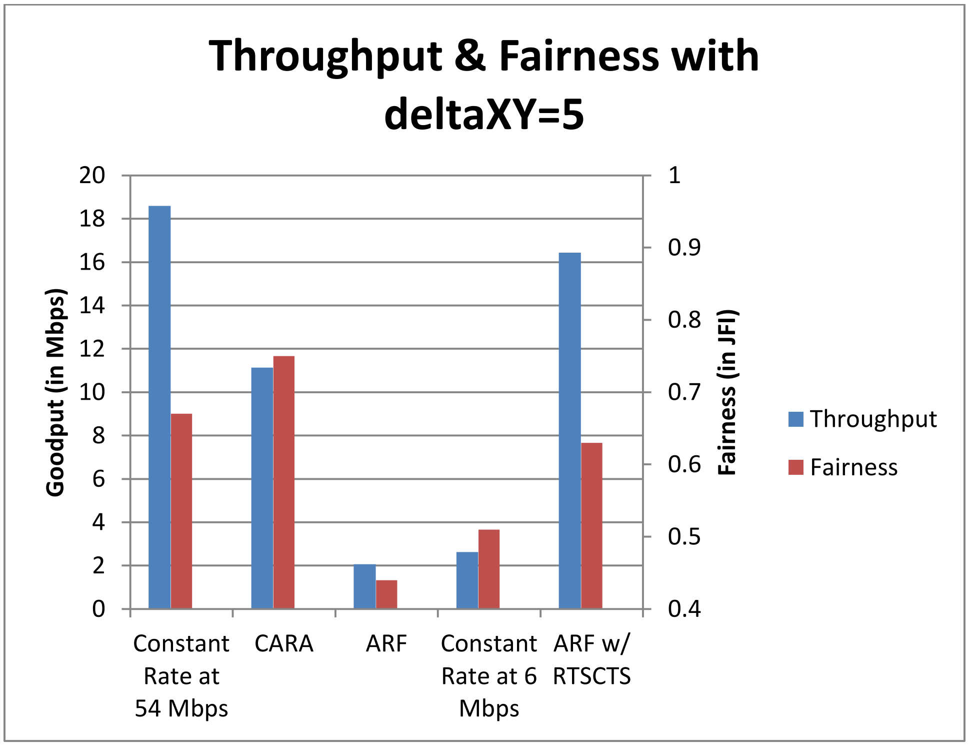


Figure 5.7: Throughput & Fairness in the Medium Distance Grid (deltaXY = 5 m)

The first impression might be that the rate adaptations with high overall throughput also have good fairness. However, upon closer inspection, not all high throughput producing adaptations have good fairness. Specifically, ARF with RTS/CTS and constant rate at 54 Mbps with RTS/CTS have high overall throughput, but have low fairness. This is an unexpected result, since use of RTS/CTS reduces number of collisions and thus leads to diminished capture effect, which causes the unfairness. In case of ARF, JFI varies little with or without RTS/CTS, but the difference is substantial in the case of constant rate at 54 Mbps. Curiously, despite differences in overall throughput, all the unfair adaptations seem to have similar JFIs.

In the previous subsection on cumulative throughput, it was observed that constant rate at 6 Mbps, ARF, and ONOE are likely to be underutilizing the channel and using the lowest rate at 6 Mbps. Assuming that most collisions in these rate adaptations occur between frames sent at 6 Mbps, similar JFIs are expected in these rate adaptations. Additionally, by default, all control packets including RTS are sent at the lowest rate at 6 Mbps. By using RTS/CTS, a typical data-to-data collision that would happen without them would now be RTS-to-RTS collision, which would also be a collision between frames that are sent at 6 Mbps. On the other hand, CARA selectively turns on RTS/CTS while maintaining higher data rate and constant rate at 54 Mbps sends every frame at the highest rate, resulting in collisions between frames that are sent at higher rates. Referring back to Chapter 3.4 and Figure 3.8, it was shown that frames with higher rates required higher SIR to be captured compared to the ones with lower rates. Given this characteristic of the capture effect, the fairness result makes perfect sense. In ARF, constant rate at 6 Mbps, or any other unfair adaptations, the closest nodes are sending at lower rates and because they typically have higher RSS at the AP when collisions happen, they are more likely to be captured and cause unfairness. In the case of constant rate at 54 Mbps or CARA, despite likely having higher RSS at the AP, because they are sending frames at higher rates, they are less likely to be captured, which leads to fairer throughput distribution among nodes.

Additional insight and further verification is made by looking at the results from same configuration, but with nodes spaced farther apart (i.e. deltaXY = 5 m). This setup will be referred to as the medium distance grid. Throughput and JFI for some of the rate adaptations can be seen in Figure 5.7. As expected, fairness goes down for most of them since farther away nodes are out of range and can no longer send data at higher rates that closer nodes can. This can be inferred from the fairness of constant rate at 54 Mbps, which plummets because Class 3 and 4 nodes have trouble sending frames at this rate. Only ARF with RTS/CTS and CARA holds their fairness despite the distance increase. RTS is sent at the lowest rate at 6 Mbps, which holds their FER even at this increased distance and JFI stays relatively same as before. CARA suffers in terms of overall throughput, but maintains fairness. This is likely due to the selective use of RTS. The selective use of RTS results in the combination of both RTS-to-RTS collisions as well as regular data-to-data collisions. In the regular data-to-data collision, the closer nodes should be sending data at higher rates due to collision awareness and have low probability of capture in case of collisions. Hence, CARA achieves intermediate fairness that is neither poor nor good.

This finding is critical in achieving any evenly well performing network in real life and should be one of the key factors considered when designing a rate adaptation algorithm. Currently, when performance of a network is discussed, it is only measured in terms of overall throughput, but unfairness should be discussed as part of the performance metric in conjunction with overall throughput. As shown, many algorithms try to increase the overall throughput at the expense of fairness of the system. A rate adaptation algorithm that achieves both cumulative throughput as well as fairness is needed.

5.2 Performance Improvement

5.2.1 Collision Study

In order to improve the performance of a WLAN network for both throughput and fairness, more knowledge on the type and frequency of collisions is needed. To study the behavior of collisions and what kind of packets are involved in collisions, new trace source was added to the core file of ns-3 PhySim-WiFi. The new trace source was implemented at the PHY layer core file and kept track of collisions through a callback method that was called when a new packet arrived while the receiver is already synchronizing or receiving the previous packet. In addition to how many collisions occurred, PhySim tag is used to access the information about each packet in the collisions. Useful information such as the timing of collision, SINR of the packets, and transmitting nodes can all be displayed for closer inspection through the new collision trace source. Together with already existing PhySim trace sources on packets, which provided general overview of errors, these two sources provided insights on the performance of the network.

The main thing that was tested through the collision study was the type of collisions that occur in regular and RTS/CTS transmissions and how often they occurred. This would help verify the performance increase that was observed as well as the unfairness issue. In repeated simulations in the same setup with constant rate at 6 Mbps with and without RTS/CTS, it was found that the data-to-data collisions in the regular transmissions are replaced by RTS-to-RTS collisions as expected. As for the number of collision occurrences, use of RTS reduced the number of collisions to about 10% of collisions experienced in a regular transmission. The small size of the RTS frame and elimination of hidden nodes are the main reasons behind this collision avoidance and improved throughput in a congested network. While it was possible to see the SINR of each packet in collision and get a sense of whether one of the packets will be captured or not, it was not possible to know for sure whether capture happened for one specific collision of interest.

Transmission statistics for constant rate at 54 Mbps with and without RTS/CTS obtained from the study are listed on Table 5.5 (a) and (b). Since use of RTS/CTS requires that every node precede a data frame with RTS frame, number of packets sent in the RTS case should roughly be twice. For nodes that are close to the AP, the number of packets sent is little less than double, but packets received is nearly triple the amount. On the other hand, nodes that are far away, namely, those that are in Class 4, send fewer packets despite the fact half of those packets are just RTS frames. By using RTS, closer nodes are able to access the channel more often, causing imbalance in throughput.

Looking at the transmission success rate increase between the two modes, Class 1 nodes in the RTS mode experience much bigger increase over the nodes in Class 4. Assuming that all the nodes have approximately equal probability of collision, nodes in Class 1 have greater increase in success rate since when collisions do happen, these nodes are more likely to be captured. This examination of packets further confirms the result presented in the previous section.

Constant Rate at 54Mbps	Distance from AP (in m)	Packets Sent	Packets Received	% of packets received
1	2.3	2094	1267	60.51%
2	2.3	2171	1367	62.97%
3	2.3	2143	1295	60.43%
4	2.3	2319	1442	62.18%
5	3.25	1721	941	54.68%
6	3.25	1700	926	54.47%
7	3.25	1757	990	56.35%
8	3.25	1517	804	53.00%
9	4.6	1241	593	47.78%
10	4.6	1428	716	50.14%
11	4.6	1476	744	50.41%
12	4.6	1395	697	49.96%
13	6.51	1192	450	37.75%
14	6.51	961	366	38.09%
15	6.51	1251	475	37.97%
16	6.51	1029	371	36.05%
Total		25395	13444	52.94%

Table 5.5 (a): Const. Rate at 54 Mbps Packet Statistics

Constant Rate at 54Mbps w/ RTSCTS	**Distance from AP (in m)**	**Packets Sent**	**Packets Received**	**% of packets received**
1	2.3	3820	3348	87.64%
2	2.3	3818	3328	87.17%
3	2.3	3634	3152	86.74%
4	2.3	3957	3472	87.74%
5	3.25	2775	2172	78.27%
6	3.25	2519	1960	77.81%
7	3.25	2524	1981	78.49%
8	3.25	2490	1936	77.75%
9	4.6	989	604	61.07%
10	4.6	961	588	61.19%
11	4.6	1052	660	62.74%
12	4.6	831	496	59.69%
13	6.51	626	302	48.24%
14	6.51	677	328	48.45%
15	6.51	652	330	50.61%
16	6.51	725	360	49.66%
Total		32050	25017	78.06%

Table 5.5 (b): Const. Rate at 54 Mbps with RTS/CTS Packet Statistics

5.2.2 Fairness Improvement

Due to the capture effect and path loss, unfairness cannot completely be avoided. The severity of unfairness, however, can be improved by careful assignment and use of different rates. From the previous sections, the possibility and potential of using rates to correct unfairness was observed. It is evident that distance and the resulting signal strength implications should be exploited to make a fair rate adaptation. In this subsection, the relationship between rates and the capture effect is explored.

The effect that a high rate has on the capture effect has already been observed. High data rate lowered the probability of capture by increasing the required SINR needed for the stronger signal to be successfully received in a collision. Having all the nodes that can support the highest rate send at that rate has shown that it improves fairness tremendously already, but it is not known whether having the more distant nodes send at lower rates will help make the system even fairer. Intuitively, it makes sense to see more captures happening in favor of the nodes with low rates, but they would also be sending data at low rates and would probably result in low individual throughputs as well. The tradeoff between these two factors makes it hard to see what impact this would have and is further investigated.

	Class 1	Class 2	Class 3	Class 4
RFTC1	54 Mbps	36 Mbps	24 Mbps	12 Mbps
RFTC2	54 Mbps	24 Mbps	18 Mbps	9 Mbps
RFTC3	54 Mbps	36 Mbps	12 Mbps	6 Mbps

Table 5.6: Rates used for RFTCs

In the same simulation testbed, different constant rates were assigned to nodes of difference classes. Class 1 nodes were assigned the highest rate at 54 Mbps, and data rates of other classes were assigned using varying combinations of lower rates. Different rate combinations used are listed in Table 5.6 and they will be referred to as rate fairness testing case (RFTC) 1, 2, and 3 for convenience. Overall throughput and fairness for each of these cases are shown in Figure 5.8 and the individual throughputs are shown in Figure 5.9. The setup using constant rate at 54 Mbps for all the nodes is also included as a reference.

Looking at these figures, fairness is excellent for all three cases with RFTC 3, which uses the lowest rates for distant nodes, performing slightly better than the other two cases. From looking at Figure 5.9, this higher fairness can be attributed to Class 1 nodes having lower percentage of the total throughput than in the case of other cases. However, comparing RFTC 1 and RFTC 2 with constant rate at 54 Mbps, there is no difference in fairness at all. This result shows that the rates used by other nodes have no or very little impact on the capture probability of stronger nodes. This result is consistent with [15], which concluded that the rates of interferer do not affect the capture probability of the sender; only the sender’s own rate affect the capture probability. Slight increase in fairness in RFTC 3 can be due to the fact that Class 4 nodes use the lowest rate, 6 Mbps. Looking back to Figure 3.8 (a), these nodes only need to be as strong as the interferer when they arrive first to the receiver to be captured.

In terms of overall throughput, these RFTCs perform badly, especially RFTC3. This is expected since the conditions allow them to send at the highest rate, but they do not do so. Using lower rates had a detrimental impact on the overall throughput. Looking at RFTC3, the only difference was that the rates used by Class 3 and Class 4 were double that of RFTC3, but the resulting overall throughput is only half compared to RFTC1. That seemed like bigger loss than what was expected, so the overall successful transmission percentages are also compared in Figure 5.10.

Comparing the percentages, all RFTCs seem to suffer incrementally with lower rates used compared to the constant rate at 54 Mbps case. The overall throughput not only suffers from sending data more slowly, it also suffers from having more collisions due to longer transmission time suffered from using lower rates. While the capture effect attempts to minimize the loss suffered from a collision, it is better to avoid collisions if possible. Using this knowledge, even the most distant nodes should send at highest rate that the channel can support in order to maximize throughput. Thus, rate selection that optimizes the overall throughput and fairness in this setup is constant rate at 54Mbps.

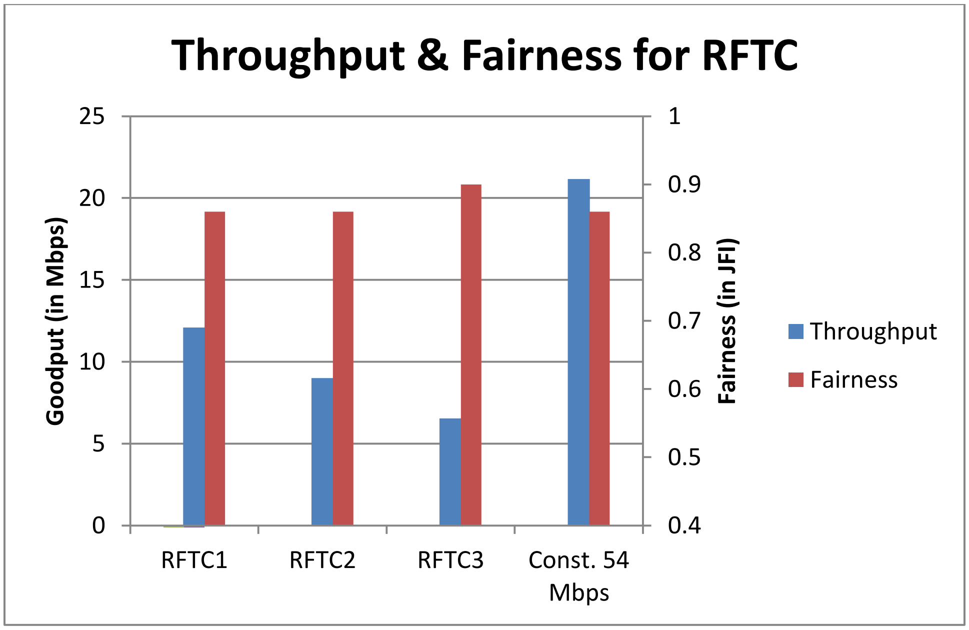


Figure 5.8: Throughput & Fairness for RFTCs

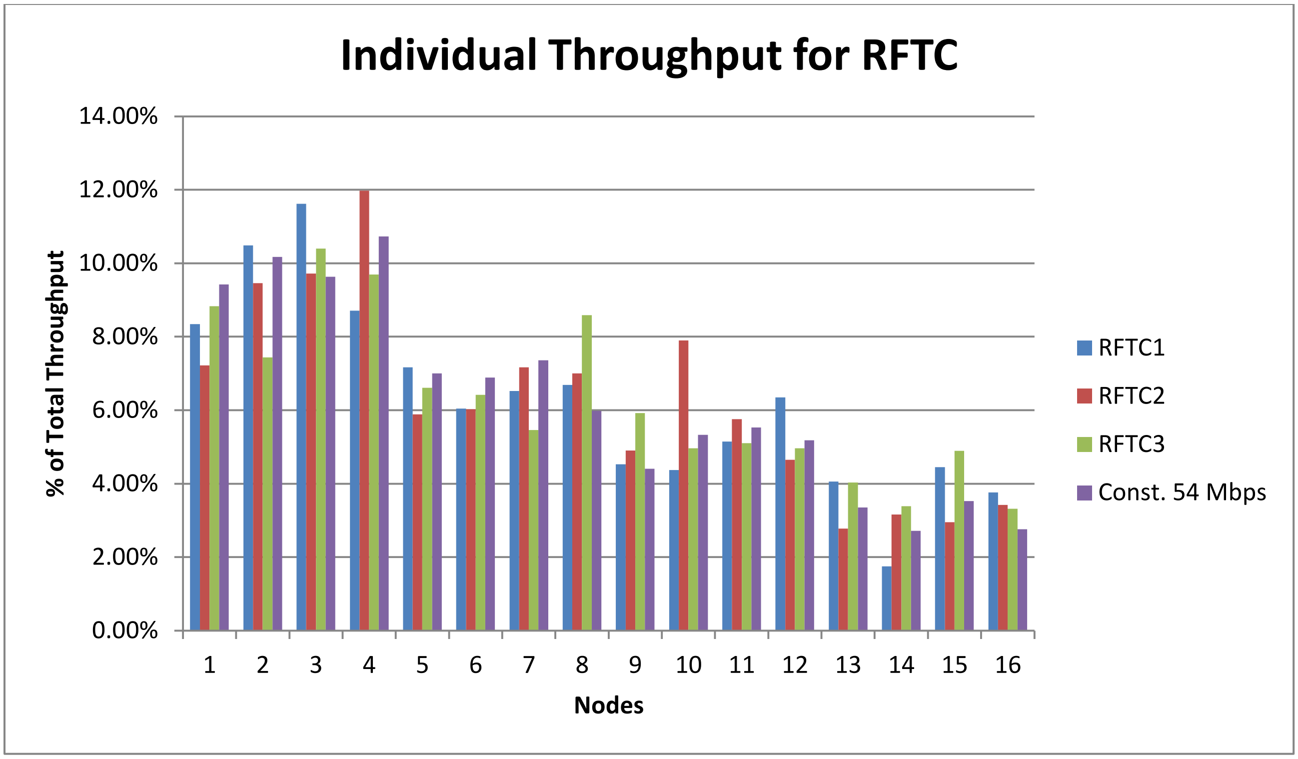


Figure 5.9: Individual Throughput of RFTCs

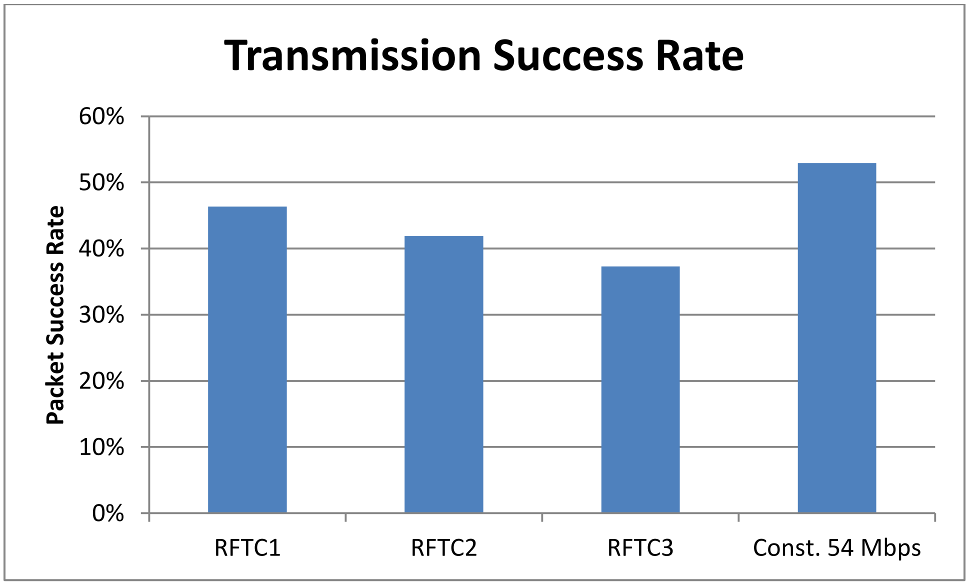


Figure 5.10: Transmission Success Rates for RFTCs

5.2.3 Use of RTS/CTS

Despite being part of the 802.11 protocol, RTS/CTS is rarely implemented even with many studies proving that the use of RTS/CTS will lead to performance gain. This is due to the overhead and latency involved with the use of RTS/CTS. RTS/CTS leads to a flat and stable throughput performance regardless of congestion level [4]. It means that the use of RTS/CTS will have consistent performance that is subpar when there are only few nodes in the network, but performs well when there are more contending nodes.

In the previous subsections, it was shown that the use of RTS significantly reduces collisions in the network due to its small size compared to a data frame. Additionally, number of collisions is reduced and fairness increased with the use of faster data rates as long as the channel supports them. One drawback to the use of RTS/CTS was that it had a huge unfairness problem since RTS, by default, is sent at the lowest rate. What happens when RTS is sent at faster rates? This concept was visited by authors of [11] when they wanted to reduce the overhead of RTS and CTS by transmitting them at higher rates. Can RTS rates be changed in a way that would improve fairness while maintaining or improving the overall throughput by reducing the number of collisions? The answers to these questions are investigated in this subsection.

The two cases of different RTS rates are considered. The rates of RTS are set according to the rate assignment in RFTC2 and data rates are varied between using ARF and constant rate at 54 Mbps. The performance results can be seen in Figure 5.11 along with results from ARF with regular RTS/CTS and constant rate at 54 Mbps with regular RTS/CTS for comparison. Total successful transmission percentages and individual successful transmission percentages are shown in Figure 5.12. As expected, the overall throughput is good. It is either on par or slightly better than the regular RTS/CTS usage. However, there is a significant improvement in fairness on par with the fairness performance seen in the previous subsection. Looking at the successful transmission percentages, the overall percentage and more importantly, the percentages of nodes in Class 1, decrease with the new RTS rates. This causes the increase in fairness. These results show the possibility of achieving a high overall throughput as well as high fairness by exploiting the capture effect.

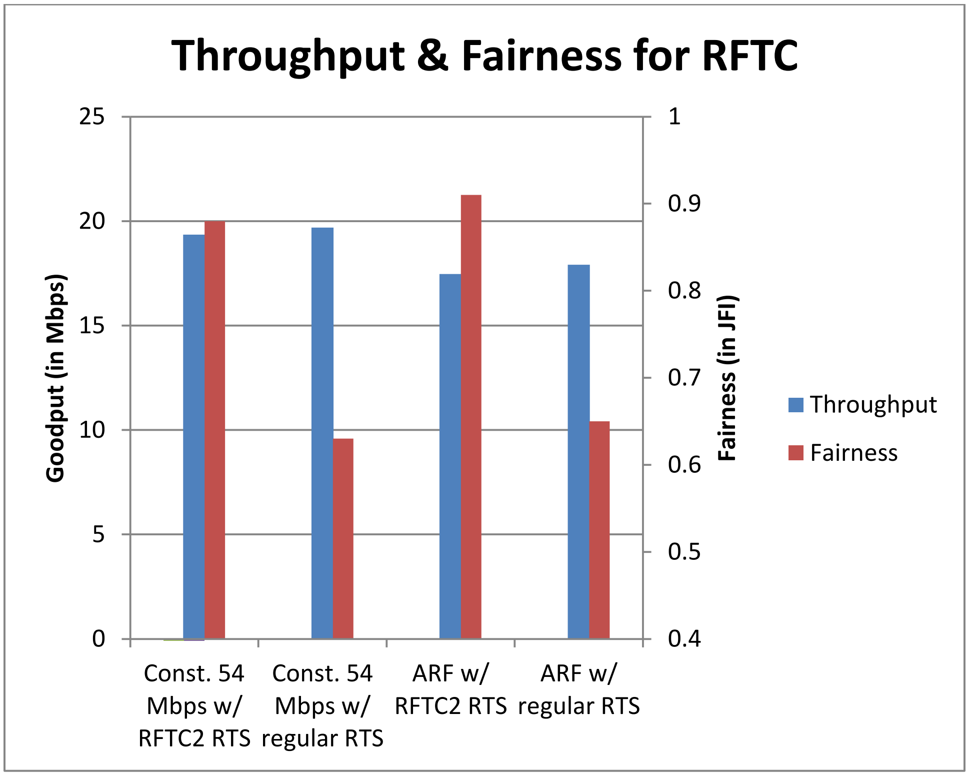
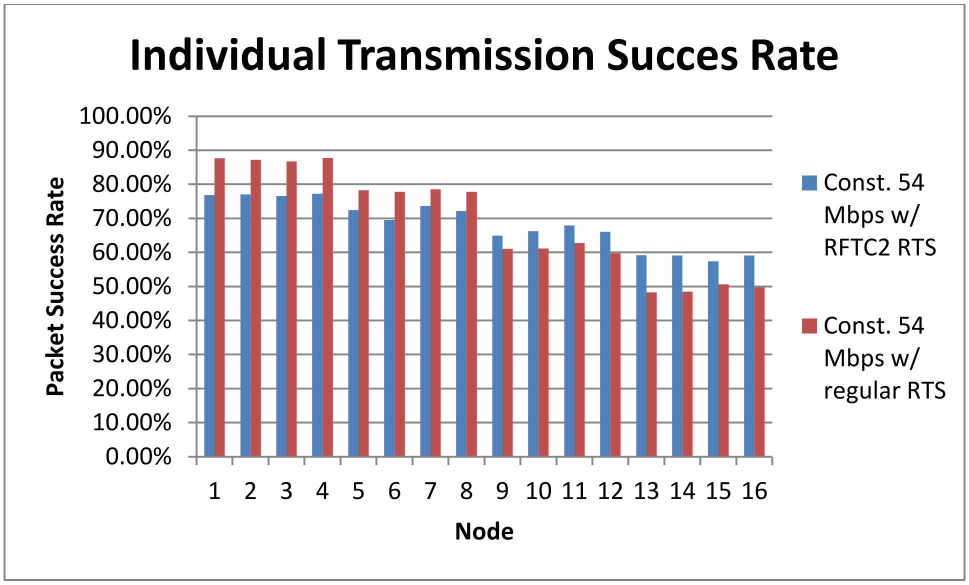
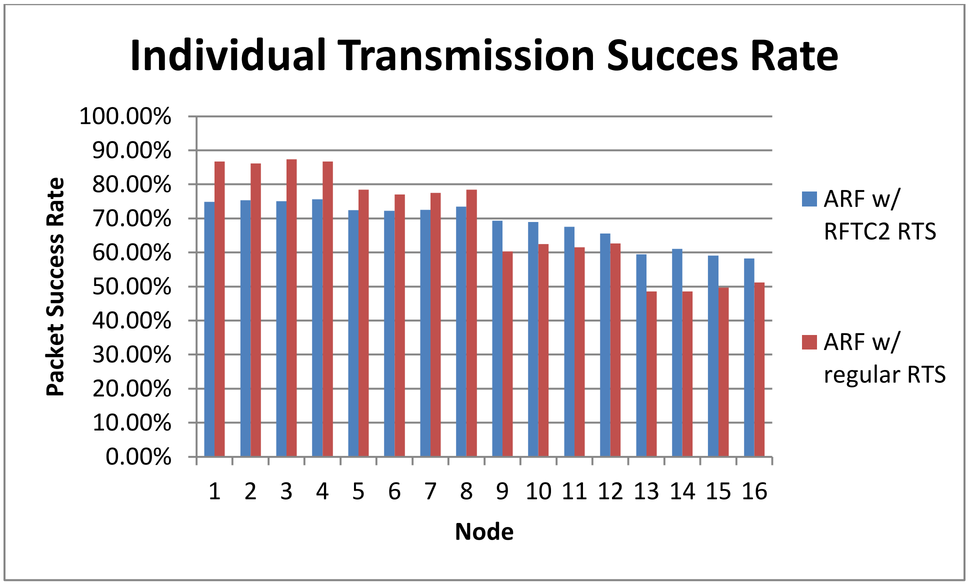


Figure 5.11: Performance Comparison of Regular RTS and RFTC2 RTS



(a) Const. 54 Mbps RTS Comparison



(b) ARF RTS Comparison

Figure 5.12: Individual Throughput Comparison of Regular RTS with RFTC2 RTS

5.3 CEFRA Implementation

A fair rate adaptation algorithm for 802.11 WLAN that utilizes the capture effect to maintain high fairness and throughput is proposed here. Findings of the previous sections are used to design a novel way to combat unfairness through rate selection and the corresponding rate adaptation is aptly named Captured-Exploited Fair Rate Adaptation (CEFRA). CEFRA is a simple rate adaptation scheme that can easily be implemented on existing devices and will provide consistent performance to all the nodes in the network regardless of the size or condition of the network.

Motivated by the impact that rates have on the capture probability, CEFRA selects appropriate rates based on the RSSI resulting from the path-loss fading to correctly counter the location-dependent unfairness caused by the capture effect. As for overall throughput, CEFRA maintains or tries to improve it by using RTS/CTS to minimize the number of collisions. RSSI is used to change the rates of RTS frame, rather than sending it at the lowest rate as it is currently implemented. By doing so, the overhead involved with the exchange of RTS/CTS is slightly decreased, but the collision probabilities are further decreased. Small size of RTS and faster RTS transmissions minimize the number of collisions that occur. If collisions do happen, since RTS rates are set according to the RSSI (distance), closer nodes, which will send at higher rates, will have lower chance of capture, increasing fairness. Data frames are reserved and collision-free due to the reservation of channel after RTS/CTS exchange and are sent at the maximum data rate possible given the channel condition.

First transmission starts off by sending RTS at the lowest rate as default, and once CTS is received back at the receiver, RSSI of the CTS is used to gauge the distance or relative RSS due to path-loss. RSSI of CTS is used as it is collision-free and would be the clearest representative of how far each node is from the AP. Through simple simulation testing, it performed well enough to be used as the indicator. Also, algorithms that try to gauge the channel by listening in on the channel and other neighbors are avoided as that adds complexity and change to the current devices. The RSSI is then mapped out to proper rate through pre-calculated lookup chart such as the one shown in Table 5.7. Since, the channel is free after the CTS, data rates are sent at the maximum rate possible, which can be computed from the CTS RSSI as well. In the next transmission, RTS is sent at the predetermined rate set by the last CTS frame. By doing so, RTS rates are set accordingly to promote fairness in the network to counter the location-dependent unfairness. This will provide both long and short term performance that adapts well to other environment conditions as RSSI is measured on every transmission. Flowchart showing CEFRA's operations is shown in Figure 5.13. Next chapter will discuss the performance of CEFRA against other rate adaptations and under different environments.

Data Rate (in Mbps)	Minimum SNR required for PER = ~5% (in dB)
54	27
48	20.5
36	16.9
24	14.1
18	10.7
12	9.3
9	7.6
6	7.3

Table 5.7: Pre-computed SNR to Data Rate Lookup Table

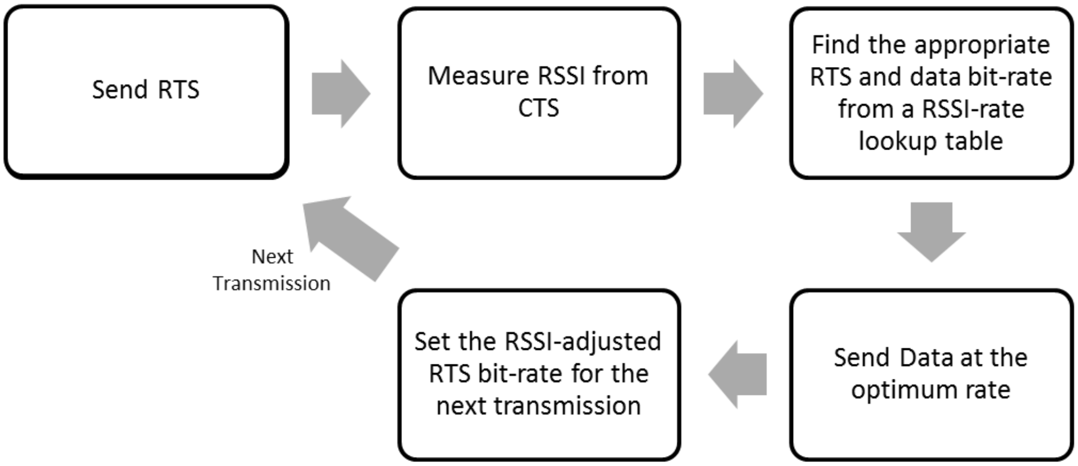


Figure 5.13: CEFRA flowchart

Chapter 6

Results/ Discussion

In this section, the performance of CEFRA is tested and evaluated through ns-3 simulations with the PhySim-WiFi add-on. Same setup of grid layout with different classes is used. As before, UDP is used to generate traffic at saturation and payload frame length is 1088 bytes. DeltaXY in the grid layout is still set to 2.3 m. Nodes are in a stationary network.

First, CEFRA is tested in the full 16 node scenario to be compared with other previously mentioned rate adaptations. The comparison of throughput and fairness is shown in Figure 6.1 and Table 6.1. Overall throughput and fairness both perform well, outperforming almost all rate adaptations in both categories. Since constant rate at 54 Mbps was found to be the ideal rate given the setup and knowledge of the channel, it is referred to as the “oracle”. It represents the near-optimum result that can be obtained in this scenario. Only constant rate at 54 Mbps or “oracle” outperforms CEFRA by about 1 Mbps in overall throughput. This shows that the overhead of RTS/CTS could not be overcome even with less time spent on retransmissions via RTS/CTS. CEFRA’s overall throughput is on par with constant rate at 54 Mbps with RTS/CTS, which suggests that CEFRA is making the right choices in terms of data rate selection. Fairness, as expected, is better than any other rate adaptations. CEFRA’s fairness can be seen with the individual throughputs in Figure 6.2.

CEFRA is next evaluated for performance under varying number of nodes. It is compared against ARF, CARA, and ARF with RTS/CTS. ARF, as the most widely used adaptation, is representative of how most current devices would perform and represents the rate adaptations without loss differentiation. CARA represents the rate adaptations with loss differentiation. ARF with RTS/CTS is the collision avoidance rate adaptation that will be used as a reference point for CEFRA. Simulations were done at 1, 2, 4, 8, 12, and 16 nodes.

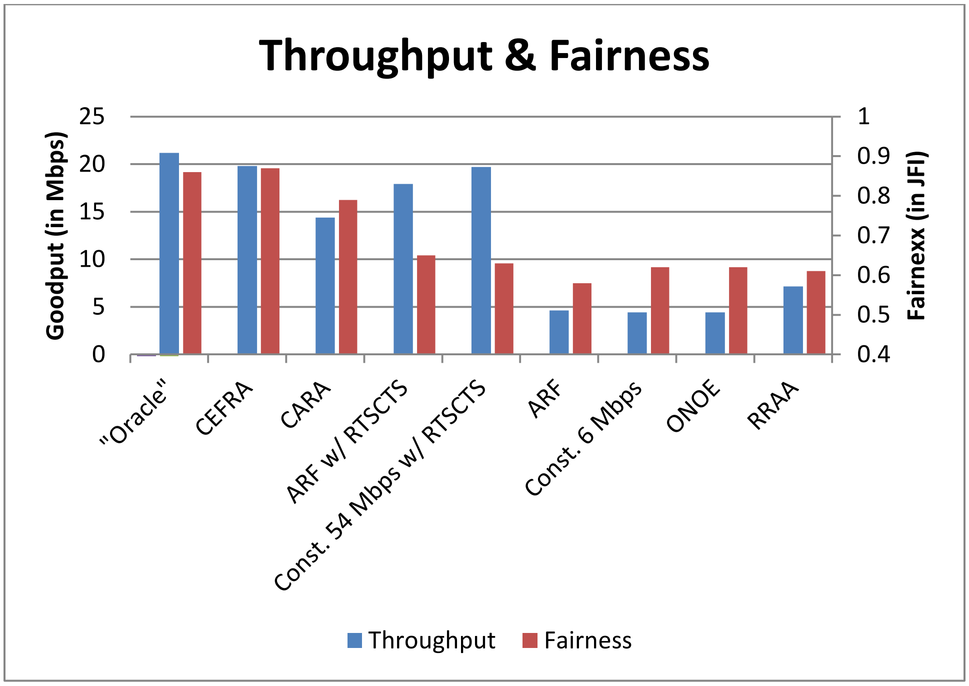


Figure 6.1: Performance of CEFRA against others in the 16 nodes close distance grid layout (deltaXY = 2.3 m)

Rate Adaptation	Throughput (in Mbps)	Fairness (in JFI)
"Oracle"	21.17	0.86
CEFRA	19.79	0.87
CARA	14.38	0.79
ARF w/ RTSCTS	17.91	0.65
Const. 54 Mbps w/ RTSCTS	19.69	0.63
ARF	4.62	0.58
Const. 6 Mbps	4.43	0.62
ONOE	4.43	0.62
RRAA	7.16	0.61

Table 6.1: Throughput and Fairness of CEFRA compared to others in the 16 nodes close distance grid layout

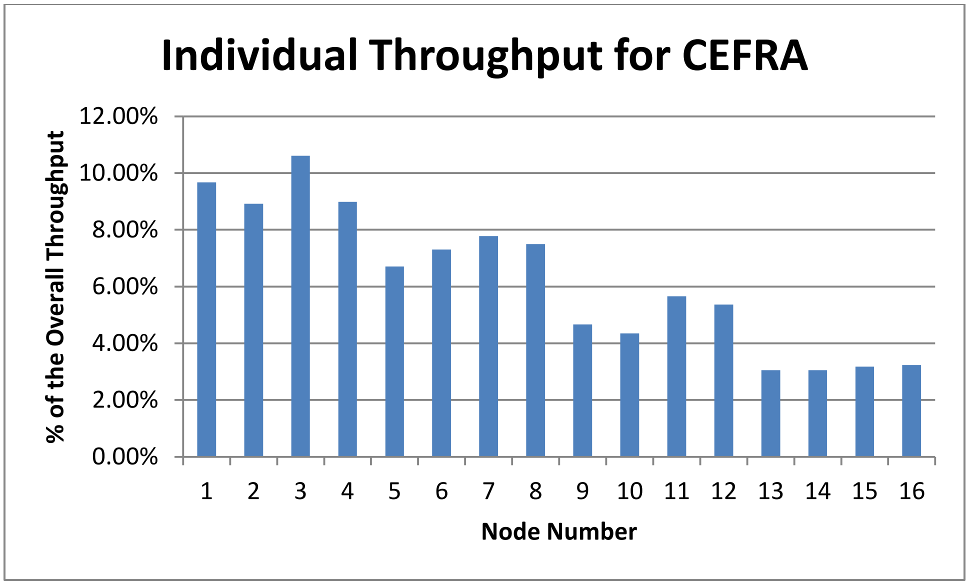
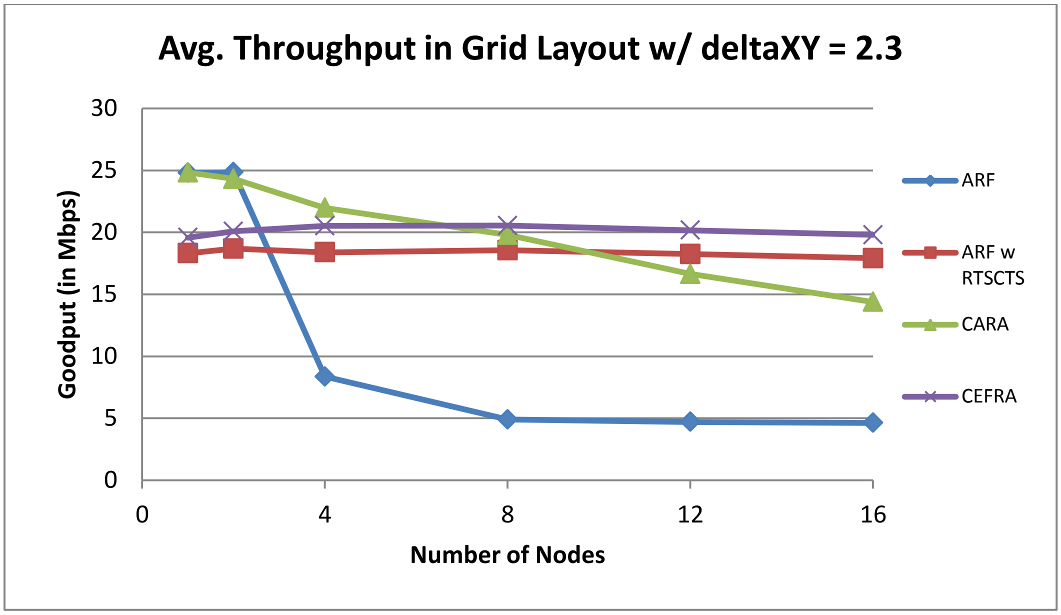
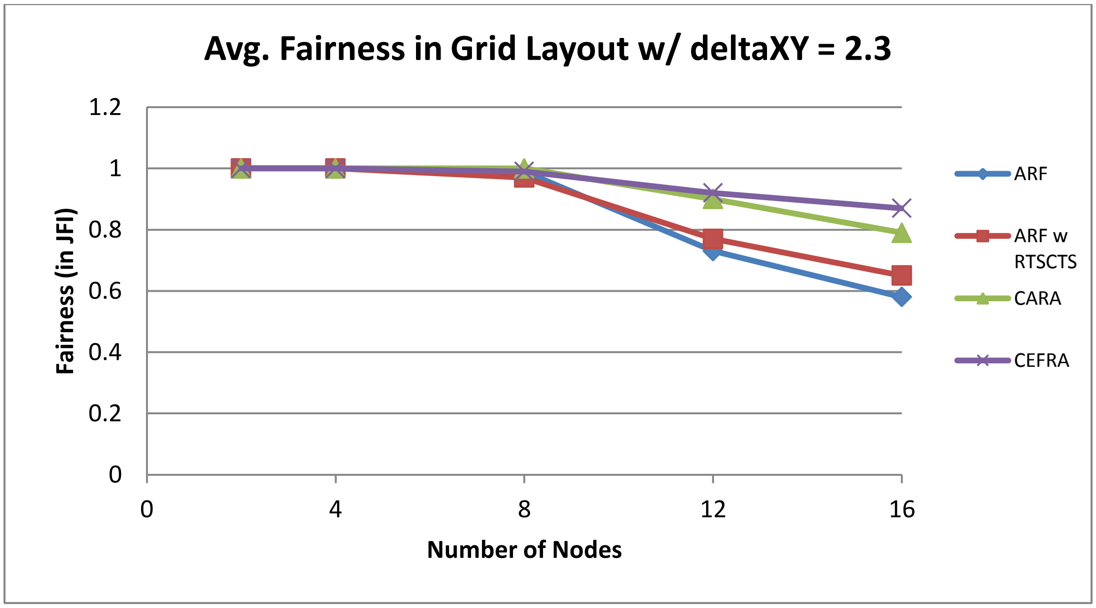


Figure 6.2: CEFRA individual throughput for 16 nodes in the close distance grid layout (deltaXY = 2.3 m)



(a) Throughput



(b) Fairness

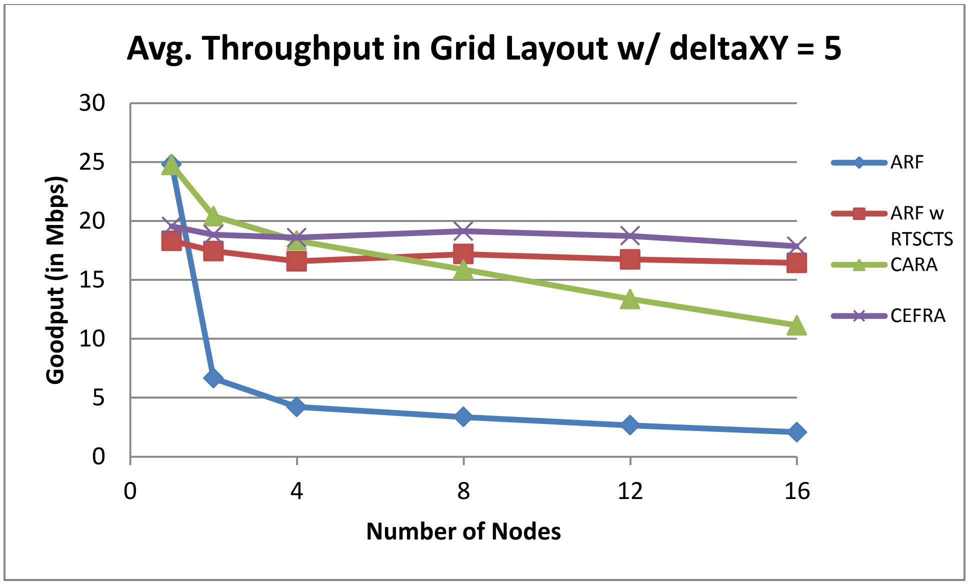
Figure 6.3: Performance with varying number of nodes in the close distance grid layout

The results are shown in Figure 6.3. CEFRA has the highest overall throughput for high contention network (i.e. 8, 12, and 16 nodes), but has a flat throughput behavior that does not increase in lower contention network. This was expected of a rate adaptation scheme that uses RTS/CTS. ARF with RTS/CTS shows similar throughput behavior but has lower throughput across all contention level. The overhead of using RTS/CTS is reduced through the use of faster RTS frames in CEFRA. ARF and CARA perform at near full capacity in low contention system with 1 or 2 nodes since they do not have RTS/CTS overhead and there are very few errors of any kind. ARF's throughput suffers badly though when there is even a slight congestion, which is evidenced by huge drop-off at 4 nodes. CARA's throughput, on the other hand, shows a linear behavior with respect to number of nodes. This is due to the selective and hybrid use of RTS/CTS for loss differentiation.

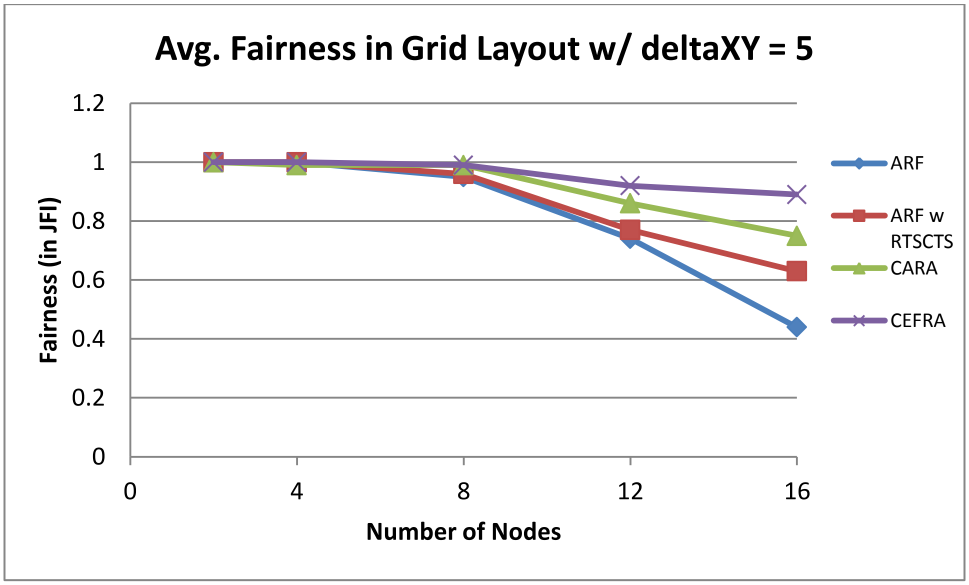
As for fairness of the system, CEFRA performs best regardless of the contention and congestion level. CEFRA has about 10% improvement in fairness over CARA and around 50% improvement over ARF in 16 nodes network. The fairness all converges towards 1 with low number of nodes due to the setup of the layout. When simulations are run with additional 4 nodes, nodes in the next class are added to show clear change in fairness. Thus, it makes sense that they all have perfect fairness (i.e. JFI of 1) for 1-4 nodes since all the nodes are located exactly same distance away from the AP. Fairness of randomly situated nodes are discussed later.

In the same layout, deltaXY is increased to 5 m to represent bigger and less densely populated network. There will be more channel errors and even bigger unfairness resulting from distant nodes not able to use higher rates. Figure 6.4 shows the performance results. Throughput-wise, all rate adaptations seem to follow the same behavior as before. However, throughput of ARF and CARA seems to suffer a bit in low contention setup and the gap between CEFRA and CARA is smaller at 1 and 2 nodes situation. CEFRA outperforms everything beyond 4 nodes. As expected, fairness decreased for all the other rates given the increase in distance, but CEFRA was able to maintain its fairness. The fairness gap is even bigger now with CEFRA having nearly 20% improvement over CARA and about 100% improvement over ARF with 16 nodes. Individual throughputs of the rate adaptations at this distance can be seen in Figure 6.5.

To represent even bigger and sparsely populated network, deltaXY is increased to 10 m. This setup is referred to as the far distance grid layout. Figure 6.6 presents the throughput and fairness of the network. Overall throughput follows the trend and the rate adaptations have similar behaviors as before. There is a slight decrease in throughput across all rate adaptations and CEFRA's and ARF with RTS/CTS's throughputs decrease linearly with increasing number of nodes. This is expected since there would be even more channel errors at these distances. Fairness shows the same trend as well and is decreased even further, especially at 16 nodes. Part of the reason is that Class 4 nodes are about 28 m away from the AP and even at the lowest rate, their FER is very high. CEFRA also suffers with 16 nodes, but still performs about 25% better than CARA and around 150% better than ARF. ARF's throughput is almost wholly dominated by Class 1 nodes. This extreme unfair distribution can be seen in Figure 6.7.

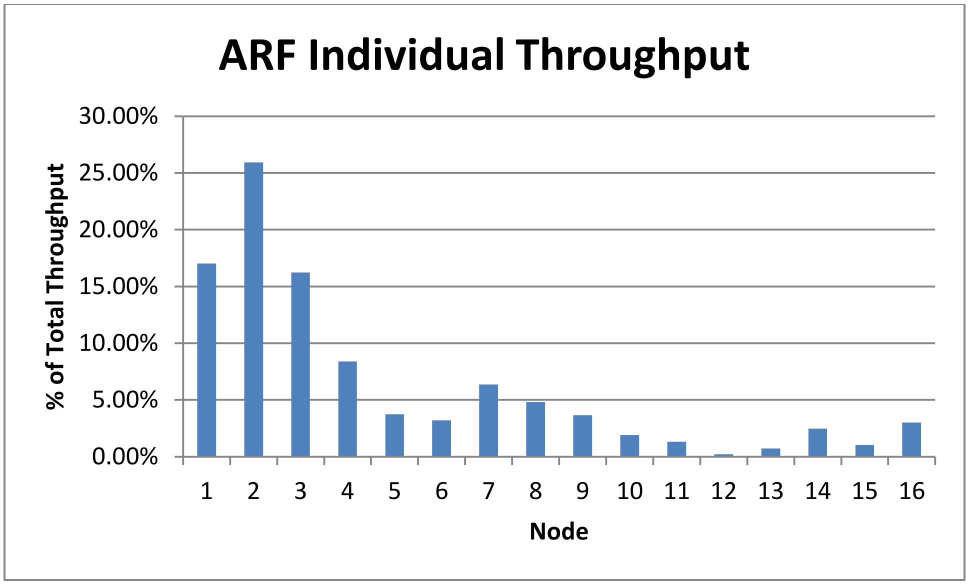


(a) Throughput

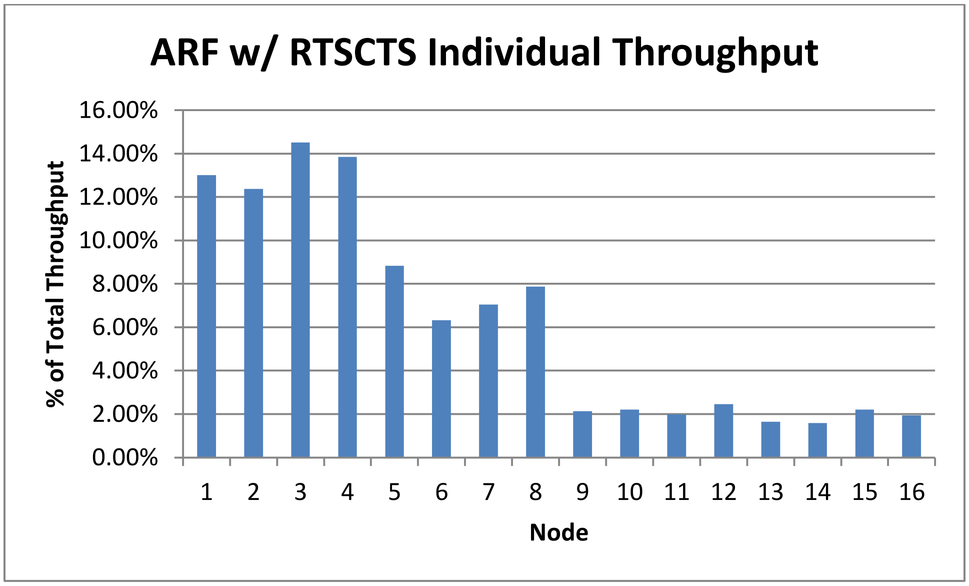


(b) Fairness

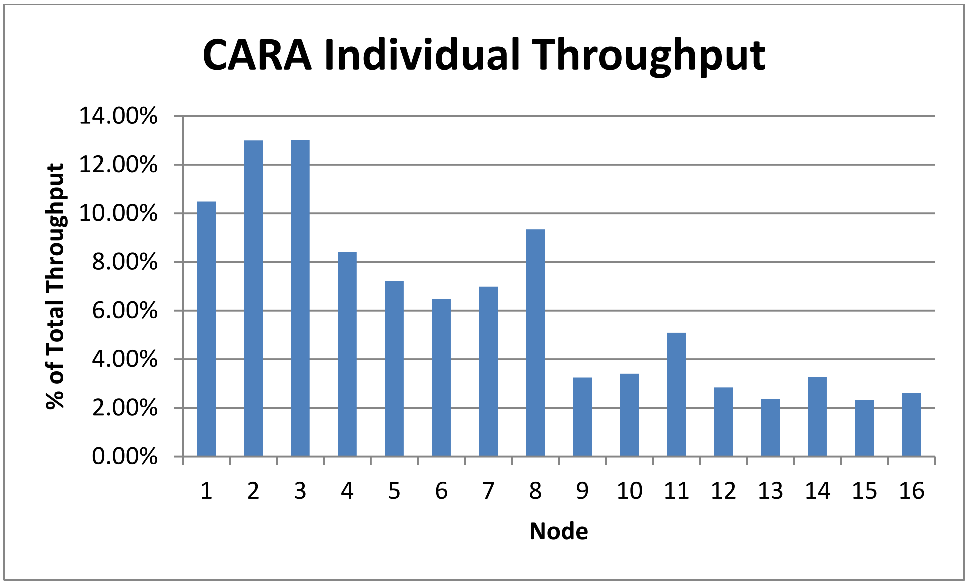
Figure 6.4: Performance with varying number of nodes for medium distance grid layout



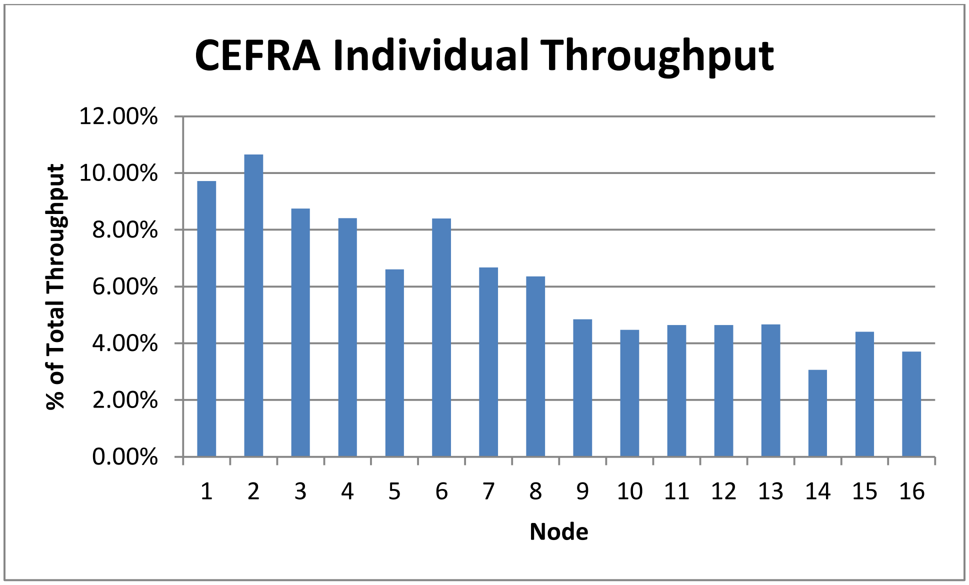
(a) ARF Individual Throughput



(b) ARF w/ RTSCTS Individual Throughput

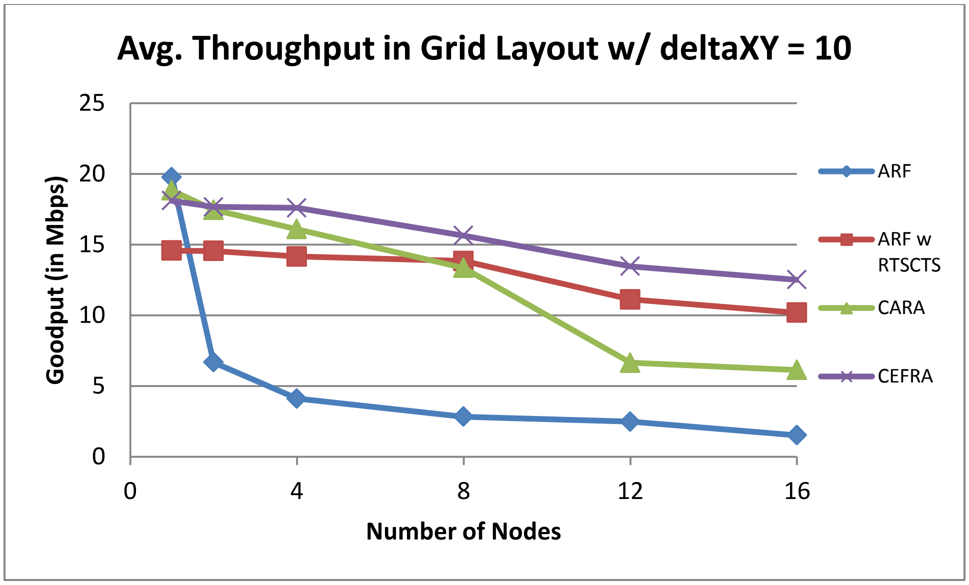


(c) CARA Individual Throughput

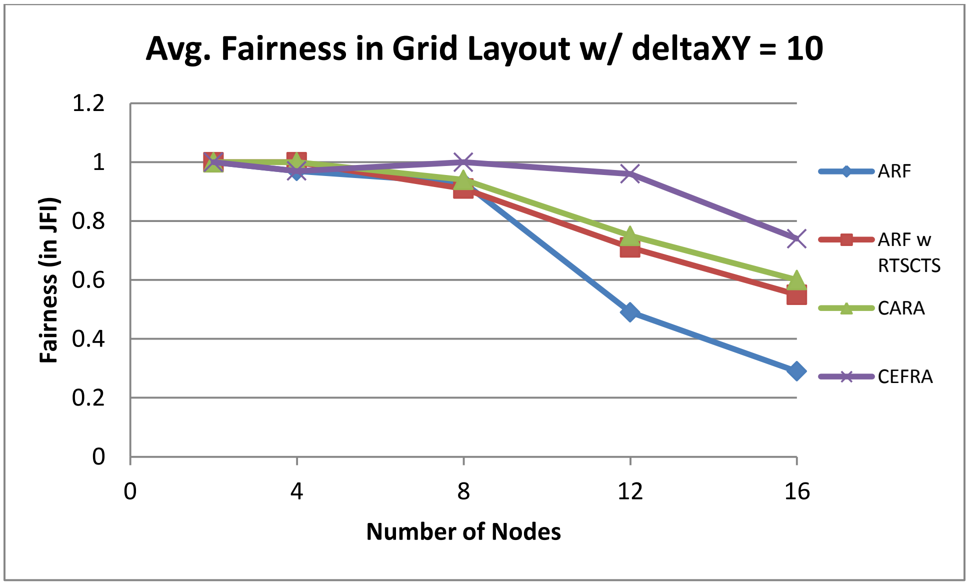


(d) CEFRA Individual Throughput

Figure 6.5: Individual Throughputs of different schemes in the medium distance grid layout (deltaXY = 5 m)

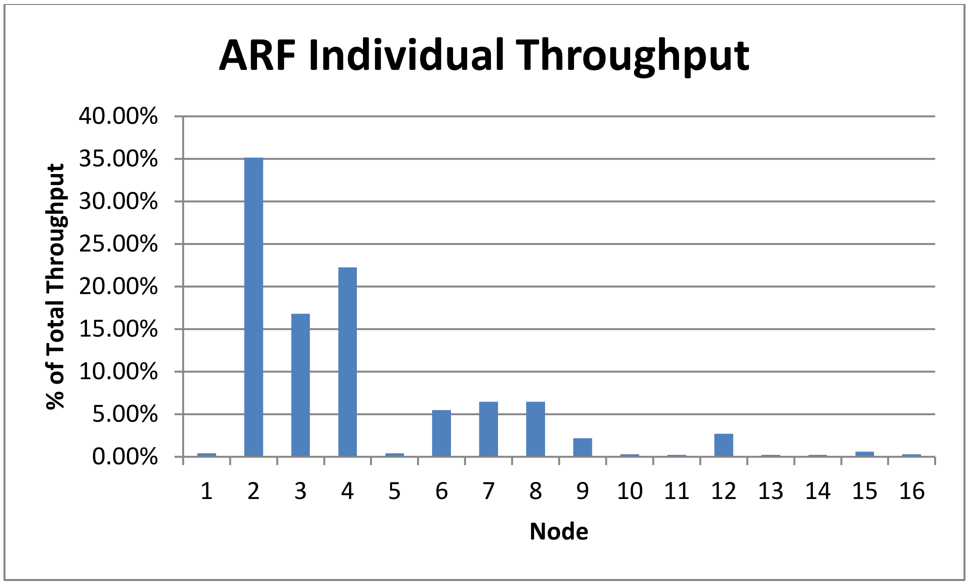


(a) Throughput

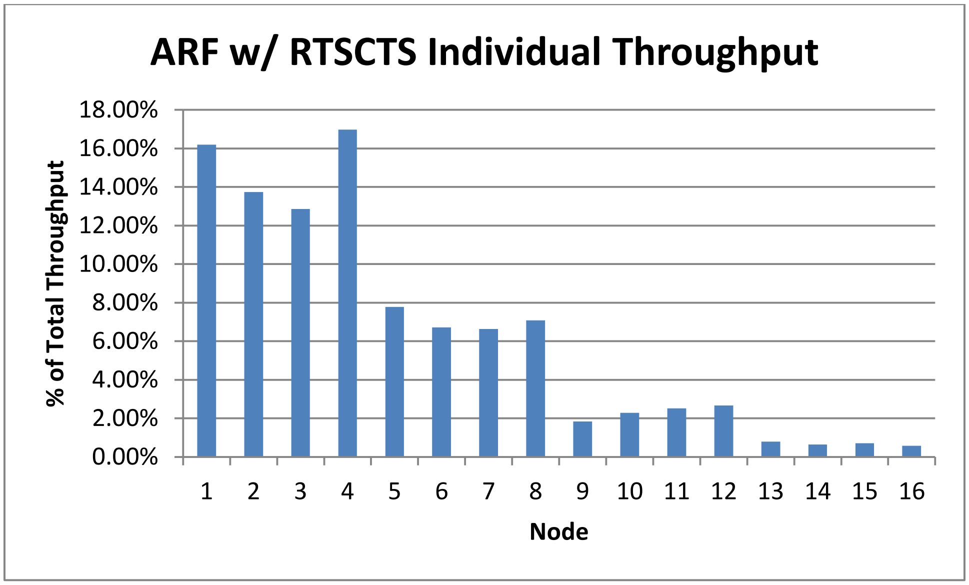


(b) Fairness

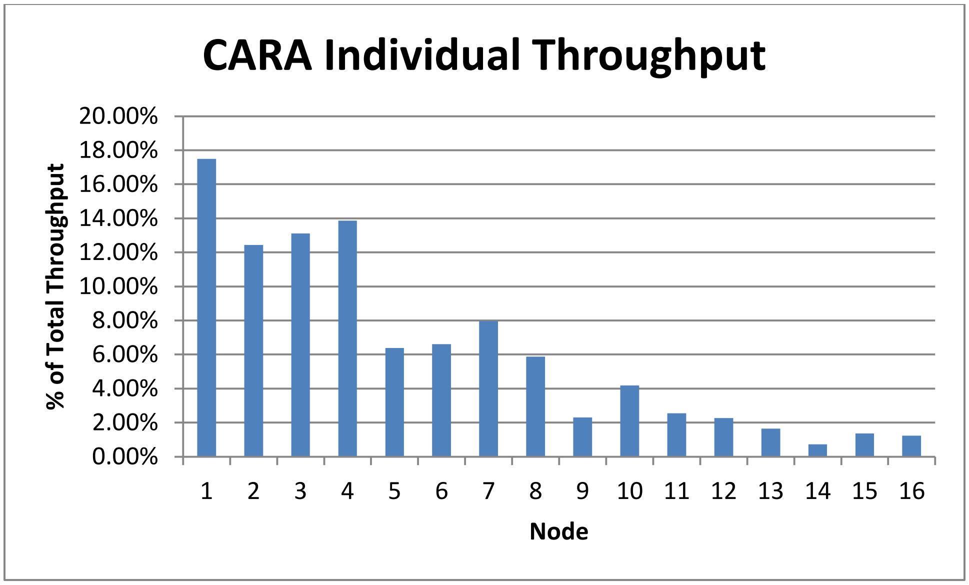
Figure 6.6: Performance with varying number of nodes in the far distance grid layout



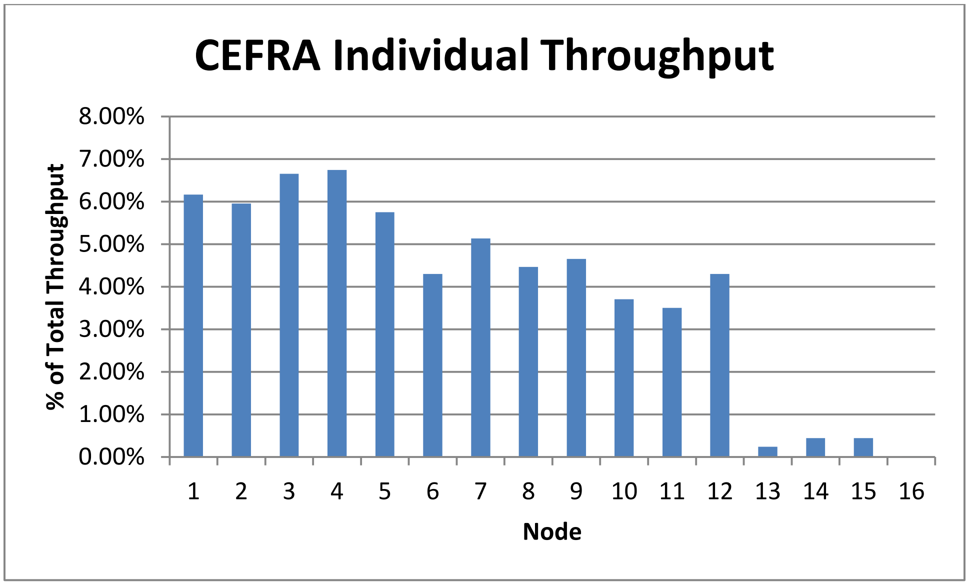
(a) ARF Individual Throughput



(b) ARF w/ RTSCTS Individual Throughput



(c) CARA Individual Throughput



(d) CEFRA Individual Throughput

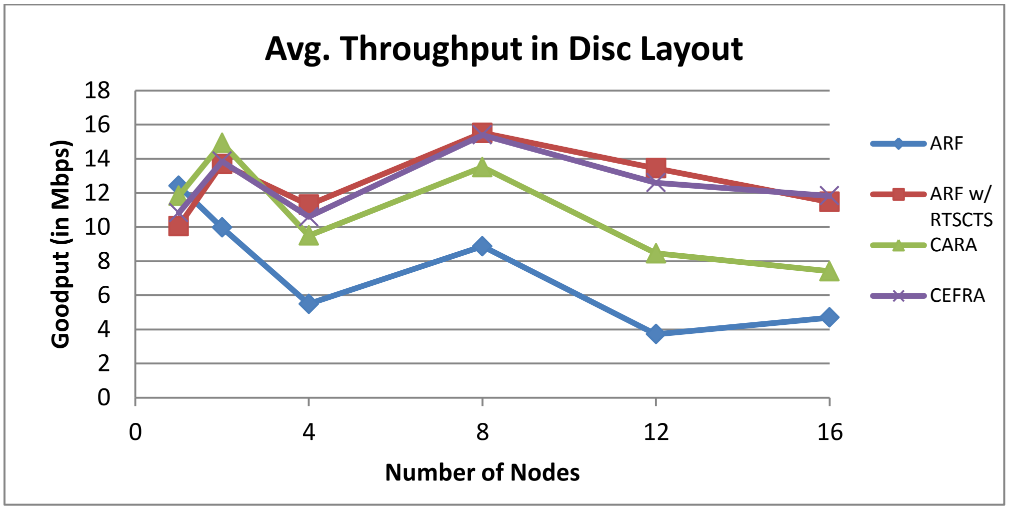
Figure 6.7: Individual Throughputs of different schemes in the far distance grid layout (deltaXY = 10 m)

In a real life WLAN network, nodes will likely not be located in a grid with set distance away from the AP with the exception of offices. They are more likely to be randomly distributed within a certain range. To emulate such environment, nodes are distributed out randomly with a uniform distribution in a disc. The radius of the disc was set to 25 m and the AP is located at the center of the disc. Simulations were repeated 5 times with different random placement of nodes. Same node locations are used for a single run of each rate adaptations for comparison purposes. The averaged throughput and fairness of the adaptations can be seen in Figure 6.8.

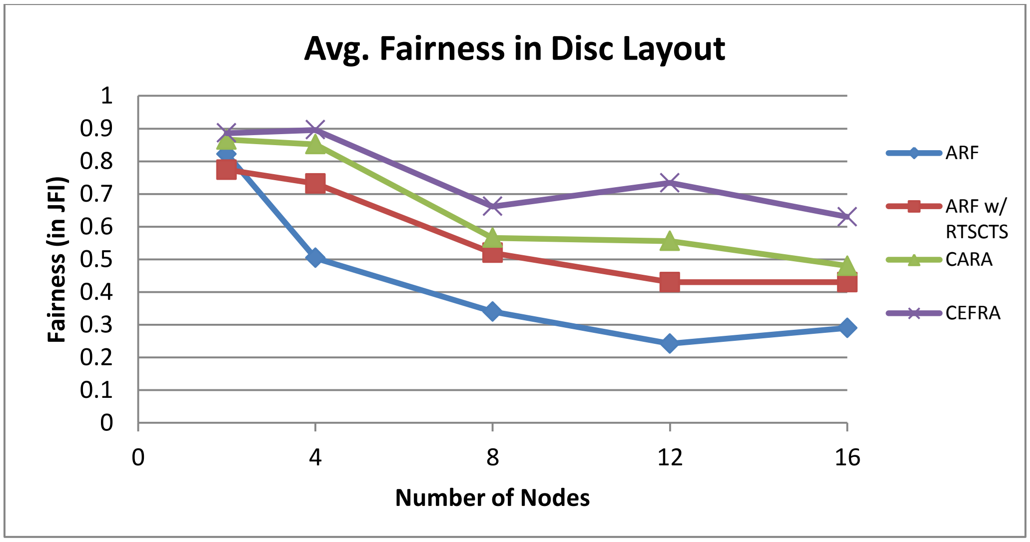
Due to the random placement of the nodes and the same node locations for all the rate adaptations, the throughput behavior is similar for all of them. CEFRA and ARF with RTS/CTS perform the best for number of nodes greater than 2. However, the gap for 1 or 2 nodes is even smaller than it was in the grid layout due to the random placements and the unequal spacing of the nodes. Curiously, ARF with RTS/CTS performs little better than CEFRA in some cases unlike in the grid layout cases. This is because random placements are likely to have few nodes that are significantly closer to the AP than all the others. CEFRA would try to mitigate these nodes from dominating the channel and preventing other nodes from accessing the channel. In these cases, other rate adaptations such as ARF with RTS/CTS would increase the overall throughput by having the close nodes dominate the channel. To illustrate such case, packet statistics of one single run of 12 nodes case for CEFRA and ARF with RTS/CTS are shown in Table 6.2. For easy comparison, a chart displaying individual throughputs is illustrated in Figure 6.9.

Even in random placements, CEFRA was able to maintain good fairness across varying number of nodes and outperformed other adaptations by significant margins.

CEFRA, on average, had about 30% improvement in fairness over CARA, which had the next highest fairness. Fairness of ARF was, by a large margin, the worst and CEFRA often had fairness that was couple of times higher than ARF.



(a) Throughput



(b) Fairness

Figure 6.8: Average performance in the uniform random disc layout

ARF w RTSCTS	Distance from AP (in m)	Packets Sent	Packets Received	% of packets received	% of total packets
1	20.28	1448	785	54.21%	4.85%
2	24.65	737	355	48.17%	2.19%
3	23.02	337	146	43.32%	0.90%
4	15.77	2004	1360	67.86%	8.40%
5	7.94	7767	7593	97.76%	46.89%
6	18.41	936	517	55.24%	3.19%
7	21.97	898	319	35.52%	1.97%
8	12.42	3162	2412	76.28%	14.90%
9	16.94	1454	878	60.39%	5.42%
10	14.93	1750	1151	65.77%	7.11%
11	24.11	532	168	31.58%	1.04%
12	22.01	1042	508	48.75%	3.14%
Total		22067	16192	73.38%	

(a) ARF w/ RTSCTS

CEFRA	Distance from AP (in m)	Packets Sent	Packets Received	% of packets received	% of total packets
1	20.28	1806	1164	64.45%	8.38%
2	24.65	1129	630	55.80%	4.54%
3	23.02	658	300	45.59%	2.16%
4	15.77	1989	1316	66.16%	9.48%
5	7.94	3896	3038	77.98%	21.88%
6	18.41	852	443	52.00%	3.19%
7	21.97	1426	713	50.00%	5.14%
8	12.42	2662	1886	70.85%	13.58%
9	16.94	1954	1241	63.51%	8.94%
10	14.93	2704	1856	68.64%	13.37%
11	24.11	799	368	46.06%	2.65%
12	22.01	1544	929	60.17%	6.69%
Total		21419	13884	64.82%	

(b) CEFRA

Table 6.2: Packet statistics for ARF w/ RTSCTS and CEFRA

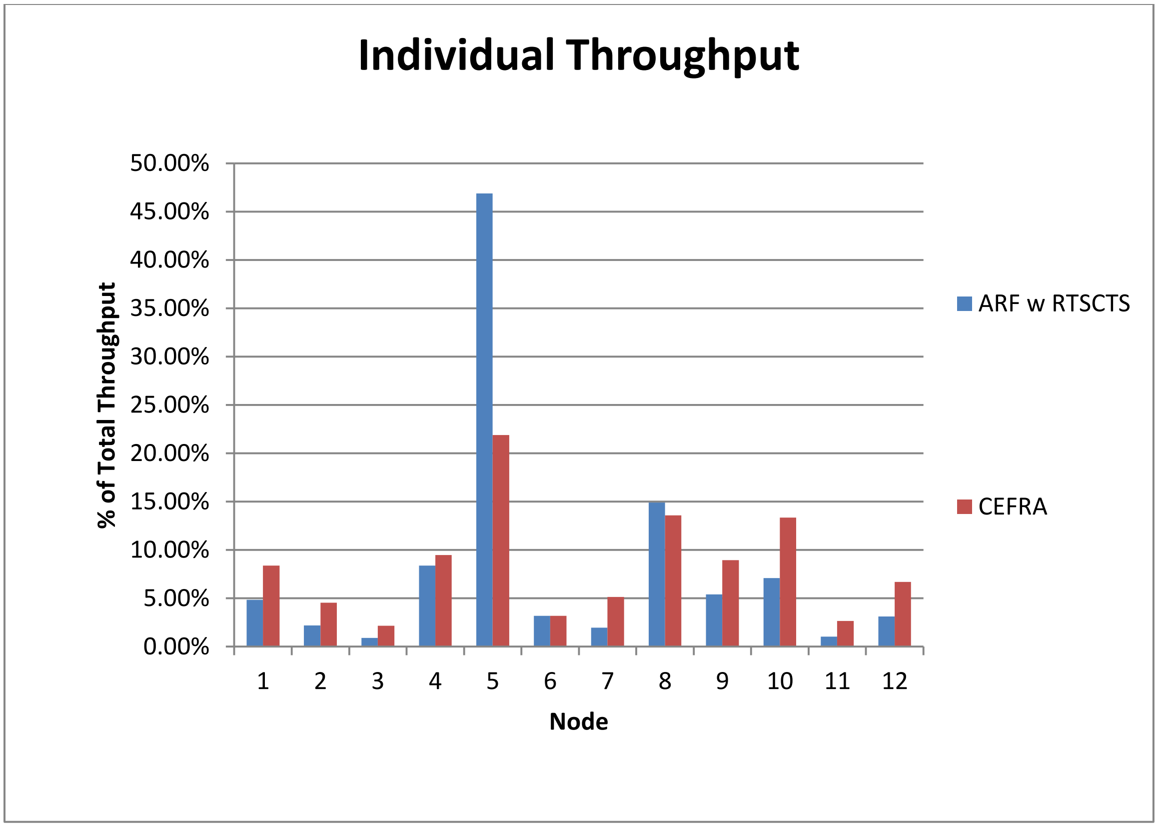
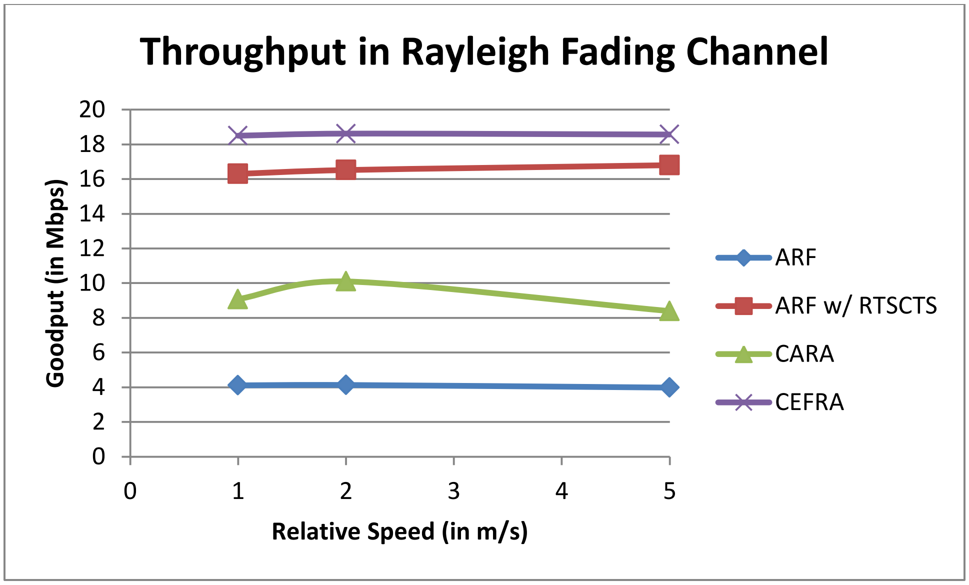


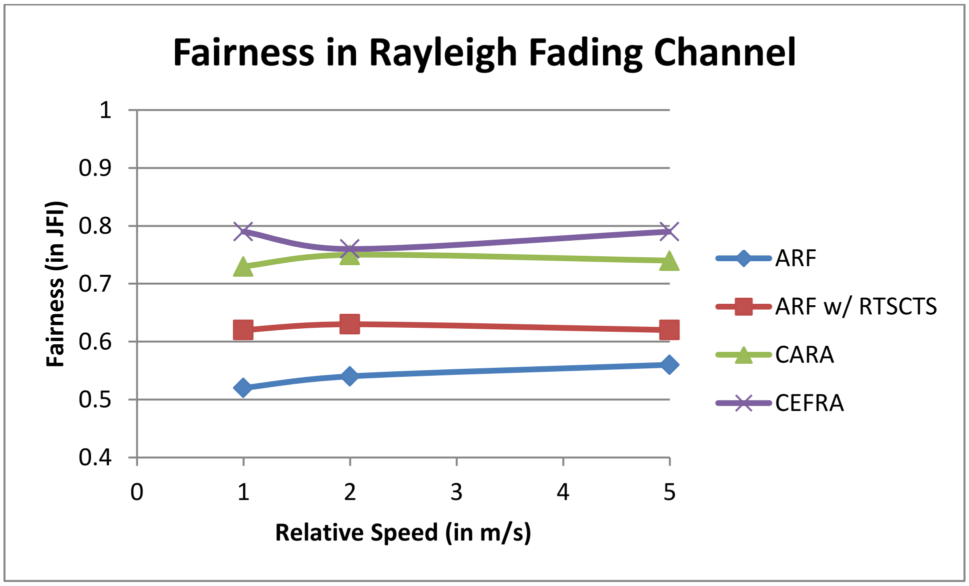
Figure 6.9: Individual throughputs of ARF w RTS/CTS & CEFRA

In a typical WLAN setting, studies have shown multipath fading to be of significant factor in performance. So far only path-loss fading and stationary setting of the nodes were considered. Multi-path fading would lead to small fluctuations in the received signals. To test the performance of CEFRA under such time-varying channel, Rayleigh fading with small relative velocities of 1, 2, and 5 m/s were added to the simulations. Although nodes are not moving, fluctuations can happen when there are other objects in the environment moving relative to the nodes. Everything else is kept the same as the original grid layout with deltaXY equal to 2.3 m and 16 nodes. The performance results are shown in Figure 6.10.

All the adaptations suffered slightly in Rayleigh fading environment in terms of throughput and fairness. CARA had the biggest drop-off in throughput with around 5 Mbps from the non-Rayleigh channel. Others including CEFRA had about 1 Mbps decrease in throughput. While fairness decreased slightly for all of them, CEFRA takes the biggest hit in fairness in the fading channel with CARA having the second biggest drop-off. Rayleigh fading seems to have bigger impact on rate adaptations that are fairer. CEFRA still has the highest fairness, but the margin is smaller in the Rayleigh channels. This drop in fairness appears to be caused by lower transmission success rates in the more distant nodes. Due to RTS being sent at higher rates based on the RSSI of the previous packet, fading causes the nodes to overestimate the quality of the channel quality and send packets at more ambitious rates that have higher chance of failing, resulting in the drop in fairness of CEFRA. Despite the drop, CEFRA had the highest throughput and fairness in the Rayleigh channels.



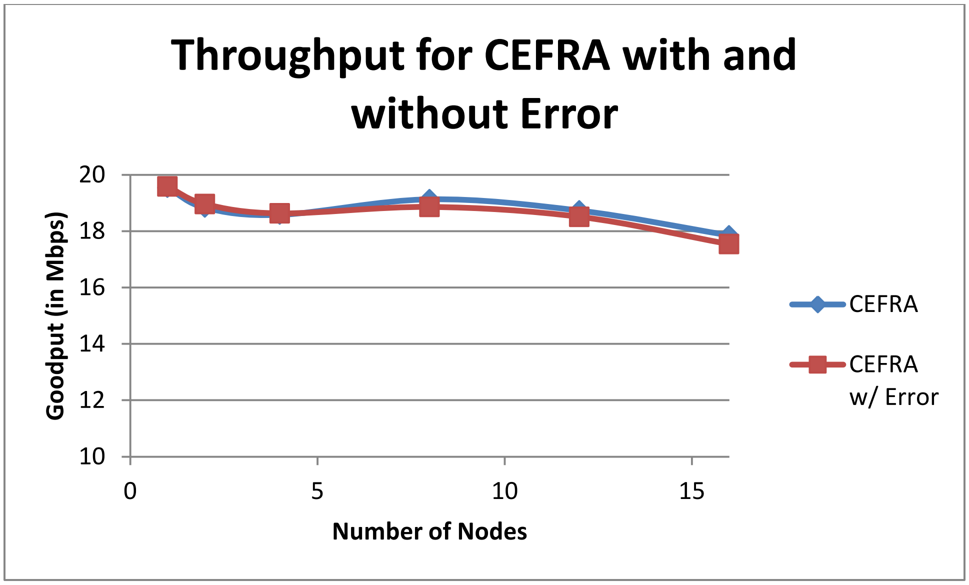
(a) Throughput



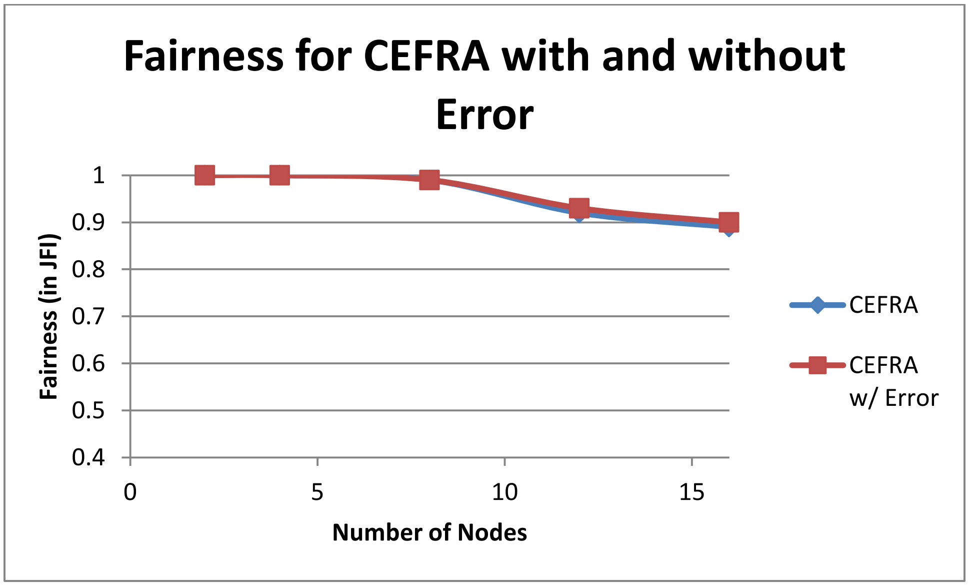
(b) Fairness

Figure 6.10: Performance in the close distance grid layout with Rayleigh Fading

One of the biggest drawbacks to using a SNR-based rate adaptation is that the RSSI measurement is not accurate enough. RSSI thresholds for all the rates are limited to a small interval of the total RSSI measurement range with one RSSI interval representing 1 dB. To test how CEFRA would perform in real-life with such constraint, measurement errors were added to the SNR measurement in ns-3. Gaussian random variable with zero mean and variance of one was added to the SNR measurement and rounded to the nearest integer to represent the typical RSSI measurement. Results for CEFRA with and without the measurement error are shown in Figure 6.11. Even with the measurement error added, performance of CEFRA does not decrease much. In fact, only throughput suffers slightly while fairness stays the same. This result illustrates that CEFRA is able to perform well even with the limitations of RSSI measurement.



(a) Throughput



(b) Fairness

Figure 6.11: Performance of CEFRA with measurement error in the medium distance grid layout

Chapter 7

Conclusion

In this paper, the effect that rates and the capture effect have on each other was investigated and discovered that there is an unfairness problem in wireless networks that use any modern day rate adaptation algorithms. Almost all of these rate adaptations were designed with increasing the overall network throughput in mind and do not take the capture effect and the location dependent unfairness into account. They simply improve throughput by giving more access to the nodes that are close to the AP and have high RSS. By doing so, the individual throughputs of these close nodes and occasionally, the overall throughput improve, but more distant nodes suffer terribly. This behavior is not ideal and does not lead to real life improvement since only select few will enjoy good throughput and a consistent performance across the network is preferred. Fairness should be a key metric in performance evaluation of rate adaptation algorithms. A rate adaptation algorithm that optimizes both throughput and fairness is needed.

Through experimental testing, the characteristic that having higher bit rate lowers the probability of capture for stronger sender is confirmed. By using higher rates for nodes that are closer to the AP, higher fairness was achieved. Using lower bit rates for weaker node had almost negligible effect on the capture probability, thus weaker nodes should use the highest rate that can be supported given the channel condition.

Using these results, a new approach to rate adaptation called CEFRA was proposed. CEFRA is designed to solve the unfairness problem while maintaining high overall throughput as well. CEFRA achieves this by altering the rates of RTS frames to use higher bit rates according to the RSSI of frames to gauge the relative strength of each node. By using RTS/CTS and at higher rates, the collision probability is minimized and system throughput is guaranteed regardless of contention level. Data frames are also sent at optimal rate using RSSI to fully utilize the channel.

The throughput and fairness performance of CEFRA was evaluated using ns-3 with PhySim-WiFi add-on for complete PHY-MAC layer simulation to provide accuracy. It was tested under various scenarios in terms of network topology and channel conditions. CEFRA outperforms all the other rate adaptation algorithms in terms of fairness and demonstrates that fairness can be maintained through proper rate adaptation. CEFRA’s overall throughput is also among the best performers especially in medium to heavily congested networks. Its only drawback is that it does not produce the best throughput result when there are only few nodes due to the overhead of RTS/CTS. It is shown that CEFRA maintains high overall throughput as well as high fairness regardless of channel conditions, providing a consistent performance that is desirable in practice, especially with increasing number of highly congested and underperforming WLANs employed today.

Bibliography

[1] “The Network Simulator – ns-3,” [Online], Available: http://nsname.org/

[2] “PhySImWiFi for NS-3,” [Online], Available: http://dsn.tm.kit.edu/english/ns3-physim.php/

[3] “MadWifi project,” Sep. 2005 [Online]. Available: http://madwifi.org/

[4] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE Journal of Selected Area in Communications*, vol. 18, no. 3, March 2000.

[5] Jaehyuk Choi, Jongkeun Na, Yeon-sup Lim, Kihong Park, Chong-Kwon Kim, "Collision-aware design of rate adaptation for multi-rate 802.11 WLANs", *Selected Areas in Communications, IEEE Journal on*, On page(s): 1366 - 1375 Volume: 26, Issue: 8, October 2008.

[6] J. P. Pavon and S. Choi, "Link adaptation strategy for IEEE 802.11 WLAN via received signal strength measurement", *IEEE ICC 2003*, pp.1108 -1113 2003.

[7] J. Kim , S. Kim , S. Choi and D. Qiao "CARA: Collision-Aware Rate Adaptation for IEEE 802.11 WLANs", *Proc. IEEE INFOCOM*, 2006.

[8] J. Bicket, "Bit-rate selection in wireless networks," Master's thesis, Masschusetts Institute of Technology, 2005.

[9] S. Biaz, S. Wu, "Rate adaptation algorithms for IEEE 802.11 networks: A survey and comparison," *Computers and Communications, 2008.*

[10] M. Lacage, M. Manshaei, and T. Turletti, "IEEE 802.11 Rate Adaptation: A Practical Approach," in *MSWiM04*, 2004, pp. 126-134.

[11] Z. Li, A. Das, A.K. Gupta, and S. Nandi, "Full auto rate MAC protocol for wireless ad hoc networks," IEEE Procedings on Communication, vol. 3, pp. 311-319, 2005.

[12] S. Wong, H. Yang, S. Lu and V. Bharghavan, "Robust Rate Adaptation for 802.11 Wireless Networks," in MobiCom'06, Angeles, California, USA., September 2006, pp. 146-157.

[13] K. Ramachandran, H. Kremo, M. Gruteser, P. Spasojevic, I. Seskar, "Scalability Analysis of Rate Adaptation Techniques in Congested IEEE 802.11 Networks: An ORBIT Testbed

Comparative Study," *World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. IEEE International Symposium on a* , vol., no., pp.1,12, 18-21 June 2007.

[14] J. Camp, E. Knightly, "Modulation Rate Adaptation in Urban and Vehicular Environments: Cross-Layer Implementation and Experimental Evaluation," *Networking, IEEE/ACM Transactions on* , vol.18, no.6, pp.1949,1962, Dec. 2010.

[15] Jeongkeun Lee, Wonho Kim, Sung-Ju Lee, Daehyung Jo, Jiho Ryu, Taekyoung Kwon, and Yanghee Choi. "An Experimental Study on the Capture Effect in 802.11a Networks," In*Proceedings of the second ACM international workshop on Wireless network testbeds, experimental evaluation and characterization* (WinTECH '07). ACM, New York, NY, USA, 2007, pp. 19-26.

[16] Noun Choi, S. Venkatesan, "Eliminating Location Dependent Unfairness in WLANs," *Vehicular Technology Conference, 2006. VTC-2006 Fall. 2006 IEEE 64th* , vol., no., pp.1,5, 25-28 Sept. 2006.

[17] P. Patras, Qi Hanghang, D Malone, "Exploiting the capture effect to improve WLAN throughput," *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2012 IEEE International Symposium on a* , vol., no., pp.1,9, 25-28 June 2012.

[18] P. Kumar, A. Krishnan, "Throughput Analysis of the IEEE 802.11 Distributed Coordination Function Considering Capture Effects," *Emerging Trends in Engineering and Technology (ICETET), 2010 3rd International Conference on* , vol., no., pp.836,841, 19-21 Nov. 2010.

[19] A. Nyandoro, L. Libman, M. Hassan, "Service Differentiation Using the Capture Effect in 802.11 Wireless LANs," *Wireless Communications, IEEE Transactions on* , vol.6, no.8, pp.2961,2971, August 2007.

[20] Xiaolong Li, Qing-An Zeng, "Capture Effect in the IEEE 802.11 WLANs with Rayleigh Fading, Shadowing, and Path Loss," *Wireless and Mobile Computing, Networking and Communications, 2006. (WiMob'2006). IEEE International Conference on* , vol., no., pp.110,115, 19-21 June 2006.

[21] Z. Hadzi-Velkov, B. Spasenovski, "An analysis of CSMA/CA protocol with capture in wireless LANs," *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE* , vol.2, no., pp.1303,1307 vol.2, 20-20 March 2003.

[22] A. Zahedi, K. Pahlavan, "Throughput of a wireless LAN access point in presence of natural hidden terminals and capture effects," *Personal, Indoor and Mobile Radio Communications, 1996. PIMRC'96., Seventh IEEE International Symposium on* , vol.2, no., pp.397,401 vol.2, 15-18 Oct 1996.

[23] A. Iyer, C. Rosenberg, A. Karnik, "What is the right model for wireless channel interference?," *Wireless Communications, IEEE Transactions on* , vol.8, no.5, pp.2662,2671, May 2009.

[24] J. Mittag, S. Papanastasiou, H. Hartenstein, E.G. Strom, "Bridging the Gap between Physical Layer Emulation and Network Simulation," *Wireless Communications and Networking Conference (WCNC), 2010 IEEE* , vol., no., pp.1,6, 18-21 April 2010.

[25] C.T. Lau, C. Leung, "Capture models for mobile packet radio networks," *Communications, IEEE Transactions on* , vol.40, no.5, pp.917,925, May 1992

[26] R. Jain, D. Chiu, W. Hawe, “A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems,” DEC Tech Report TR-301, Tech. Rep., September 1984

[27] A. Kamerman, LMonteban, “WaveLAN II: A high-performance wireless LAN for the unlicensed band,” *Bell Labs Technical Journal*, pp. 118-133, 1997

[28] A. Goldsmith, *Wireless Communications*. New York: Cambridge University Press, 2005.