Title
Multi-Channel and Multi-Rate Adaptation for High-Throughput Wireless Networks

Author
Nguyen, Duy Duc

Publication Date
2012

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
SANTA CRUZ

MULTI-CHANNEL AND MULTI-RATE ADAPTATION FOR HIGH-THROUGHPUT WIRELESS NETWORKS

A dissertation submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER ENGINEERING

by

Duy Duc Nguyen

December 2012

The Dissertation of Duy Duc Nguyen
is approved:

J.J. Garcia-Luna-Aceves, Chair

Katia Obraczka

Cedric Westphal

Tyrus Miller
Vice Provost and Dean of Graduate Studies

Table of Contents

List of Figures

Abstract

Multi-Channel and Multi-Rate Adaptation for High-Throughput Wireless Networks

by

Duy Duc Nguyen

As more mobile devices are becoming connected everyday, we face an unprecedented demand and growth for bandwidth in wireless networks. The gradual shift from personal computers (PCs) to tablets and mobile phones brings new opportunities as well as challenges. Users tend to be mobile and demand instant access to their contents. Bandwidth usage ranges from simple browsing of web pages to audio and video streaming. Due to the limited and scarce resources, care must be taken in order to utilize the spectrum more efficiently, to mediate access to a shared common medium, and, especially, to disseminate information. This dissertation focuses on designing a new channel access scheme as well as using its channel access time more efficiently through multiple channels and dynamic rate adaptation for high-throughput wireless networks.

First, it provides a new multi-channel medium access control (MAC) for bandwidth exploitation. Current IEEE 802.11 uses one common channel for both control and data packets because it is simple even though many other adjacent channels are available and left intact. We can improve the throughput significantly by simply allowing other channels to be used. The added complexity includes channel switching and additional control packets to ensure that the neighbors are aware of the channel selection. Collision freedom is difficult in wireless environment, but I show that it can be achieved for a single transceiver without requiring temporal synchronization among nodes through an asynchronous split phase together with an observation phase as well as a unique handshake.

Second, it provides an in-depth study and analysis of rate adaptation for single and multiple antenna systems, together with a new throughput enabled approach. Throughput-based

approach eliminates complexity of sender-initiated rate adaptation by incorporating errors and multiple access interference (MAI) implicitly. Its main job is to simply find the best attainable throughput at the sender and keep using it.

My contributions include providing a collision-free multiple channels access MAC with no temporal synchronization and an efficient, robust, yet simple, throughput enabled rate adaptation for wireless networks. Each proposed approach to multi-channel MAC and multi-rate adaptation is validated through analysis and extensive simulations and prototype implementations. My work provides a better understanding on the current limitation of the wireless systems and new insights for further improvements, from the time node accessing the medium through multi-channel MAC to making the best use of its access time through an efficient and robust rate adaptation.

Dedicated To

My mother for being the sunshine of my life

Acknowledgments

I would like to thank my advisor JJ Garcia-Luna-Aceves for his supervision and mentorship. Of course, being an apprentice to such a famous advisor can be a burden sometimes. There is more pressure than I could imagine even though he simply wanted his students to be successful. I may be thinking that my “PhD boat” has set sail upon the submission of this thesis, but I feel I will still need his guidance in order to discover more wonderful hidden treasures of research. I am always amazed at his knowledge and insights. At all the conferences that I have been to, researchers in the community often give praise about my advisor. How he has done some very interesting work. I feel humble and honored to be one of his students. I hope to continue collaborating and publishing papers with him in the future because simply I just cannot get enough of his knowledge.

I would like to thank Prof. Cedric Westphal for many long discussions and critiques on my research. The commute from San Jose to Santa Cruz is often filled with brainstorming sessions. Thank you for helping me refining many of my ideas. I would like to thank Prof. Katia Obraczka for being my thesis committee member and a co-author of my first paper. I really value her insights, critiques and reviews of my work, especially on the multi-channel MAC paper.

I would like to thank my colleagues in CCRG Lab. I would like to thank all my wonderful friends in Santa Cruz and San Jose. Being tied down by the workload of graduate school, it may seem like I am avoiding everyone, but it is the opposite. Everyone is so special to me.

Finally, I would like to thank my parents and my sister for their support. Without them, I would not have made it this far. Even though they have no clues what I am doing and why I am in school for so long, they are the main motivation that keeps me going every day.

The text of this dissertation includes reprints of the following previously published material in [52], [12], [13], and [11]. These are collaborated work with my advisor JJ Garcia-Luna-Aceves, Cedric Westphal, and Katia Obraczka.

Chapter 1

Introduction

Bandwidth demand in wireless networks continues to rise as existing and emerging applications become increasingly popular, including multimedia streaming, emergency response and disaster rescue operations, smart environments, and others. With the popularity of social networks, users tend to spend more time on their mobile phones and tablets and demand seamless Internet experience. As more devices are connected and online, they place heavy burden on the wireless networks to support them. As the networks get more crowded and the number of users increases, scarce bandwidth and resources become an issue.

In wireless networks, Medium Access Control (MAC) plays a central role in coordinating access to the medium in an orderly fashion. It lays out a set of rules and policy that participating nodes must follow in order to avoid collisions and resolving collisions. Indeed, while IEEE 802.11's Distributed Coordinate Function (DCF) has been originally designed for one common channel, the IEEE 802.11 PHY offers multiple channels transmission capability. Multiple channels transmissions allow us to take advantage of the untapped resources that are available to us while, at the same time, improving the congestion bottleneck of single channel. A number of multi-channel MAC protocols have been proposed as a way to increase performance and network utilization. However, transmission over multiple channels raises a number of challenges including hidden terminals over multiple channels and node synchronization.

I introduce the Asynchronous Multi-Channel Medium Access Control protocol, or AM-MAC for short, an asynchronous collision-free multi-channel medium-access control protocol.

AM-MAC's main features include its efficient utilization of the medium and energy efficiency without the need for time synchronization. Additionally, AM-MAC's simplicity and ease of implementation makes it well suited for low-cost, limited-capability devices.

Once having access to the medium, node now must make full use of its channel access time. In order to accomplish this, it must transmit as quickly as possible so that it can return the medium to others. The main challenge is how to transmit as quickly as possible given interference, collisions, and natural radio waves phenomena. This is exactly the job of rate adaptation, an important component IEEE 802.11 physical layer. Unlike wired networks, wireless networks have to deal with fading, attenuation, and especially interference. Rate adaptation consists of attempting to find the best rate for sending data packets in adverse, unexpected, and interference-prone environments.

Different bit rates in IEEE 802.11 are made possible by the type of modulation they carry, which is obtained by varying changes in the amplitude, phase, and frequency of the waveform. Higher bit rates allow more data to be transmitted on high quality links but suffer low throughput on lossy links; lower bit rates allow lower packet loss even in the presence of lossy links.

Multiple access interference (MAI) and natural phenomena associated to radio wave propagation are the key reasons for throughput reduction in wireless networks. Adapting to them is complicated by the unpredictability of interference. A network may be subject to little or a lot of interference, depending on the characteristics of the environment, the network density, and node movement, and environmental mobility. A major concern with MAI is that it increases very rapidly with node density and impacts the network layer, which causes MAI to spread over multiple hops as nodes attempt to route around congestion. Given that MAI depends on the time that each transmission occupies the channel, nodes must attempt to transmit at the highest data rates that render the largest throughput. Because the algorithm used for rate adaptation must operate without any a-priori information on the state of the system, nodes must increase or decrease their rates quickly if they perceive significant degrees of successful or failed transmissions, and must change their rates smoothly otherwise.

Rate adaptation can be categorized into implicit and explicit approach. The former requires the sender to estimate the channel condition of the receiver by means of acknowledgement packets or received signal strength indicator (RSSI). The latter requires the receiver to feed its channel condition back to the sender through signal and noise (S/N) measurements. They both have their advantages and disadvantages depending on the applications. Because of the simplicity and ease of modification on real systems, I opted for the implicit approach. I proposed a packet error rate adaptation by relying only on acknowledgement packets. Performance is monitored over a packet window and a time window. The time window ensures that we do not wait forever for gathering enough packets. Based on these error rate ratio, rates are adapted accordingly.

Due to the difficulty of tuning many different parameters for packet error rate approach, I proposed an alternative rate adaptation that relies solely on throughput measurements at the sender. End-user applications care about the throughput attained, rather than the transmission rate used or the loss ratios. This approach eliminates the complexity of setting different parameters and threshold.

My work on collision-free multi-channel MAC protocol shows we can take advantage of multiple unused channels for significant throughput gains with minimal overhead. My simulations confirm the advantage of using multiple channels even with single transceiver. With increasing number of devices being connected, multiple channel usage can help alleviate the backlog and traffic congestion while, at the same time, minimizing multiple channel access interference.

My work on link and rate adaptation provides a simple yet intuitive solution to a very complex problem of rate adaptation. The insight is that implicit approach to rate adaptation may work very well. I show that we can have a robust rate adaptation with very minimal overhead and changes to the hardware. I extend my work from single input single output (SISO) to cover multiple input multiple output (MIMO) technology.

Together, this dissertation contributes by providing new efficient approaches for nodes to access the channel efficiently and make the best use of its channel access time.

Chapter 2

Background

2.1 Wireless Networks

Wireless technology is all around us. Often we take it for granted. From the music beaming from our radios to the images and sound on our televisions, from the voice and text messages on our mobile phones to the hand-free Bluetooth headset, the wireless technology affords us all the convenience of being connected in today's world.

Wireless networks allow many computers and devices to be connected without the need for cables. It allows them to be mobile. Generally, it is implemented through radio communication. Unlike wired networks, where everything is physically connected with clearly defined paths, wireless network works by encoding information bit into radio waves. The wireless medium is shared among all the participants. As a result, it is susceptible to interference which can make it hard for the receiver to decode the distorted information correctly. As radio wave propagates in space, it may encounter many obstacles that impacts its properties, either through absorption or reflection.

There are many types of wireless networks such as wireless personal area networks (PANs), wireless local area networks (WLAN), wireless mesh networks, wireless metropolitan area networks (MAN), and cellular network. These networks are differentiated by the distance they communicate. Wireless Mesh Networks refers to the type of ad hoc networks where each device can forward the messages on behalf of other nodes. Advanced medium access control

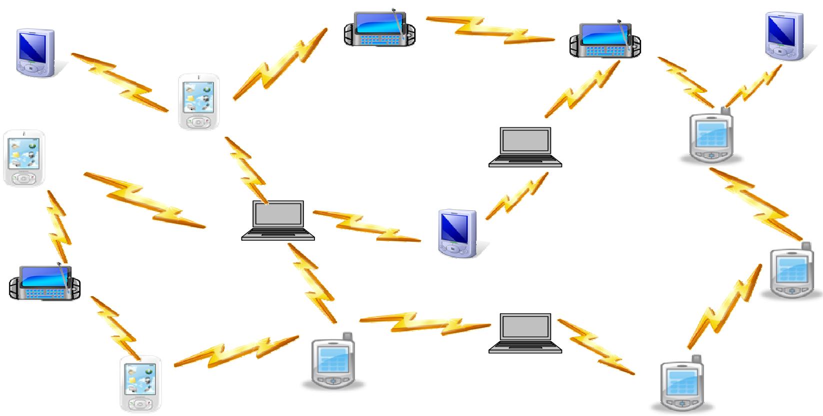


Figure 2.1: Wireless Mobile Ad Hoc Networks

(MAC) and routings are required for the maintenance of wireless mesh networks. Wireless Mesh Networks perhaps is one of the most active research area due to its ad hoc nature and the susceptibility of link failures due to natural phenomena of radio waves' interface.

2.2 IEEE 802.11 Physical Layer

In order for all devices to communicate with one another, we need a unified agreement among all the vendors. IEEE 802.11 Working Group plays this central role by providing this bridge in maintaining and creating IEEE 802.11 DCF standard, a widely deployed protocol for wireless networks. This set of standards allows for wireless communication in the 2.4, 3.6 and 5 GHz frequency bands. Each base version of the standard also has subsequent amendments for improvements and changes to the standard.

IEEE 802.11 Distributed Coordination Function (DCF) [30] plays a key role in mediating access to the medium. DCF employs Carrier Sense Multiple Access and Collision Avoidance (CSMA/CA) with binary exponential backoff algorithm. Each station that wishes to transmit must listen for the channel status for DCF Interframe Space (DIFS) interval. DIFS is equal to SIFS plus twice the slot time. Short Interframe Space (SIFS) is time interval between the data

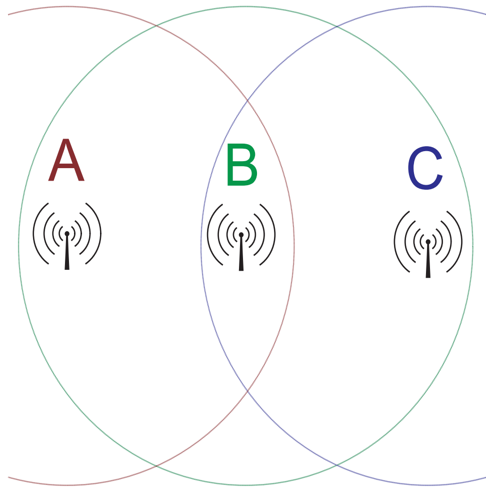


Figure 2.2: Hidden Terminal: A and C can each communicate with B, but are hidden from each other [28]

frame and its acknowledgement.

If the channel is sensed busy, the station defers its transmission until the ongoing transmission terminates based on a slotted binary exponential backoff algorithm. A random backoff interval is chosen uniformly from $[0, CW - 1]$ for initializing the backoff timer, where CW is the current contention window size. The backoff timer runs if the channel is sensed idle, otherwise it is paused during the data transmission and resumed when the channel is sensed idle again. If the transmission is unsuccessful, CW is set to CW_{min} in the first transmission attempt and is doubled for each subsequent retransmission up to a maximum value of CW_{max}.

Simply, nodes avoid collisions by attempting to transmit when the channel is sensed to be "idle". Upon collisions, it must back off appropriately. This back-off mechanism as described above is conservative and sub-optimal, but it does its job in limiting the number of collisions. Due to the differences in nodes' locations and receiver sensitivity, some nodes may not be able to hear one another's broadcast. Request To Send (RTS) and Clear To Send (CTS) control packets may be employ to mitigate the possibility of collisions. Stations receive control packets such as RTS/CTS will set their Network Allocation Vector (NAV) accordingly. Duration field is extracted from the MAC layer frame headers for setting the NAV. NAV acts as a virtual carrier sensing to tell a station how long it should defer from accessing the medium. However, RTS/CTS handshake still does not completely eliminate the hidden terminal problem.

The IEEE 802.11 standard makes it possible for all devices from different vendors to communicate with one another. It plays an indispensable role in wireless networks. Any modification to the standard must be agreed and ratified by the standard or risking losing its compatibility.

2.3 Multi-channel MAC

In this section, I will provide the background to the medium access control for multiple channels. Because of the broadcast nature of wireless networks, nodes need to access the medium in an orderly fashion. If any node can transmit at any time with no policy in place, it will create a chaotic wireless network where there are many collisions, interference, congestion, and backlog of data packets.

IEEE 802.11 standard is designed for single channel usage at any given time even though there are non-overlapping and orthogonal channels available. IEEE 802.11b provides 3 orthogonal channels in 2.4 GHz spectrum [32], IEEE 802.11a provides 12 orthogonal channels in the 5 GHz spectrum [31], and IEEE 802.11n provides 9 non-overlapping 40MHz channel in 5GHz of 802.11n [34]. With MIMO systems, non-overlapping channels can provide the capacity up to Gigabit throughput. If we do not use other available channels, we are not taking advantage of the spectrum or bandwidth that are available to us. Given the significant performance improvement of using multiple channels and smaller channel switching delays, it is attractive to take advantage of these low cost solutions. The goal is to revisit the topic of multi-channel MAC and to show that we can modify 802.11 MAC protocol to take advantage of the whole spectrum, and at the same time, to show that multi-hidden terminal problems can be solved to provide a collision-free multi-channel access in a multi-hop environment.

Multi-Channel MAC approaches can be classified into the following categories: dedicated control channel, split phase, common hopping, and parallel rendezvous.

In the dedicated control channel approach, a channel is reserved exclusively for exchanging control information. Usually, a node is equipped with 2 radios, in which one is solely used for control information exchange. For example, the DCA protocol [82] maintains a dedicated radio for control messages and the other for data transmission. Since each node has two transceivers, it can always monitor the control channel. The multi-channel hidden terminal problem simply does not exist in this case. However, an obvious drawback is the low spectrum utilization since one radio is reserved exclusively for control data exchange. Other drawbacks of the dedicated control channel approach include higher cost and energy consumption associated with equipping and operating nodes with two radios. Recent work on full-duplex wireless design [36] may make the dedicated control channel approach more attractive by allowing wireless nodes to transmit and receive at the same time and on the same channel.

Split phase approaches avoid the need for multiple radios by having a dedicated common control channel and having nodes alternate between channel negotiation phases on the common channel and data transfer phases on the negotiated channels. This approach, however, requires tight time synchronization among all nodes in order to agree on the switching time between the control and data phases. Additionally, during the control phase, other channels become under-utilized. Another drawback is that if a node missed the negotiation period or did not succeed contending during this period, it would have to wait until the next negotiation phase to contend again. This approach is used in MMAC [65] and Wiflex [47].

MMAC protocol [65] is a multi-channel MAC based on the power management of the Power Saving Mechanism in 802.11, which is Ad Hoc Traffic Indication Messages (ATIM). Time is divided into beacon intervals, periodic beacon transmissions are used to synchronize every node in the network. During the ATIM window, nodes will exchange ATIM packets for transmission during data window, nodes that did not exchange any ATIM packets during this window would have to wait until the next beacon time. Because of the split phase approach, nodes have to hold back their transmission until the end of control data phase, and they cannot utilize all the channels during this period; it results in the inefficient use of bandwidth with increasing number of available channels. Besides, the common channel bottleneck is more serious than the dedicated channel approach because all nodes must contend during the negotiated phase.

Wiflex [47] is another multi-channel MAC for OFDM-like PHY that uses split phase, but does not require nodes to be synchronized. It is a multi-channel cooperative protocol using asynchronous split-phase with dynamic priority support, designed for OFDM-like multi-channel physical layer. The protocol is based on three phase: Observe, Review, Access. Each device has its own phases and associate durations. Node begins with an observation phase, and its purpose is to monitor others RTS/CTS exchange, to collect information to avoid potential conflicts. Then, node may begin its Review phase by transmitting RTS/CTS packets. However the handshake RTS/CTS is not sufficient to provide collision free data transmission and to cover all cases of hidden terminals. Nodes in collisions may not be able to keep up-to-date with channel selection by their neighbors.

Protocols that follow the common hopping approach have all idle nodes follow a common hopping sequence. A pair of nodes will stop hopping as soon as they agree on a hopping sequence to communicate; they will go back to hopping at the end of their exchange. CHMA [75] is an example of the common hopping approach. It eliminates the need for carrier sensing and code assignment by allowing the source-destination pair to agree and remain in the same hopping frequency in order to communicate. The main drawbacks are frequent channel switching and tight synchronization requirement.

With the parallel rendezvous approach, each node publishes its own channel hopping schedule. Senders learn about the receiver's current hopping sequence via a seed broadcast mechanism. Senders, then, must adopt the published sequence if they want to increase the time spend on the channel with the receiver. The advantage is that there is no single channel bottleneck. This approach ,however, does require tight synchronization. Examples include [5, 51].

BTMA(Busy-tone Multiple Access) [74] extends CSMA to eliminate the hidden terminal problem by dividing the bandwidth into a message channel and a busy-tone channel. The hidden terminal problem can seriously degrade the performance of CSMA. BTMA with hidden terminals is able to perform almost as well as CSMA without hidden terminals [74].

FAMA(Floor Acquisition Multiple Access) [19] is a MAC protocol for single-channel packet radio networks that influence our design for collision freedom; it consists of carrier sensing and collision avoidance dialogue between the sender and the receiver. The control of the channel (the floor) is assigned to at most one station at a time; objective is to ensure no packet collisions at the receiver side. FAMA accomplishes through a three-way RTS/CTS/DATA handshake similar to the Busy Tone Multiple Access strategy.

Figure 2.3: Mobile devices communicate with Access Point

2.4 Multi-Rate Adaptation

Once having access to the medium, nodes must make the best use of it. For example, they must transmit quickly in order to free the medium for others. Rate adaptation address this challenge by adapting rates to the dynamic channel condition in order to "pump" as many packets out as possible. Rate adaptation schemes can be classified based on whether explicit or implicit feedback to the transmitters is used. Explicit feedback requires the receiver to explicitly communicate the channel condition on the receiver's side back to the sender. Implicit feedback looks at acknowledgment (ACK) packets or other channel information (i.e., received signal strength indicator (RSSI)) to infer the channel conditions on the receiver's side. Note, I use the term rate control and rate adaptation interchangeably.

2.4.1 Explicit Feedback Approaches

Explicit feedback approaches can be viewed as receiver-driven rate adaptation, because the receiver dictates the rate that should be used. The receiver obtains its current channel condition and relays this information back to the sender.

Receiver Based Auto-Rate (RBAR) [29] selects the bit rate based on the S/N measurements. Upon processing a request to send (RTS) packet, the receiver calculates the highest bitrate and piggybacks this selected bit rate on the clear to send (CTS) packet. However, RBAR needs an accurate mapping between S/N values rates for different hardware.

Collision-Aware Rate Adaptation (CARA) [44] combines the RTS/CTS packets for Clear Channel Assessment (CCA) functionality to differentiate frame collisions and frame failures. The excessive use of the control packets in CARA may not be necessary.

Effective SNR [26] presents a delivery model by taking RF channel state as input and predicts packet delivery for the links based on the configuration of the Network Interface Controller (NIC). It takes advantage of the channel state information (CSI) either from feedback or estimated from the reverse path and computes its effective SNR by averaging the subcarrier BERs in order to find the corresponding SNR, where BER is a function of the symbol SNR and OFDM modulations.

Cross-layer wireless bit rate adaptation [78] uses confidence information from the physical layer to estimate the bit error rate (BER). The receiver sends a BER estimate to the sender in a link-layer feedback frame. The sender then uses this per-frame BER feedback to select the best transmit rate. The limitation with this approach is that it requires sending link-layer feedback frames at the lowest bit rate and in a reserved time slot.

Frequency-Aware Rate Adaptation (FARA) [60] computes the optimal choice of bit rate on each sub-band and sends it back to the sender in ACK packets. A sequence number is also added to acknowledgment (ACK) and data packets to prevent sender and receiver from going out of synchronization. However, the need to modify ACK packets and to synchronize makes this approach incompatible with the IEEE 802.11 standard.

The drawback of using CSI is that SNR needs to be measured instantaneously, and feedback delay may not allow mode adaptation on an instantaneous basis [23]. Because CSI itself is an approximation of the wireless channel, it may need to incorporate other information, such as higher-order statistics of SNR and Packet/Bit Error Rate or both for improving its accuracy and robustness [23].

2.4.2 Implicit Feedback Approaches

Implicit feedback approaches can be viewed as a sender-driven rate adaptation, given that the sender adapts its rate by inferring the channel conditions on the receiver side.

The Automatic Rate Fallback (ARF) scheme is one of the earliest rate control algorithms designed for WaveLAN-II [41]. Upon encountering a second missed acknowledgement of data packets, then it falls back to a lower rate. A counter is used to track the number of good and bad acknowledgement packets for upgrading rates accordingly. However, the limitation of ARF is that it was designed for a few rates and does not work well with current IEEE 802.11 implementation.

ONOE is the credit-based rate control algorithms originally developed by Atheros [4]. It extends ARF [41] to current IEEE 802.11. However, its limitation is that the credit-based system tends to be too conservative and often gets "stuck" using lower rates.

The Adaptive Multi Rate Retry (AMRR) scheme [46] introduces a Binary Exponential Back-off and adaptive threshold value depends on the feedback obtained from the number of attempted packets. The limitation of this approach is that binary exponential back-off tends to be too conservative in adapting rates.

The Sample rate control algorithm [39] begins by sending the data at the highest bit rate. Upon encountering four successive failures, the scheme decreases the bitrate until it finds a usable bitrate. At every tenth data packet, the algorithm picks a random bitrate that may do better than the current one. Minstrel [50], a widely deployed and popular Linux rate control, is an improved version of Sample, which takes into account the exponential weighted moving average statistics for sorting throughput rates. Unfortunately, Minstrel still spends 10 percent of transmitted frames in trying random rates when its current rate is working perfectly.

Robust Rate Adaptation Algorithm (RRAA) [80] uses short-term loss ratios to opportunistically adapt the rates. Like CARA [44], it employs an RTS filter to prevent collision losses from rate decreases. However, enabling RTS filtering upon encountering failed transmissions might not work as well as simply transmitting the data at lower rates. Besides, this adds an additional control overhead. Due to the nature of air interface, it is complex and difficult to predict the cause of the packet collisions.

Woo and Culler [81] propose an adaptive rate control that uses loss as collision signal to adjust the transmission rate for sensor networks. They assume the nodes have a notion of descendants or parents as in sensor networks. This rate control is probabilistic: the probability p is either incremented by an additive factor or multiplied by a multiplicative factor.

TARA [73] tries to estimate the per-packet transmission times and use these measurements to provide network state and identify network congestion. It approximates the time spent by monitoring station backoff or transmitting together with the busy channel time due to non-transmission and backoffs. With these measurements, TARA looks to distinguish between collision-dominated or noise-dominated environments. TARA periodically probes data rates that may perform better by monitoring the gain factor. However, protocol operating on per-packet basis may be too sensitive to environment changes.

2.4.3 Explicit and Implicit Approaches

In addition to incurring overhead by requiring the receiver to relay its channel state information back to the sender, the drawback with explicit approach is the possible stale feedback due to the dynamic channel conditions during data transmissions. If the channel coherence time is very short, the receiver is unable to relay accurate information to the sender. In the worst-case scenario, the receiver ends up sending feedback information to the sender continuously, which occupies the channel with feedback packets and prevents the sender from transmitting data. However, explicit feedback works well if the channel conditions do not change rapidly.

Indeed, each explicit and implicit approach has its own advantages and disadvantages. The effectiveness of our protocols shows that ACK packets are all that is needed for adapting rates robustly.

Chapter 3

Multi-Channel Adaptation

From the previous discussion on the state-of-the-art of multi-channel MAC protocols, we note that most existing approaches require tight temporal synchronization among nodes and when that requirement is relaxed, collision-freedom cannot be guaranteed. As a result, it is more desirable to provide a simple solution to achieving collision freedom without the need for temporal synchronization.

Designing a multi-channel MAC in wireless networks can be challenging because the hidden terminal problem now extends to multiple channels. The main challenge is to ensure that neighbors are aware of channel selection so that there will not be any conflicts in scheduling later. Because tight synchronization requirement may be hard to accomplish, I rely on an asynchronous split phase approach as it is simple and easy to implement on real systems. Each node has its own schedule, and it does not need to hold back their data transmission or to wait until the end of a fixed and predefined control period. Instead, it may transmit immediately once it satisfies the rules and the arrangement is made with the destination node. I will describe in details the sets of rules that are required for this to happen.

3.1 AM-MAC Protocol

The assumptions for the protocol is as follows:

- Each node is equipped with a single transceiver such that it can either transmit or receive,

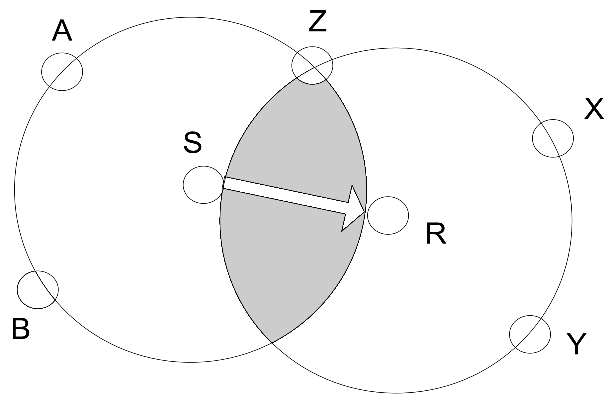


Figure 3.1: Sender S and receiver R are transmitting the ATS packets. The shaded region shows the anticipated noise

but not both simultaneously.

- A node's transceiver has N orthogonal channels of equal bandwidth; orthogonal channel means that simultaneous transmissions do not interfere with one another.

3.1.1 AM-MAC Handshake

For collision freedom (see Section 3.3), nodes must observe the control channel for a specified period of time. Any nodes interested in data transmissions must perform the handshake exchange accordingly.

The protocol borrows the basic CSMA handshake mechanism and carrier sensing for channel negotiation. However, the regular RTS/CTS handshake is insufficient to provide collision-free data transmission. For example, consider the scenario in Figure 3.5. R sends a CTS to S after having received a RTS request; however X, which is hidden from S, begins to send a RTS to Y at the same time. X has no knowledge of R's channel selection. Thus a collision in the data channel may occur.

In order to guarantee collision-freedom without the need for nodes to be synchronized, I introduce the ATS (for Announce To Send) control frame. ATS is used to inform the sender's and receiver's neighbors of the ongoing data transmission or channel selection and, at the same time, serves as a jamming signal to prevent possible interference and to prevent hidden terminal problems.

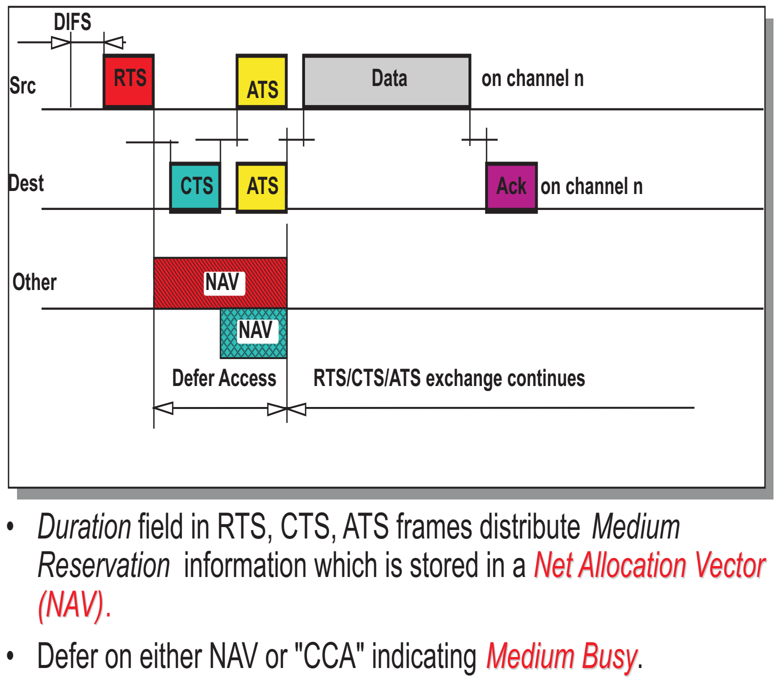


Figure 3.2: The RTS/CTS/ATS Based Access

Both sender and receiver need to broadcast their ATS packets which act like a busy-tone in order to broadcast channel selection information and prevent any potential interfering neighbors. This strategy is similar to the work done in [74] using a separate busy-tone channel. As for neighbors, they must adhere to back-off rules stipulated by the control frame exchange. This is illustrated by an example in Figure 3.1 which shows the ATS packet transmissions by both S and R concurrently. Some neighbors of S, such as A and B, can receive the ATS packet; the same applies to neighbors X and Y of R. However, the mutual neighbor Z of S and R (the shaded region) will only hear noise. Depending on the proximity of S and R's transmission, this overlapping region may shrink or expand. On the sender side, ATS informs neighbors who have not heard the CTS to aware of the channel selection. Simply, the sender announce its intention to transmit twice, once through an RTS and once through an ATS. On the receiver side, the ATS acts like an extended CTS to announce the channel selection and jam any potential interfering neighbors. In the following section, I will show that this unique handshake ensures collision-free data transmission with some additional conditions.

The AM-MAC's control frames are:

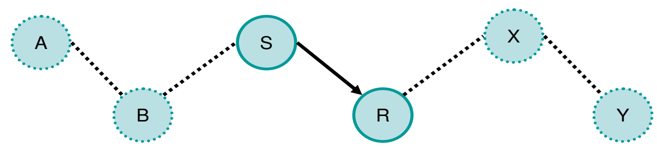


Figure 3.3: Sender S and receiver R together with potential interferers from the neighbors

- RTS (Request to Send): is used when a sender has data to send. I modified the RTS to contain some additional fields such as the available channel list, and data transfer time.
- CTS (Clear To Send): is used to acknowledge reception of the RTS. I introduce some additional fields, i.e., selected channel, and data transfer time.
- ATS (Announce To Send): is used to inform neighbors of the sender/receiver of the ongoing transmission and the data channel selection. It also acts as a jamming signal or a busy tone. Besides the standard RTS/CTS fields, some additional fields of ATS are: selected channel and data transfer time.

3.1.2 AM-MAC Collision free conditions

Figure 3.3 illustrates possible cases of hidden terminal involving the source-destination pair S and R. Node B represents any neighbor of S that is hidden from R while node A represents any neighbor of B hidden from S but can cause collision at B and prevents B from following the conversation between S and R. Similarly, node X is a neighbor of R but is hidden

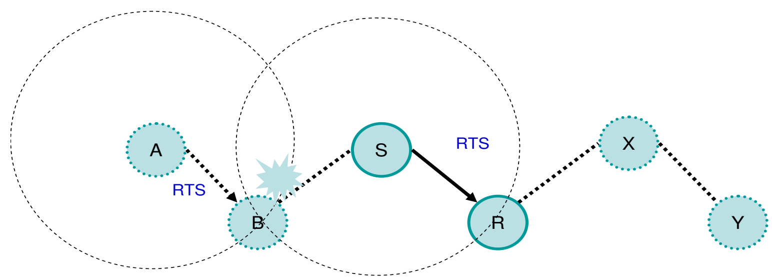


Figure 3.4: RTS collision: S sends a RTS packet to R while, at the same time, A sends a RTS packet to B

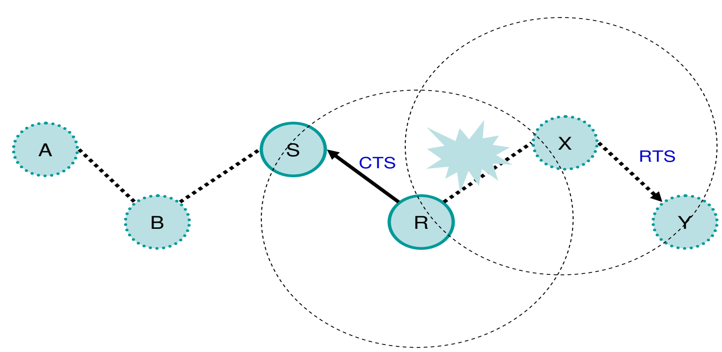


Figure 3.5: CTS collision: R sends a CTS packet to S while, at the same time, C sends a RTS packet to D

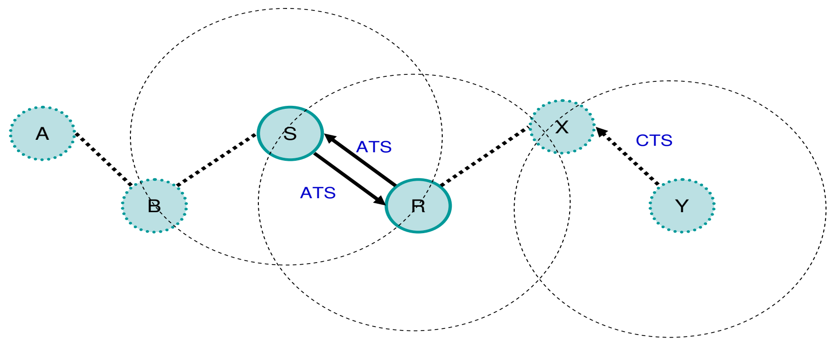


Figure 3.6: Collisions in which Y is hidden. Y switches to the data channel and, simply, times out and returns to the control channel

from S and may cause collision at R. Node Y represents any neighbor of X that is hidden from R and can prevent X from following the conversation between S and R

Let us also define the following notations:

- The maximum end-to-end propagation time in the channel is $\tau < \infty$.
- The transmission times of the RTS, CTS, and ATS are $\gamma, \gamma^{'}, \gamma^{''}$, respectively; the transmission time of a data packet is δ; the channel switching delay is ε, and $\gamma, \gamma^{'}, \gamma^{''} < \delta < \infty$

Theorem 1. *AM-MAC provides correct data channel acquisition in the presence of hidden terminals, provided that $\gamma > \tau$ and $\gamma + 2\tau + \varepsilon < \gamma^{'}, \gamma^{''} < \infty$*

Proof. Consider the illustration in Figure 3.3. For S to send data to R, S must receive a CTS from R confirming the request. Without loss of generality, assume that at time t_0 S sends an RTS to R. Because the channel has minimum propagation delay, any neighbor of S must begin receiving the RTS at time $t_0^B > t_0$. If the RTS arrives at B with no errors, B must back off for a period larger than $2\tau + \gamma^{'} + \gamma''$ after receiving the RTS, or for a total time of $3\tau + \gamma + \gamma^{'} + \gamma^{''}$

after t_0. If the RTS arrives at B in error (e.g., because of possible interference from B), B must also back off for a period larger than $2\tau + \gamma^{'} + \gamma^{''}$ after receiving the RTS. It follows that the RTS sent by S at time t_0 forces any neighbor of S other than R to back off until time $t_1 > t_0 + \gamma + \gamma^{'} + \gamma^{''} + 3\tau$.

If R receives the RTS from S in error, R will simply ignore or drop it. Assume that the RTS is received correctly at time t_2. R begins to reply with a CTS to S at time $t_2 \leq t_0 + \gamma + \tau$. Within one propagation delay, S receives the CTS from R at time $t_3 \leq t_2 + \gamma^{'} + \tau = t_0 + \gamma + \gamma^{'} + 2\tau$.

As for the receiver side, any neighbor of R must begin receiving CTS at time $t_2^X \leq t_2 + \gamma^{'} + \tau$. If CTS from R arrives at X with no errors, X then must back off for a period of time greater than $\tau + \gamma^{''}$ after t_2^X. If R's CTS arrives at X in error, X must also back off for a period greater than $\tau + \gamma^{''}$ after t_2^X. It follows that CTS sent by R at time t_2 forces X and any neighbor of R other than S to back off until $t_4 > t_2^X + \tau + \gamma^{''} = t_0 + \gamma + \gamma^{'} + \gamma^{''} + 3\tau$.

S begins its ATS broadcast at time $t_3 \leq t_2 + \gamma^{'} + \tau \leq t_0 + \gamma + \gamma^{'} + 2\tau$. Similarly, R begins its ATS broadcast at time $t_2^X \leq t_2 + \gamma^{'} + \tau \leq t_0 + \gamma + \gamma^{'} + 2\tau$. Because $t_1 > t_3$ and $t_4 > t_2^X$, any potential interfering neighbors of S must back off long enough to allow the transmission of ATS packets. Otherwise, ATS transmissions will jam any potential interfering neighbors and let them know of the channel selection since ATS packet is longer than RTS packet.

Neighbors should hear at least a CTS packet or a ATS packet. This is important because ATS and CTS carry the data channel selection information. Thus, any neighbor in the vicinity of S and R must have heard either CTS or ATS because they are both longer than RTS. □

Having complete knowledge of channel selection is crucial to provide a collision free protocol. Nodes which just completed data transmission will often have no knowledge of the channel status. Thus, nodes must observe for the maximum data transmission time before initiating any request for transmission.

Let T_0 be the observation period before a node initiates a request in the common channel and T_{MAX} be the maximum data access allowed.

Theorem 2. *Assuming a single collision domain network. If $T_0 \geq T_{MAX}$,*

Then AM-MAC is collision free.

Proof. Consider a node A just coming back from the data channel, and a node B switching to the data channel simultaneously. Obviously, A and B are not aware of each others' activities. If node A has observed the common channel for T_0 observation period, node A will have complete knowledge of the channel selected by B since B's data access is upper bounded by T_0. If there is a collision, two things can happen. Either node A did not observe for a sufficient time T_0 or node B held onto the data channel for period of $T_B > T_0$. This contradict our assumptions that $T_B \leq T_{MAX} \leq T_0$. □

3.1.3 AM-MAC Operation

Having described all the basic mechanisms that form the core of AM-MAC, its complete operation is presented here. Suppose A has data destined for B and A gets access to the common channel, the procedures used by A and B is described as follows:

1. A transmits RTS to B, assuming A had already listened in the common channel for the required observation time T_0.
2. B replies by sending a CTS to A. B starts a timer for the CTS so that upon its expiration, it sends the ATS.
3. On receiving the CTS, A sends ATS packet .
4. After the concurrent transmission of the ATS packets, nodes switch to the negotiated channel.
5. A begins sending data to B and B acknowledges each data packet it receives from A.
6. After data transmission has been completed, A and B will go back to observing the channel for a full waiting period T_0 before transmitting or responding t o a request.

As for any neighbors of A or B, they will observe the control channel and update their channel usage list accordingly.

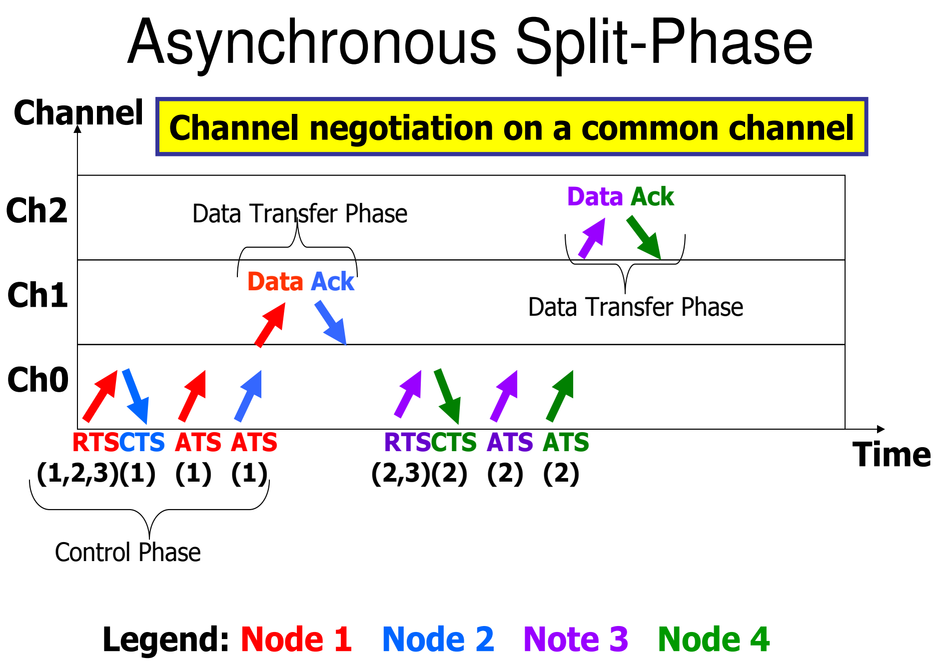


Figure 3.7: AM-MAC Split Phase Illustration

3.2 Performance Evaluation

I evaluate AM-MAC's performance through extensive simulations using the ns-2 [77] network simulator with CMU's wireless extension [69]. I evaluate AM-MAC against IEEE 802.11 and MMAC. As performance metrics, I use aggregate throughput and average packet delay.

Throughput is calculated as the number of successful packets received multiply by the packet length and divided by the total simulation time. Packet delay is calculated as the difference between the time a packet arrives at the queue and the time a packet gets transmitted successfully.

For the simulations, I used two network scenarios, namely a fully connected mesh and multi-hop topologies. In the fully connected mesh scenario, every node is within range of one another and is able to reach any destination in one hop. In the multi-hop scenarios, a packet may travel several hops before it reaches the destination.

3.2.1 Simulation Setup

The simulation setup is based on MMAC's setup model because I consider the protocol a variant of MMAC protocol. The difference is that each node follows its own observation phase instead having a single control phase for all nodes. In the simulations, the bit rate for each channel is 3Mbps and the transmission range is set to 250m. Each source generates and transmits constant-bit rate traffic. I use 3 or 4 available channels depending on the scenarios, packet size of 512 bytes, drop-tail queues with maximum queue length of 50 packets, omni-directional antenna, and TwoRayGround propagation model. Each data point in the graph is an average of 10 different simulation runs.

In the fully connected mesh scenario, every node is within each other's range. I simulate two scenarios, one with 36 nodes and 18 concurrent flows and another with 64 nodes and 32 concurrent flows, in a 400x400m area. Nodes move according to the random way-point model with speeds varying between 0 and 10m/s and with no pause. In each scenario, approximately half of the nodes are sources and half are destinations. All nodes begin transmission at the same time.

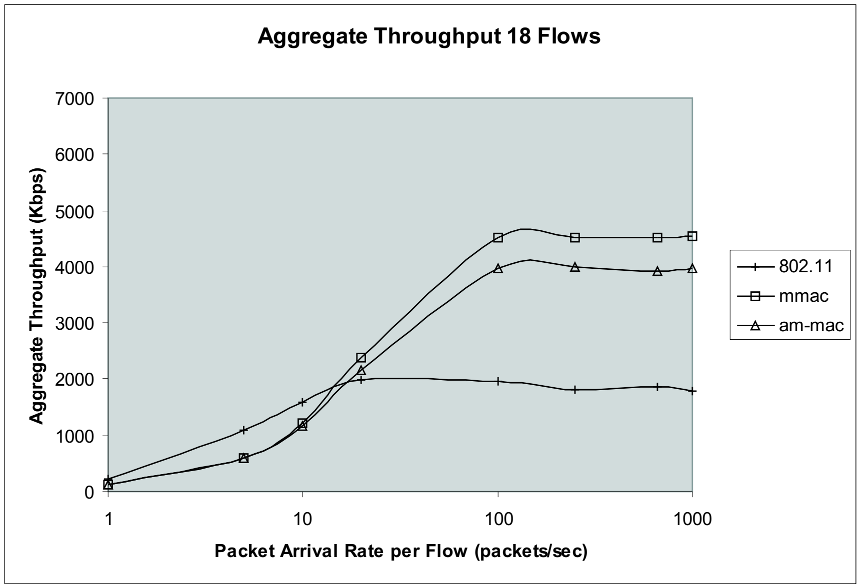
In the multi-hop network scenario, 121 nodes are placed randomly in a 1000x1000m area. Sources and destinations are randomly selected such that a node may be the source for multiple destinations and a node maybe a destination for multiple sources. At any given point in time, 42 concurrent flows are active. Nodes move according to the random way-point model with speeds varying between 0 and 10m/s with no pause time. For these simulations, I use 3 and 4 available channels.

3.2.2 Simulation Results

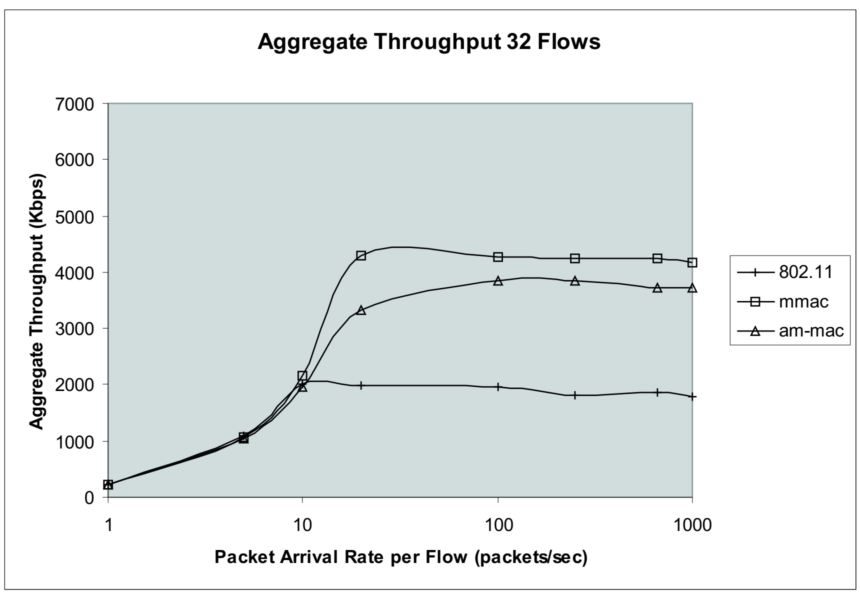
Figure 3.8(a) shows the aggregate throughput for the fully connected mesh scenarios. When the network load is low, all protocols have similar performance. When the network approaches saturation, AM-MAC performs significantly better than 802.11 and yields a slightly lower performance compared to MMAC. Figure 3.8(b) shows the result for the same mesh scenario but with great node density(64 nodes) and more flows(32 concurrent flows). The gain in performance over IEEE 802.11 is mainly due to multi-channel transmission versus a single channel transmission. I observe that, in the fully-connected mesh environments, AM-MAC performs slightly lower than MMAC. This is because when a pair of nodes is negotiating for a channel, all nodes in the networks must delay their requests for data transmissions anyway. As a result, perfect synchronization seems to give MMAC a slight advantage in this environment. Also, because of full connectivity, ATS packets are not needed and may introduce an additional overhead.

Figure 3.10(a) and Figure 3.10(b) show the simulation results for the multi-hop scenario. With 3 available channels, I vary the constant-bit rate traffic from low to very high. For low traffic loads, all protocols have similar performance. However, under moderate to high traffic load conditions, AM-MAC performs better than 802.11 and slightly better than MMAC in terms of aggregate throughput. I repeat the same experiment with 4 available channels, and notice a similar performance trend.

Figure 3.9(a) and Figure 3.9(b) report the average delay results for fully connected networks. Figure 3.11(a) and Figure 3.11(b) show the average packet delay in the multi-hop environment. When the number of contending nodes is moderate, AM-MAC has significantly better average packet delay than 802.11 and has a comparable performance to MMAC. The

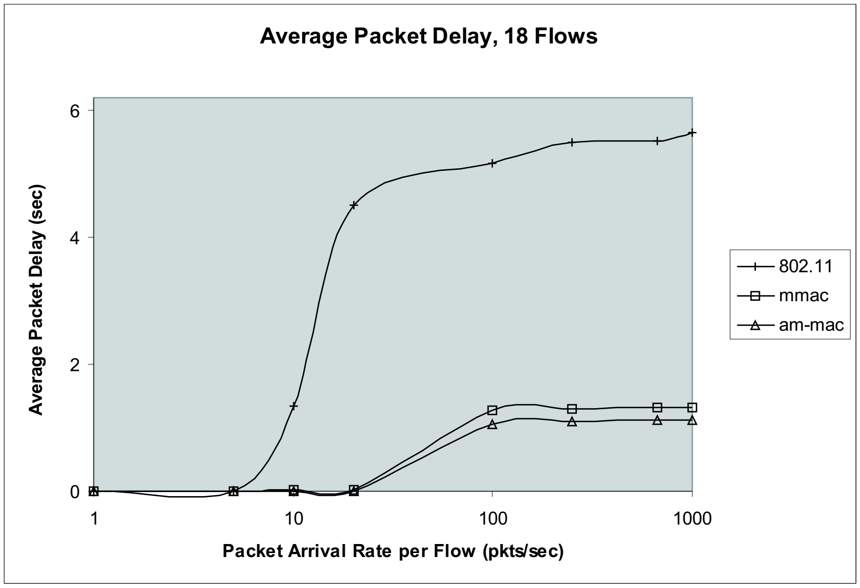


(a) 18 Flows

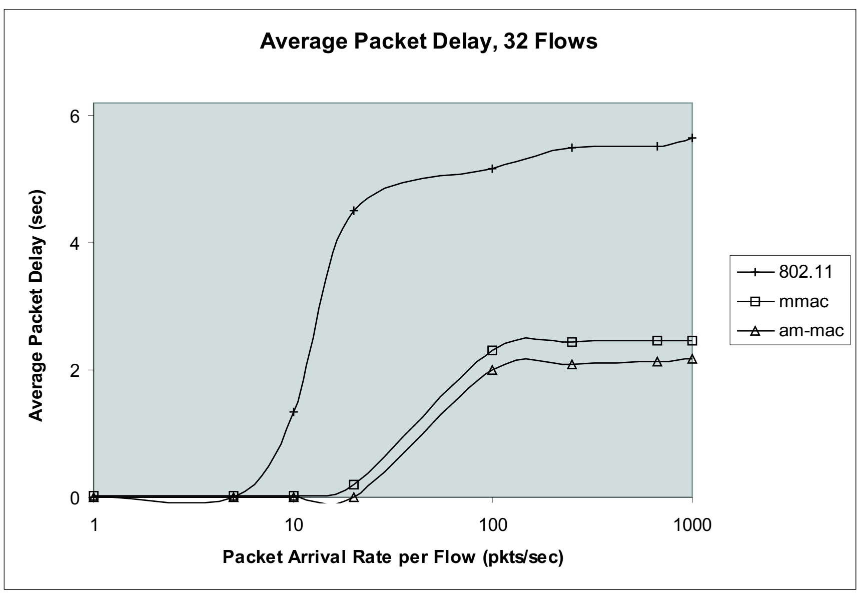


(b) 32 Flows

Figure 3.8: Connected Mesh: Aggregate Throughput vs Packet Arrival Rate

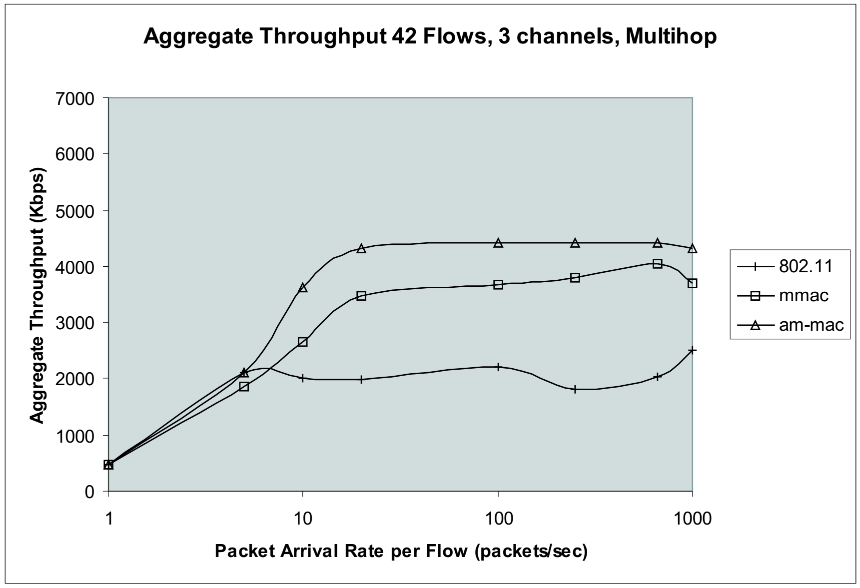


(a) 18 Flows

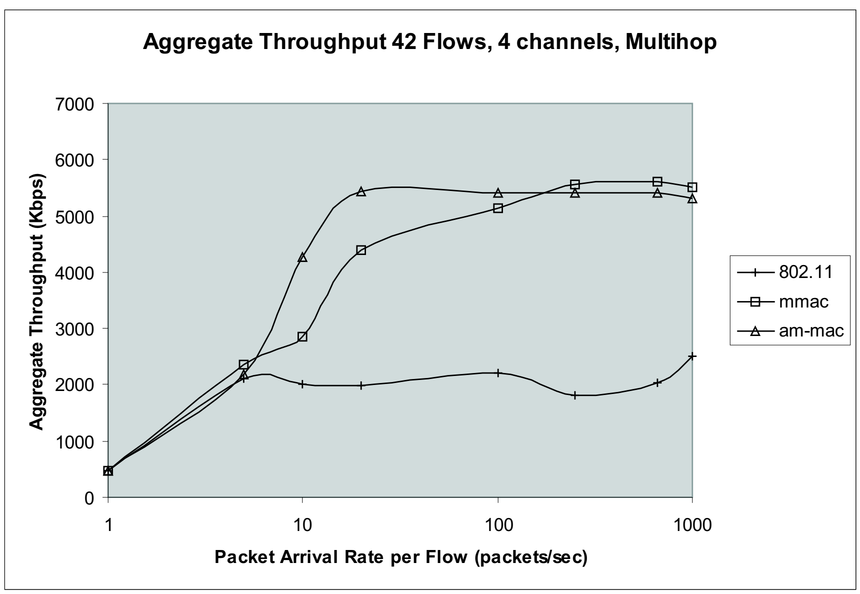


(b) 32 Flows

Figure 3.9: Connected Mesh: Average Packet Delay vs Packet Arrival Rate

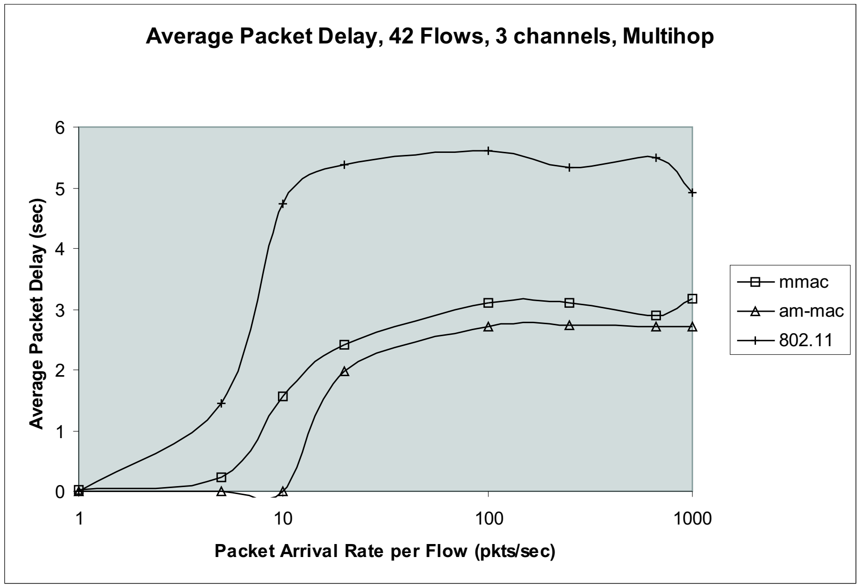


(a) 3 Channels

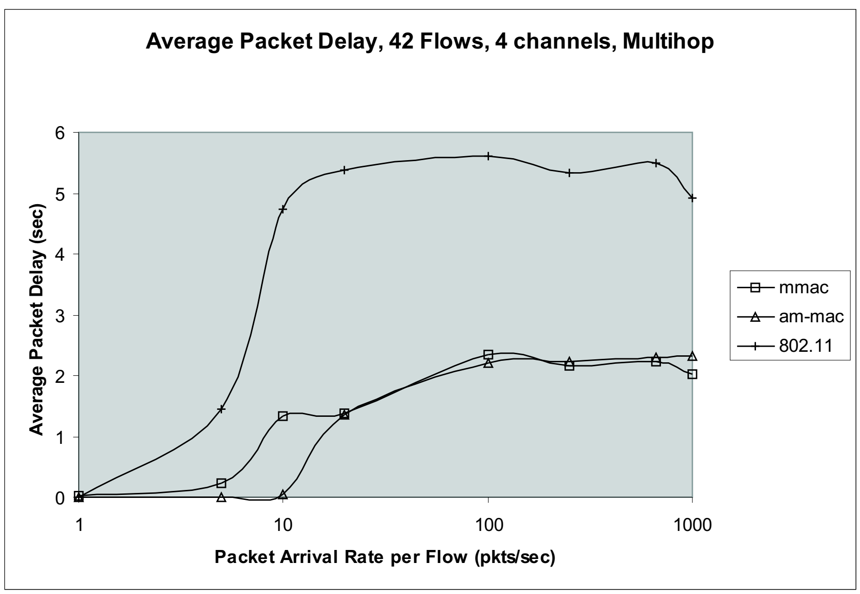


(b) 4 Channels

Figure 3.10: Multi-hop Ad hoc: Aggregate Throughput vs Packet Arrival Rate



(a) 3 Channels



(b) 4 Channels

Figure 3.11: Multi-hop Ad hoc: Average Packet Delay vs Packet Arrival Rate

same result applies when I increase the number of contending nodes, flows, and packet arrival rate.

In the fully connected mesh scenario, AM-MAC achieves a slightly slower throughput than MMAC's but gains an advantage in terms of average packet delay. I conjecture that the perfect synchronization gives MMAC an advantage in throughput but suffers in average packet delay because of the control-phase wait time. As a result, on average, AM-MAC has a slightly better average packet delay than MMAC. However, in the multi-hop scenario, our protocol AM-MAC achieves better or comparable performance in both throughput and average packet delay.

3.3 Summary

I presented AM-MAC a novel solution to multi-channel medium access for single-transceiver nodes. AM-MAC employs a simple, yet efficient approach to collision-free data transmission over multiple channels without the need of temporal synchronization among nodes. It uses a unique three-way handshake to ensure collision freedom and broadcasting channel selections. The simulation results show that AM-MAC significantly improves performance when compared against IEEE 802.11, and yields comparable performance to MMAC [65], which requires tight synchronization. The insight is that we can take advantage of multiple channels with minimal overhead even with single transceiver.

Chapter 4

Multi-Rate Adaptation: Packet Error Approach

Previous chapter deals with medium access control. In this chapter, I will address the challenge of making the best use of channel access time through rate or link adaptation. Rate adaptation plays an important role in attaining bandwidth for both the sender and receiver. I begin by proposing new approach to rate adaptation based solely on packets errors. The simplicity of the implicit approach is employed because of the challenges in querying the receiver for channel conditions.

4.1 Multi-Rate Adaptation with Interference Awareness (MAICA)

The role of a rate adaptation mechanism is to select the proper rate for transmission. Given that the available rates are constrained to the deterministic values of the transmission rate vector of Table I, this requires selecting the index corresponding to the adequate rate value.

4.1.1 Rate Adaptation in Multiple Access Interference

Multiple Access Interference (MAI) and natural phenomena associated to radio wave propagation are the key reasons for throughput reduction in wireless networks. Adapting to them is complicated by the unpredictability of interference. A network may be subject to little or a lot of interference, depending on the characteristics of the environment, the network density, and node movement, and environmental mobility. A major concern with MAI is that it increases very rapidly with node density and impacts the network layer, which causes MAI to spread over multiple hops. As the network becomes congested, nodes adapt at the network layer by trying to find routes around the congestion points. This draws even more intermediate nodes into carrying data. The more nodes that are involved in data transmission, the more interference they generate.

Let $x_i(t)$ be the rate index of the ith user during time slot t, and let $y_i(t)$ be the feedback derived from the success or failure of the transmission. $y_i(t)$ can take four values: decrease multiplicatively (M_D); decrease incrementally (A_D); increase incrementally (A_I); do nothing (N). The ith user's in the system may increase or decrease its demand by a function $f(x_i(t), y_i(t))$ of the previous demand, and system feedback, such that:

$$x_i(t+1) = \begin{cases} x_i(t) & \text{if } y_i(t) = O; \\ x_i(t) + 1, & \text{if } y_i(t) = A_I; \\ x_i(t) - 1 & \text{if } y_i(t) = A_D; \\ x_i(t) \times M_D & \text{if } y_i(t) = M_D; \end{cases}$$

The notation M_D to denote both the feedback decision, and the coefficient in $(0, 1)$ by which we multiply $x_i(t)$ to effect this decision.

4.1.2 MAICA Rate Adaptation

MAICA rate adaptation makes its rate control decisions by keeping track of the number of successes over a *rate adaptation window* w corresponding to the number of packets transmitted.

The first step of the MAICA algorithm is to transmit the number of packets specified

Rate Index	IEEE 802.11a Data Rates (Mbps)	IEEE 802.11b Data Rates (Mbps)	IEEE 802.11g Data Rates (Mbps)
0	6	1	1
1	9	2	2
2	12	5.5	6
3	18	11	9
4	24	n/a	12
5	36	n/a	18
6	48	n/a	24
7	54	n/a	36
8	n/a	n/a	48
9	n/a	n/a	54

Table 4.1: Rate Index and Data Rates Conversion Table [30]

in our window of at least w distinct packets (our recommended value is 10). w only counts new packets, and not retransmission. I also consider a time window ω and make a decision when either one of the transmission or time window concludes first. I use $\omega = 100$ millisecond as an implementation guideline. The rational for using a time window is to ensure proper reactivity in case the sender does not transmit w packets within ω seconds.

During the transmission window, MAICA increments three counters, one each for three transmission cases: σ for packet success; ϵ for packet error; and ρ for packet retransmission.

The second step of MAICA happens upon conclusion of either the w-window or the ω-window. The algorithm checks the performance of each window w packets against some pre-defined thresholds.

MAICA defines two thresholds: τ_ϵ and τ_γ; and one credit counter γ. There are a number of scenarios depending upon the number of success, errors and retransmissions.

- If the number of packet errors $\epsilon \leq \tau_\epsilon$, then the transmission rate is of good quality, and a credit is added to the credit counter γ. Once γ reaches the threshold τ_γ, then the rate is increased: $x_i(t+1) = x_i(t) + 1$ and then γ is reset to 0. The credit counter allows us to increase more progressively (i.e. more conservatively) to avoid erratic rate variations.

- If the number of packet errors $\epsilon > \tau_\epsilon$, then the transmission rate is of poor quality, and

Algorithm 1 MAICA Algorithm for Up-shifting and Down-shifting Rates

```
τ_γ = credit threshold for promoting to the next rate
w = sampling of packets window
τ_ε = success packets times acceptable error rates
rateIndex = the rate index as shown in Table I.
packetsCount = 0
credit γ = 0; retransmitPackets ρ = 0;
successPackets σ = 0; errorPackets ε = 0
//comment: in addition to sample window w, time window ω is required
while (packetsCount < w) do
  if (packetIsSuccess) then
    σ++
  else if (packetIsError) then
    ε++
  end if
  if (packetIsRetried) then
    ρ++
  end if
  packetsCount = σ + ε
end while
packetsCount = 0
//comment: success packets with many retries
if (σ < ρ) then
      rateIndex−−
      γ = 0
end if
//comment: within acceptable error threshold
if (ε ≤ τ_ε) then
      γ++
end if
//comment: downgrade rate
if (ε > τ_ε) then
      rateIndex−−
      γ = 0
  //comment: multiplicative downgrade
  if (ε > σ) then
        rateIndex ← rateIndex * M_D
        γ = 0
  end if
end if
//comment: ensure stability before upgrading
if (γ ≥ τ_γ) then
      rateIndex++
      γ = 0; ρ = 0
      σ = 0; ε = 0
end if
ρ = 0; σ = 0; ε = 0
```

the rate is decreased; the credit counter γ is reset to 0. $x_i(t+1) = x_i(t) - 1$. Note that I decrease right away rather than subtracting a credit, as it is more conservative in my rate utilization.

- If the number of packet errors ϵ is greater than the number of success τ, then the transmission rate is of very poor quality, and the rate is decreased multiplicatively: $x_i(t+1) = M_D x_i(t)$, and γ is reset to 0. This case preempts the above case.

- Finally, if the number of success $\sigma < \rho$ (the number of re-transmissions), then the channel retransmits too much at the current rate and the rate is decreased: $x_i(t+1) = x_i(t) - 1$, and γ is reset to 0.

At the end of the window, σ, ϵ and ρ are re-set to 0.

The exact method to increase and decrease rates is described in Algorithm 1.

There are a number of parameters in the above algorithm, such as the credit threshold, the multiplicative decrease component, and the acceptable error rate. From testbed experiments and simulations, there are large fluctuations in throughput if the threshold is set below 10. On the other hand, the throughput remains stagnant if the threshold is set higher than 10. As for the acceptable error rate, the throughput reduces if the acceptable error rate is set to 0. The same applies to allowing a higher acceptable error rates. For the multiplicative decrease component, I find a big gain by employing it, especially when subject to heavily loaded interference. Figure 4.1 show the simulation results for 100 nodes in a multi-hop networks with 45 exponentially distributed flows. It confirms the advantage of selecting the index using $B_D = 3/4$ or $B_D = 1/2$ over selecting consecutive indices ($B_D = 0$). Note that all the rates are adapted in terms of their rate indices as shown in Table I. For example, transmitting at rate index 9 means that we are transmitting at 54Mbps. Additive decrease rate index 9 to 8 translates to transmitting at 48 Mbps. Multiplicative decrease rate index 8 by $B_D = 3/4$ equals rate index 6 translates to transmitting at 24 Mbps. Due to the mapping and rate index conversions, $B_D = 3/4$ usually multiplicatively decreases data rates by half (i.e. from 48Mbps to 24Mbps). Note, the multiplicative component does carry the same meaning as Additive Increase and Multiplicative Decrease (AIMD) scheme. Its main purpose is to decrease to several steps of indices instead of linearly. This approach shortens the probing time for rate adaptation. This experiment confirms the need for downgrading and upgrading the rate as quickly as possible in order to get to the optimal rate. The maximum rate index depends on the corresponding physical layers(PHY). Due to limited space, I only present the results of the multiplicative decrease component of MAIA (See Figure 4.1).

4.2 Multi-rate Retry

According to IEEE 802.11 standard [30], a data packet is allowed up to seven retries before being dropped altogether. A special retry chain in Atheros chipset [4] is used to combat the delay of the next data packet. Our modified multi-rate retry is as follows:

Algorithm 2 MAIA Multi-rate Retry Pseudo-Code

if $--RateIndex \geq 0$ **then**
  $firstRetryRate \leftarrow RateIndex$
else
  $firstRetryRate \leftarrow LowestRate$
end if
$RateIndex \leftarrow \lfloor RateIndex - 1 \rfloor$
if $RateIndex \geq 0$ **then**
  $secondRetryRate \leftarrow RateIndex$
else
  $secondRetryRate \leftarrow LowestRate$
end if
$thirdRetryRate \leftarrow LowestRate$

Given a bit rate, Algorithm 2 returns three other retries rate that can be used for feeding to the hardware multi-rate retry chain. Each retry rate will have 2 retries. For example, if transmissions at the current rate index fail after two retries, the rate index is decremented. If this decremented rate index fails again after two retries, the rate index is additively decreased. Finally, if this rate does not work again, the lowest rate index is used. The rate index and data rates conversion is shown in Table I. Note that these retry rates are only used for retrying attempts.

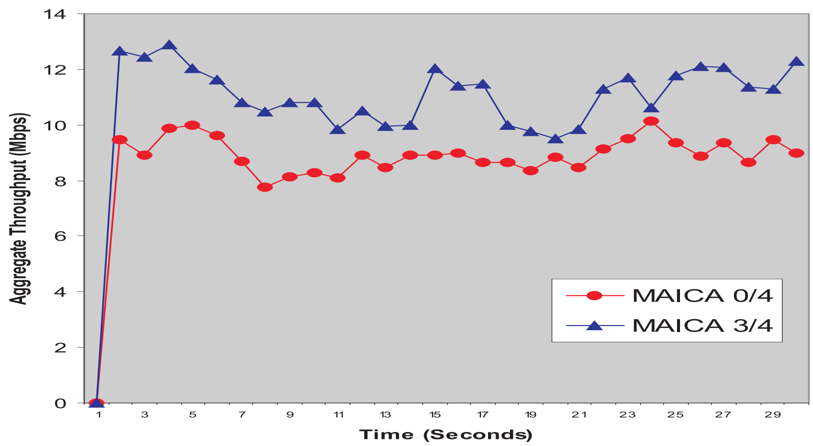


Figure 4.1: The advantage of multiplicative component in congested networks

4.3 Performance Evaluation

I evaluate MAICA's performance against many well-known multi-rate adaptation schemes through extensive network simulations, analytical analysis, and real-world and testbed experiments with an actual implementation.

4.3.1 Analytical Model

4.3.1.1 Markov Chain Model

A general analytical model for multi-rate Auto Rate Fallback (ARF) [41] has been analyzed and studied by Choi [8] and Singh [63]. I use these models as the baseline comparison due to their simplicity and because they represent the behavior of most rate-adaptation algorithms.

If we assume that each trial in w window samples is independent identically distributed, then MAICA follows a Binomial distribution. Figure 4.2 illustrates the main operation of MAICA. There are 8 Markov states for eight possible different corresponding rates of 6, 9, 12, 18, 24, 36, 48, and 54 Mbps to represent IEEE 802.11a. Analysis for IEEE 802.11g is similar and omitted here due to space constraints.

This Markov chain follows the birth-death process with self-loops at the initial and ending state. In addition, each state can transition to a previous state or multiplicatively to its previous state which is approximately half of its rate. For example, state with 12 Mbps has a transition to state 6 Mbps. We made an exception for state 54 Mbps so that it fits our model. Also, each state has only one transition to the next higher rate since our protocol does not have multiplicative increase.

Following the Bianchi [6] model, we assume that each station transmits a frame with probability π. Given that there are n stations contending for the channel, we define probability of transmission P_t as

$$P_t = 1 - (1 - \pi)^{n-1} \tag{4.1}$$

With m number of IEEE 802.11 rates, we have

$$r_1 < r_2 < ... < r_m$$

subject to the following frame error conditions

$$e_1 \leq e_2 \leq ... \leq e_m$$

The probability P_s that a transmission is successful [6] is

$$P_s = \frac{n\pi(1 - \pi)^{n-1}}{P_t}$$

Thus, the conditional frame success probability at rate r_i is simply

$$p_i = P_s * (1 - e_i) \tag{4.2}$$

With the Markov chain being irreducible and aperiodic for $p_i > 0$, we are interested in finding the stationary probability for our system, Π_i for 1 ≤ i ≤ N such that $\Pi_i = \Pi_i P$. Finally, we assume full network connectivity.

4.3.1.2 Solution of MAICA Markov Chain

MAICA makes its rate control decisions by keeping track of the number of successes in w trials. With our assumption that each trial is independent identically distributed, this operation follows a binomial distribution with parameters w and p_i, or $X_k \sim B(w, p_i)$. Its probability mass function is given by:

$$P(K = k) = \binom{w}{k} {p_i}^k (1 - p_i)^{w-k} \tag{4.3}$$

$$\text{for } k = 0, 1, 2, ..., w$$

Let $P_{i,j}$ = the transition probability from state i to j. From the description of Algorithm 1, it follows that

$$P_{i,j} = \begin{cases} \prod_{k=1}^{\tau_\gamma} \sum_{c=w-\tau_\epsilon}^{w} P_k(K = c), & \text{if } j = i + 1; \\ 1 - \sum_{c=\frac{w}{2}}^{w} P(K = c), & \text{if } i > j;\ i - j = 2; \\ \sum_{c=1}^{\tau_\epsilon} P(K = c), & \text{if } i > j;\ i - j = 1; \\ 0, & \text{otherwise}; \end{cases} \tag{4.4}$$

P_k denotes k number of $P(K = c)$ required for reaching the credit threshold τ_γ. Using global balancing equations at each state together with given condition $\sum_{i=0}^{k} \Pi_i = 1$, we easily obtain numeric solution for the stationary probability at each state.

I evaluate our model with NS-3 simulations. Frame error rates (e_i) are imported from the simulations and fed into our model. I use the payload size of 256 bytes since it places extra load on rate adaptation to delivery packets. Each station's frame transmit probability τ is set to the value observed in the simulations. The analytical result(Figure 4.3) shows that MAICA has better performance throughput than ARF even as the number of nodes increases. The large gain is obtained when the number of nodes is small. As I increase the number of nodes this gain becomes smaller because of the congestion and interference, at which point, the medium is not accessible for any rate adaptation to work. Overall, the analytical model provides a good match with the results obtained with our simulations.

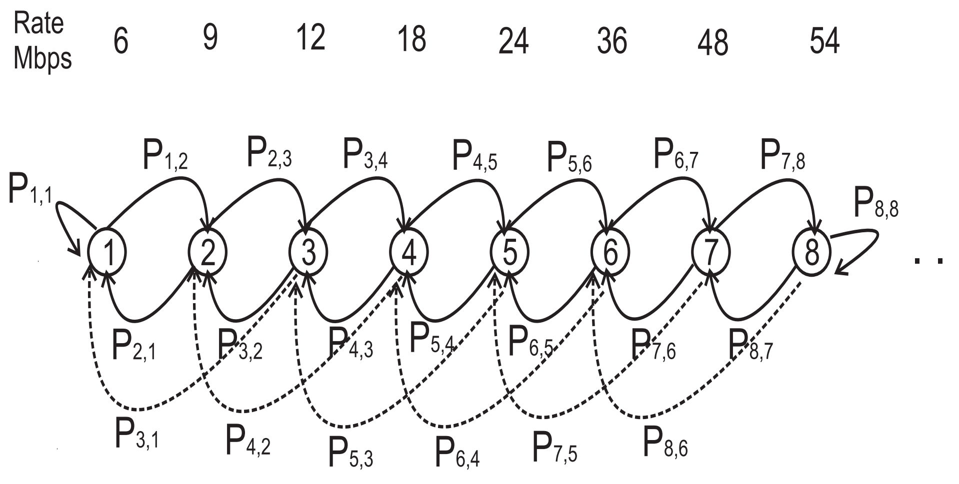


Figure 4.2: MAICA operation

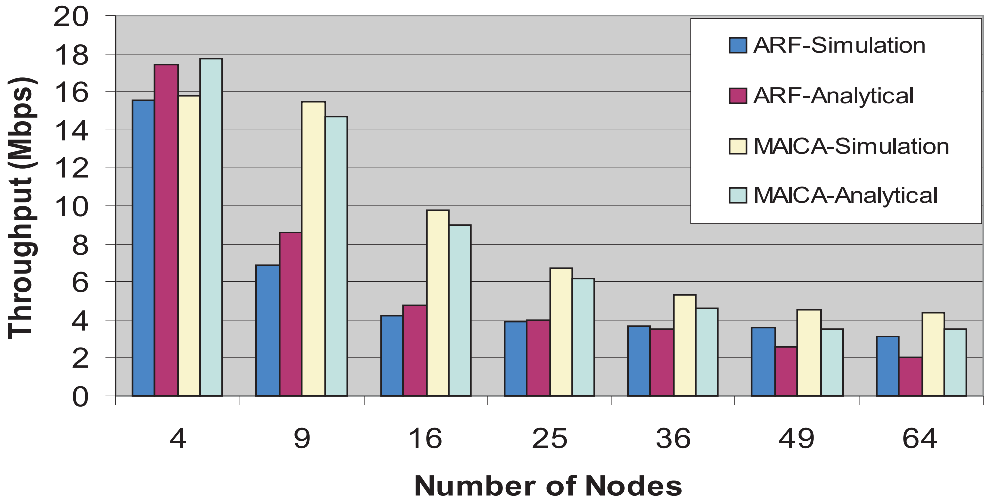


Figure 4.3: Analytical Results for ARF and MAICA

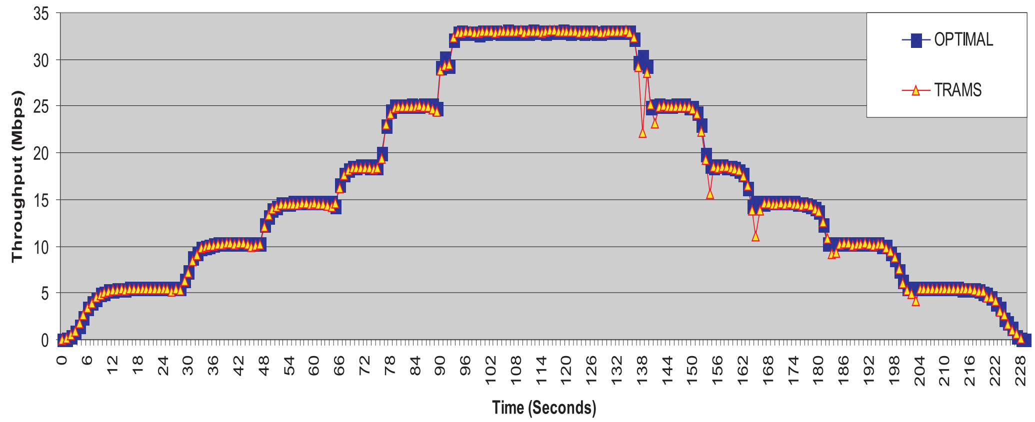


Figure 4.4: Two-Node Fading

4.3.2 Network Simulation Setup

I evaluate and simulate several relevant rate adaptation schemes using NS-3 [72]. Unless otherwise specified, I assume a packet size of 512 bytes, a drop tail queue with a maximum length of 100, the IEEE 802.11a MAC model, a constant speed propagation model, a log distance propagation loss model ($L = L_0 + 10n \log_{10} \frac{d}{d_0}$), a transmission range of 140m, and UDP throughput. Each simulation was performed for a duration of 60 seconds. 60 seconds and longer durations produced similar results in benchmarking runs.

First, I study the case of two nodes' movement during data transmission. The source node moves at a speed of 1 m/s away from the target with no pause. The objective is to see how decreasing signal strength and fading affect performance.

Second, I set up 100 nodes in a 10x10 grid topology with a default distance between nodes of 20m. I select nine sources regularly placed in the 10x10 grid topology and assign them 25 target nodes with the flows being exponentially distributed with mean of 3 seconds, for a total of 225 (25x9) distinct flows. Then I vary the propagation loss models, PHY layers, number of nodes, packet size, and distance between the nodes in the grid topology. I choose the grid topology and set it up this way because I do not want to employ any specific routing protocol, which might influence the results of rate adaptations. Flows are exponentially distributed to ensure that this scenario does not favor any approach.

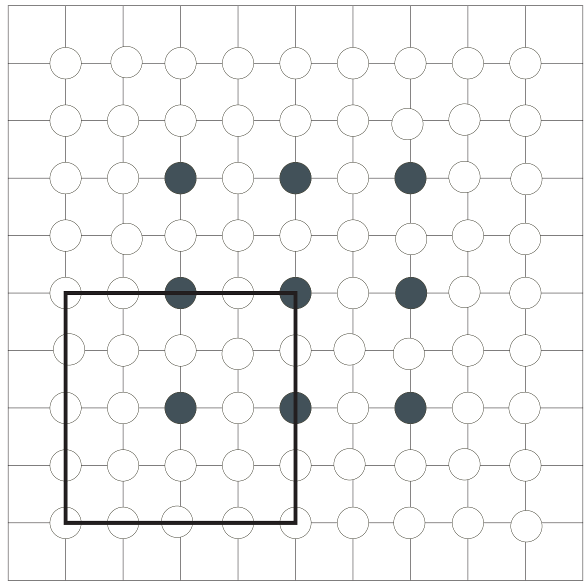


Figure 4.5: Each source sends to 25 target nodes in exponentially distributed time

Third, I include the Jain's fairness index and average aggregate throughput per node for the 100-node and 16-node grid topology with 50 flows and 8 flows respectively. Finally, I set up 50 flows for 100 nodes placed randomly in a 500m x 500m square area with global routing knowledge. I also experiment with a mobility scenario for 30 flows in 500m x 500m square area such that a node may be the source for multiple destinations and a node may be the destination for multiple sources. This experiment is performed with global routing knowledge and random 2D mobility with speed of 1 m/s and 0.2s pause.

4.3.3 Network Simulation Results

Figure 4.4 reports the results for the two-node movement scenario. The throughput results are at the lowest when nodes are farthest apart at 140m and greatest when they are in close proximity. ONOE, being conservative in raising rates, takes some time before transmitting at the optimal rate. AMRR has many sporadic dips throughout the experiment, this is probably due to the exponential backoff mechanism. RRAA does not perform well in this fading scenario because it lowers its rate quickly due to employing short-term loss. Minstrel performs well but it takes dips during transition such as 30m, 50m, 90m, and 115m due to its probing and trials and

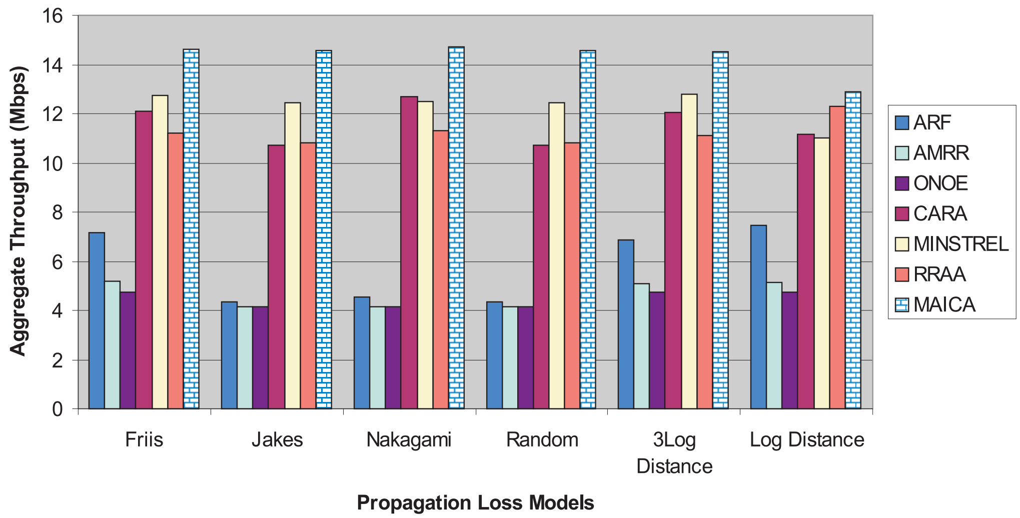


Figure 4.6: Throughput vs. Propagation Loss Models

errors nature before achieving the optimal rate. CARA's control packets probing for collision detection suffers a slight performance decrease. The fading pattern works well for ARF due to gradual increasing and decreasing signal strength; however it still does not perform well as MAICA. During the fading transition, MAICA lowers its rate accordingly to adapt to it.

Figure 4.5 presents a 10x10 grid scenario with exponentially distributed flows. For varying propagation loss models scenario, AMRR, ARF, and ONOE do not perform well due to conservativeness in raising rates. I find that RRAA follows MAICA's performance closely, though it fails in many propagation loss scenarios because short-term loss ratio does not work well for different types of propagation packet loss.

I continue by varying the physical layers such as IEEE 802.11b and a rate-adaptation friendly MAC protocol designed by Holland et al. [29]. For different packet sizes (Figure 4.8), I find that RRAA and MAICA have very similar performance and they both perform better than all the other schemes. Throughput increases with larger packet size but AMRR and ONOE continue to have the same performance. As the distance between nodes in the grid increases, all schemes begin to exhibit similar performance due to the minimizing effects of interference.

Figure 4.1 show the results for 100 nodes in a multi-hop networks with 45 exponentially

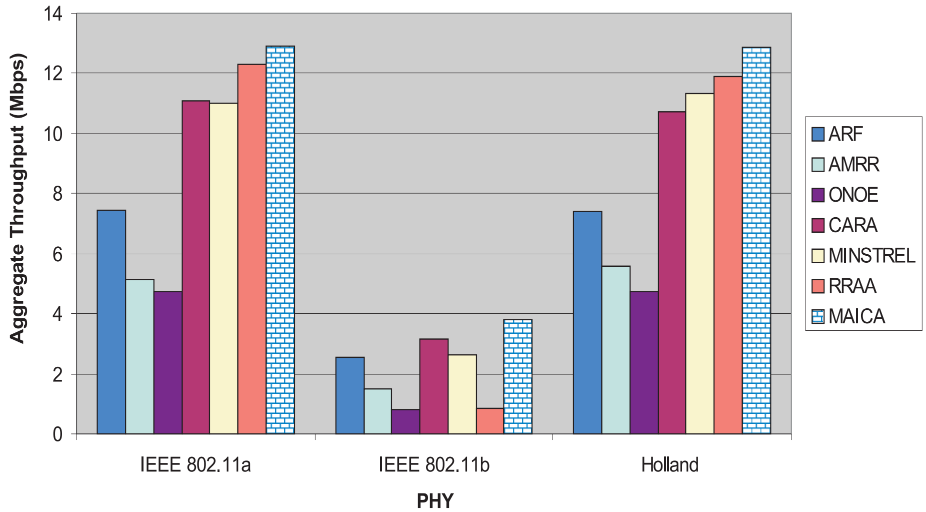


Figure 4.7: Throughput vs. PHY

distributed flows. It confirms the advantage of selecting the index using $B_D = 3/4$ over selecting consecutive indices ($B_D = 0$). Note that all the rates are adapted in terms of their rate indices as shown in Table I. For example, transmitting at rate index 9 means that we are transmitting at 54Mbps. Decrease of the rate index from 9 to 8 translates into transmitting at 48 Mbps. Decrease of the rate index from 8 by $B_D = 3/4$ means shifting to rate index 6. This translates into transmitting at 24 Mbps. Due to the mapping and rate index conversions, $B_D = 3/4$ usually multiplicatively decreases the data rates by a half (i.e. from 48Mbps to 24Mbps). The maximum rate index depends on the corresponding physical layers(PHY).

Figure 4.11 reports the Jain's fairness index and average aggregate throughput per node for sparse and dense networks. I show only a short period so I can highlight the results for average aggregate throughput gain per node. For parse networks, MAICA attains slightly higher fairness and outperforms all protocols in terms of average aggregate throughput per node. With 8Mb gain per node over the second highest, MAICA gains approximately 8% average network-wide throughput over CARA. AMRR and ONOE performs well in this scenario due to its implicit multiplicative component in their protocol due to binary exponential back-off and rapid rate downgrade respectively. For denser networks, I observe that most protocols suffer.

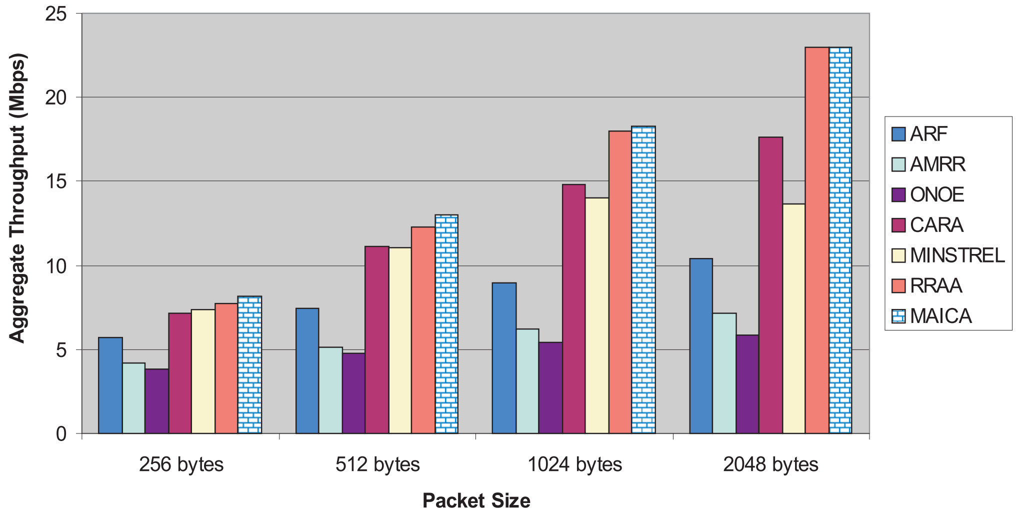


Figure 4.8: Throughput vs. Packet Size

MAICA still provides better fairness and average aggregate throughput per node. With 5Mb gain per node over the second highest, MAICA gains approximately 25% average network-wide throughput over CARA and RRAA. Note that, being fair does not translate to the best aggregate throughput per node.

Figure 4.10 reports the results when all 100 nodes are placed in a connected Wireless LAN, any node is reachable in one hop. In this scenario, MAIA performs significantly better than all the protocols, especially when subject to heavily loaded interference.

Figure 4.12 shows the results for the 50-flow scenario where nodes are randomly placed. ONOE, ARF, and AMRR do not perform well due to their being conservative in adapting rates. Figure 4.13 presents the mobility scenario where we set up 30 flows for all 100 nodes placed randomly in 500mx500m square area. A node may be the source for multiple destinations and a node may be the destination for multiple sources. MAICA gains an average of 1Mbps aggregate throughput over the next highest rate CARA.

Network simulations results show that MAICA works well in all scenarios, from fading, random loss models, different packet sizes and PHY layers to random network topology and mobility, especially in heavily loaded interference scenarios. In the following section, I evaluate

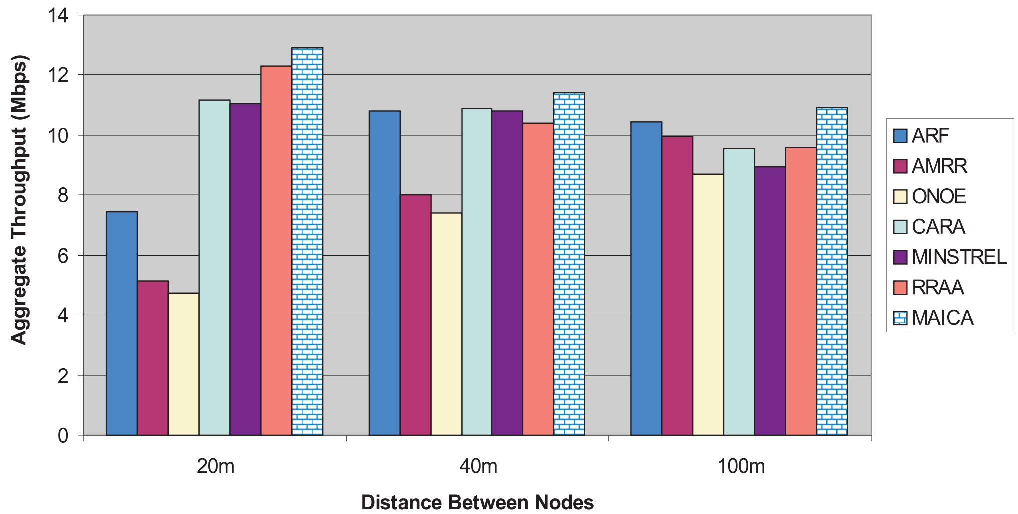


Figure 4.9: Throughput vs. Distance Between Grid Nodes

the performance of MAICA using testbed and real-world settings subject to various conditions that are hard to simulate realistically.

4.3.4 Experimental Setup

I use MadWifi [71] for the implementation of MAICA. I also implemented MAICA in the Linux Kernel Wireless Stack [70] to make sure that our design can work correctly and independently of the Atheros chipset. I only compare our approach against ONOE, SAMPLE, AMRR, and MINSTREL, because these are available and widely deployed in many real-world settings. Many schemes in our network simulations are either not publicly made available or only exist as simulation code.

Each node in our testbeds is a Mini-ITX Board AMD LX800 equipped with R52 802.11 a/b/g based on Atheros chipsets AR5212 and AR5213 [4] in addition to a few laptops running the same chipset. All of the testbeds nodes run Debian Linux [14] kernel version 2.6.33. I build a custom radio shield using layers of tin foil paper wrapped around 2x2 meters plastic sheet of 7mm thickness for blocking the line-of-sight propagation. All testbed experiments are carried out in the 2.4GHz frequency.

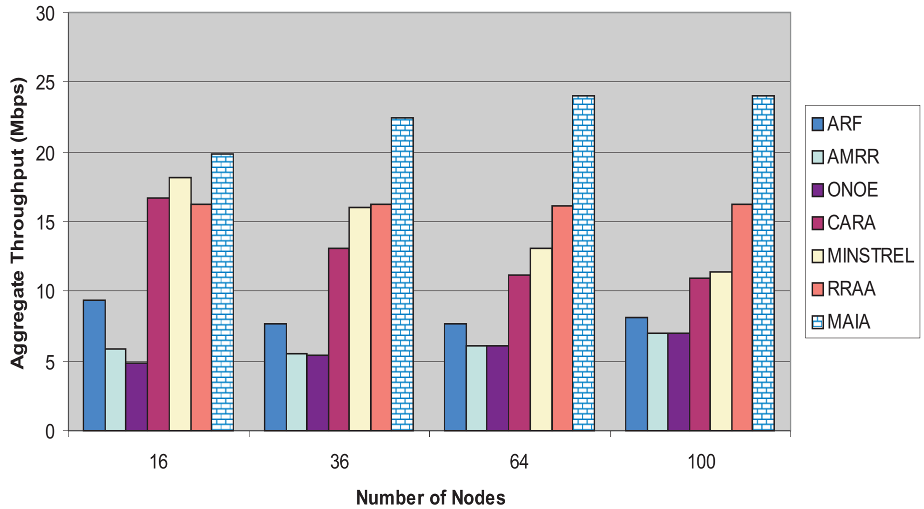
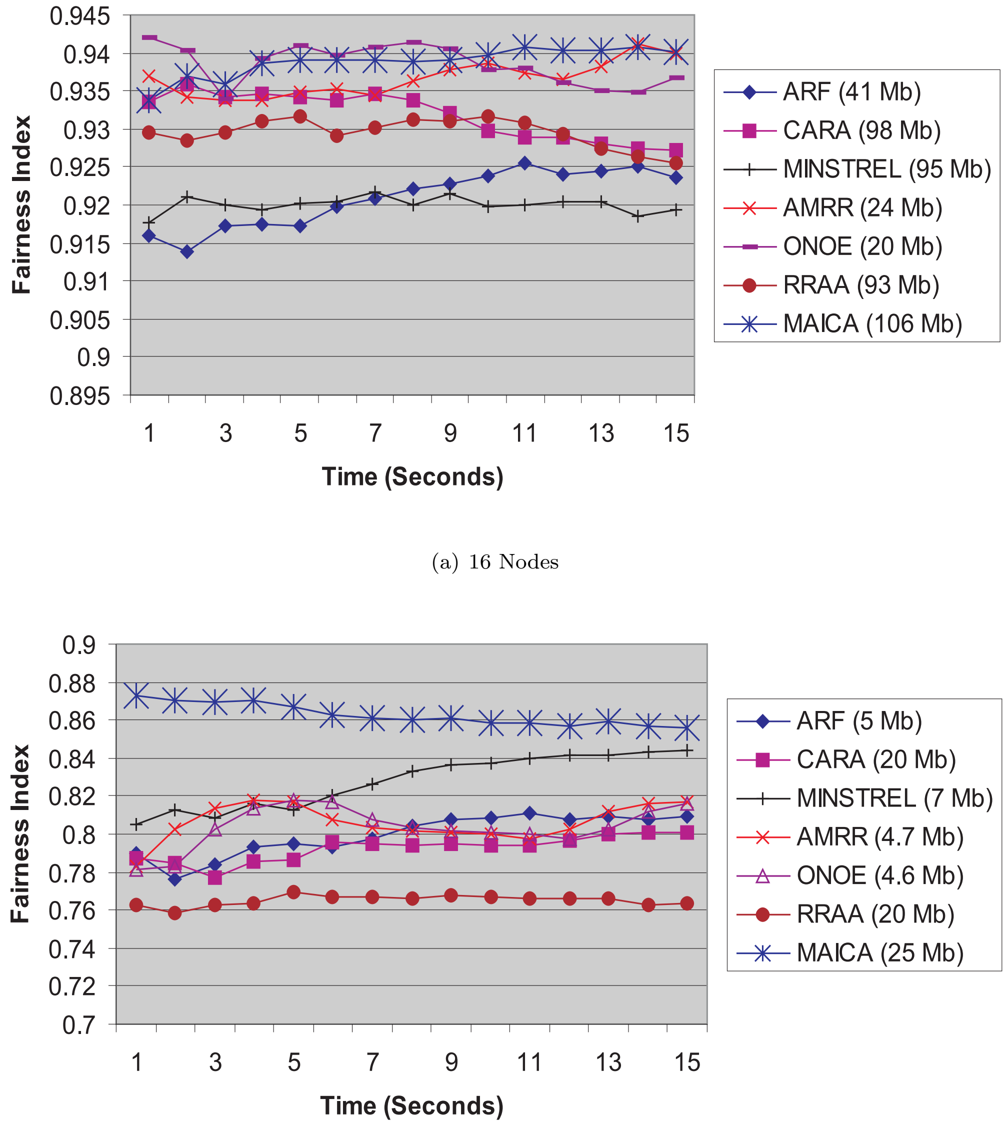


Figure 4.10: Fully Connected Wireless LAN

First, I perform a simple testbed experiment by blocking the line-of-sight between two nodes in concurrent transmission. I block the line of sight for 2, 4, and 8 seconds. In between each event, I allow 10s to see how protocols recover. The goal is to create the dynamics of lossy links and to see how the schemes respond when good links abruptly turn bad and good again.

Second, I used the school's wireless networks, which provides coverage for the whole building. I measure throughput at various locations around our building. A limited mobility scenario is included where a node moves from location A to location D and back to A at speed of 1 m/s (see Figure 4.15(a)).

Third, I brought our laptop to some nearby public WiFi hotspot such as public library, coffee shop, and the airport for further testing. I set up a node at home running Iperf server on port 5001, then I measure the TCP throughput at these public places. Our goal is to study the performance of MAICA in real-world settings where there are many clients vying for access to the wireless medium. All of the experiment results are taken from an average of 10 runs. By using public networks where we have no control over the access points, I demonstrate the backward compatibility of our implementation.



(a) 16 Nodes

(b) 100 Nodes

Figure 4.11: Jain's Fairness Index and Average Aggregate Throughput per Node

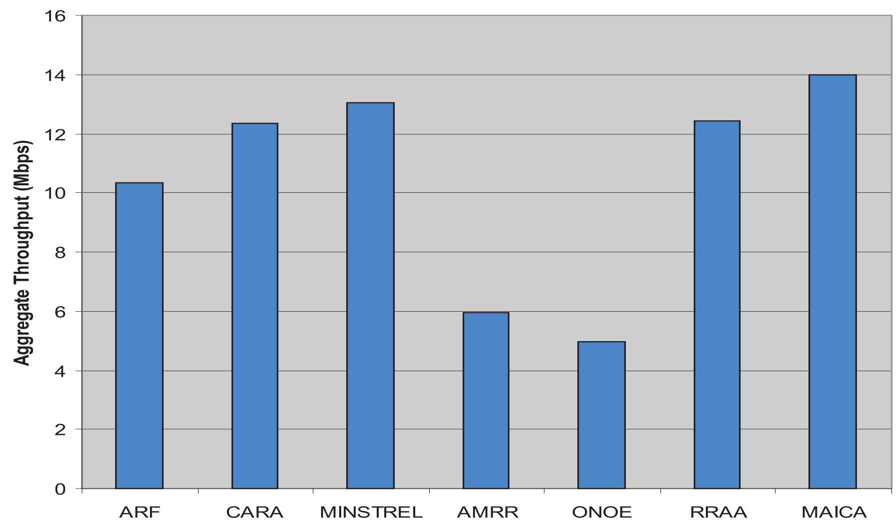


Figure 4.12: 50 Flows in Random 500m x 500m

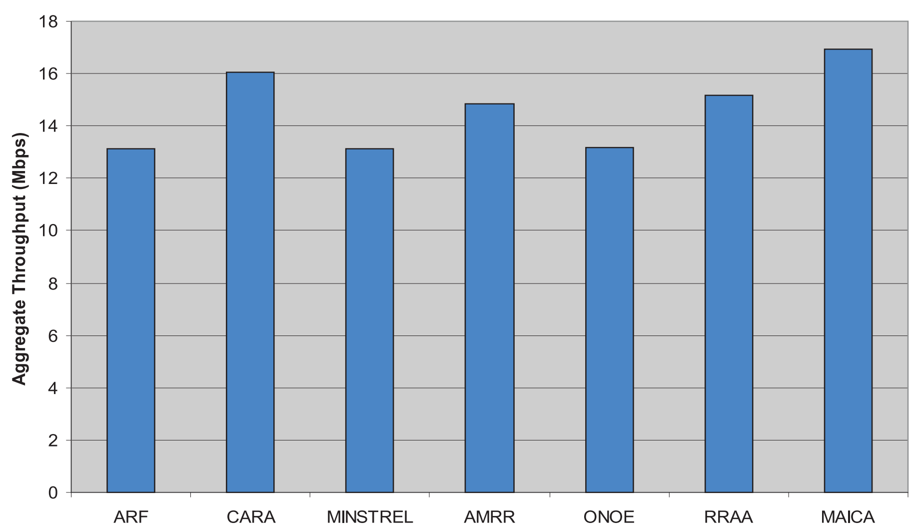
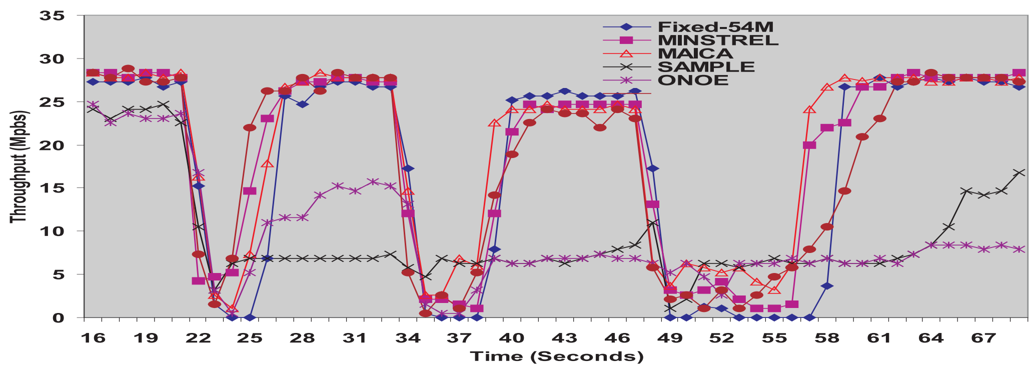
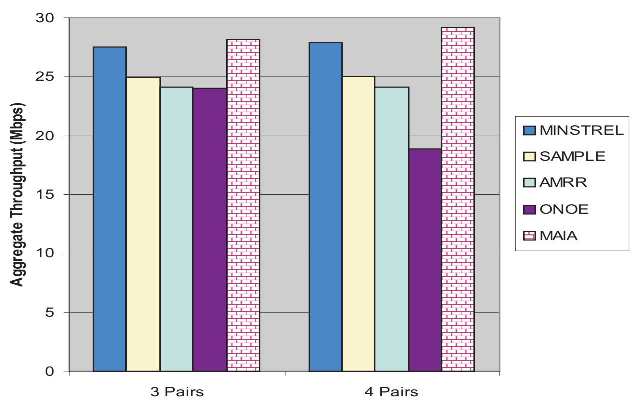


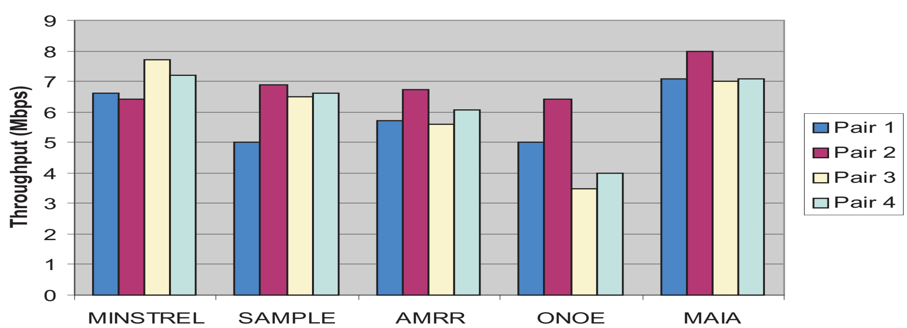
Figure 4.13: 30 Flows and Mobility in Random 500m x 500m



(a) Block Line-of-sight

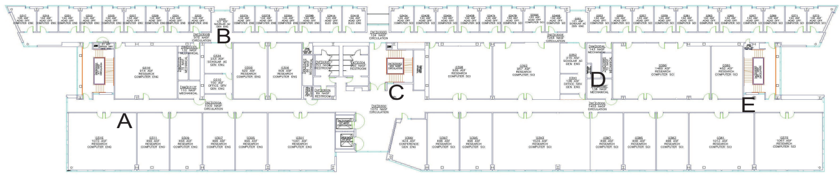


(b) Node Density

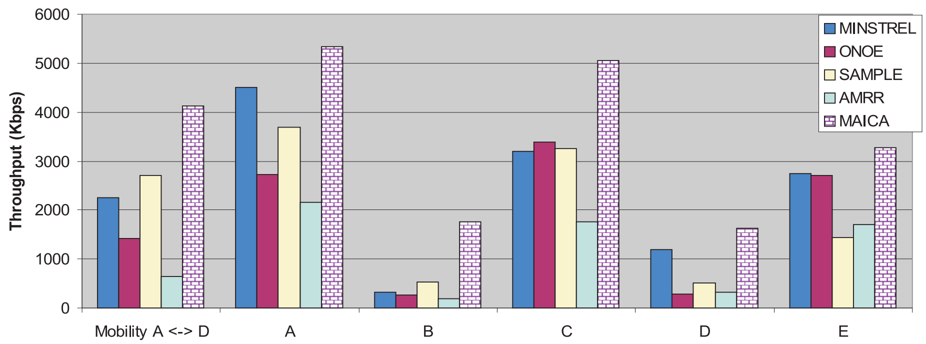


(c) Throughput Distribution

Figure 4.14: Experiments I



(a) Engineering Building 2 Map (30m x 150m)



(b) Throughput vs Location

Figure 4.15: Experiments II

4.3.5 Experimental Results

Figure 4.14(a) shows the throughput for each approach over a sample period of time. I use this scenario for recreating the dynamics of the ever-changing interference (how it comes and goes). Observe that there are three major dips in the graph, each corresponds 2, 4, and 8 seconds block of the line-of-sight between two nodes. All approaches have similar performance except ONOE and SAMPLE which are not able to return to optimal throughput. During the 8 seconds block, AMRR takes longer to return to the optimal throughput. ONOE and SAMPLE fail because they are slow to adapt to the optimal rates due to its trial and error nature.

Figure 4.15(b) reports the results conducted for various locations around our building during the day with many users accessing the school's wireless networks. Figure 4.18 reports the results in public places where there are many users connecting to AP router. At the public library, the throughput obtained is less than 500Kpbs. I conjecture that there was either a bandwidth cap or outdated hardware. As for places such as the coffee shop and the airport, the schemes achieve throughput in excess of 1Mbps. Overall, there is a big improvement in

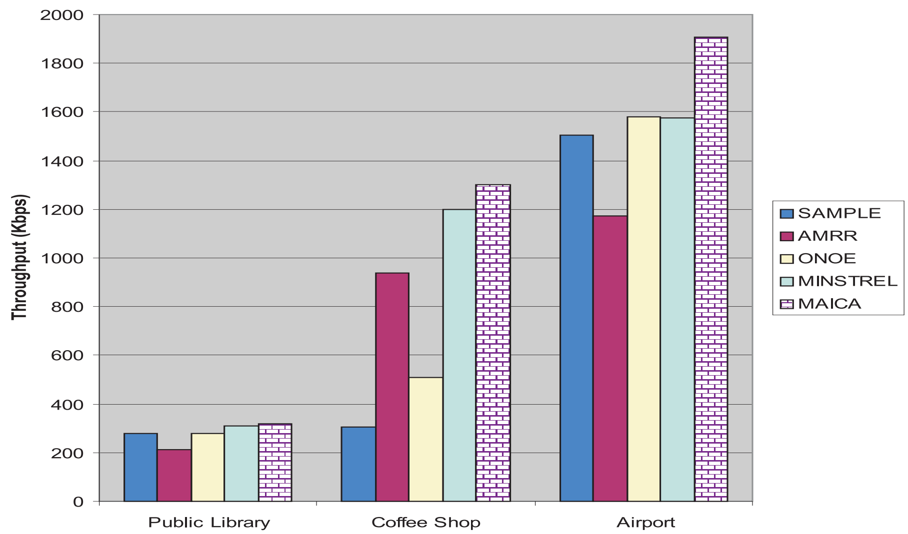
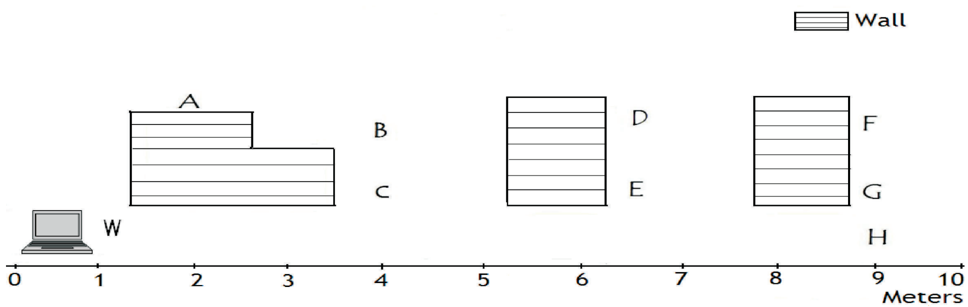


Figure 4.16: Experiments in Public Places

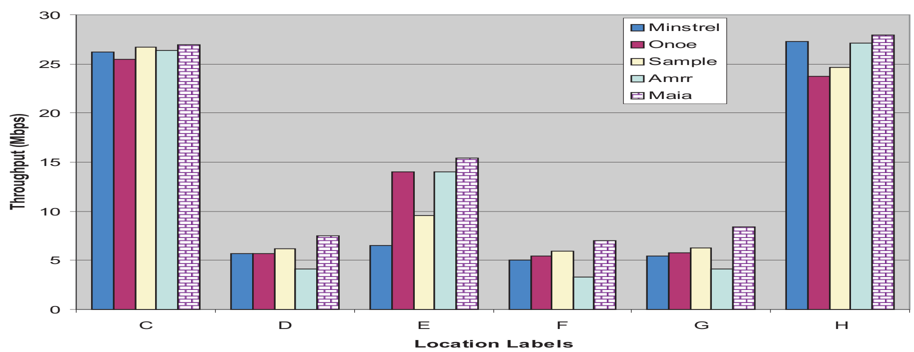
performance of MAICA over all approaches in large public places with many clients, such as the airport. I find that many algorithms fail because they do not consider the effects of multiple clients on rate adaptation.

Figure 4.17 shows the throughput for each node according to its placement in the setup map(see Figure 4.17(a)). Recall that I place a stationary testbed node at Location W and measure the throughput for various locations around home (A to H). Overall protocols have similar performance except at locations D, E, F, and G where a node is farther away and tucked behind the walls. Observe that MAIA performs much better than all other protocols in these locations. Figure 4.17(c) presents the throughput measurements of a node communicating with the target node connected either via wired or wireless connection to AP router. In both scenarios, MAIA performs slightly better than all protocols.

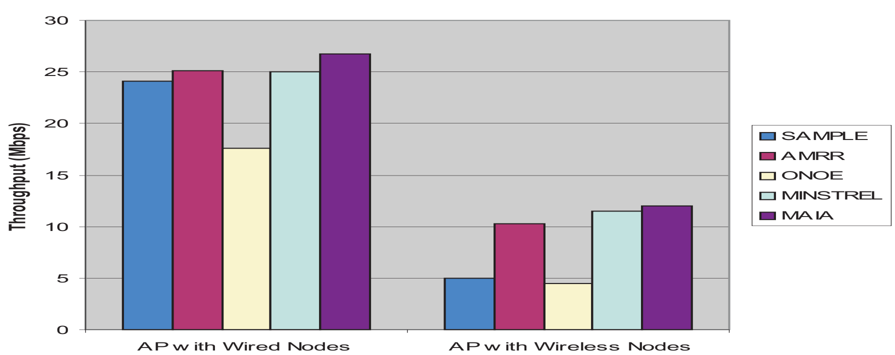
In all scenarios, I observe that MAICA performs significantly better than all other protocols in places where there is weak signal strength or where it has to compete with many clients for access to AP routers (as evident through results in Figure 4.15(b) and Figure 4.18).



(a) The map of experiments with labeled locations from A-H



(b) Throughput of bit rates depending on their locations(Ad HocSettings)



(c) AP Settings

Figure 4.17: Experiments at Home

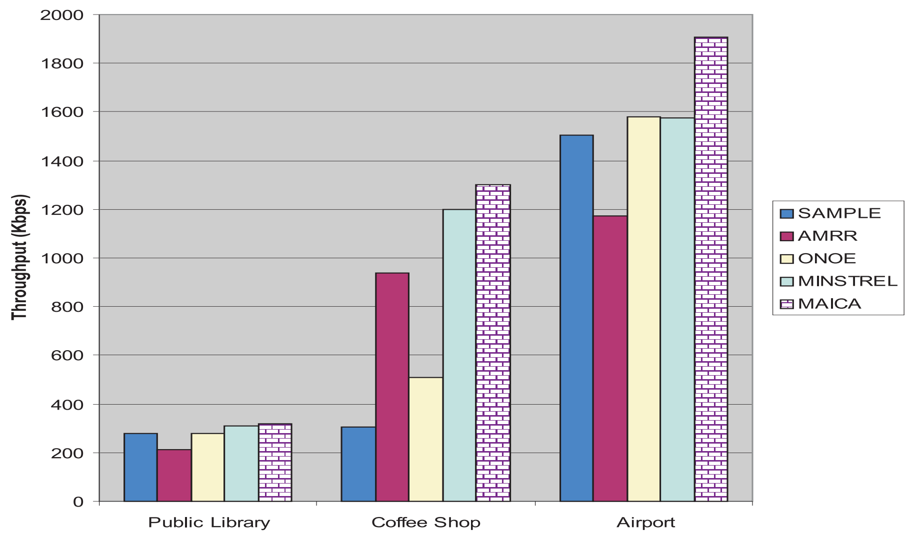


Figure 4.18: Experiments at Public Places

4.4 Summary

I evaluated MAICA extensively via network simulations, analytical model, and real-world experiments in public places and in the lab with many different scenarios for fading, interference collisions, and user density. The results show that MAICA performs consistently better than many multi-rate adaptation schemes that are widely used and deployed today, especially in dense networks with many clients. In the following chapter, I redesign MAICA with a throughput enabled approach.

Chapter 5

Multi-Rate Adaptation: Throughput Enabled Approach

5.1 Throughput-Enabled Rate Adaptation (TERA) [13]

MAICA [12] adapts the data rates used for transmission based on packet loss and a credit-based system. The only limitation in MAICA is the need to tune several parameters needed for its error threshold and credit-based system.

Rate adaptation that operates solely on throughput is desirable, because it would require no configuration of performance parameters, and would incorporate MAI and channel effects implicitly. This is precisely the motivation behind TERA. The design challenge for TERA is that the transmitter must: (a) increase its transmission rate quickly as long as the perceived throughput increases, (b) avoid drastic fluctuations in data rates resulting from instantaneous measurements of throughput, (c) reduce its transmission rate quickly when the channel conditions are poor, and (d) must operate without any a-priori knowledge of the state of the channel. Algorithm 3 specifies the way in which TERA attains rate adaptation based on throughput measurements taking into account the four constraints just mentioned.

This paper addresses rate adaptation for Single Input Single Output (SISO) systems only, but the approach can be easily adapted to IEEE 802.11n or other Multiple Input Multiple Output (MIMO) systems by including the new MIMO enhancements as described in [11].

Algorithm 3 Modulation Group Adaptation

```
ω = time window
ω_probinglow = probe less because channel is bad (900ms)
ω_probinghigh = probe more because channel is good (100ms)
idx = the rate index as shown in Table 6.1.
check(ω) = check whether time window ω is expiring
Multiplicative = successive successful probes
Oscillate = rates oscillate in a see-saw fashion state
Γ_(t−1) = previous throughput
prev_idx = previous rate index
while check(ω) do
  if Probing then
    if Γ < Γ′ and idx! = prev_idx then
      idx = prev_idx
      if Multiplicative then
        ω = ω_probinghigh
      else
        ω = ω_probinglow
      end if
    else if Γ > Γ_(t−1) and idx! = prev_idx then
      ω = ω_probinghigh
    end if
    return
  end if
  if Δ ≥ 1 then
    // Multiplicative Increase
    if Multiplicative and !Oscillate then
      if idx + idx < max_idx then
        idx = idx + idx
      else
        idx = max_idx − 1
      end if
      // Additive Increase
    else if !Oscillate then
      if idx + 1 < max_idx then
        idx + +
      end if
    end if
    Probing = true
  else if Δ ≤ 0.90 and Δ ≥ 0.75 then
    // Additive Decrease
    if idx > 0 then
      idx − −
    end if
  else if Δ < 0.75 then
    // First occurrence: Additive Decrease
    if first and idx > 0 then
      idx − −
      // Successive occurrences: Multiplicative Decrease
    else if second and idx > 0 then
      idx ← idx ∗ M_D
    end if
  end if
end while
```

TERA adjusts the transmission rate based on a periodic review of the performance of transmissions over a time window ω. The rationale for using a time window is to ensure that transmission rates do not fluctuate too much and that transition between rates can proceed as seamlessly as possible. This is particularly important for such bandwidth sensitive applications as audio and video. I use $\omega = 100$ millisecond as an implementation guideline, because it provides sufficient information about the performance attained at given rates, and is also sufficient as a reactive time for adjusting rates.

Let $x_i(t)$ be the rate index of the ith user during time slot t. The throughput attained by a node (Γ) is calculated as in MINSTREL [50] by taking into account the probability of success and the packet transmission time, i.e., $\Gamma = P * M/\tau$, where $P =$ *(number of successful packets)/attempts* denotes the probability of successful transmission using rate index $x_i(t)$, M denotes the Megabits of data transmitted, and τ denotes the time for one try of one packet over the air using rate index $x_i(t)$.

To adjust rates, the ratio of the current throughput is monitored over the reference throughput, which is obtained by applying an exponential weighted moving average (EWMA) filter. This filter can be easily adjusted depending on how sensitive the reaction should be to the surrounding environment. Let $\Delta = \Gamma_t/\Gamma'_t$ be the ratio of the current throughput over the reference throughput with exponential weighted moving average (EWMA) at time t, where Γ'_t is defined as $\Gamma'_t = \alpha * \Gamma_t + (1-\alpha) * \Gamma'_{t-1}$ for $t > 1$ and $\Gamma'_t = \Gamma_t$ for $t = 1$. The coefficient α represents a constant smoothing factor between 0 and 1. A higher α value discounts older observations faster. I recommend a value of α between 0.75 and 0.95 to ensure sufficient reactiveness to changes in throughput and the surrounding environment.

During the transmission window, TERA keeps track of the packet successes as well as the number of attempts. These are used for calculating the probability of success needed to compute the attained throughput at time t (Γ_t). The EWMA filter is used on the current throughput to prevent TERA from being too sensitive to throughput fluctuations.

The initial step in Algorithm 3 consists of determining if the node is in the probing state. The *Probing* flag is used to signal that the protocol is in the probing state. The objective of having such a state is that, in some cases, increasing the transmission rate does not necessarily translate into a higher throughput. The transmission rate is increased only when the node is in the probing state, where the node searches to find the highest rate for the best throughput as quickly as possible with a combination of two probing frequencies.

To determine whether a higher rate produces a better throughput, the node keeps track of the last throughput $\Gamma_{(t-1)}$ it attained using the last rate index, so that the node can fall back to the best throughput. While the node is in probing state, the node sets the frequency of its probing rate to $\omega_{probinghigh}$ if the throughput at the higher rate is larger than or equal to the reference throughput Γ'_t. This allows the node to increase its rate quickly as long as the resulting throughput does not deteriorate. Conversely, if the probing of a higher data rate produces a deterioration in throughput, the frequency with which a new rate is probed is set to $\omega_{probinglow}$. In my implementation, I set $\omega_{probinghigh}$ to 100ms and $\omega_{probinglow}$ to 900ms. The rational for a 900ms delay is to keep the probing window under 1s, so that the transmitter can react to improving channel conditions resulting in better throughput without undue delays. When the current throughput is better than the reference throughput, the transmitter attempts to increase the rate either additively or multiplicatively. If the increased rate provides a better throughput, it is adopted and the algorithm exits the probing state.

To decrease the data rate, the node attempts to find a lower rate that provides better throughput and rely on its probing rate increase to bring it back to the highest transmission rate possible for the best throughput. After ensuring that TERA is not in a probing state for increasing rate, TERA checks the value of Δ, which is the ratio of the current throughput over the reference EWMA throughput. If Δ is at least 1, this is an indication that the transmission rate is of good quality. Accordingly, if there is no oscillation in rate selections, then the rate is increased multiplicatively with successive successful probing increase: $x_i(t+1) = x_i(t) + x_i(t)$; otherwise, the rate is increased incrementally (i.e., $x_i(t+1) = x_i(t) + 1$).

If Δ is between 0.90 and 0.75, then the quality of transmission rate is marginal, and the rate is decreased incrementally (i.e., $x_i(t+1) = x_i(t) - 1$), so that the node may be able to find a better throughput by transmitting at lower rates.

If Δ condition is less than 0.75, then the transmission rate is of poor quality, and the rate is decreased incrementally with the first occurrence (i.e., $x_i(t+1) = x_i(t) - 1$), and is decreased multiplicatively with each successive occurrence (i.e., $x_i(t+1) = M_D x_i(t)$, with $0 < M_D < 1$ a multiplicative factor). The rationale for the initial incremental decrease of transmission rate is to probe the receiver at a slightly lower data rate. If that fails to improve throughput, it is a clear indication that the channel conditions are poor and that the data rate must be lowered drastically.

If Δ value is between 1 and 0.90, the algorithm does nothing in order to account for the fact that the maximum throughput of the next lower rate is still lower than the current throughput.

The *Multiplicative* flag in Algorithm 3 denotes that the node has experienced successive successful p robes (e.g., two successive additively increases of the throughput). In this case, the algorithm enters the *Multiplicative* state, which allows the rate to be increased multiplicatively. The *Oscillate* flag in Algorithm 3 is used to detect scenarios in which rate selection is alternating back and forth. The typical scenario is one in which the transmitter keeps trying to increase its rate, but fails to provide a better throughput doing so every time. When this happens, the frequency of probing is adjusted by resetting ω *Time Window* appropriately to every 100ms or every 900ms.

Note that multiplicative increases and decreases in data rate are needed for TERA to respond quickly to drastic changes in channel conditions. Using only incremental changes to transmission rates would make the transmitter too slow in responding, given that there are 10 possible rates in IEEE 802.11g and rate adaptation must take place over a transmission window.

5.2 Performance Evaluation

5.2.1 Simulation Results

I compared the performance of TERA against several relevant rate adaptation schemes using NS-3 [72]. Unless otherwise specified, the following parameters are used in the simulations: a packet size of 512 bytes, a drop tail queue with a maximum length of 100, the IEEE 802.11a MAC model, a constant speed propagation model, a log distance propagation loss model ($L = L_0 + 10n \log_{10} \frac{d}{d_0}$), a transmission range of 150m, interference range of 250m, and UDP

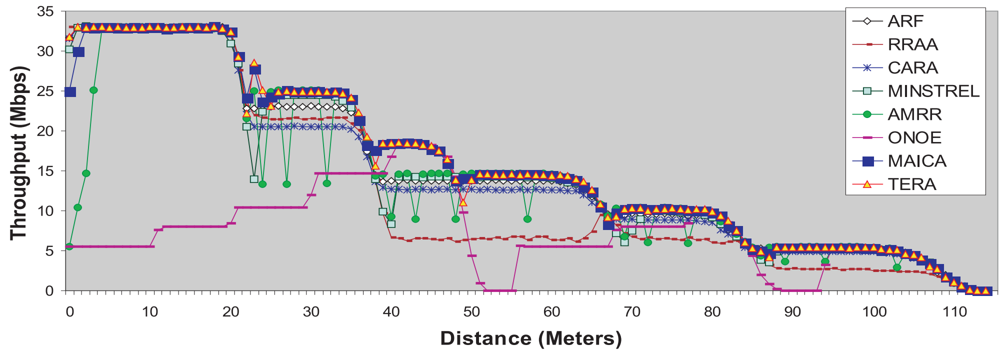


Figure 5.1: Node Moves Away From Target

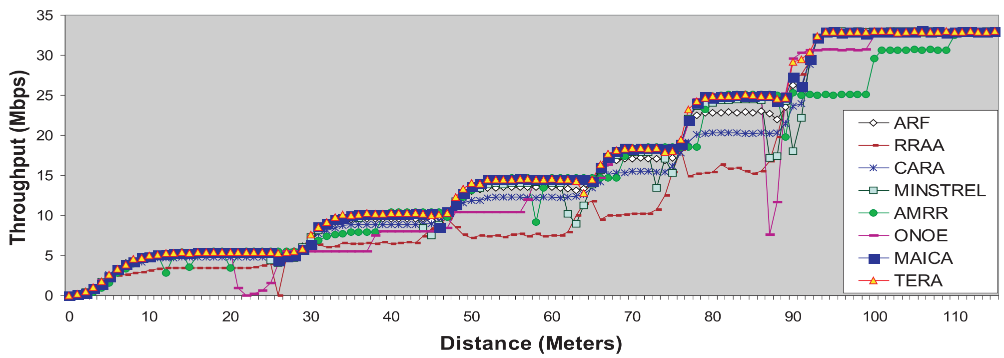


Figure 5.2: Node Moves Towards Target

throughput. Each simulation was performed for 60 seconds; simulations lasting longer than 60 seconds produced similar results in benchmarking runs.

First, I considered the case of two nodes moving during data transmission. The source node moves at a speed of 1 m/s away from the target with no pause. The objective is to see how decreasing signal strength and fading affect performance. Similarly, I have the source node moves at a speed of 1 m/s towards the target with no pause. The objective is to study the reactiveness of protocols with increasing signal strength.

Figure 5.1 reports the results for this scenario. The throughput results are at the lowest when nodes are farthest apart at 140m and greatest when they are in close proximity. ONOE, being conservative in raising rates, takes some time before transmitting at the optimal rate. AMRR has many sporadic dips throughout the experiment, this is probably due to the exponential back-off mechanism. RRAA does not perform well in this fading scenario because it lowers its rate quickly due to employing short-term loss. MINSTREL performs well but it takes dips during transition such as 30m, 50m, 90m, and 115m due to its probing and trials and errors nature before achieving the optimal rate. CARA's control packets probing for collision detection suffers a slight performance decrease. The fading pattern works well for ARF due to gradual increasing and decreasing signal strength; however it still does not perform well as TERA. During the fading transition, TERA lowers its rate accordingly to adapt to it. The same applies to MAICA. However, due to the credit threshold, MAICA is slow at the initial start because it does not employ multiplicative increase like TERA. Figure 5.2 provides the opposite scenario where a node is moving toward the target node. Ideally, protocol should be able to recognize the increasing signal strength and increase their rates accordingly. Again, TERA has a smooth transition between rates due to its reactiveness.

Second, I select nine sources placed in a 10x10 grid topology and assign them 25 target nodes with the flows being exponentially distributed with mean of 3 seconds, for a total of 225 (25x9) distinct flows (see Fig 5.4(a)). I choose the grid topology and set it up this way because I did not want to employ any specific routing protocol, which could have influenced the results on rate adaptation. Flows are exponentially distributed to ensure that this scenario does not favor any approach. Finally, I have 50 flows in a grid 10x10 topology where each node transmits to its immediate neighbor.

Figure 5.3 shows the results for MAICA and TERA in a 100-node,10x10 grid topology with each node transmitting to each of its 8 immediate neighbors with exponentially distributed flows lasting an average of 3 seconds. In these scenarios, I show that TERA can perform as well as, if not better than, MAICA without the complexities incurred in MAICA for the careful tuning of parameters. Given that TERA is a better approach than MAICA, the rest of the subsequent scenarios and experiments do not include MAICA.

Figure 5.4(b) reports the results for the 10x10 grid scenario described in Figure 5.4(a) with exponentially distributed flows. AMRR, ARF, and ONOE do not perform well due to conservative approach with which they increase transmission rates. RRAA does not perform well because short-term loss ratio is unpredictable in this experiment.

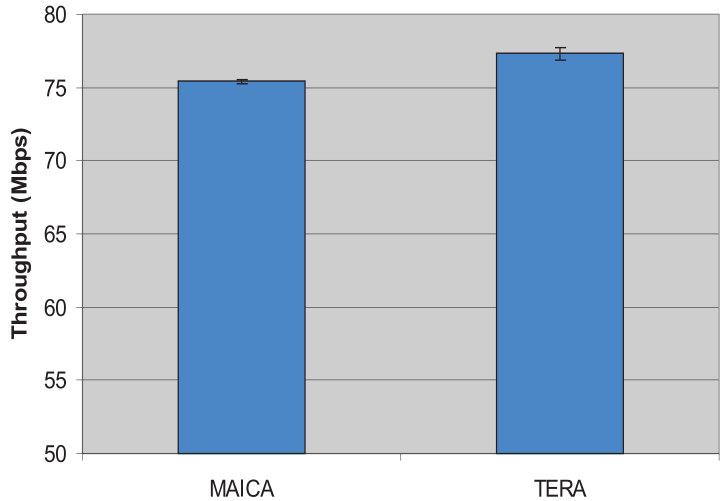


Figure 5.3: MAICA and TERA

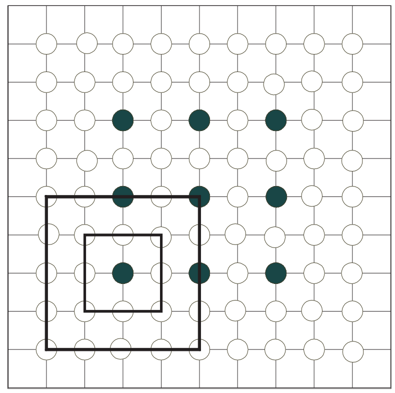
Figure 5.5 shows the results for the 50-flow scenario in which nodes are paired up with its immediate neighbors. In this scenario, progressive increase seems to outperform other aggressive probing protocols, as in ARF, CARA, and MIN STREL.

5.2.2 Experimental Results

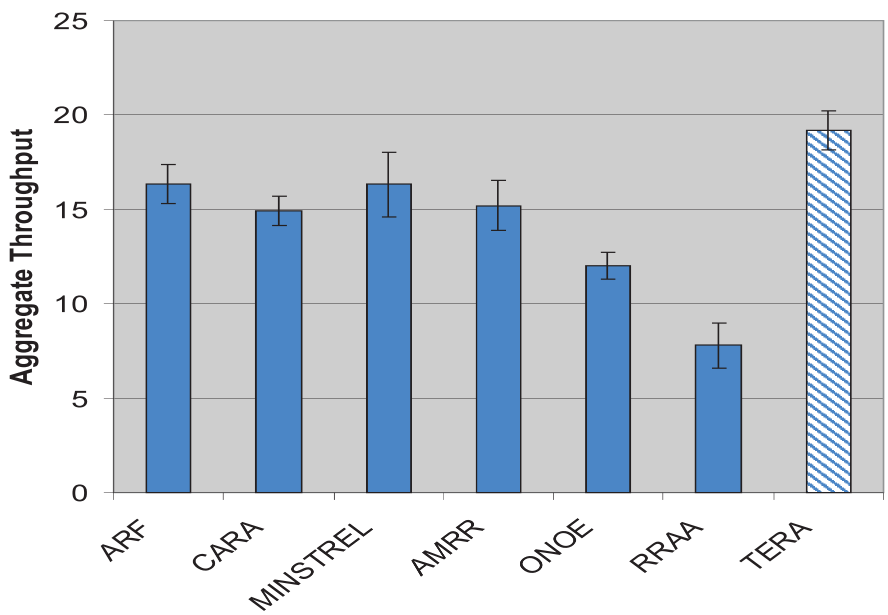
I implemented TERA in the Linux Kernel Wireless Stack [70] to demonstrate that it can work correctly and independently of the Atheros chipset. We compare our approach against the default ATH5K Atheros chipset and a popular Linux rate control MINSTREL, given that these are available and widely deployed in many real-world settings. Many schemes in our network simulations are either not publicly made available or only exist as simulation code.

I set up our wireless access point in our laboratory as shown as a star symbol in Figure 5.6(a). I measure throughput at different locations around the building.

Figure 5.6(b) reports the experiment results of our protocol compared at two other widely popular and deployed Linux rate adaptations. At location A, nodes are about 10m from the access point (AP) router and they are all able to obtain high throughput approximately 20 Mbps. I repeat the experiment on average of seven times and find that MINSTREL has a large



(a) Setup



(b) Aggregate Throughput

Figure 5.4: Exponentially Distributed Flows (mean=3s)

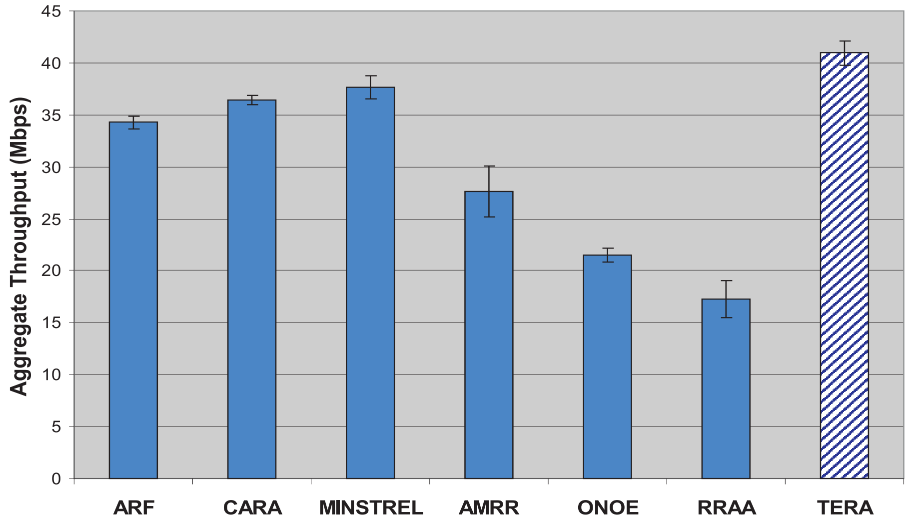
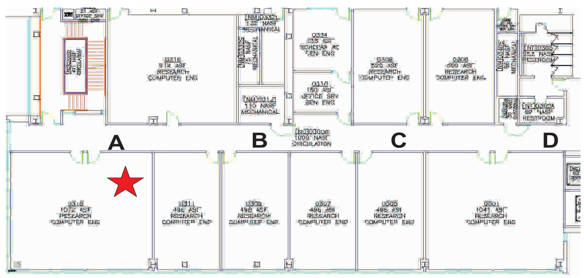
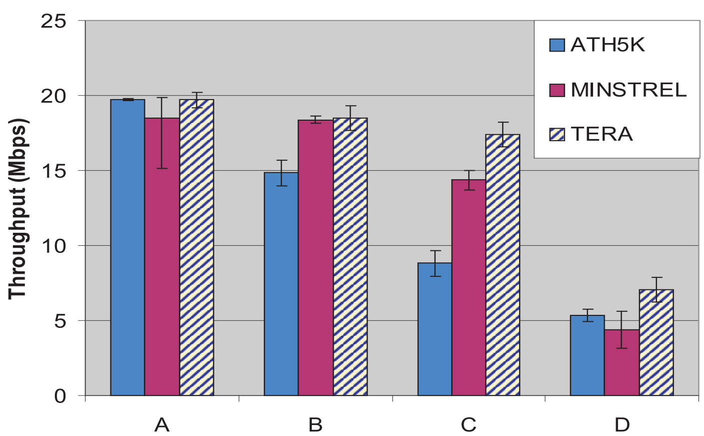


Figure 5.5: 50 Flows

standard deviation, perhaps due to its spending 15 percent of the time trying random rates even though its current is working perfectly fine. At location B, 20m away from the AP, protocols suffer only slight degradation in throughput. ATH5K, however, gets stuck in sub-optimal rates. At location C and D, which are 40m and 65m away from the AP router respectively, TERA was able to perform much better than ATH5K and MINSTREL. At location C, TERA is 50 percent better than ATH5K and 15 percent better than MINSTREL. At location D, TERA is 20 percent better than both ATH5K and MINSTREL.



(a) Engineering Building (30m x 75m)



(b) Throughput vs Location

Figure 5.6: Experiment Results: Throughput at Different Locations

5.3 Summary

I proposed a new throughput enabled TERA protocol, based on implicit feedback. The key insight of my work is that throughput can provide a great measurement tool for adapting rates robustly. I evaluated TERA extensively via network simulations and real world settings with many different scenarios for fading, interference collisions. The results show that TERA performs consistently better than many multi-rate adaptation schemes that are widely used and deployed today, especially in dense networks. Furthermore, TERA is surprisingly simple, practical, and is compatible with today's WiFi networks.

Chapter 6

Multi-Rate Adaptation for MIMO Systems

6.1 Introduction

The introduction of spatial multiplexing and transmit diversity, types of guard interval, and channel width complicates even more the job of rate adaptation. Due to these complexities for selecting rates in IEEE 802.11n, the standard [34] defines Modulation Coding Scheme (MCS) Index ranges 1 to 76 to simplify rate selections for up to a maximum of four antennas.

To simplify the job of adapting rates for IEEE 802.11n, I categorize different types of modulation into a modulation group and spatial multiplexing, transmit diversity, types of guard interval, and channel width into an enhancement group. Then, I adapt these two groups concurrently. The combination of these two groups are mapped back to MCS for adapting rates in IEEE 802.11n.

6.1.1 New Enhancements in MIMO IEEE 802.11n and Current Approaches

The difference between IEEE 802.11n and its predecessor is the new enhancements for higher throughput such as Multiple Input Multiple Output transmissions (MIMO), shorter

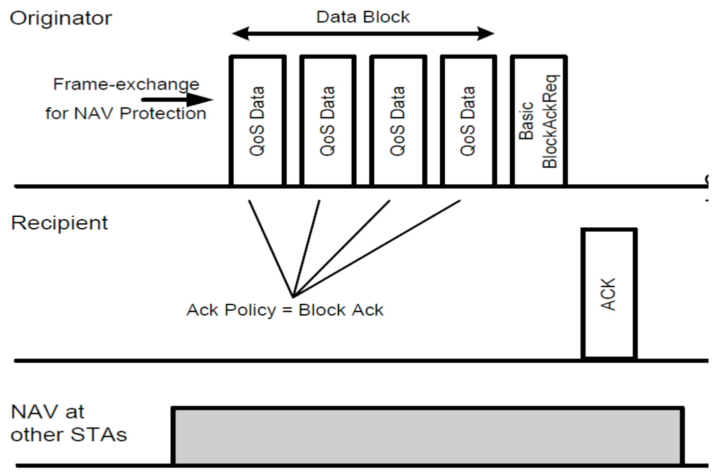


Figure 6.1: Block Ack

guard intervals, wider channel width, frame aggregation, and block ACKs.

The use of MIMO provides great performance improvements due to array gain, diversity gain and spatial multiplexing gain. Array gain refers to an increase in average receive SNR due to a coherent combining effect. Apart from depending on the number of transmit and receive antennas, channel knowledge at the transmitter and receiver is required in order to yield this array again [56]. Transmit diversity is a technique used to overcome fading in wireless links and increase robustness; it allows transmitting the signal over multiple fading paths for redundancy (e.g., [2,67]).

Spatial multiplexing allows nodes to send multiple streams of data independently and concurrently; hence, it can attain a significant increase in throughput. This is due to MIMO channels achieving a linear increase in capacity [18,57,68].

Additional performance gains are made possible in IEEE 802.11n through the use of reduced guard intervals, a wider channel width, frame aggregation, and block acknowledgments (ACK). Channel bonding has a big impact on throughput in IEEE 802.11n and careful management is needed [15]. Many of the rate adaptation schemes designed for IEEE 802.11 are not applicable to the enhancements available in IEEE 802.11n. As a result, they need to be redesigned to take full advantage of these enhancements. Recently, there have been a number of efforts trying to address these problems and challenges for rate adaptation in MIMO systems.

Table 6.1: Modulation Group

m_index	Modulation	Coding Rate
0	BPSK	1/2
1	QPSK	1/2
2	QPSK	3/4
3	16-QAM	1/2
4	16-QAM	3/4
5	64-QAM	2/3
6	64-QAM	3/4
7	64-QAM	5/6

In [45], the authors address the complexity of deciding between spatial multiplexing and diversity by proposing the Demmel condition, which is based on instantaneous channel state and the computed minimum Euclidean distance at the receiver, as an indicator of spatial structure of MIMO channels [27]. The proposed rate adaptation is based on a two-dimensional look-up table of the average SNR per subcarrier and the average Demmel condition number per subcarrier, which are exchanged through RTS and CTS packets [45].

ATH9K [4] is an emerging open-source driver for IEEE 802.11n. It contains its own rate adaptation based on probing, and sorts the rates that provide the best throughputs, which takes into account of the packet error rates (PER). The PER for each rate is defined as the ratio of the number of bad frames over the number of frames transmitted. The PER for each rate is maintained using an exponential weighted moving average (EWMA). The main disadvantage of ATH9K is the random probing used and searching for the optimal rate. In some cases, rates are "stuck" in sub-optimal rates.

Minstrel High Throughput (HT) [17] extends Minstrel [50] for IEEE 802.11n rate adaptation by constructing two dimensional array tables with sample columns and MCS group rates. Because of the large sample space, Minstrel High Throughput divides the MCS group rate into smaller 8 sub-groups with varying number of streams and channel width. It then populates the

Table 6.2: Enhancement Group

e_index	No. of Spatial Streams	Guard Interval	Channel Width
0	1	800 ns	20 MHz
1	1	400 ns	20 MHz
2	1	800 ns	40 MHz
3	1	400 ns	40 MHz
4	2	800 ns	20 MHz
5	2	400 ns	20 MHz
6	2	800 ns	40 MHz
7	2	400 ns	40 MHz
8	3	800 ns	20 MHz
9	3	400 ns	20 MHz
10	3	800 ns	40 MHz
11	3	400 ns	40 MHz
12	4	800 ns	20 MHz
13	4	400 ns	20 MHz
14	4	800 ns	40 MHz
15	4	400 ns	40 MHz

samples table with rates randomly chosen from these groups and uses them for sampling rates. With the division of search space, Minstrel HT still suffers the same drawback as ATH9K due to its randomness in search.

MiRA [58] is based on zigzagging and sampling between intra- and inter-mode rate options or the mode selection between single spatial stream and double spatial streams. After this sampling period, it sorts through the sampled rates to find the best rate. The drawback of MiRA is that it expenses resources exploring other rates when its first sample rate may be the best rate. MiRA, in its current state, is limited to single and dual spatial streams and does not address multiple spatial streams. It is unclear how its adaptive probing has an effect on the reactiveness of the scheme. In our experiments, I find that short probing interval does not provide sufficient information about the probing rate. We will explain our approach to the probing time interval in Section 6.3.

Effective SNR [26] presents a delivery model by taking RF channel state as input and predicts packet delivery for the links based on the configuration of the Network Interface

Controller (NIC). The drawback of using CSI is that SNR needs to be measured instantaneously, and feedback delay may not allow mode adaptation on an instantaneous basis [23]. CSI itself is an approximation of the wireless channel and has many parameters. To improve its accuracy, CSI may need to incorporate other information, such as higher order statistics of SNR and Packet/Bit Error Rate or both for robustness [23]. However, CSI may work well if the channel condition does not change instantaneously.

Table 6.3: MCS Index Mapping

MCS Index	Modulation (M) and Enhancement (E)	Guard Interval (ns)		Channel Width (MHz)	
0	M0 :: E0-E3	800	400	20	40
1	M1 :: E0-E3	800	400	20	40
2	M2 :: E0-E3	800	400	20	40
3	M3 :: E0-E3	800	400	20	40
4	M4 :: E0-E3	800	400	20	40
5	M5 :: E0-E3	800	400	20	40
6	M6 :: E0-E3	800	400	20	40
7	M7 :: E0-E3	800	400	20	40
8	M0 :: E4-E7	800	400	20	40
9	M1 :: E4-E7	800	400	20	40
10	M2 :: E4-E7	800	400	20	40
11	M3 :: E4-E7	800	400	20	40
12	M4 :: E4-E7	800	400	20	40
13	M5 :: E4-E7	800	400	20	40
14	M6 :: E4-E7	800	400	20	40
15	M7 :: E4-E7	800	400	20	40
16	M0 :: E8-E11	800	400	20	40
17	M1 :: E8-E11	800	400	20	40
18	M2 :: E8-E11	800	400	20	40
19	M3 :: E8-E11	800	400	20	40
20	M4 :: E8-E11	800	400	20	40
21	M5 :: E8-E11	800	400	20	40
22	M6 :: E8-E11	800	400	20	40
23	M7 :: E8-E11	800	400	20	40
24	M0 :: E12-E15	800	400	20	40
25	M1 :: E12-E15	800	400	20	40
26	M2 :: E12-E15	800	400	20	40
27	M3 :: E12-E15	800	400	20	40
28	M4 :: E12-E15	800	400	20	40
29	M5 :: E12-E15	800	400	20	40
30	M6 :: E12-E15	800	400	20	40
31	M7 :: E12-E15	800	400	20	40

6.2 Packet Error Approach to Multi-Rate Adaptation for MIMO

The amendment enhancements for higher throughputs to the IEEE 802.11 standard brings much promise as well as many challenges in adapting rates. For example, how do we switch between spatial multiplexing and transmit diversity? When should we fall back to using a smaller channel width (20 Mhz vs 40 Mhz)? Which guard intervals should we employ? For this reason, IEEE 802.11n divides and groups all different MIMO configurations into 76 MCS indices to help facilitate rate adaptation. However, MCS does not provide a monotonic relation between loss and index rates. This is why many rate adaptation schemes, including Ath9k [4], Minstrel HT [17], and MiRA [58], resort to variations of random sampling. Our goal is to show that one can take advantage of the monotonic relation between loss and modulation types and build a mapping with the new enhancement features of IEEE 802.11n. This mapping allows us to adapt rate in an orderly fashion instead of random sampling. With this motivation, I take a drastic approach by separating rate adaptation into two groups: (a) the modulation group, which consists of different types of modulation; and (b) the enhancement group, which consists of the new enhancement features for IEEE 802.11n such as spatial multiplexing, transmit diversity, guard interval, and channel width. Each group has its own rules for upgrading and downgrading indices, but they are adapted concurrently. These indices are then mapped back to MCS for adapting rates in IEEE 802.11n. Because modulation group consists of modulation in varied degrees of redundant bits, it is natural that I choose it greedily for the highest throughput. As for the enhancement group, I adapt them based on stream error detection and delivery ratio. I describe the modulation group first and follow by the enhancement group. Note that Minstrel [50] separates rate indices and groups differently for efficient random sampling.

6.2.1 Adapting Modulation-Type Group

Table 6.1 lists a table of modulation schemes with its corresponding coding rate. For example, index 7 has a modulation type of 64-QAM with a coding rate of 5/6 (one redundant bit is inserted every six bits). It is natural that we want to increase to the highest index or the

Algorithm 4 Adapting Modulation Group

```
w = sampling of packets window
τ_ϵ = success packets times acceptable error rates
τ_γ = credit threshold for promoting to the next rate
m_index = modulation group index as seen in Table 6.1
credit γ = 0; retransmitPackets ρ = 0
successPackets σ = 0; errorPackets ϵ = 0
//comment: in addition to w, time window ω is required
while (σ + ϵ < w) do
  if (packet_is_success) then
    σ++
  else if (packet_is_error) then
    ϵ++
  end if
  if (packet_is_retried) then
    ρ++
  end if
end while
//comment: zero success packets or with many retries
if (σ == 0||σ < ρ) then
      m_index = m_index − 1
      γ = 0
end if
//comment: downgrade modulation group index
if (ϵ > τ_ϵ) then
      m_index = m_index − 1
          γ = 0
end if
//comment: within acceptable error threshold
if (ϵ ≤ τ_ϵ) then
      γ++
end if
//comment: ensure stability before upgrading
if (γ ≥ τ_γ) then
      m_index = m_index + 1
      γ = 0; ρ = 0
      σ = 0; ϵ = 0
end if
```

highest modulation type for the best throughput.

6.2.1.1 Modulation-Group Adaptation Rules

Adapting the modulation group is similar to rate adaptation in SISO systems. The separation into modulation group allows us to concentrate only on varying modulation types, which further reduces complexity and adaptation rules.

Modulation-group adaptation makes its decision by keeping track of the number of successes over a *rate adaptation window* w corresponding to the number of packets transmitted. First, it makes sure that packet transmission of w distinct packets (our minimum recommended value is 30), excluding retransmission, have been reached. Then, I enforce time window ω and make a decision when the time window concludes. I use $\omega = 100$ millisecond as an implementation guideline. The rational for using a time window is to ensure proper reactivity in case the sender does not transmit w packets within ω seconds. This time window also allows us more flexibility in choosing w parameters.

During the transmission window, RAMAS keeps track of three counters, one each for three transmission cases: σ for packet success; ϵ for packet error; and ρ for packet retransmission.

- If the number of packet errors $\epsilon \leq \tau_\epsilon$, a credit is added to the credit counter γ. Once γ reaches the threshold τ_γ, then modulation group index is increased and γ is reset to 0. The credit counter allows us to increase more progressively to avoid erratic variations.

- If the number of packet errors $\epsilon > \tau_\epsilon$, index is decreased; the credit counter γ is reset to 0. Note that we decrease right away rather than subtracting a credit.

- Finally, if the number of success $\sigma < \rho$ (the number of re-transmissions), then the channel retransmits too much and the index is decreased.

At the end of the window, σ, ϵ and ρ are re-set to 0.

The exact method to increase and decrease index for modulation group is described in Algorithm 4.

Unlike many credit-based systems, which assign credits based on consecutive successes, our credit-based system is based on obtaining k successes out of N trials, where each trial is

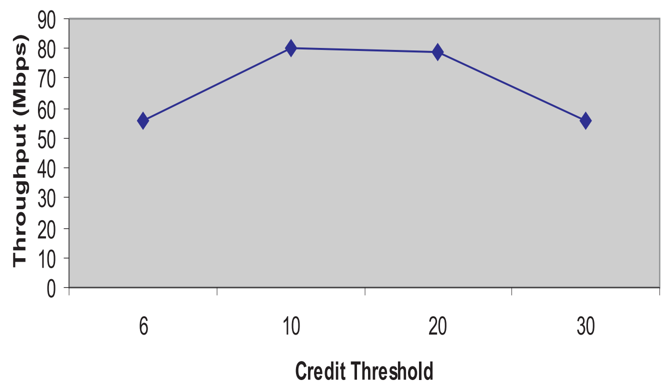


Figure 6.2: Credit Threshold

affected by the common "air interface" and contention errors.

Because stability plays in important roles in adapting rates, this Binomial-like property loosens the restrictions of consecutive successes and helps to facilitate the flow of credits and, at the same time, prevent rates from being "stuck" as experienced in ARF [41].

6.2.1.2 Guideline for Setting Parameters

Note that there are a few other parameters in Algorithm 4. First, what is the acceptable error threshold? Its purpose is to loosen requirement of consecutive successes and to facilitate the flow of credits. In our experiment, I find that requiring consecutive packet success can cause rates being "stuck" and throughput starvation. Between 15% to 25% acceptable error threshold, there is almost no impact on throughput (Our recommended value is 20%).

As for the credit threshold, its main objective is to ensure stability before upgrading the next modulation index. Figure 6.2 plots the impact of different credit threshold on throughput for various scenarios. The credit threshold is nothing more than an additional mechanism to prevent false positives. Values between 10 an d 20 have almost no effect on throughput (Our recommended value is 10).

Algorithm 5 Adapting Enhancement Group

```
e_index = enhancement group index as as seen Table 6.2
α = Aggregate MPDU length
β = Aggregate MPDU ACKs length
Φ = stream failure threshold
Ψ = packets delivery threshold for upgrading
η = number of streams;
ε = stream errors transmission
//comment: for multiple streams
if (η > 1) then
  //comment: check and register stream errors
  if (α − β + ρ/2 > β) then
    ε++
  else
    ε = 0
  end if
  //comment: making sure stream errors still persist
  if (ε > Φ) then
    e_index = e_index − 1
    ε = 0
  end if
  //comment: for upgrading index
  if (success/attempts > Ψ) then
    e_index = e_index + 1
  end if
end if
//comment: for single stream
if (η == 1) then
  if (success/attempts > Ψ) then
    e_index = e_index + 1
  end if
end if
```

6.2.2 Adapting Enhancement Group

Table 6.2 shows the ranking of indices for the enhancement group. These are the new features and enhancements in IEEE 802.11n that allows node to achieve a much higher throughput. Number of spatial streams denotes the number of independent streams in transmissions. Guard interval denotes the space between each symbol and is enforced to reduce inter-symbol interference (ISI). Guard Interval can be switched between 400ns or 800ns depending on hardware support. The same applies to channel width with 20 Mhz and 40 Mhz channel. This enhancement group table covers up to a maximum of four antennas; however, our experiment uses off-the-shelf wireless cards that only supports 3 spatial streams and MCS indices up to 23.

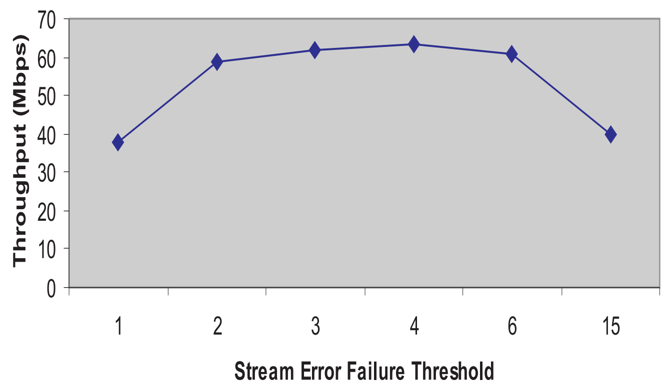


Figure 6.3: Stream Error Failure Threshold

It is natural that we adapt the enhancement group index to the highest index, in this case, 4 spatial streams, 400ns guard interval, and 40Mhz channel width.

6.2.2.1 Spatial Multiplexing vs Transmit Diversity

The selection between spatial multiplexing or transmit diversity can have an adverse impact on performance. Spatial multiplexing allows nodes to send multiple streams of data independently. Transmit diversity allows nodes to replicate data on different fading paths to increase robustness.

In [45], the authors point out that spatial multiplexing works best when there exists a multiplexing structure; otherwise, it may reduce the reliability and worsen the throughput. But how can we identify this multiplexing structure? In [27], the authors propose the Demmel condition as an indicator for figuring out this multiplexing structure. Based on instantaneous channel state and the computed minimum Euclidean distance at the receiver, the Demmel condition of the matrix channel can be obtained and provides sufficient condition for selecting between multiplexing and diversity. This Demmel condition needs to be relayed back to the sender by control packets.

Because RAMAS is centered on simplicity and implicit feedback, I strive to provide a simpler approach. The intuition revolves around spatial multiplexing and how symbols are

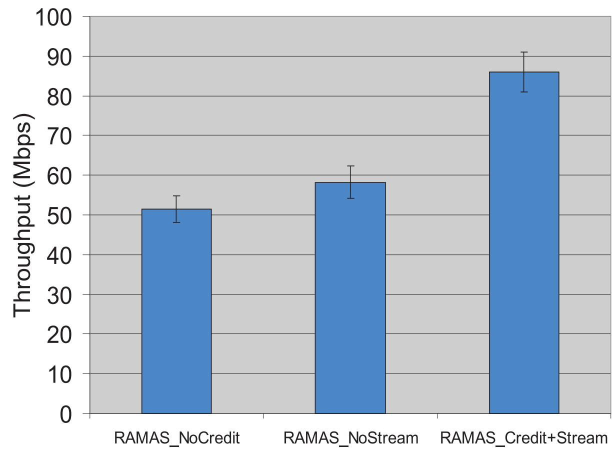


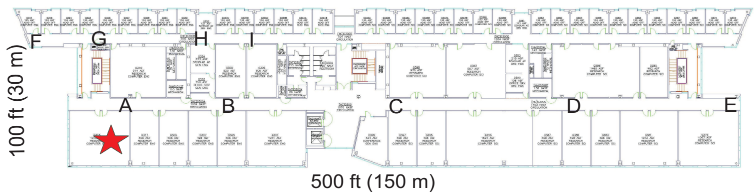
Figure 6.4: RAMAS Performance Decomposition.

spread out among each antennas. Therefore, RAMAS keeps track of the number of packets transmitted and acknowledged through spatial multiplexing. If the number of loss packets is greater than the number of ACKs, RAMAS deduces that one or more of the multiple antennas is having difficulty transmitting packets. If this event persists, RAMAS lowers the number of spatial streams until there is only one stream. Our approach allows nodes to gradually lower the number of spatial streams and switch to transmit diversity completely when the situation warrants it. The goal is to use spatial multiplexing whenever possible because that is where most of the additional throughput capacity can be harvested.

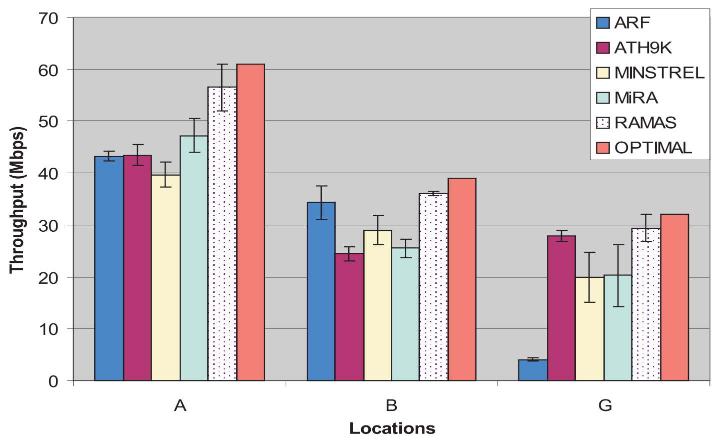
6.2.2.2 Enhancement-Group Adaptation Rules

Adaptation for the enhancement group is centered on stream error detection. Our goal is to determine when one of the antennas is not doing well in terms of transmitting the packets. This is the case where the number of loss packets is greater than ACKs in a Block ACK bitmap. I rely on aggregate MAC protocol data unit (AMPDU) length and block ACK length for stream error detection. MPDU aggregation comes in various sizes, and more aggregation leads to higher throughput.

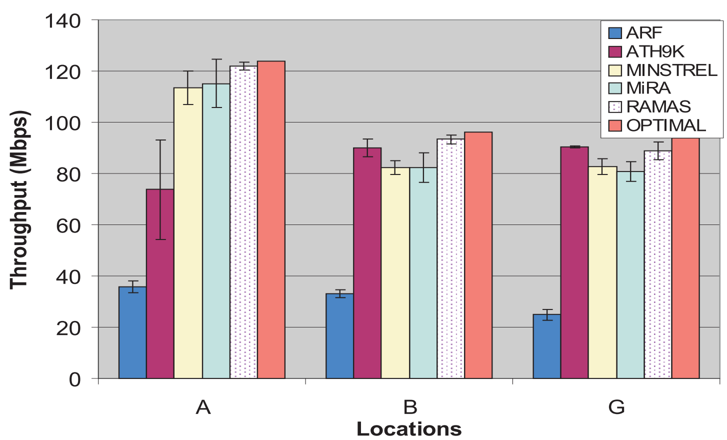
Note that the Enhancement Group adaptation has no packet window w or time window ω, instead, it is checked with the reception of block ACK packets. I divide it into two cases:



(a) Map of Engineering 2 Building

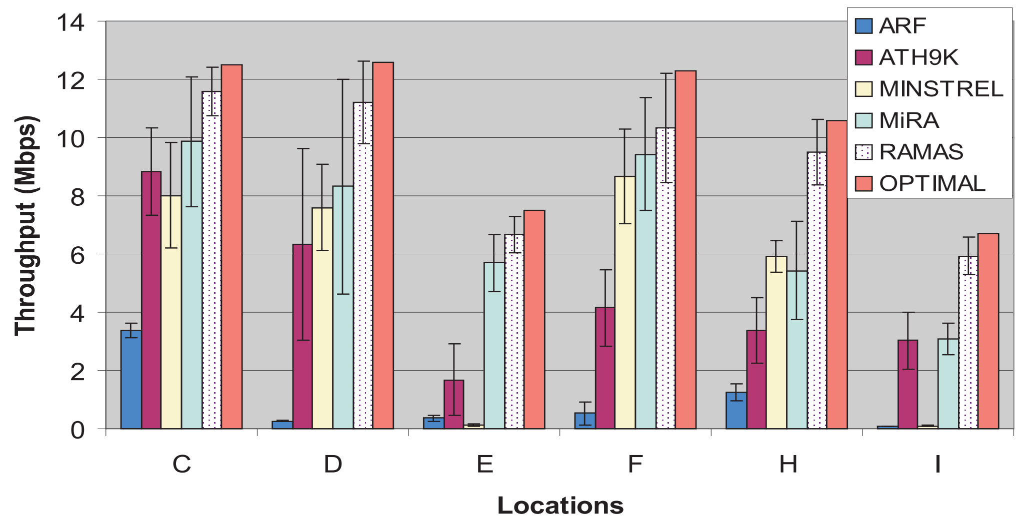


(b) 2x2 MIMO High Throughput Locations

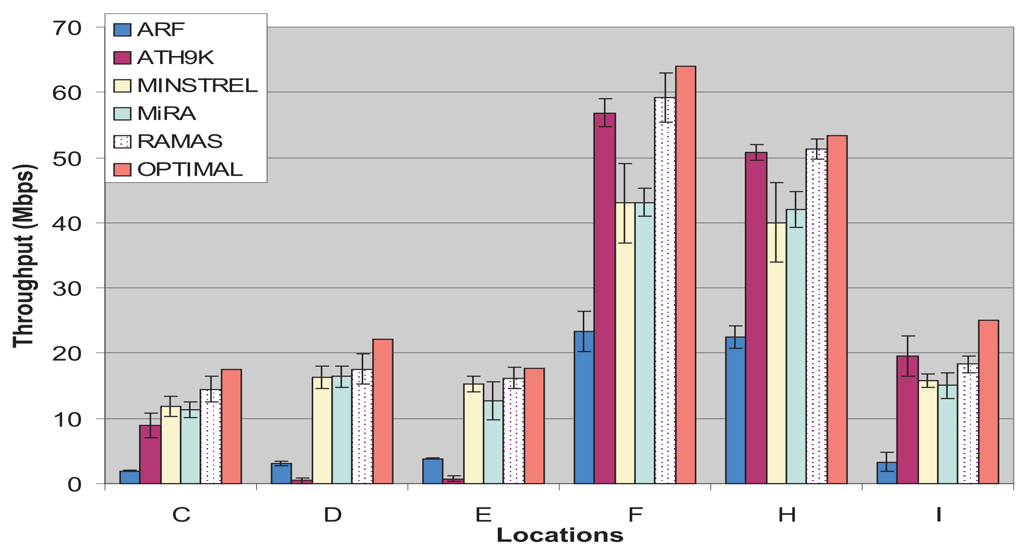


(c) 3x3 MIMO High Throughput Locations

Figure 6.5: Experiments with Different Locations Around Our Engineering Building II



(a) 2x2 MIMO Low Throughput Locations



(b) 3x3 MIMO Low Throughput Locations

Figure 6.6: Experiments with Different Locations Around Our Engineering Building II

multiple streams and single stream. Obviously, for single stream or complete transmit diversity, there is no need for stream error detection or further downgrading enhancement index since it is already at the lowest.

a) For multiple streams, RAMAS checks the following:

- If there are more errors than ACK packets during AMPDU transmission, one or more of multiple streams must be having problems transmitting packets. RAMAS relies on the AMPDU length α and block ACK length β for checking this stream error by comparing unACKed packets $\alpha - \beta + \rho/2$ and ACKed packets β. If this is the case, RAMAS increments the stream error counter ε by 1. Otherwise, it resets it to 0. Note that RAMAS considers two retransmitted packets ρ as one error because too many retransmissions can have adverse impact on performance.
- Given that stream errors can come and go, RAMAS should not act prematurely based on one single instance of a stream error. For this reason, a stream failure threshold Φ is implemented to ensure its persistence before downgrading the enhancement index and reducing the number of streams.
- Upgrading the enhancement index is based on the packet delivery ratio. Given that stream errors create more error packets than good packets, RAMAS relies on a high delivery ratio of Ψ to ensure stability for upgrading enhancement index.

b) For a single stream, or complete transmit diversity, RAMAS applies the same delivery ratio method as mentioned above for upgrading index.

The exact method to increase and decrease index for enhancement group is described in Algorithm 5

6.2.2.3 Guideline for Setting Parameters

Stream Error Failure Threshold is used to ensure that the stream error persists before I downgrade the number of streams. Figure 6.3 shows the stream error threshold and its corresponding throughput. Setting the stream error threshold to 1 suffers a great throughput degradation compare to other choices. The same applies to other extreme end of setting it to 15 since it is waiting too long to downgrade index.

In Algorithm 5, notice that I decrement the enhancement index instead of lowering it to the corresponding stream. Knowing that the stream is corrupted, the objective is to see if we can try anything else before resorting to complete transmit diversity, because transmit diversity yields the lowest throughput. This is why our algorithm begins by relaxing the guard interval, and channel width as well as lowering the number of streams before falling back to transmit diversity.

Stability plays an important role to adapting enhancement group. The objective is to ensure that there are no more stream errors before upgrading it to higher number of streams. In my experiments, I find that a high delivery ratio provides good indicator for upgrading enhancement index. If I set the acceptable delivery ratio too low, the protocol tend to encounter stream error again much sooner. Our recommended value for acceptable delivery ratio is 90% over the same recommended w sample window.

6.2.3 Mapping Modulation Group Index and Enhancement Group Index to MCS Index

I adapt modulation group and enhancement group concurrently and simultaneously. Each comes with its own rules and methods. These indices together are mapped back to the MCS index for adapting rates in IEEE 802.11n. Table 6.3 shows mapping to MCS index for up to four antennas even though our current software and hardware only support up to 3 antennas or MCS indices up to 23. For example, MCS Index 23 $M7 :: E8 - E11$ refers to a modulation index of 7 (or $64 - QAM$) and enhancement group index of 8 to 11 in which these features' availability is depending on hardware support).

Figure 6.4 shows the performance decomposition of RAMAS. First, I modified RAMAS with no Binomial-like credit system by setting parameter τ_γ to 1. With no credit system, RAMAS performs erratically. Second, I remove stream error detection and use only delivery ratio for upgrading and downgrading enhancement index. I observe that without stream error detection, the throughput with RAMAS drops sharply. Finally, with both credit system and stream error detection, RAMAS performs robustly.

6.2.4 Performance Evaluation

Aside from comparing our protocol against two widely used and popular open-source rate adaptation ATH9K [4] and MINSTREL High Throughput [17], we implemented a modified ARF [41] and MiRA [58] for more rigorous baseline comparisons. I modified MiRA algorithm to have it zigzag through an extra triple streams for the 3x3 MIMO experiment.

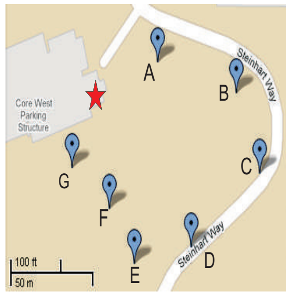
Our experiments include wireless cards and router with Atheros wireless chipsets AR9380, AR5416+AR2133, AR5418+AR5133. I implemented RAMAS using open-source software from the Linux Kernel Wireless Stack and using the Wireless-Testing Git Tree [49]. Our implementation of RAMAS in the Linux Kernel Wireless Stack allows it to run on many different chipsets and independent of any specific hardware vendors.

All of the experiments are conducted in the 2.4 Ghz frequency due to its long range and being more suitable for many of our experiments, which range from forest settings to car mobility. All of our experiments are set to Greenfield mode so that only devices with IEEE 11n capability can participate. Because of the nature of real-world experiments, I run at least 7 repetitions for each of our data point and provide error bars. Our experiment scenarios include indoor fading, outdoor fading, limited mobility, and interference collisions. I provide the best or optimal MIMO throughput for indoor fading locations where I cycle through different MIMO configurations. I will discuss each scenario and its result in detail next.

6.2.4.1 Indoor Fading Scenario

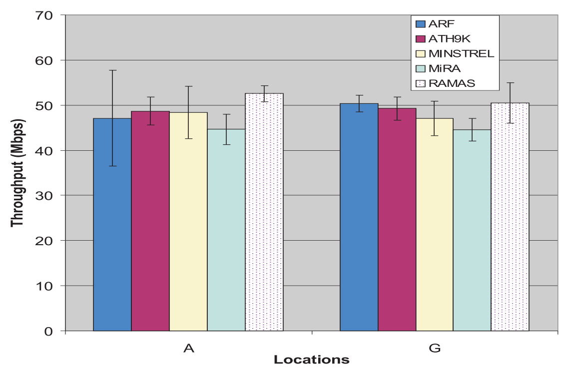
Setup I place an AP router at a far end corner of the building shown as a star symbol in Figure 6.5(a); then, I measure throughput at different locations. It is atypical to place the AP router near one end of the building, but our goal is to see the maximum coverage of 802.11n router. I vary AP router with 2 and 3 antennas as well as clients with 2 and 3 antennas.

Indoor Fading Results Figure 6.12 reports the result for various locations around our building. On one side of the building from location A to E, AP router does not have any problems extending coverage as opposed to location F to I where there are multiple thick walls.

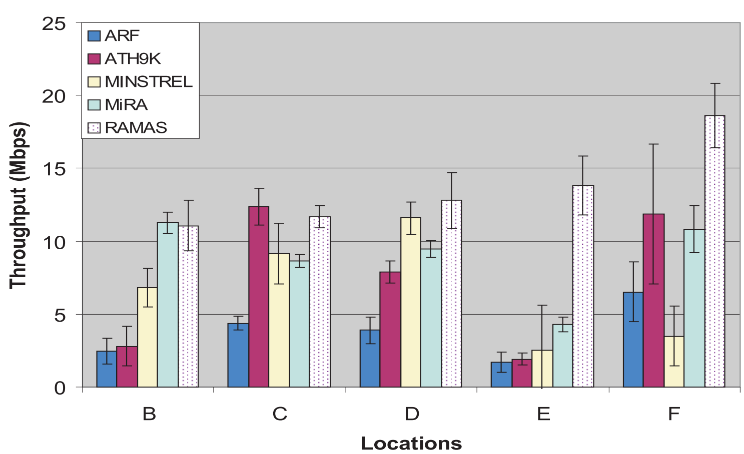


(a) GoogleMaps

(b) Redwood Forest

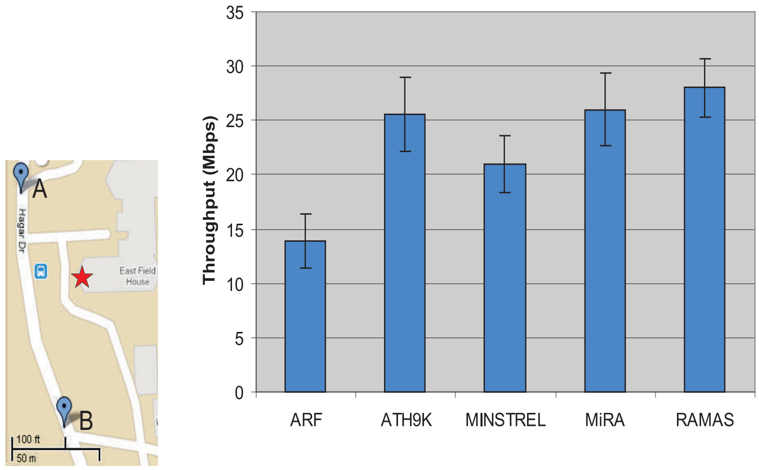


(c) High Throughput Locations



(d) Low Throughput Locations

Figure 6.7: Outdoor Redwood Forest Experiments



(a) GoogleMaps

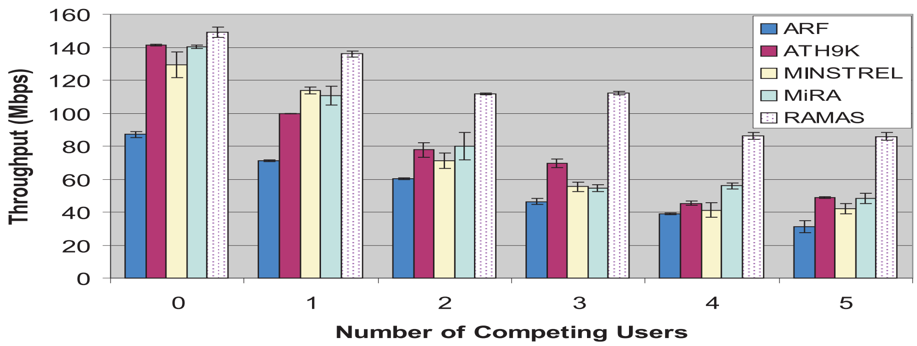
(b) Throughput

Figure 6.8: Mobility 20mph (32km/h) from point A to B

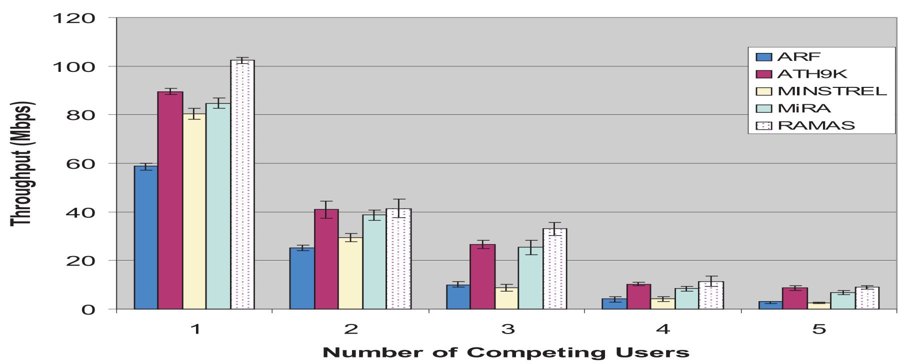
The result for 2x2 MIMO scenario in which both client and AP equipped with two antennas is shown in Fig 6.5(b) and Figure 6.6(a). I separate the results into high- and low-throughput locations for readability. ARF generally suffers from throughput degradation due to its conservative nature and the requirement of consecutive successes. I find that its rates tend to get stuck at lower throughputs.

ATH9K's performance is unpredictable and its throughput tends to depend on certain locations. For example, at location G, ATH9K performs better than MINSTREL and MiRA, and it performs exceptional well at location I (the farthest distance from the AP). At location D and E, its throughput fluctuates greatly. At a few occasions, I find that ATH9K becomes confused and ends up transmitting at Kbps. This unpredictable performance can be attributed to the way ATH9K samples random rates. It tends to keep using the rates that offer reasonable throughput, and it often fails to explore optimal rates.

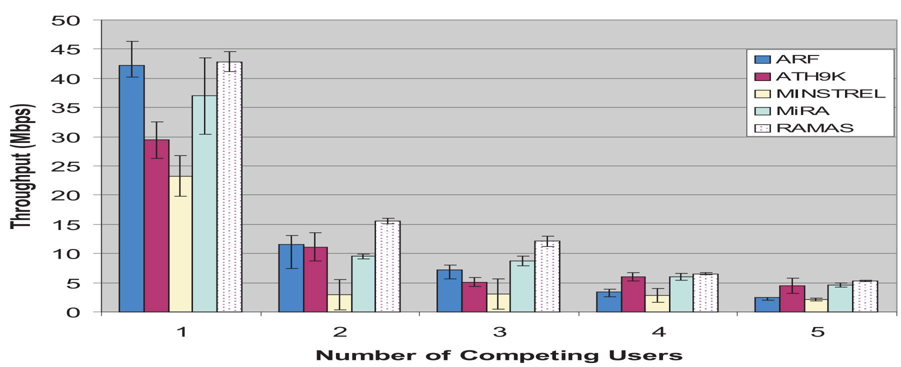
MINSTREL and MiRA have similar performance at locations near AP but MiRA performs better than MINSTREL at other locations. Both MINSTREL and MiRA sample many rates and sort through these sampled rates to find the best throughput. MiRA is able to perform better at long distance fading because it is able to switch to other mode appropriately.



(a) Users sending 10Mbps



(b) Users sending 50Mbps



(c) Users sending 100Mbps

Figure 6.9: Multi-user Experiments

However, MiRA's performance still trails that of RAMAS. I find that MiRA resorts to sampling every time its throughput slightly fluctuates. At times, it tends to sample other rates when its current rate is working perfectly fine since it needs to make sure that the current rate provides better throughput.

Figure 6.5(c) and Figure 6.6(b) repeats the same experiment around the building but for 3x3 MIMO where both clients and AP are equipped with three antennas, for a maximum of three spatial streams. For this scenario, I had to modify MiRA to make it support triple streams by zigzagging through another stream space. Because of the extra space that MiRA has to sample, in addition of frequent sampling due to small changes in throughput, it still does not perform as well a s RAMAS.

RAMAS succeeds because it adapts rates in an orderly fashion instead of using random sampling. In the presence of minor errors or fluctuations, RAMAS lowers the number of spatial streams and modulation types gradually and appropriately to harvest the best throughput. Taking into account of the margin of errors, RAMAS's performance is close to the best throughput when we cycle through different MIMO configurations.

6.2.4.2 Outdoor Fading Scenario

Setup Apart from indoor scenario, I carried our experiments out into the redwood forest due to its rich dynamic of multiple path wireless link in an area of 330x330 feet square (100x100 meter square) with varying degree redwood tree density, slopes, and steeps. I place our AP router on the edge of a parking garage structure overlooking the forest (marked as a star symbol in Figure 6.7(a)).

Outdoor Fading Results Figure 6.7 reports the results for outdoor experiments performed at various location around the 330x330 feet square area (100x100 meter square) of the redwood forest. At location A and G near AP, all protocols have similar performance. At location B to F where there is a varying degree of fading distance, density of the redwood forest, and elevations, I find varied performance among protocols.

ATH9K performs well in location C and F but does not fare well at other locations. MINSTREL and MiRA have similar performance except at location F where MiRA is able to perform much better. RAMAS performs much better than the other protocols at location E and F where there are steep elevations, dense redwood trees, and no visible AP in sight. At these locations, I observe that MINSTREL and MiRA resorts to sampling very often due to the dynamic changing of multi-path links. RAMAS, on the other hand, maintains transmit diversity and only varies modulation types. This component of RAMAS gives it an edge over all other protocols.

6.2.4.3 Mobility Scenario

Setup Mobility is an important requirement in wireless networks. Therefore, I set up a limited mobility scenario on our campus. I test mobility with speed limit of 20 mph (32 km/h), driving from point A to point B as shown in Figure 6.8(a). I could not drive too slow or too fast due to safety on public streets.

Mobility Results Figure 6.8(b) shows the result for a limited mobility scenario with mobile speed of 20 mph (32 km/h). Overall, all protocols have similar performance. I cannot conclude much except to defer it for further research in the future.

6.2.4.4 Interference Collisions

Setup Interference collision is very common in wireless environment. Because of the scarce resources, many nodes have to compete for access to the medium or share the medium with other no des. I have multiple nodes sending traffic to the target nodes on AP router. First, I set up a transmission between two nodes. Then I vary the number of interferers from 1 to 5. These interferers, in turn, vary their sending rates using 10, 50, and 100 Mbps. Finally, I create a scenario where all five interferers assume a different sending traffic rate from 10, 30, 50, 70, and 100 Mbps. These nodes serve both as interferers and users competing for resources. The objective is to see how protocols perform in an interference collision and multi-user environment where other nodes are competing for access to AP router.

Interference Collisions Results Figure 6.9 reports the result for the interference collision and multi-user scenario where multiple nodes are sending traffic to AP . In Figure 6.9(a), we gradually add the number of interferers where each sending a fixed traffic of 10 Mbps to AP. When there are no interferers, ATH9K, MINSTREL, MiRA have similar performance. As I increase the number of interferers competing for resources at AP, the performance of ARF, ATH9K, MINSTREL and MiRA degrade drastically. I observe that these protocols responded prematurely to small traffic disturbance. RAMAS has no problem recognizing this small traffic disturbance and is able to perform significantly better than all other protocols.

Figure 6.9(b) and Figure 6.9(c) repeat the same experiments but have each interferers sending a higher fixed traffic of 50 Mbps and 100 Mbps respectively. With the fourth and fifth added interferers, RAMAS's performance gains become less significant. This is because the interference collision is so great that none of the rate adaptations can work normally.

Figure 6.10 present a scenario where I have five interferers sending five different traffic rates to AP with 10, 30, 50, 70, and 100 Mbps. Again, I observe that all protocols tend to suffer with this interference collision scenario where there are many users competing for resources with varying degree of traffic among the interferers. RAMAS performs significantly better in this scenario. I observe that most protocols fail this scenario because they do not consider the impact of multiple users on rate adaptations.

Figure 6.11 reports Jain's fairness index [37] for multi-user scenario. In our initial test, I find that nodes with more processing power tend to grab the biggest share of the bandwidth, i.e. laptop versus desktop. As a result, I provide the same machine architecture for all nodes. For two and three competing users, all protocols have similar fairness index. As I increase to four competing users, MINSTREL's fairness becomes unpredictable. It is not surprising that ARF performs as well as RAMAS in terms of fairness because, like RAMAS, ARF increases its rate index incrementally and progressively. of users for our future work.

6.2.5 Summary

I presented RAMAS, a multi-rate adaptation approach for multi-antenna systems by separating it into two groups, namely a modulation group and an enhancement group. These groups are adapted concurrently and their indices are mapped back to an MCS index. This mapping allows us to adapt rates in an orderly fashion instead of chaotic random sampling. I

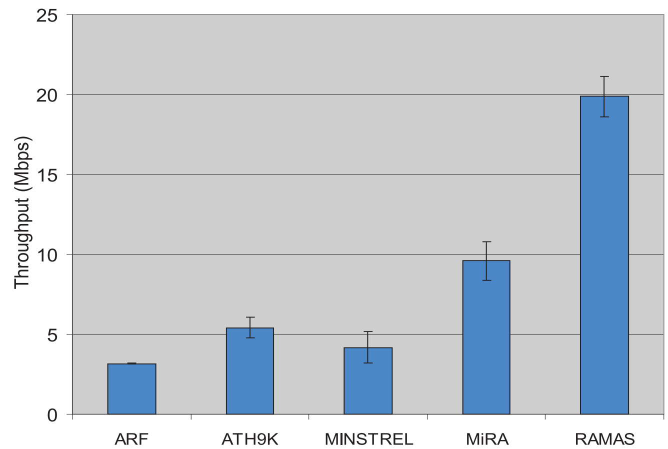


Figure 6.10: Multi-user Experiments: Mixed Users with 10, 30, 50, 70, and 100 Mbps

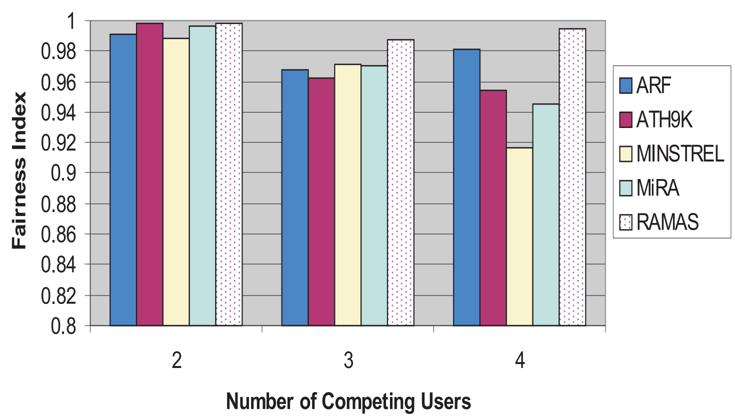


Figure 6.11: Jain's Fairness Index

implemented the scheme using open-source software and carried out an extensive performance evaluation. In the next section, I will address an improvement to RAMAS by using a throughput-centric approach.

6.3 Throughput-centric Rate Adaptation for MIMO Systems (TRAMS)

Due to the difficulty in tuning different parameters in RAMAS, a new approach is needed in order to make it easier to facilitate rate selection. I adopt RAMAS [11] approach by separating IEEE 802.11n rate adaptation into two groups: a modulation group and an enhancement group. These two groups are adapted concurrently. The difference is that TRAMS replaces RAMAS with a throughput-centric design that provides efficiency, stability, and reactiveness. I adapt between the multiplexing and transmit diversity by relying on a simple observation that bit symbols are spread equally among antenna in spatial multiplexing. If the number of un-acknowledged packets is greater than the number of ACKs in block ACK packets, I consider switching it to diversity. When this occurs, it means that one or more of the multiple antennas is not doing very well.

Like RAMAS, TRAMS rate adaptation will be divided into two groups the modulation group and the enhancement group. Each group will be adapted concurrently based on its own policy. The only difference is that TRAMS bases its modulation group on throughput enabled TERA instead of the packet approach MAICA. As a result, I will only show the algorithms for modulation group and enhancement group.

6.3.1 Performance Evaluation

I compare TRAMS against RAMAS [11] and two widely used and popular open-source rate adaptation ATH9K [4] and MINSTREL High Throughput [17].

Our experiments include wireless cards and router with Atheros wireless chipsets AR9380, AR5416+AR2133, AR5418+AR5133. I implemented TRAMS using open-source software from the Linux Kernel Wireless Stack and using the Wireless-Testing Git Tree [49]. Our

Algorithm 6 TRAMS Modulation Group Adaptation

```
ω = time window
ω_probinglow = probe less because channel is bad (900ms)
ω_probinghigh = probe more because channel is good (100ms)
idx = the rate index as shown in Table 6.1.
check(ω) = check whether time window ω is expiring
Multiplicative = successive successful probes
Oscillate = rates oscillate in a see-saw fashion state
Γ_(t−1) = previous throughput
prev_idx = previous rate index
while check(ω) do
   if Probing then
      if Γ < Γ′ and idx! = prev_idx then
         idx = prev_idx
         if Multiplicative then
            ω = ω_probinghigh
         else
            ω = ω_probinglow
         end if
      else if Γ > Γ_(t−1) and idx! = prev_idx then
         ω = ω_probinghigh
      end if
      return
   end if
   if Δ ≥ 1 then
      // Multiplicative Increase
      if Multiplicative and !Oscillate then
         if idx + idx < max_idx then
            idx = idx + idx
         else
            idx = max_idx − 1
         end if
         // Additive Increase
      else if !Oscillate then
         if idx + 1 < max_idx then
            idx + +
         end if
      end if
      Probing = true
   else if Δ ≤ 0.90 and Δ ≥ 0.75 then
      // Additive Decrease
      if idx > 0 then
         idx − −
      end if
   else if Δ < 0.75 then
      // First occurrence: Additive Decrease
      if first and idx > 0 then
         idx − −
         // Successive occurrences: Multiplicative Decrease
      else if second and idx > 0 then
         idx ← idx ∗ M_D
      end if
   end if
end while
```

Algorithm 7 TRAMS Adapting Enhancement Group

```
e_index = enhancement group index as as seen Table 6.2
α = Aggregate MPDU length
β = Aggregate MPDU ACKs length
Φ = stream failure threshold
Ψ = packets delivery ratio for upgrading
η = number of streams;
ε = stream errors transmission
skip20Mhz = to skip 20Mhz channel when upgrading
//comment: for multiple streams
if (η > 1) then
    //comment: check and register stream errors
    if (α − β > β) then
        ε++
    else
        ε = 0
    end if
    //comment: making sure stream errors still persist
    if (ε > Φ) then
        e_index = e_index − 1
        ε = 0
    end if
    //comment: for upgrading index
    if (success/attempts > Ψ) then
        e_index = e_index + 1
    end if
end if
//comment: for single stream
if (η == 1) then
    if (success/attempts ≥ Ψ) then
        e_index = e_index + 1
    end if
    if (success/attempts < 1 − Ψ) then
        e_index = e_index − 1
        skip20Mhz = false
    else
        skip20Mhz = true
    end if
end if
```

implementation of TRAMS in the Linux Kernel Wireless Stack allows it to run on many different chipsets and independent of any specific hardware vendors.

All of the experiments are conducted in the 2.4 Ghz frequency due to its long range and being more suitable for many of our experiments. All of our experiments are set to Greenfield mode so that only devices with IEEE 11n capability can participate.

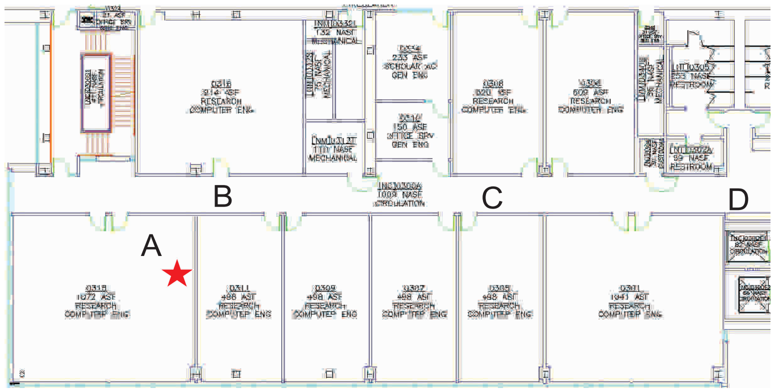
Because of the nature of real-world experiments, I run at least 7 repetitions for each of our data point and provide error bars. Higher error bars mean that the performance of a scheme tends to be erratic and fluctuates greatly. Our experiment scenarios include fading and interference collisions, which are the most common scenarios in the real-world settings. I will discuss each scenario and its result in detail next.

6.3.1.1 Fading Scenario

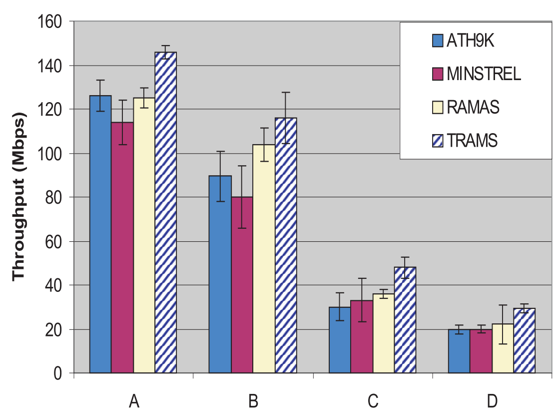
I set up our wireless access point in our laboratory, shown as a star symbol in Figure 6.12(a). I measure throughput at different locations around our building. Figure 6.12(b) reports the result for various locations around our building with 3x3 MIMO where both clients and AP are equipped with three antennas, for a maximum of three spatial streams. At locations A and B, TRAMS performs better than all other approaches for the case of close proximity. In this close range, TRAMS picks the highest modulation group and the highest enhancement group and keeps using it, whereas ATH9K and MINSTREL tends to sample other rates that are detrimental to their performances. ATH9K performs better than MINSTREL at location A and B, but falls behind other schemes at location C and D. ATH9K tends to keep using the rates that offer reasonable throughput, and it often fails to explore optimal rates.

MINSTREL performs well at far-away locations such as C and D but only obtains marginal throughput compared to others at locations A and B. MINSTREL samples many rates and sort through these sampled rates to find the best throughput. At times, it tends to sample other rates when its current rate is working perfectly fine since it needs to make sure that t he current rate provides better throughput.

TRAMS is able to maintain stable throughput throughout. RAMAS does not perform as well as TRAMS because of its progressive rate increase through credit-based systems. Because



(a) Map of Engineering Building



(b) 3x3 MIMO Locations

Figure 6.12: Experiments with Different Locations Around Our Engineering Building

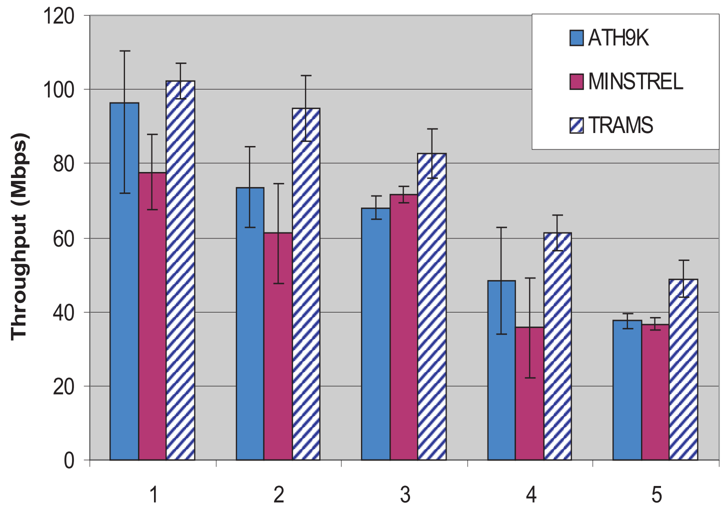


Figure 6.13: Multi-user: Number of users sending 10 Mbps

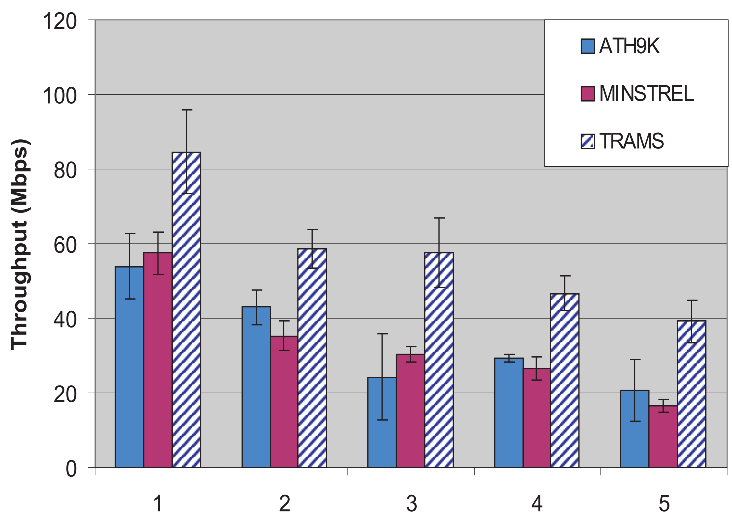


Figure 6.14: Multi-user: Number of users sending 50 Mbps

of the structure of the enhancement group in Table 6.2, RAMAS tends to sample the 20Mhz channel width even though its 40Mhz channel is working perfectly. TRAMS fixes this problem by allowing rates to bypass this channel width.

TRAMS succeeds because it adapts rates in an orderly fashion instead of using random sampling. In the presence of minor errors or fluctuations, TRAMS lowers the number of spatial streams and modulation types gradually and appropriately to harvest the best throughput. Given that TRAMS is a better approach than RAMAS, the rest of the subsequent scenarios and experiments do not include RAMAS.

6.3.1.2 Interference Collisions

Interference collision is very common in wireless environment. Because of the scarce resources, many nodes have to compete for access to the medium or share the medium with other nodes. The performance in this multi-user environment is an important component for evaluation of rate adaptations, which I find lacking in many recent IEEE 802.11n rate adaptations.

I have multiple nodes sending traffic to the target nodes on AP router. First, I set up a transmission between two nodes. Then I vary the number of interferers from 1 to 5. These interferers, in turn, vary their sending UDP traffic of 10 and 50 Mbps. Finally, I create a scenario where all five interferers assume a different sending traffic rate from 10, 30, 50, 70, and 100 Mbps. These nodes serve both as interferers and users competing for resources. The objective is to see how the various schemes perform in an interference collision and multi-user environment where other nodes are competing for access to AP router.

Figure 6.13 reports the result for the interference collision and multi-user scenario where multiple nodes are sending traffic to AP. We gradually add the number of interferers where each is sending a fixed traffic UDP traffic of 10 Mbps to AP. When there are 1 interferer or 2 interferers, I find that the performance of ATH9K and MINSTREL fluctuates greatly. This is due to the fact that these approaches are sampling different rates but could not decide which one is optimal. As I increase the number of interferers competing for resources at AP, the performance of all approaches degrade but more drastically in ATH9K and MINSTREL. I

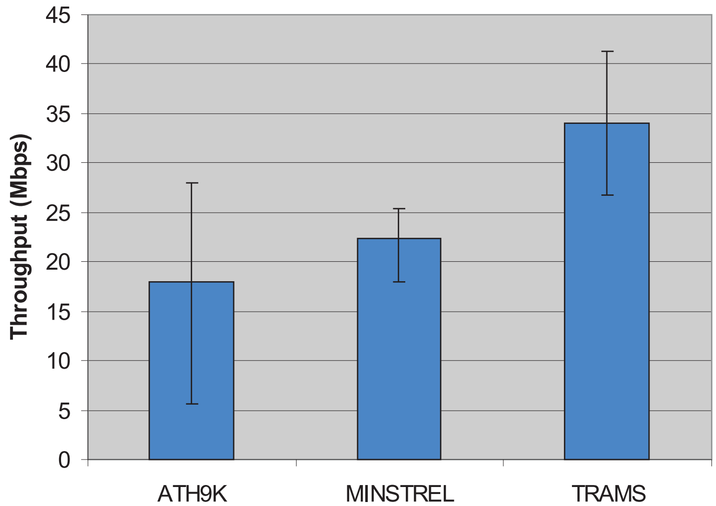
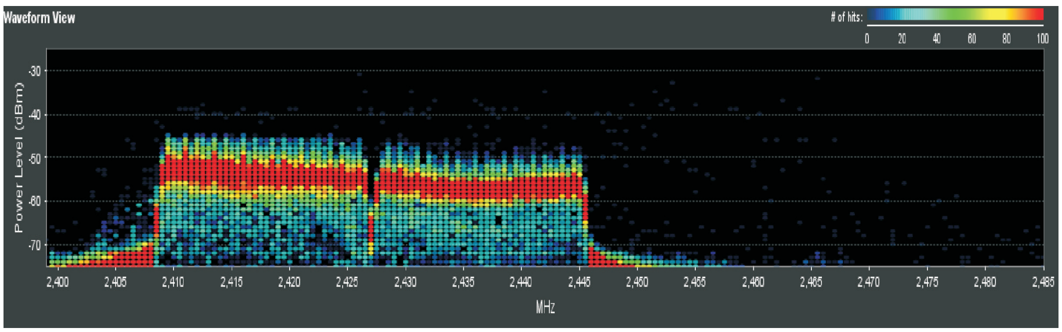


Figure 6.15: Multi-user: Mixed Users with 10, 30, 50, 70, and 100 Mbps Traffic

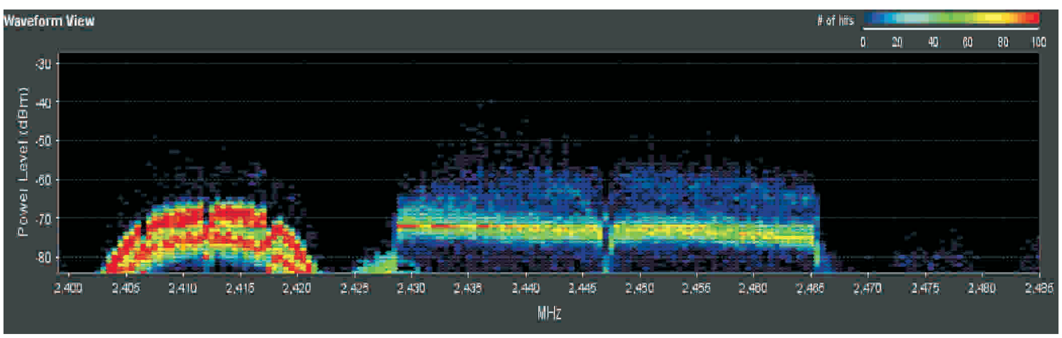
observe that these schemes responded prematurely to small traffic disturbance. TRAMS has no problem recognizing this small traffic disturbance and is able to perform better than all other schemes with very stable performance.

Figure 6.14 repeats the same experiments with each interferer generating 50Mbps. I have tried traffic higher than 50Mbps but the results are the same after 50Mbps UDP traffic. In this scenario, TRAMS performs significantly better than all schemes. About 30 percent better than ATH9K and MINSTREL with 1 or 2 interferers and almost 50 percent better with 3 to 5 interferers involved.

Figure 6.15 presents a scenario where I have five interferers sending five different traffic rates to AP with 10, 30, 50, 70, and 100 Mbps. This scenario is very common in the real world where many users have different traffic needs. Again, I observe that all schemes tend to suffer with this interference collision scenario where there are many users competing for resources with varying degree of traffic among the interferers. TRAMS performs significantly better in this scenario. I observe that most schemes fail this scenario because they do not consider the impact of multiple users on rate adaptations.



(a) All interferers on the same channel during interference experiment



(b) All interferers on another channel during interference experiment

Figure 6.16: Spectrum Views

6.3.2 Summary

I presented TRAMS, a novel and simple throughput-centric rate adaptation approach for MIMO systems. TRAMS simplifies the complexity of rate adaptation in IEEE 802.11n by separating it into two groups, namely a modulation group and an enhancement group. TRAMS addresses the shortcoming of RAMAS by replacing its modulation group adaptation with a throughput enabled rate adaptation. These groups are adapted concurrently and their indices are mapped back to an MCS index. Simplicity is attained by using only implicit feedback. I implemented our scheme using open-source software and carried out an extensive performance evaluation including fading experiments to heavy interference collision with many users competing for resources. The results show that TRAMS consistently performs better than all the well-known prior schemes for rate adaptation in multi-antenna systems.

Chapter 7

Conclusion and Future Work

In this dissertation, I addressed the bandwidth demand in wireless networks by proposing new approaches for accessing the medium more efficiently as well as making the best use of channel access time through an efficient and robust rate adaptation.

I propose a collision-free multi-channel medium access control (MAC) that does not require tight synchronization through a asynchronous split phase and an observation phase. Each node has the freedom to follow its own schedule as long as it adheres to a few sets of rules such as observation phase and a unique three-way handshakes. The observation phase ensures that node has enough time to observe the surrounding before selecting a channel for rendezvousing.

Multiple channels inherit all the problem of the single channel MAC. The hidden terminal problem now extends to multiple channels. I include an additional control packet Announce to Send (ATS) packet to inform neighbors of channel selection. ATS packets works by broadcasting the schedules among nodes within its range. Very minimal changes the current standard IEEE 802.11 is required for supporting the addition of ATS packets as well as changing the duration of CTS and ATS to be longer than RTS to avoid scheduling conflicts.

I also propose two new approaches for node to take the best advantage of its channel access time: packet error rate approach and throughput enabled approach. Packet error approach relies only on ACKs and packet errors. Rate performance is measured through a time window and packet window. At the conclusion of each window, the rate is adjusted accordingly.

The disadvantage is that there are many parameters to tune.

In the throughput enabled approach, throughput of the rate is monitored over a period. Rate is adapted based on its performance. This approach is more desirable because it takes into account of channel interference and errors implicitly.

The complexity of rate adaptation increases with multiple input and multiple output technology (MIMO). Rate adaptation now must decide between many different rate indices instead of simply a dozen ones as in SISO systems. Simply, we need to pick the best rate as quickly as possible with minimal overhead. Because of the complexity of MIMO, I separate it into two groups, namely, a modulation group and an enhancement group. Each group has its own policy for adapting its own indices before being mapping it back to MCS. For future work, one may explore whether it may be more beneficial to divide the MCS into other groups or other dimensions and to improve the policies for adapting the current modulation group and enhancement group.

Rate adaptation is still an open and challenging problem in the community. Ideally, we could use channel state information (CSI) for adapting rates accordingly. However, CSI may be susceptible to short channel coherence time and signaling overhead. Different measurements of CSI from different vendors may make it hard to map to the correct rate uniformly. One improvement could be made by applying statistical learning methods to rate adaptation in order to guide it in selecting the optimal rate or to ensure that it learns from its past mistakes.

I have presented current and challenging problems in improving the bandwidth in wireless networks. I have shown that the channel access scheme and its rate adaptation component can have a great impact on wireless networks. My proposed solutions ensure that node have access to the medium at a higher throughput while, at the same time, making the best use of its channel access time by maintaining the highest attainable throughput. All of the work presented in this dissertation from simulations to prototype implementation is made available on github [55].

Bibliography

[1] A. Adya, P. Bahl, J. Padhye, A.Wolman, and L. Zhou. A Multi-Radio Unification Protocol for IEEE 802.11 Wireless Networks. In *In IEEE International Conference on Broadband Networks(Broadnets)*, 2004.

[2] S. M. Alamouti. *A simple transmit diversity technique for wireless communications.* IEEE J. Select. Areas Commun, vol. 16, 1998.

[3] D.G. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris. Resilient overlay networks. In *Proc. SOSP, 2001.*

[4] ATHEROS Communications. `http://www.atheros.com`.

[5] P. Bahl, R. Chandra, and J. Dunagan. SSCH: Slotted Seeded Channel hopping for capacity improvement in ieee 802.11 adhoc wireless networks. In *In Proc. of ACM MobiCom*, 2004.

[6] G. Bianchi. *Performance Analysis of the IEEE 802.11 Distributed Coordinated Function.* IEEE J. Selected Areas in Commun., vol. 18, no. 3, 2000.

[7] D-M Chiu and R. Jain. *Analysis of the increase and decrease algorithms for congestion avoidance in computer networks.* Computer Networks and ISND Systems, 1989.

[8] J. Choi, K. Park, and C. Kim. Cross-Layer Analysis of Rate Adaptation, DCF and TCP in Multi-Rate WLANs. In *IEEE INFOCOM*, 2007.

[9] D. E. Comer. *Internetworking with TCP/IP: Principles, Protocols, and Architecture.* Prentice Hall, 2006.

[10] D. De Couto, D. Aguayo, J. Bicket, and R. Morris. A high-throughput path metric for multi-hop wireless routing. In *Proc. Mobicom 2003*.

[11] D. Nguyen and J.J. Garcia-Luna-Aceves. *A Practical Approach to Rate Adaptation for Multi-Antenna Systems.* Proc. IEEE ICNP, 2011.

[12] D. Nguyen and J.J. Garcia-Luna-Aceves and C.Westphal. *Multi-Rate Adaptation with Interference and Congestion Awareness.* Proc. IPCCC, Orlando, Florida, 2011.

[13] D. Nguyen and J.J. Garcia-Luna-Aceves and C.Westphal. *Throughput Enabled Rate Adaptation.* Proc. International Conference on Computing, Networking and Communications, San Diego, California, 2012.

[14] Debian Linux. `http://www.us.debian.org/`.

[15] L. Deek, E. Garcia-Villegas, E. Belding, S. Lee, and K. Almeroth. The Impact of Channel Bonding on 802.11n Network Management. In *CoNEXT*, Tokyo, Japan, December 2011.

[16] R. Draves, J. Padhye, and B. Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *Proc. Mobicom 2004*.

[17] F. Fietkau. linuxwireless.org/en/developers.

[18] G. J. Foschini and M. J. Gans. *On Limits of Wireless Communications in a Fading Environment When Using Multiple Antennas.* Wireless Personal Communications, 1998.

[19] C. Fullmer and J.J. Garcia-Luna-Aceves. Floor Acquisition Multiple Access for Packet-Radio Networks. In *Proc. ACM SIGCOMM*, Cambridge, MA, August 28-September 1 1995.

[20] C. Fullmer and J.J. Garcia-Luna-Aceves. Solutions to Hidden Terminal Problems in Wireless Networks. In *In Proceedings of ACM SIGCOMM*, 2007.

[21] Chane L. Fullmer. Phd thesis: Collision avoidance techniques for packet-radio networks.

[22] J.J. Garcia-Luna-Aceves, Hamid R. Sadjadpour, and Zheng Wang. Challenges: Towards truly scalable ad hoc networks. In *In Proc. ACM MobiCom*, 2007.

[23] S. Gatreux, V. Erceg, D. Gesbert, and R. W. Heath. *Adaptive Modulation and MIMO Coding for Broadband Wireless Data Networks*. IEEE Communications, 2002.

[24] M. Grossglauser and D. Tse. Mobility Increases the Capacity of Ad Hoc Wireless Networks. *IEEE/ACM Transactions on Networking*, 10(4):477–486, August 2002.

[25] P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388–404, 2000.

[26] D. Halperin, W. Hu, A. Sheth, and D. Wetherall. Predictable 802.11 Packet Delivery from Wireless Channel Measurements. In *In Proc. of SIGCOMM*, New Delhi, India, 2010.

[27] R. W. Heath and A. J. Paulraj. *Switching Between Diversity and Multiplexing in MIMO Systems*. IEEE Transactions on Communications, 2005.

[28] Hidden Terminal. `http://en.wikipedia.org/wiki/Hidden_node_problem`.

[29] G. Holland, N. Vaidya, and P. Bahl. A Rate-Adaptive MAC Protocol for Multi-Hop Wireless Networks. In *Proc. ACM MOBICOM*, Rome, Itaty, 2001.

[30] IEEE 802.11 Working Group. *Wireless LAN Medium Access Control(MAC) and Physical Layer(PHY) specifications*, 2007.

[31] IEEE 802.11a Working Group. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications - Amendment 1: High-speed Physical Layer in the 5GHz band*, 1999.

[32] IEEE 802.11b Working Group. *Wireless LAN Medium Access Control(MAC) and Physical(PHY) Layer Specification: High Speed Physical Layer Extensions in the 2.4 Ghz Band*, 1999.

[33] IEEE 802.11g. *Further Higher Data Rate Extension in the 2.4 Ghz Band*, 2003.

[34] IEEE 802.11n Working Group. *Enhancements for Higher Throughput*, 2009.

[35] V. Jacboson. Congestion Avoidance and Control. In *Proc. of SIGCOMM '88*, Palo Alto, CA, 1988.

[36] M. Jain, J. Choi, T. Kim, D. Bharadia, K. Srinivasan, S. Seth, P. Levis, S. Katti, and P. Sinha. Practical, Real-time, Full Duplex Wireless. In *In Proc. ACM Mobicom*, 2011.

[37] R. Jain, D-M Chiu, and W. Hawe. *A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems*. DEC Research Report TR-301, 1984.

[38] M. Jankiraman. *Space-Time Codes and MIMO Systems*. Artech House, 2004.

[39] J.C Bicket. *M.S thesis: Bit-rate selection in wireless networks*. MIT Press, 2005.

[40] G. Judd, X. Wang, and P. Steenkiste. Efficient Channel-aware Rate Adaptation in Dynamic Environments. In *Proc. of the ACM MobiSys*, June 2008.

[41] A. Kamerman and L. Monteban. *WaveLAN-II: A High-performance wireless LAN for the unlicensed band*. Bell Lab Technical Journal, 1990.

[42] S. Keshav. A control-theoretic approach to flow control. In *ACM SIGCOMM 1991*.

[43] A. Khanna and J. Zinky. The revised arpanet routing metric. In *Proc. SIGCOMM, 1989*.

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

[45] Wonsoo Kim, O. Khan, K. T. Truong, S. Choi, R. Grant, H. K. Wright, K. Mandke, R. C. Daniels, R. W. Heath, and S. M. Nettles. An Experimental Evaluation of Rate Adaptation for Multi-Antenna Systems. In *IEEE INFOCOM*, 2009.

[46] M. Lacage, M. H. Manshaei, and T. Turletti. *IEEE 802.11 rate adaptation: A practical approach*. INRIA Research Report.

[47] J Lee, J Mo, T. M. Trung, J. Walrand, and H-S.W So. WiFlex: Multi-Channel Cooperative Protocols for Heterogeneous Wireless Devices. In *Wireless Communications and Networking Conferences(WCNC)*, pages 1489–1494, March 31 - April 3 2008.

[48] Alberto Leon-Garcia. *Probability and Random Processes for Electrical Engineering*. Addison-Wesley Publishing Company, 1994.

[49] Linux Wireless Stack. `http://wireless.kernel.org/`.

[50] Minstrel Linux Wireless. http://linuxwireless.org/en/developers/Documentation/mac80211/RateControl/minstrel.

[51] J. Mo, H-S W. So, and J. Walrand. Comparison of multi-channel MAC protocols. In *8-th ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, Montreal, Qc., Canada, October 2005.

[52] D. Nguyen, J.J. Garcia-Luna-Aceves, and Katia Obraczka. Collision-free asynchronous multi-channel access in ad hoc networks. In *In Proc. IEEE Globecom 2009 AdHoc, Sensor and Mesh Networking Symposium*, 2009.

[53] NIST/SEMATECH e-Handbook of Statistical Methods: EWMA Control Charts. http://www.itl.nist.gov/div898/handbook/pmc/section3/pmc324.htm.

[54] Nuttcp. http://www.lcp.nrl.navy.mil/nuttcp/.

[55] Open Source Code. https://github.com/dnlove/.

[56] A. J. Paulraj, D. Gore, R. Nabar, and H. Bolcskel. *An Overview of MIMO Communications-A Key to Gigabit Wireless*. Proceedings IEEE, 2004.

[57] A.J. Paulraj and T. Kailath. *Increasing Capacity in Wireless Broad-cast Systems Using Distributed Transmission/Directional Reception*. U.S. Patent 5 345 599, 1994.

[58] I. Pefkianakis, Y. Hu, S.H.Y Wong, H. Yang, and S. Lu. MIMO Rate Adaptation in 802.11n Wireless Networks. In *In Proc. ACM MOBICOM*, Chicago, USA, 2010.

[59] PID Linux Wireless. http://linuxwireless.org/en/developers/Documentation/mac80211/RateControl/PID.

[60] H. Rahul, F. Edalat, D. Katabi, and C. Sodini. Frequency-Aware Rate Adaptation and MAC Protocols. In *Proc. ACM MOBICOM*, 2009.

[61] S. Rangwala, R. Gummadi, R. Govidan, and K. Psounis. Interference-aware fair rate control in sensor networks. In *Proc. SIGCOMM 2006*, Pisa, Italy.

[62] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly. Opportunistic media access for multirate ad hoc networks, 2002.

[63] A. Singh and D. Starobinski. A semi-Markov-based analysis of rate adaptation algorithms in WLANs. In *IEEE SECON*, 2007.

[64] H-S W. So and J. Walrand. Technical report: Design of a multi-channel mac protocol for ad hoc wireless networks.

[65] J. So and N. Vaidya. Multi-channel Mac For Ad Hoc Networks: Handling Multi-channel Hidden Terminals using a single transceiver. In *In Proc. ACM MobiHoc*, 2004.

[66] J. L. Sobrinho. Algebra and algorithms for qos path computation and hop-by-hop routing in the internet. *IEEE/ACM Transactions on Networking*, 10(4):541–550, 2002.

[67] V. Tarokh, N. Seshadri, and A. Calderbank. *Space-time Codes for High Data Rate Wireless Communication*. IEEE Trans. Inform. Theory, vol 44, 1998.

[68] E. Telatar. *Capacity of Multi-Antenna Gaussian Channels*. European Transactions on Telecommunications, 1999.

[69] The CMU Monarch Project. Wireless and mobility extension to ns.

[70] The Linux Kernel. `http://www.kernel.org/`.

[71] The MadWifi project. `http://madwifi-project.org/`.

[72] The ns-3 project. `http://www.nsnam.org/`.

[73] E. Ancillotti and R. Bruno and M. Conti title =.

[74] F. A. Tobagi and L. Kleinrock. Packet switching in radio channels: Part II-The hidden terminal problem in carrier sense multiple-access and the busy-tone solution. *IEEE Trans. Commun.*, COM-23:1417–1433, Dec 1975.

[75] A. Tzamaloukas and J.J. Garcia-Luna-Aceves. Channel-Hopping Multiple Access. In *Proc. IEEE ICC*, New Orleans, Louisiana, June 2000.

[76] A. Tzamaloukas and J.J. Garcia-Luna-Aceves. A Receiver-Initiated Collision-Avoidance Protocol for Multi-Channel Networks. In *In Proc. Of IEEE INFOCOM*, 2001.

[77] VINT GROUP. Ucb/lbnl/vint network simulator(version 2).

[78] Mythili Vutukuru, Hari Balakrishnan, and Kyle Jamieson. Cross-Layer Wireless Bit Rate Adaptation. In *ACM SIGCOMM*, Barcelona, Spain, August 2009.

[79] Wireless Networks. `http://en.wikipedia.org/wiki/Wireless_network`.

[80] S.H.Y Wong, H. Yang, S. Lu, and V. Bharghavan. Robust Rate Adaptation for 802.11 Wireless Networks. In *Proc. MOBICOM*, 2006.

[81] A. Woo and D. Culler. A transmission control scheme for media access in sensor networks. In *Proc. Mobicom 2001.*

[82] S.L. Wu, C.Y. Lin, Y.C. Tseng, and J.P. Sheu. A New Multi-Channel MAC Protocol with On-Demand Channel Assignment for Multi-Hop Mobile Ad Hoc Networks. In *In'l Symposium on Parallel Architectures, Algorithms and Netwroks(I-SPAN)*, 2000.

[83] Q. Xia and M. Hamdi. Smart sender: a practical rate adaptation algorithm for multirate IEEE 802.11 WLANs. pages 1764–1775. IEEE Transactions on Wireless Communications, 2008.