RATE CONTROL ALGORITHMS FOR IEEE 802.11 WIRELESS NETWORKS

by

BODHISATWA CHAKRAVARTY

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

August 2007

ACKNOWLEDGEMENTS

I would like to express my deep gratitude and thank my advisor, Prof. Kalyan Basu, for the constant support and guidance throughout my Master's program. I thank him for putting up with me patiently and correcting me at every instance and for urging me to think on my own, analyze situations and find the best action based on that. His in depth knowledge in wireless technology and statistical mathematics helped me develop the novel algorithm. Without his guidance and support this thesis would not have been possible at all.

I would also like to thank Dr. Sajal Das for the invaluable guidance in developing the LeZiRate algorithm from his previous work and providing the basis of LeZiRate algorithm. I would also like to thank my friends in the CReWMaN lab for supporting me through the thesis work.

I would like to thank my family for sending me to USA for pursuing my dream to work in wireless technology, especially my father and mother. I would like to dedicate this work of mine to my late father, Mr. Bijan Bikash Chakravarty whose dream I would try to realize, whose words would always guide me in every task I take up in my life.

July 20, 2007

ABSTRACT

RATE CONTROL ALGORITHMS FOR IEEE 802.11 WIRELESS NETWORKS

Publication No. ______

Bodhisatwa Chakravarty, MS

The University of Texas at Arlington, 2007

Supervising Professor: Prof. Kalyan Basu

This thesis evaluates software driven rate selection algorithms used in IEEE 802.11 wireless network interface cards. The motive of the bit-rate selection techniques is to optimize the throughput over the wireless network out of the many rates that are supported by the IEEE 802.11 link. The decision to switch from one rate to another with the changing link conditions to optimize the throughput is the primary focus of the bit-rate selection algorithms. Due to the uncertainty in the channel quality, it is a challenge to make the correct decision so as to minimize the wastage of network resources and achieve the highest throughput.

This thesis also presents a novel learning bit-rate selection algorithm called the LeZiRate. LeZiRate uses the measurements of the quality of the signal received at the device to learn and predict the future quality of signal in near time. This thesis also presents a novel learning bit-rate selection algorithm called the LeZiRate. LeZiRate uses the measurements of the quality of the signal received at the device to learn and predict the future quality of signal in near time. Based on the prediction of channel condition it also selects the best bit-rate that would achieve the maximum throughput. LeZiRate does not monitor the network packets and therefore does not use any of the network resources. It monitors the signal quality received at the station and makes its decision based on that. It also makes the corrections to the prediction values by inducting the actual measurement of the signal quality on a real time basis so as to enable itself to make better predictions in the future.

The LeZiRate algorithm monitors the signal quality and maps that to the received signal strength values, these values are quantized to map into a set of symbols. The frequency of occurrences of the string of these symbols is used to build a tree based on first order Markov model. It then selects the best bit-rate it believes would fetch the highest throughput. LeZiRate has a short setup and learning time to predict the first symbol.

This thesis provides the simulation study of the LeZiRate algorithm and also presents simulation results of an existing bit-rate selection algorithm currently being used in the MADWiFi project for the multi-band Atheros wireless network interface cards. The simulation results show that LeZiRate is more sensitive to the changing link conditions of the wireless media although it does not use the network resources at all.

The simulation study involved some measurements done through experimentation to collect realistic data and running the algorithms against that data.

TABLE OF CONTENTS

Chapter

LIST OF ILLUSTRATIONS

Figure Page

LIST OF TABLES

CHAPTER 1

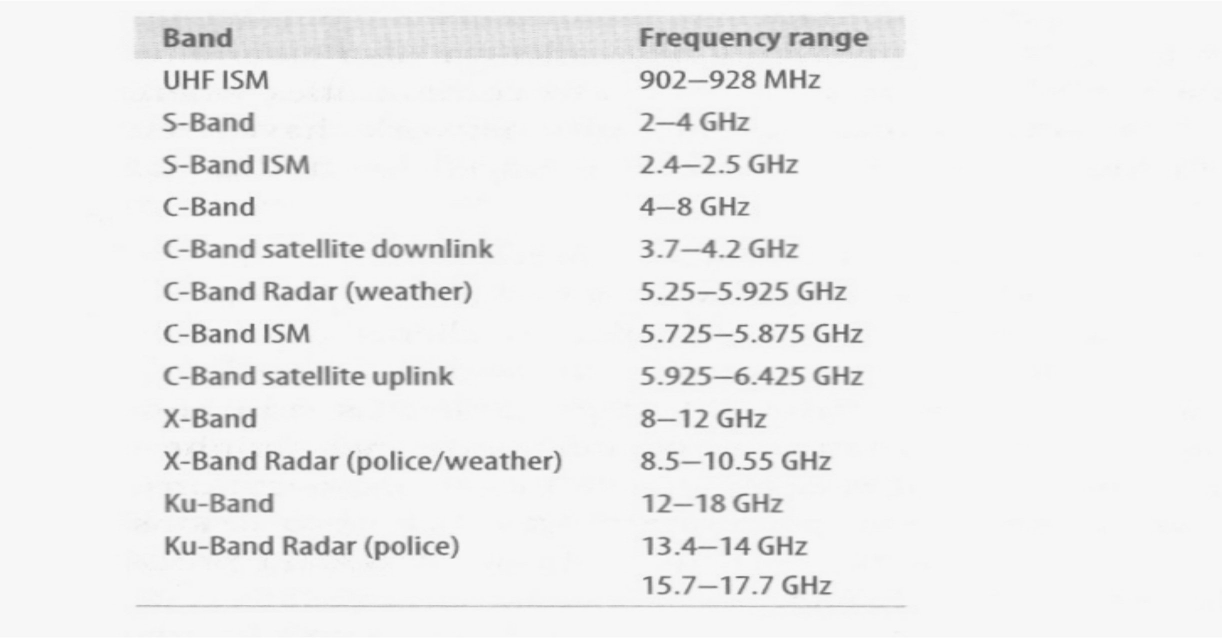
INTRODUCTION

One of the biggest advantages that wireless networking provides over the wired networking is *mobility.* The users can connect to existing networks and can then move about freely without having to worry about getting disconnected. This frees the users from the requirement of having to sit close to a place where Ethernet cable is available. The second advantage that wireless data networks provide is the *flexibility.* What this means is that the modern day wireless networks can be deployed in very short span of time and it does so without creating much clutter. The infrastructure needed to setup a wireless network consists of base station(s) and antennas. Once this is setup the users can connect to the network easily through their wireless NIC cards present in their PCs or laptops or other wireless capable handheld devices. Scalability is also an advantage that these networks provide. Adding a user to the existing network does not require any changes to the existing infrastructure design most of the time. In majority of the cases the user just has to go through an authentication process to authenticate the user to the wireless network. After this the user can connect to the internet seamlessly. A normal user changing from a wired to a wireless and vice versa doesn't even notice the difference in performance. Due to this flexibility we have witnessed proliferation of wireless networks all over the world in a very short span of time.

Today wireless internet access is offered from street side café's, airports, coffee shops to hotels, motels, school campuses and other public gathering spots. While serving users on the go through the wired networks is possible but it nevertheless proves to be problematic for several reasons. Drawing cables is time-consuming and much more expensive and could potentially require changes be made in the form of constructions. It is almost impossible to estimate the number of cable drops that would suffice in order to support such a group. With a wireless network in place, there is no need to perform any of these operations and gives the users and providers much more freedom, especially in a place like a café or an airport. A simple wired infrastructure connects the base station(s) to the internet. The only limiting factor in supporting group like this would likely to be the total bandwidth on the WAN side. But this is comparable to the cost incurred in a wired local area network too. Availability of cheap equipment for the wireless network infrastructure setup coupled with flexibility that the 802.11 networks provide, has empowered people to provide internet access to many places where high speed access was only dream once. This idea has been highly successful in places where the terrain was hostile and too rugged for traditional wired network to be setup.

Wireless networks have one thing in common. And that is their ability to transmit data over the "air" in the form of electromagnetic signals. Ideally the wireless technology should free the user of any physical boundaries so that the user can freely roam about anywhere without worrying about loosing connectivity and performance degradation. For 802.11 networks two sorts of medium were conceived in the electromagnetic spectrum. The first being the infrared light and the second being the radio frequency. Infrared technology became popular initially but had some issues when it came to user mobility. Devices like mobile phones, printers connected to each other through infrared and exchanged data. But infrared had some limitations and challenges. Infrared signals would easily be blocked by walls, or any opaque objects in their. Infrared is more over limited to line-of-sight scenarios only where the devices connected to each other should be visible to each other at all times. But this still had issues because even if the devices were placed at a fair distance from each even though satisfying the line-of-sight requirement would experience degradation in signal quality due to weather conditions and other factors. Radio waves on the other hand did not suffer from this kind of limitation as it could easily penetrate through most solid objects thereby offering much wider coverage and freedom to the user. It is therefore very obvious that most of the 802.11 equipments we see today use the radio frequency as the transmission media.

Wireless devices are required to use only a specific frequency band of the entire radio frequency spectrum. This frequency band plays a major role in determining the bandwidth as each band unique propagation characteristics. This propagation characteristic determines the amount of data that can be transmitted through air without any loss of information, called the bandwidth of the spectrum band. The bandwidth mainly measures the amount of data that could be exchanged or transmitted over the link. This frequency band and the bandwidth hold the key to the capacity of the wireless network. The wider the frequency band is the higher the bandwidth becomes thereby increasing the capacity of the wireless link. It has been proved using mathematical techniques, information theory and signal processing techniques that higher bandwidth slices have the capability of sending higher amount of data through. The use of radio frequency spectrum is controlled by the government regulatory authorities through the licensing process. For example mobile phones operate over 900, 1800 or 1900Mhz frequency band. The carrier companies like Cingular, T-Mobile acquire permission from these regulatory authorities to use this band for their operations and services. In turn they pay a license fee to the regulatory authority for this permission. In United Stated, Federal Communications Commission (FCC) is the regulatory authority that controls and provides this permission. The figure below shows the common frequency bands commonly used in the United States in a tabular manner. [6]



Band	Frequency range
UHF ISM	902—928 MHz
S-Band	2—4 GHz
S-Band ISM	2.4—2.5 GHz
C-Band	4—8 GHz
C-Band satellite downlink	3.7—4.2 GHz
C-Band Radar (weather)	5.25—5.925 GHz
C-Band ISM	5.725—5.875 GHz
C-Band satellite uplink	5.925—6.425 GHz
X-Band	8—12 GHz
X-Band Radar (police/weather)	8.5—10.55 GHz
Ku-Band	12—18 GHz
Ku-Band Radar (police)	13.4—14 GHz 15.7—17.7 GHz

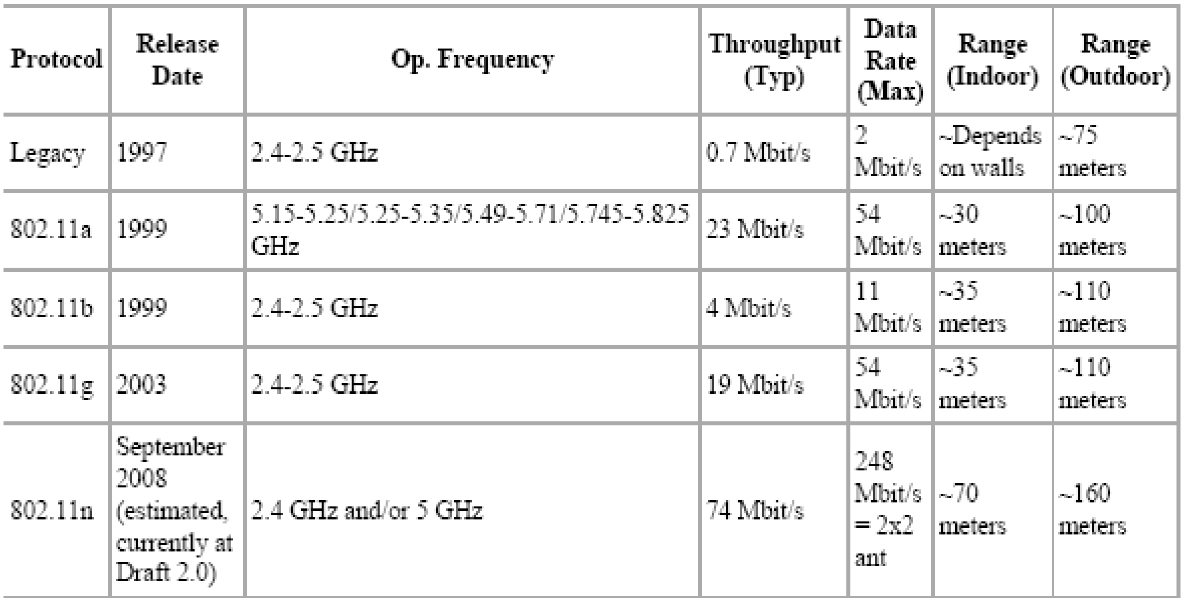
Figure 1.1: List of common frequency bands used in USA

There are a few frequency bands called the ISM (Industrial Scientific and Medical) bands that are available for use without acquiring any license. ISM is an acronym for industrial, scientific and medical. These bands were especially reserved for industrial, scientific and medical equipments. Therefore to use any device or equipment using the ISM band does not require a license to operate unlike the cell phones. The most common example of equipment using the ISM band that is found in almost all households would the microwave oven. The microwaves generated to heat the object placed inside have their frequency in the 2.4-GHz ISM band. Other equipments using the same 2.4-Ghz band would be some of the modern cordless phones. All these devices share the common frequency spectrum for their operation. But there is a restriction that FCC puts on devices operating in the ISM band. The restriction is in the form of the operating power of the device. That means the devices using this band can freely operate as long as their transmitting power remains below a certain number decided by the FCC.

In spite of having multiple advantages wireless networks do not serve to replace wired networks. The maximum data transfer rate achievable for the wireless networks is heavily restricted in comparison the wired ones. Ethernet these days can boast speeds up to 1Gbps (Gigabit) but the maximum speed on the current wireless networks reach only up to a maximum of 54 Mbps (108 Mbps in some proprietary cards with proprietary "*Turbo*" mode). This limitation comes from the fact that the current available bandwidth on the ISM band is wide enough to support this speed only. Unless the regulatory authorities make way for wider band or the current network technology moves to the next level of the frequency spectrum and transfers to the UWB (Ultra Wide Band) spectrum. Second challenge faced while using wireless network is the high BER (Bit Error Rate). High BER is an inherent property of the wireless networks. When comparing the BER in the wired domain with the BER in the wireless domain we find that the two respective values differ by a magnitude of hundreds if not thousands. This is also one of the primary reasons why the conventional TCP protocol doesn't work very well in the wireless networks. Because the moment TCP experiences a loss of packet it assumes loss due to network congestion (congestion at the router) and starts backing off. While this assumption is reasonably true for wired networks but does not hold true at all for the wireless networks. This presents a challenge for the protocol designers. The next concern which often becomes a big issue for any wireless network is the security. Since these networks do not require a person to be physically present at a particular place, therefore a user who is sitting outside the building potentially has the same capabilities of accessing the network as any user sitting inside the building. In comparison a wired network resources are only available to people physically authorized. The wired network can easily be protected using doors, locks etc. wireless are therefore vulnerable to eavesdropping, unauthorized network resource access, virus attack etc. over the past couple of years efforts have been made to make the wireless networks more and more secure. WEP (Wired Equivalent Privacy) is a method of securing access to a wireless network by sharing a key between the exchanging hosts. This key is used to encrypt the messages and then exchange the encrypted messages. Some researchers at University of California at Berkeley devised a method to break this code and expose this key. In the past couple of years wireless networks have become more secure by employing more sophisticated methods of security like WPA, WPA2, MAC filtering etc. The next challenge for wireless networks is energy efficiency. The driving force for wireless networks has been mobility and with that in mind it is reasonable to assume that most of the devices accessing wireless resources would be running on battery power, which happens to be a limited resource and a limiting factor. Therefore reduction of power consumption has also been a constant motivation. The wireless NIC cards should be able to provide network services to the user using minimum power resources without affecting the quality of service much. Research has been going on in this field too for many years now. Many types of wireless network technologies have emerged in the past few years to improve the power usage efficiency of the wireless devices. Wireless broadband access through the cell phone networks is a constantly evolving field through many innovations. With the emergence of 3G (Third generation) wireless networks and WCDMA (Wideband CDMA) also coming in to play, users can now access internet through their cell phones at high speeds. We have also seen the emergence of wireless personal area networking (WPAN) in the form of Bluetooth technology. Bluetooth was mainly developed to aid short range communication. Bluetooth has been fairly successful in connecting devices inside home like cell phone, Bluetooth headset etc. Bluetooth uses the principle of Ad-Hoc networking for communication. Bluetooth has limited data speed in comparison to cell phone networks or the 802.11 networks providing maximum speeds of up to 752 Kbps.

The table in the next page gives a basic comparison between different 802.11 standards. [13]

Table 1.1: Different IEEE 802.11 standards



Protocol	Release Date	Op. Frequency	Throughput (Typ)	Data Rate (Max)	Range (Indoor)	Range (Outdoor)
Legacy	1997	2.4-2.5 GHz	0.7 Mbit/s	2 Mbit/s	~Depends on walls	~75 meters
802.11a	1999	5.15-5.25/5.25-5.35/5.49-5.71/5.745-5.825 GHz	23 Mbit/s	54 Mbit/s	~30 meters	~100 meters
802.11b	1999	2.4-2.5 GHz	4 Mbit/s	11 Mbit/s	~35 meters	~110 meters
802.11g	2003	2.4-2.5 GHz	19 Mbit/s	54 Mbit/s	~35 meters	~110 meters
802.11n	September 2008 (estimated, currently at Draft 2.0)	2.4 GHz and/or 5 GHz	74 Mbit/s	248 Mbit/s = 2x2 ant	~70 meters	~160 meters

CHAPTER 2

IEEE 802.11 WIRELESS NETWORKS

This chapter serves as the introduction to the IEEE 802.11 wireless networks. It covers the physical layer (PHY) and the medium access control (MAC) layer.

The IEEE 802.11 wireless networks consist of two types of basic architecture. The first one is Ad-Hoc configuration where two or more stations communicate with each other without any access point present. These stations normally communicate with each other and not the outside internet. If they want to connect to the outside internet then one of the stations is made to act like a pass through router or the users will have to use the second type of configuration called the infrastructure based. This kind of network needs an access point to be acting as the common point of contact and all the stations connect to the access point and not each other directly. The access points acts as the router also providing internet access to the connected stations. The figures below depict the Ad-Hoc as well as Infrastructure based IEEE 802.11 network configuration. Figure 2.1 shows the Ad-Hoc scenario. [14]

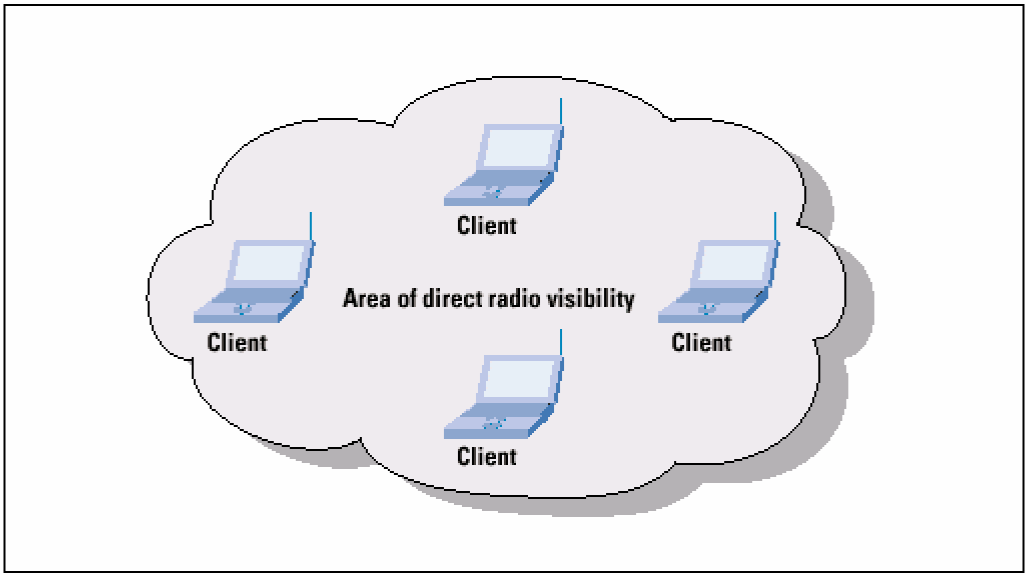


Figure 2.1: Typical IEEE 802.11 Ad-Hoc network setup

Figure 2.2 shows the Infrastructure based 802.11 wireless networks.

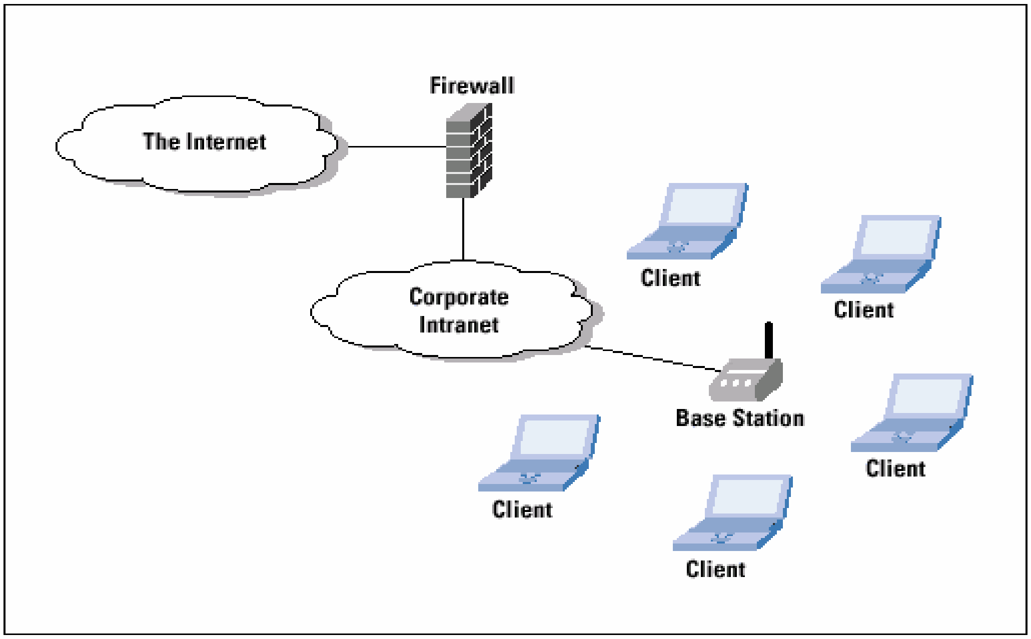


Figure 2.2: Typical IEEE 802.11 Infrastructure based network setup

2.1 Medium Access Control (MAC)

Medium access control layer is the layer which is primarily responsible for controlling the transmission of data over the physical media. It is the key to the IEEE 802.11 specification. It plays a major role in the discussion of the thesis because the rate control algorithms mainly use the MAC for its operations. The MAC acts as the brain of the specification. Different physical layers might provide varied types of speeds and encoding techniques but they still have to be interoperable.

The MAC layer in the 802.11 specifications is not very different from the MAC in the existing IEEE 802 standards. It uses similar principles on the radio link. The wired 802 standards use *Carrier Sense Multiple Access* (CSMA) protocol along with collision detection control the shared physical media access by the stations. The 802.11 wireless NIC continuously measures the energy level on the radio link. The radio link has the energy threshold. Energy level over the threshold indicated activity on the link. In comparison to the IEEE 802 wired standards 802.11 uses *Collision Avoidance* in place of *Collision Detection* along with CSMA. This is due to the fact that collisions waste the packets transmitted by the transmitting stations and this translates to valuable energy loss for the battery powered wireless station(s). 802.11 MAC uses the same principle of distributed coordination for the access where all the stations use the same method to control its access to the physical medium without the presence of any central coordinator.

2.1.1 Challenges for the MAC

The difference in the physical layer of the Ethernet and IEEE 802.11 networks pose many challenges for the MAC layer. The main component of the 802.11 networks is the radio link. The radio link quality therefore determines the "quality" of air that the station will see. In comparison to the Ethernet where irrespective of physical presence of the station on the network the "quality" of the physical media doesn't change with distance or presence of objects like wall, partitions etc the radio link quality changes drastically from one place to another according the distance and also with the presence of walls and other obstructions. Radio link quality also matters with the kind of frequency that is employed. Since the 802.11 networks use the unlicensed ISM band along with other devices like microwave oven and cordless telephones as mentioned before, therefore presence of noise and interference in the medium is expected. Presence of walls and other obstructions give rise to multi-path fading which could potentially degrade the signal quality received at the station. The 802.11 devices also have to deal with the limited range of each radio and therefore the famous hidden node problem. Figure 2.3 in the next page depicts the hidden terminal problem. [15]

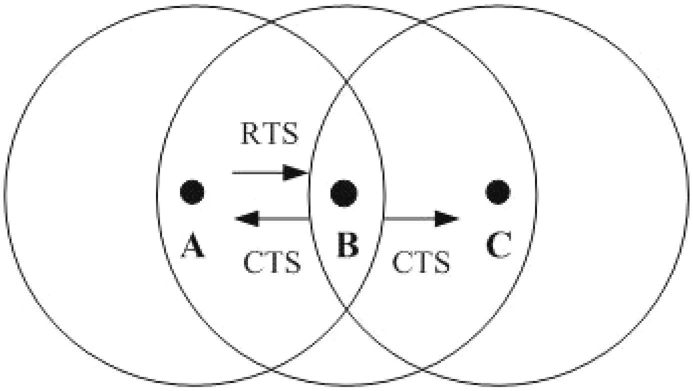


Figure 2.3: Hidden terminal problem and RTS/CTS message exchange mechanism

Collisions resulting from the hidden node are difficult to detect as the wireless transceivers only work in the half-duplex mode. This limits their ability to transmit and receive at the same time. Therefore the nodes tend to avoid collisions. To employ *Collision Avoidance* the 802.11 network devices use two kinds of messages, *Request to Send (RTS) and Clear to Send (CTS)*. The RTS message is sent by the station wishing to transmit data to the station it wishes to send data. And the CTS message is sent by the prospective receiver station once that RTS request is received. This procedure achieves two things, (1) it serves as an acknowledgement that the prospective station is ready to receive the data and (2) this informs another station that might be hidden from the sender that it is already expecting to receive data. This way the hidden node "backs off" for a period of time. Once the exchange of RTS/CTS has taken place the sender can freely send the data to the receiver and expects positive acknowledgement from the receiver. This exchange of RTS/CTS messages before any transmission takes place induces wastage of time and also uses some network bandwidth. This is the reason why driver allows the user to control the RTS/CTS message exchange by setting the RTS *threshold*. RTS/CTS messages are exchanged before transmission only for packet sizes exceeding this threshold value. Setting a larger value for threshold would mean much lower chances of bigger packets getting destroyed due to collision with only added overhead of exchanging small RTS/CTS messages. This is a tradeoff that the user can make keeping in mind the various factors like average size of packets, importance and number of contending stations before setting the RTS threshold value. 802.11 MAC uses two kinds of access modes for the operation. They are called the DCF (Distributed Coordination Function) and the PCF (Point Coordination Function). The first one is more widely used and is more popular than the later. DCF is mainly based on the principle of CSMA/CA, where as the name suggests there is no coordinator for the medium access. All the stations present in the network contend for the medium access. Just like Ethernet the stations first check to see if the medium is free for transmission. Each station also chooses a random back-off time it uses to defer transmission attempt after each packet. As mentioned earlier, it can also use the RTS/CTS message exchange along with this before data transmission. Both of the techniques are to avoid and to minimize the chances of collision. In comparison, the PCF uses a central coordinator to coordinate access to the physical medium by the stations. This coordinator normally resides inside the access point in the network. The coordinator polls each station for any transmission that station might intend to perform. This method ensures contention free media access to each station and therefore almost eliminates the chances of any collision. But due to the fact that the PCF resides inside the access point the availability of PCF is restricted to managed (infrastructure based) networks only. There are few types of short time intervals that 802.11 network uses to separate frames. These are mainly SIFS (Short Inter Frame Spacing), PIFS (PCF Inter Frame Spacing), DIFS (Distributed Inter Frame Spacing) and EIFS (Extended Inter Frame Spacing). The length of each of these is in the order mentioned starting from the shortest one first. The PIFS is restricted to use only when PCF is being used for managing the media access. And EIFS is not a fixed interval and is only used when there is an error in the frame transmission. The 802.11 Wireless NIC cards use two types of carrier sensing. First type is the physical carrier sensing. In this the antenna monitors the energy level on the medium constantly to determine the availability of the medium. But the physical device has a limitation; the device cannot sense as well transmit at the same time as these are made to be half duplex. Therefore the stations also use software based carried sensing which is called the *Virtual Carrier Sensing*. If any of these indicate a busy medium the MAC translates that to a busy medium and registers that information. Virtual carrier sensing makes use of something called Network Allocation Vector (NAV). It is nothing but a number that indicates the amount of time that a station considers it would take to complete a transmission and therefore it also indicates the amount of time that the medium will be reserved for the station. The NAV is included inside the RTS/CTS packets. As an example, when a station intends to transmit a packet it sends the RTS message to the intended recipient and includes the NAV as the amount of time it considers the medium would become free after it has completed sending the packet. The other stations hearing this RTS message back off for the time period equal to the NAV specified. The receiver then sends out the CTS message with a new NAV in the message which accounts for the time spent between the RTS/CTS message exchange. The hidden node oblivious of the original RTS message now hears the CTS message with the NAV specified and backs off for the NAV specified in the reply (CTS message). The stations use a countdown timer to count down from NAV to 0. They assume the medium to be busy till the NAV hasn't reached 0. Once the timer goes down to 0 the contention period starts after the DIFS time interval. Each station replies to a message after SIFS time interval.

2.1.2 Error Recovery and Backoff

Every frame sent should be positively acknowledged. The receiver must send an acknowledgement of the frame it receives but it is up to the sender to detect error and recover from it. Normally the sender infers an error or loss of packet when it doesn't get an acknowledgement from the sender. As in most of the error recovery schemes the sender then tries to retransmit the frame back. There is a single retry counter associated with every frame and that gets incremented with every failed transmission (retransmission attempt). Stations on the other hand two retry counters associated with them. One is short retry counter and that is used for packets shorter than the RTS threshold value and the other one is the long retry counter associated with packets longer than the RTS threshold. Most of the presently used bit-rate selection algorithms use the statistics of retry count, failed transmissions. Therefore this topic is relevant to this thesis. The data transmissions are supposed to be atomic in nature. Either the whole frame is transmitted and acknowledged or the transmission is termed as failed and retransmission takes place. With the frame retry counter we also have a retry limit or maximum retry number. If the MAC is unable to transmit the packet successfully till it reaches the maximum retry limit then the frame is discarded and the higher layers are notified of the loss and then it is up to the transport layer (TCP/UDP) to decide whether to retransmit the data or to discard it.

The Backoff used by the 802.11 stations is principally similar to the one used in the Ethernet. After the DIFS time period is over the contention window starts. Contention window is divided into time slots and length of each slot is dependent on the speed of the physical media in use, i.e., the transmission speed. The stations then choose a random number slot between 0 and contention window size to transmit data. If two stations choose the same number slot then there is a collision. After every collision the size of the contention window increases. The size of contention window is always 1 less than a power of 2. Every time a collision takes place the contention window moves to the next higher value. This reduces the chances of collision by reducing the chances two stations trying to transmit at the same time. The size of the contention window is again limited by the transmission capabilities of the physical medium. Once the contention window reaches its maximum size it is only reset if the frame is discarded or frame is successfully transmitted.

2.1.3 Connection establishment and Roaming

In an infrastructure based setup the stations connect to the access point as the point of contact to the outside wired network. The process of establishing connection with the access point follows some standard steps. The station first scans the area for existing signals. There are two methods of scanning used by the stations. The first one is called *Passive Scanning* and the second one is known as *Active Scanning*. In the passive scanning mode the station listens for beacons transmitted by the access point(s) on all the existing channels. The beacons normally contain the basic information about the network like the SSID, encryption type used etc. and the second method involves the station sending out broadcast packets to the access points on all the existing channels. The access points then either reply to the request or just ignore it according to the security settings of the network, whether to reply with the SSID or not. Once the station has the basic information about the network(s) it again reports to the user to select the network the user wishes to connect to. After that there is a series of message exchanges in which authentication is done along with IP address allocation is done using DHCP in most cases. Once the station is connected to the network, it can now start communicating with the internet. If the network that the station connects to consist of multiple access points spread over a physical area then the station could be connected to only one access point at any point of time. The IEEE 802.11 standards specify the MAC and the PHY aspects but don't specify the inter-access point communication standards or protocol. It is up to the vendor to implement that. Different vendors have different protocols of communicating between multiple access points on the same network. The user has the choice of moving around and it is possible that at some point of time the station moves far away from the access point and loses connectivity. At this point it might be getting better signal quality from a different access point on the same network. The station can roam from one access point to another at this point. The station then issues an association request after it evaluates that an association with a different access point will be better then maintaining with the old one. The new access point then asks for the credentials from the station and communicates with the older access point. Once it gets the authentication clearance the access points update their tables indicating the stations current access point connection and also the wireless bridge interfacing with outside internet is notified of the access point to which the station is connected. All the packets destined for the station are forwarded to the new access point from that point of time onwards.

2.2 Physical Layer (PHY)

Below the MAC layer is the physical layer which deals with the actual transmission of the bits received from the MAC layer above into electromagnetic signals. In the IEEE 802.11 network terminology it is abbreviated as PHY. This layer is important to the thesis discussion because much of the algorithm robustness and accuracy depends on the working of the PHY.

The physical layer in the 802.11 standards has been divided into two sub-layers. The PHY consists of *PLCP* (Physical Layer Convergence Procedure) sub-layer and the *PMD* (Physical Medium Dependent) sub-layer. The PLCP layer acts as the go-between the MAC (Medium Access Control) layer and the PHY (Physical Layer). It adds its own header information to the frames passing from the MAC to the PHY layer. A preamble is added to the frames to help synchronize the timing for the incoming transmissions. Different modulations techniques might place different kinds of preamble sequences. The PMD has the responsibility of converting the frames received from the PLCP into bits for transmission over the physical media. The physical layer also has an additional function. It is called the *CCA* (Clear Channel Assessment) function. Its job is to inform the MAC layer about signal detected on the channel.

The 802.11 specifications standardized some physical layers in two stages. First stage had

- Frequency Hopping (FH) using spread spectrum
- Direct sequence (DS) using spread spectrum
- Infrared signals (IR)

The second stage saw two new physical layers being added. They were

- Orthogonal Frequency Division Multiple Access (OFDMA)
- High-Rate Direct Sequence (HR/DS or HR/DSSS)

As mentioned before IR did not find wide scale implementation due to its heavy limitations over range and quality.

Spread spectrum technology forms the base used for data transmission in the ISM band. Unlike the traditional form of radio communication where as maximum amount of signal is pushed into a narrow band of frequency. Spread spectrum works quite similar to the concept of *CDMA* (Code Division Multiple Access) technology. In this, the signal power is spread over the whole available frequency band using mathematical operations. This way the receiver performs an inverse operation at the other end and reconstructs the original signal from the spread signal. The reconstructed signal is identical to the one transmitted. Using this method the noise is separated to a large extent. To a narrow band receiver the transmission would still seem to be something similar to noise. Also, narrow band receivers cannot reconstruct the whole signal fully. Although the spread spectrum technique separates noise to a large extent, it does not eliminate it. And as the number of devices using this technique increases in a specific area the signal to noise ration keeps decreasing and thereby limiting the range of practical operation of the network.

Different Spread Spectrum Techniques

Frequencies Hopping over Spread Spectrum (FH or FHSS) - in this technique the devices hop from one frequency to another in the available spectrum at a definite interval. The sequence of jumps is also predefined and agreed upon by the participating stations. The stations remain on the frequency for a short period of time called the Dwell Time and transmit a burst of information. It is one of the standard techniques used in the modern *Bluetooth* networks.

Direct Sequence over Spread Spectrum (DS or DSSS) - the stations using DSSS spread the signal power over the whole available frequency spectrum with the help of mathematical coding techniques. It multiplies the data to be transmitted by a “noise” signal. This noise signal is pseudorandom sequence of 1 and -1 values.

Orthogonal Frequency Division Multiplexing (OFDM) – station using OFDM uses a technique of dividing the available channel into multiple sub-channels and encodes a part of the signal and transmits that each part into each of those sub-channels simultaneously. This is similar to the Discrete Multi-Tone (DMT) technique used by some DSL modems.

Direct Sequence systems require more sophisticated digital signal processing techniques; therefore they need more complex hardware than the Frequency Hopping systems. Precise timing coordination is required for the FH techniques to control the hopping pattern etc.

There are some theoretical and practical considerations for the physical media that are important and relevant to this thesis.

2.2.1 Theoretical aspects of PHY

Wireless networks allow different modulation techniques for different link qualities. This allows the links to choose the modulation technique that best suites the conditions and thereby optimizes the throughput as the link quality could vary by large amount. Each rate uses a modulation technique to transform the incoming data into a stream of symbols which are then encoded by varying the amplitude, frequency or the phase of the electromagnetic signal being used. The amount of information that a particular modulation technique can transmit depends upon the number of distinct symbols the technique can use to represent the data assuming all the techniques transmit symbols at a constant rate. This set of unique symbol values is called a *Constellation.* The minimum distance between any of the two unique values in a constellation determines the amount of noise it takes to confuse or cause a bit-error. The lesser the distance between these values the higher the chances of bit-error. Sparse constellations tend to experience less bit-error rates than the denser ones. Sparse constellations are resilient to noise interference and experience bit-error at a much lesser signal-to-noise (S/N) ratio than the dense constellations. The table below shows some of the modulation techniques used in the 802.11 networks. [13]

Table 2.1: Different modulation techniques used with OFDM

Data rate *(Mbit/s)*	Modulation	Coding rate	Number of data bits per symbol	1472 byte transfer duration *(µs)*
6	BPSK	½	24	2012
9	BPSK	¾	36	1344
12	QPSK	½	48	1008
18	QPSK	¾	72	672
24	16-QAM	½	96	504
36	16-QAM	¾	144	336
48	64-QAM	2/3	192	252
54	64-QAM	¾	216	224

The amount of information that is received over a specific link is dependent upon the number of bits successfully decoded by the receiver. Channel can experience different types of interference and noise due to which different types of distortions occur. For theoretical purposes we consider only additive Gaussian white nose (AGWN) to be the only one present. Figure 2.4 shows the theoretical bit error rate (BER) against the signal-to-noise ratio for some of the modulation techniques keeping the theoretical assumptions and equations from [7]. The figure [3] shows the Signal-to-noise (S/N) ratio on the X-axis and the Y-axis shows the BER in log scale(Only AGWN has been assumed).

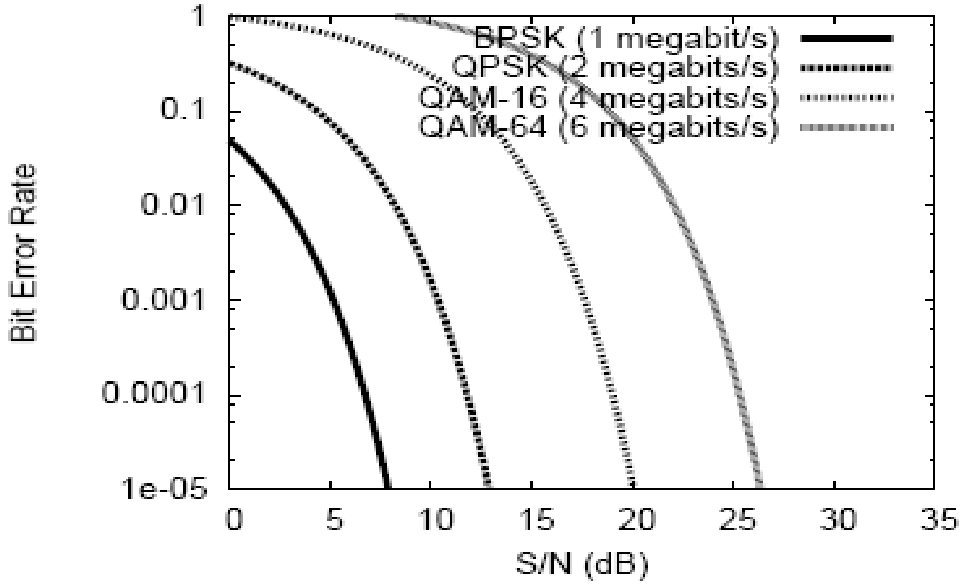


Figure 2.4: Different modulation schemes and their performance against signal power

The IEEE802.11 a/g networks make use of a new techniques called the Orthogonal Frequency Division Multiplexing (OFDM). It encodes a single transmission into multiple sub-carriers derived from one wide carrier. The wide frequency channel is divided into multiple sub-carriers of smaller bandwidth and not necessarily non-overlapping channels. OFDM has enabled 802.11 networks to achieve data rates of up to 54 mbps.

2.2.2 Practical aspects of PHY

Along with the theoretical aspects we need to take into account some of the practical aspects of the PHY. The theoretical considerations overlook some of the practical challenges that are encountered. It has been shown from earlier work [9, 8] that link quality cannot be accurately ascertained from the signal-to-noise ratio (S/N) where there is high density of obstacles that tend to reflect the waves of them thereby giving rise to multi-path fading. This degrades the performance of the link due to phase error even though the signal-to-noise (S/N) [10, 11] may not degrade in the same way. This adds a new section on the problem of rate control in IEEE 802.11 wireless networks. The present devices support many rates at different coding and modulation scheme. According to Figure 2.4, as the link condition changes the S/N ratio dynamically, if the modulation and coding is changed properly, it is possible to get the best capacity from the system.

CHAPTER 3

RELATED WORK

This chapter gives a brief description of the various rate control algorithms that have been developed and are being used in wireless network interface cards today.

3.1 Auto Rate Fallback (ARF) and Adaptive ARF (AARF)

This algorithm was developed at the Bell Labs and published in a journal in the summer of 1997 [12]. This was mainly developed for use in the WaveLAN-II 802.11 WNIC. These cards were one of the earliest cards used for the 802.11 networks and could transmit at rate of 1 and 2 Mbps. The idea behind ARF is to adapt to the variable conditions in the wireless link by monitoring the amount of packet loss that the link experiences. It was designed to work with many different rates for the future IEEE 802.11 standards of the WaveLAN cards.

This algorithm works by monitoring the current rate that is being used along with the number of lost packets. It makes use of the property of 802.11 networks that all the packets need to be positively acknowledged by the receiver. The card reports this information to the MAC layer. The station tries to retransmit the packet repeatedly till it is either acknowledged or the number of retransmissions exceeds the maximum retry count set before hand. After which it discards the packet and reports it to the higher layers, namely the transport layer.

The algorithm starts by transmitting at the highest rate listed. It then starts building the statistics for number of retransmissions and number of successful transmissions. It then adjusts the rate using the statistics it built. There is a time limit for the adjustment to be completed and the algorithm purely depends upon the packets to be transmitted during that time. The decision to adjust the rate is taken using the following rules:

1) It steps down to the next available lower rate when the packet was lost, i.e., the retransmissions exceeded the maximum retry limit without being acknowledged. It moves to the next available higher rate if the there has been ten successive packet transmissions without having to make any retransmission attempts.
2) It stays at the current rate if none of the previous conditions are true.

This is a very simple algorithm to understand and implement and does not maintain any information from the past. It jumps to the higher after 10 successful transmissions. This enables it to work well in situations where there is a change in the link conditions over a period of time and it takes some time before it can shift to the next higher rate. The algorithm however only steps down if there is a packet failure. It does not step down for packets which get successfully transmitted after some number of retries. Successive packets needing retry could potentially mean degraded link condition but ARF does not take that into account. This can waste significant amount of network resources and time. This also does not guarantee optimum throughput. ARF also pushes up to the next higher bit rate based on the performance of the current rate. It assumes that the higher bit rate would not perform better than the current one; this assumption however is not always true. There could be a higher rate that might be able to achieve a better throughput then the current one. ARF steps down quite quickly in conditions where there is high probability of packet loss but it could take a significant amount of time before it can jump to a higher rate and moreover all the shifts it makes are in a step wise manner. This could translate to wastage of time and network bandwidth again.

Adaptive ARF (AARF) [5] is a variant of the ARF algorithm. It has a step up parameter that is increased to twice the current one every time the algorithm steps up the rate and experiences a packet failure subsequently. This is to ensure that the algorithm does not step up to the next level again with the same ease as it did the last time if the higher rate proved to experience a packet failure in the first attempt itself. This is a good strategy and increases the throughput quite well in higher bit-rates where the back-off penalty is quite high. Both of these algorithms work well with the unicast packets.

3.2 Receiver Based Auto Rate (RBAR)

Receiver Based Auto Rate (RBAR) makes use of the signal quality of the link by tracking the signal-to-noise (S/N) values. It assumes that the channel quality at the receiver's end determines the success of the packet delivery. It also relies on the working of the RTS/CTS mechanism. Therefore for the algorithm to work, the RTS/CTS message exchange mechanism should be turned on.

The receiver calculates the highest rate that the link could achieve with a Bit Error Rate (BER) of less than 10^{-5} based on the received information from the signal-to-noise (S/N) ratio. It then piggybacks that bit-rate in the reply (CTS) to the receiver. The sender then uses that rate to transmit the packet at that rate. This algorithm works well in networks where rate needs to be determined on a per-packet basis. It works well in the environments where the average packet length is high and the penalty to retransmit that packet is high. But it assumes that the receiver is effectively able to calculate the best bit-rate at the sender's end from the RTS packet. This also induces computational need at the receiver end. RBAR makes the assumption that the rate calculated at the receiver is valid for the sender. Although it has been suggested and observed before that signal-to-noise (S/N) does not indicate actual quality of the channel and is not an accurate indicator of the channel condition.

3.3 Onoe

Onoe is one of the open-source Rate Control Algorithm which has been made available by the MADWiFi (Multi-band Atheros Driver for Wireless Fidelity) [4] project that has been developed to work with multi band 802.11 networks. The MADWiFi driver is currently available for Atheros chipset for the Linux distributions. This algorithm is one of four algorithms available in the source. This thesis uses an implementation of this algorithm to compare the results of the simulations conducted.

The idea behind this algorithm is to use a system of credits for each bit-rate. It steps up by one level every time the credit for that bit-rate reaches the max_credit limit and steps down by one level if the number of credits for the current bit-rate drops below zero. At every new bit-rate it resets the credit to zero. This algorithm does not rely on the packet failures only. It considers the number of retries to be a more accurate indicator of the channel quality. The algorithm takes a decision to adjust the bit-rate every 1000 milliseconds (default configuration).

The algorithm starts off by setting the initial bit-rate to 24 Mbps and the credits for this rate set at 0. For the 802.11 b mode it sets the initial bit-rate at 11 Mbps. It then starts tracking the packet statistics at the current rate. The decision to step down, step up or continue at the current rate is based on the following criteria:

1) If all the packets failed then step down to the next lowest bit-rate available.
2) If during the observation period 10 or more packets were sent with an average number of retries exceeding one then step down to the next lowest bit-rate available.
3) If more than 10% of all the packets transmitted during this observation period needed retry then it decrement the number of credits by 1 as long as the value of credits is more than 0.
4) If less than 10% of all the packets transmitted during the observation period required a retry then increment the value of credit by one as long as it does not exceed the upper limit of the credit (which is 10 in the default configuration).
5) If decrementing the number of credits takes the credit count below 0 then step down to the next lowest bit-rate available.
6) If the current rate has accumulated more then upper limit of the credits then step up to the next higher available bit-rate.
7) If none of the above are true then continue at the current rate.

Onoe again uses steps to move up and down the available list of bit-rates as in the ARF or AARF. Onoe is comparatively insensitive to individual packet failure which could have occurred due to a short transient condition of for one packet. It takes some amount of time to decide about the change in the bit-rate. Onoe works well in situations where there is a gradual change in network conditions. And once it has stepped down from a rate it will not try that rate for at least another 10 seconds (10000 milliseconds). Therefore it is relatively quite conservative in nature. And minimum amount of time it will take to step down is at least 1 second (1000 milliseconds). Therefore it will take some time and network resources before Onoe can find the ideal rate if Onoe starts of at a much higher rate than this one. This happens in the reverse case too where the ideal is much higher than the one it initially chooses.

3.4 Adaptive Multi Rate Retry (AMRR)

AMRR uses an adaptive method to control the time period after which the transmission count and rate pair value are changed. It adapts by using binary exponential back off (BEB) mechanism to change the length of time the software takes to change the values for the pairs. It makes use of probe packets that algorithm sends at different rate and retry pairs. It then tracks their performance and their transmission status it adapts the time threshold. It also applies simple heuristics to capture short term variation is the link by setting the rate and transmission parameters appropriately. This whole mechanism ensures that there is fewer numbers of unsuccessful transmission and retransmission attempts. It also achieves better throughput by increasing or decreasing the time by using back off mechanism which in turn does not let it switch to a higher rate easily if there have been failures. This algorithm is also a part of the MADWiFi project.

3.5 Sample

Sample [3] algorithm was developed by a student at MIT about two years back. The aim of this algorithm was to meet some of the challenges that were not taken care of by the existing algorithms. it took into account a) higher bit-rate may not necessarily perform worse than the lower ones just because the lower one is performing poorly, b) it should be ale to adapt to the changing link conditions, c) sampling all the available bit-rates would not yield optimum results.

Based on the above considerations Sample was developed to transmit packets and periodically (every 10th packet) picks up a random rate other than the current one and collects the statistics. Average transmission time plays a major role in the working of this algorithm. Sample stops probing the bit-rates that have a poor history, it stops sampling the bit-rates out of the list available with 4 successive failed transmissions. The average transmission time is calculated using packet size, the bit-rate and the number of retries needed to transmit the packet. Sample chooses to transmit data at the rate which it predicts to have the lowest average transmission time including the time needed for any retransmissions that are needed. It uses a time frame called averaging window during which it calculates the statistics. It also provides mechanism to remove stale samples from the list by calculating average transmission times for only those packets that were used in that time window.

3.6 Minstrel

Minstrel [4] is one of the latest additions to the existing list of algorithms used in the MADWiFi project. It derives its name from the Minstrel who wanders around at different places (different rates in this case). The basic idea behind this algorithm is transmit at different rates whenever possible other than the current one and switch to the rate that provides the best opportunity for maximum throughput.

The Minstrel autorate selection is a EWMA based algorithm [14]. It uses similar idea as used the Sample algorithm. It uses a formula [14] to compute the successfulness of packet transmission. This measure of successfulness is used to adjust the transmission speed to the optimum level. It dedicates a particular percentage of data packets to be transmitted at different rates than the current one and is set to 10% in the default configuration and the algorithm fires at a definite time interval which is set at 100 milliseconds (10 times per second) in the default configuration. A table is then constructed populated with the success history of each of the rates that were tried during the time period. The rates are then ranked in terms of their throughput performance and a decision is made by selecting the rate that performed and ranked the highest among the list.

The table in the next page shows a summary of all the existing algorithms that have been discussed so far in this chapter.

Table 3.1: Summary of different Rate control algorithms in use

NAME	***PROPERTIES***
ARF/AARF	Uses packet transmission characteristics to make decisions. Waits for the ten successive successful transmissions to move to a higher rate and moves to a lower rate at the first failed transmission. Assumes higher rate cannot do better than the lower rate at any time.
RBAR	Receiver calculates the best rate possible based on the signal-to-noise ratio information from the RTS packet received. Piggybacks the rate on the CTS message to the sender. The calculation is based on the rate that could achieve BER of less than ten in a million.
Onoe	Uses credit based system and assigns credit value to each rate. Increments/decrements the credit based on the transmission statistics of the current rate and makes the decision of shifting the rates only when the credit value has reached the higher/lower threshold.
AMRR	Employs *Binary Exponential Backoff* mechanism to adjust the time period after which the transmission count/rate pair would be changed. Each time there is a failure the time period is doubled to have a higher penalty.
Sample	Uses probing mechanism by transmitting every tenth packet at a different rate randomly chosen from a set of sampled rates. Selects the rate that has the lowest average transmission time including the retransmission attempts.
Minstrel	Uses EWMA based mechanism along with calculating success probability of each rate. Transmits a percentage of the total packets in a time period at different rates to collect statistics. Selects the rate that has the highest probability of success.

CHAPTER 4

DESIGN AND IMPLEMENTATION

This chapter provides the basic idea behind the LeZiRate algorithm proposed in this thesis along with the simulation structure that was built to evaluate the performance. The main motivation of LeZiRate algorithm is to design a learning based adoptive algorithm for the rate selection of the IEEE 802.11 devices. The learning algorithm is based on robust variable window LeZi compression scheme, though in the initial implementation, the window size of the algorithm is kept constant to reduce computational complexity.

4.1 Design

LeZiRate algorithm uses a two stage process to decide upon the rate to optimize the throughput. The algorithm initially takes some time to train itself. This algorithm tries to address some of the issues that were noticed in the existing algorithms discussed in the earlier chapter. The key issues that this algorithm needed to address while adding some functionalities were:

1) Minimal network resource wastage: most of the existing algorithms use the network resources to transmit packets and based on their transmission status make their decisions.

2) Step based upgrade or downgrade: algorithms using step based method take some time before it can achieve the optimal transmission rate in cases where the optimal is far away from the initial one or there are heavy sudden changes to the radio link quality. This wastes both network resources as well as time.

3) Sensitivity to changes: algorithms should be sensitive to short term as well as long term changes in the radio link quality. Short term changes should not trigger changes in the transmission rate under normal circumstances.

4) History information: algorithms should have some amount of state information or past record so as to remove the chances of non-useful rates from being considered.

5) Proper selection: algorithms should not assume poor performance of a higher bit-rate based on the poor performance of the lower bit-rates.

This algorithm addresses all of these issues by passively learning about the channel quality and predicting the quality. Based on the prediction of channel quality it sets the transmission rate. It also keeps measuring the channel quality and calculates the error in actual observed quality and the predicted quality. It incorporates the observed value to the current data and moves ahead, this sliding movement is similar to the window mechanism used in different places. This mechanism also ensures that only recent samples are used to predict the channel quality. For theoretical and simulation purposes some fundamental assumptions have been made. They are:

1) The whole simulation was based assuming Infrastructure based network and not an Ad-Hoc network. This assumption does not affect the LeZiRate adversely.

2) The underlying physical media is reliable at high signal strength values. This is reasonable assumption to make as seen from empirical evidence and BER vs. S/N graph presented in chapter 2.

3) The application level program is not producing data packets at a very high rate and therefore there is no transmission queue formed. This is again a fair assumption and does not affect the LeZiRate mechanism in any way.

4) There is only one station transmitting and receiving in the infrastructure network. This is to ensure there is no overloading of the access point with data packets.

5) There are no packet drops at the router/ access point level. In majority of the cases this is true for wireless network. Packets are only dropped when the network is heavily loaded, and moreover there is only one station in the network.

6) The mobility of the user is not high. In other words we assume that the user is not moving about too fast and too randomly. This is a practical assumption keeping practical scenarios in mind. A user would be at most walking around in the network periphery.

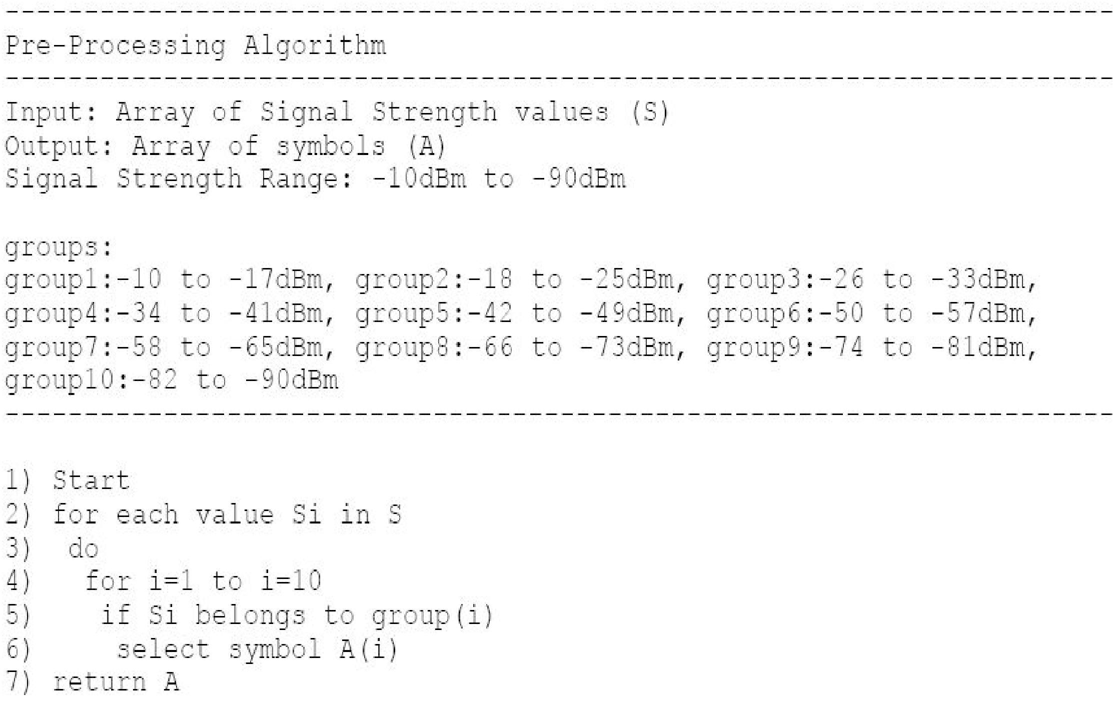
7) The network is assumed to consist of only one access point and therefore there are no handoffs from one access point to another. This is a restrictive assumption to avoid the hand-off complexities. This was a condition kept specifically while experimentation and therefore simulation too. Roaming capability and roaming parameters induce complexity into the simulation.

8) Slight variation in the signal strength values does not alter the channel quality considerably. This assumption is true as it is evident from the graph in chapter 2 that BER does not change much for a specific modulation technique with slight variation in Signal-to-Noise (S/N) ratio.

4.2 Algorithm

The LeZiRate Algorithm is divided into multiple parts. The first part being the preprocessing that is needed to convert the measured signal-strength values into symbols. The procedure is given below:

Algorithm 1: this algorithm mainly does the conversion of raw signal strength data into symbol values for the LeZiRate to run on. In this algorithm, the actual measurement dBm values are classified into a number of groups' as shown below:

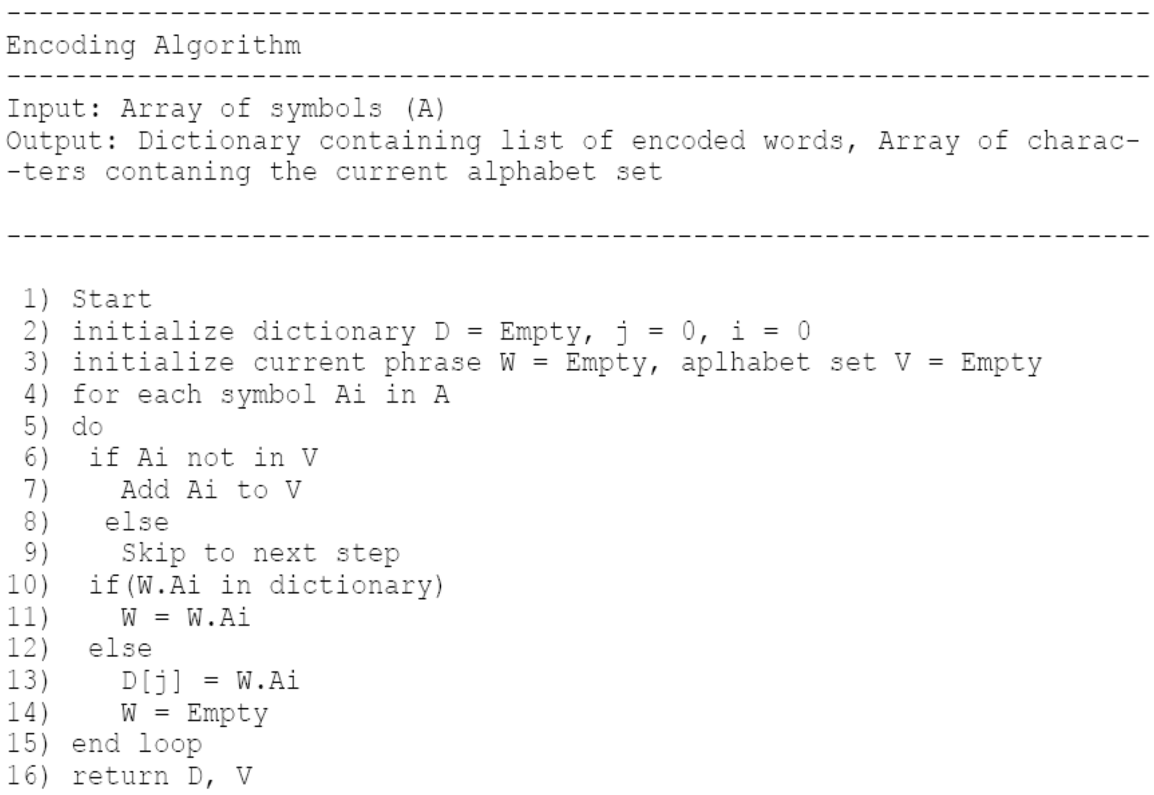


```
----------------------------------------------------------------------
Pre-Processing Algorithm
----------------------------------------------------------------------
Input: Array of Signal Strength values (S)
Output: Array of symbols (A)
Signal Strength Range: -10dBm to -90dBm

groups:
group1:-10 to -17dBm, group2:-18 to -25dBm, group3:-26 to -33dBm,
group4:-34 to -41dBm, group5:-42 to -49dBm, group6:-50 to -57dBm,
group7:-58 to -65dBm, group8:-66 to -73dBm, group9:-74 to -81dBm,
group10:-82 to -90dBm
----------------------------------------------------------------------

Start
for each value Si in S
 do
  for i=1 to i=10
   if Si belongs to group(i)
    select symbol A(i)
return A
```

Algorithm 2: this is the second one. This does the encoding of the symbol values into the dictionary. This is a part of the LeZi- update algorithm from [1]. The encoding process starts off with a phrase (empty) in the beginning or after adding a word to the dictionary and keeps appending a symbol to it every time a new symbol is encountered in the stream. It then matches the appended phrase with the list of words already present in the dictionary and adds it to the list if not found else moves ahead to the next symbol till all the symbols have been exhausted in the stream.



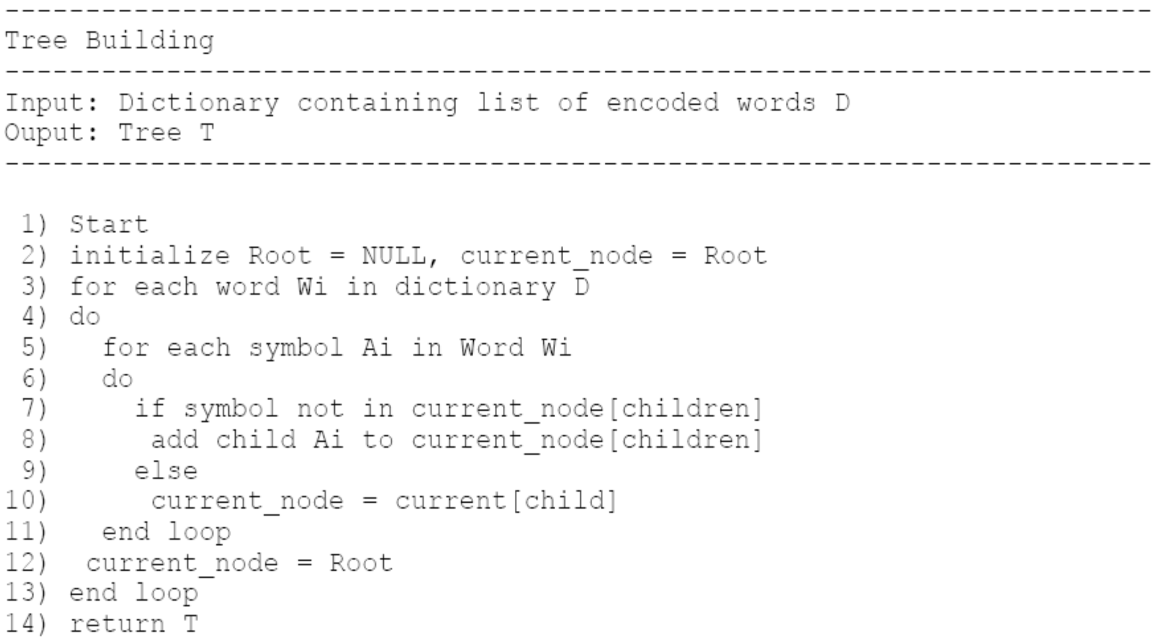
```
-----------------------------------------------------------------
Encoding Algorithm
-----------------------------------------------------------------
Input: Array of symbols (A)
Output: Dictionary containing list of encoded words, Array of charac-
-ters contaning the current alphabet set

-----------------------------------------------------------------

Start
initialize dictionary D = Empty, j = 0, i = 0
initialize current phrase W = Empty, aplhabet set V = Empty
for each symbol Ai in A
do
 if Ai not in V
   Add Ai to V
  else
   Skip to next step
 if(W.Ai in dictionary)
   W = W.Ai
 else
   D[j] = W.Ai
   W = Empty
end loop
return D, V
```

Algorithm 3: this gives the tree building process from the dictionary build in the previous step. The dictionary is scanned for each word and the tree is build depth wise first starting from the root node at the top and adding nodes if needed depicting a

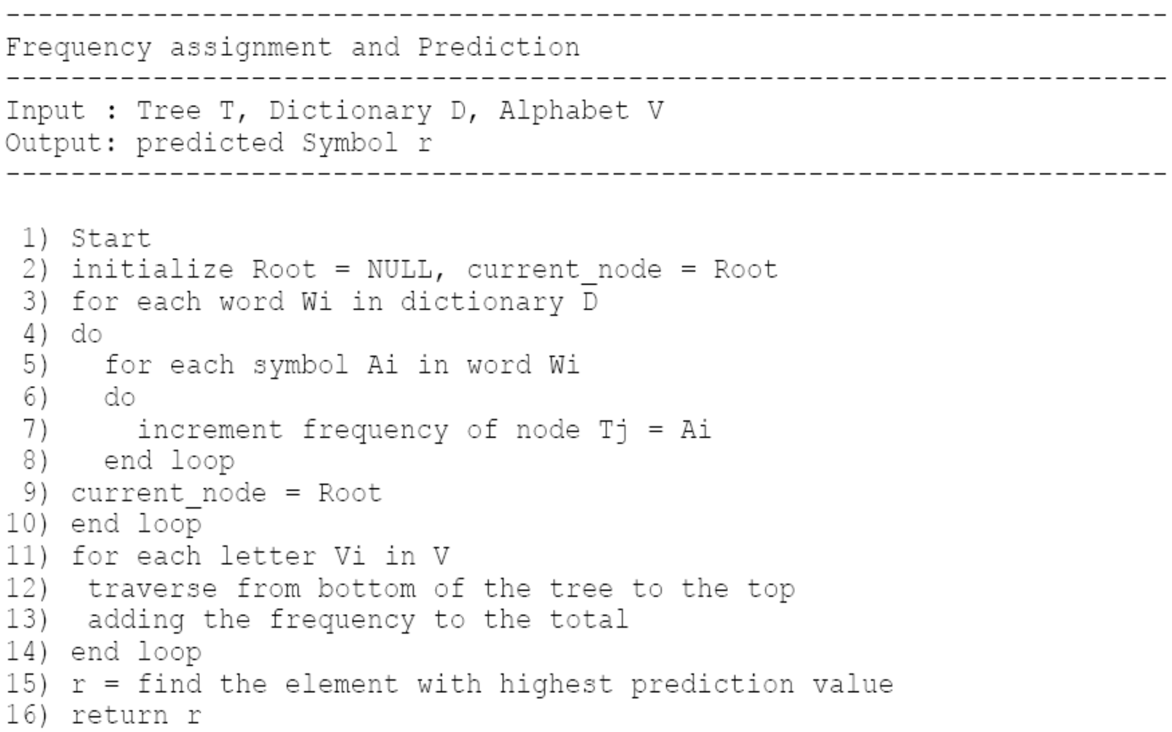
symbol as the control moves down to the leaf node. Therefore each word in the dictionary is depicted as a unique path from the root node to a leaf node.



```
----------------------------------------------------------------------
Tree Building
----------------------------------------------------------------------
Input: Dictionary containing list of encoded words D
Ouput: Tree T
----------------------------------------------------------------------

Start
initialize Root = NULL, current_node = Root
for each word Wi in dictionary D
do
  for each symbol Ai in Word Wi
  do
    if symbol not in current_node[children]
     add child Ai to current_node[children]
    else
     current_node = current[child]
  end loop
 current_node = Root
end loop
return T
```

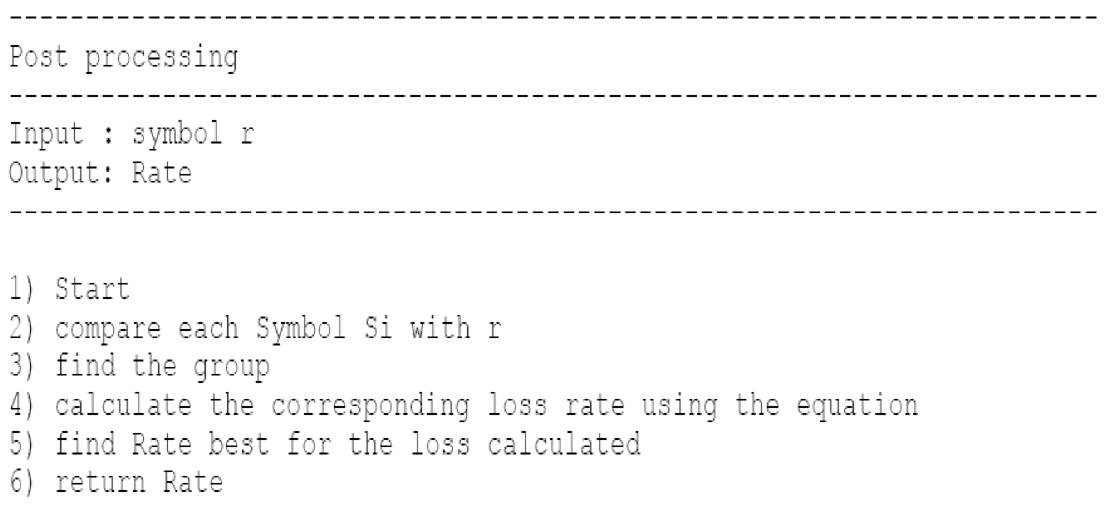
Algorithm 4: this algorithm demonstrates the decoding and the prediction phase. This part can be subdivided in to two stages. The first stage involves frequency assignment of each node in the tree build in the previous step. The frequency of a node depends upon the frequency of occurrence of a particular symbol in the initial stream of input symbols. The second stage involves taking the last word from the dictionary and calculating the probability of occurrence of each symbol in the alphabet *V*. probability calculation is started from right to left scanning of the last word and bottom to top scanning of the tree.



```
----------------------------------------------------------------------
Frequency assignment and Prediction
----------------------------------------------------------------------
Input : Tree T, Dictionary D, Alphabet V
Output: predicted Symbol r
----------------------------------------------------------------------

Start
initialize Root = NULL, current_node = Root
for each word Wi in dictionary D
do
  for each symbol Ai in word Wi
  do
    increment frequency of node Tj = Ai
  end loop
current_node = Root
end loop
for each letter Vi in V
 traverse from bottom of the tree to the top
 adding the frequency to the total
end loop
r = find the element with highest prediction value
return r
```

Algorithm 5: the last phase of the LeZiRate bit-selection technique which involves selecting the best rate based on the symbol predicted. This is done by selecting the symbol that the algorithm has predicted and matching it against the groups used in the first step. Once the group is selected the median value in dBm from the group is selected and corresponding loss rate is calculated using the loss function. The loss function approximately determines the optimum behavior of the modulation schemes and coding rates of the protocol at that loss point. The loss percentage determines the best rate that is possible under the conditions.



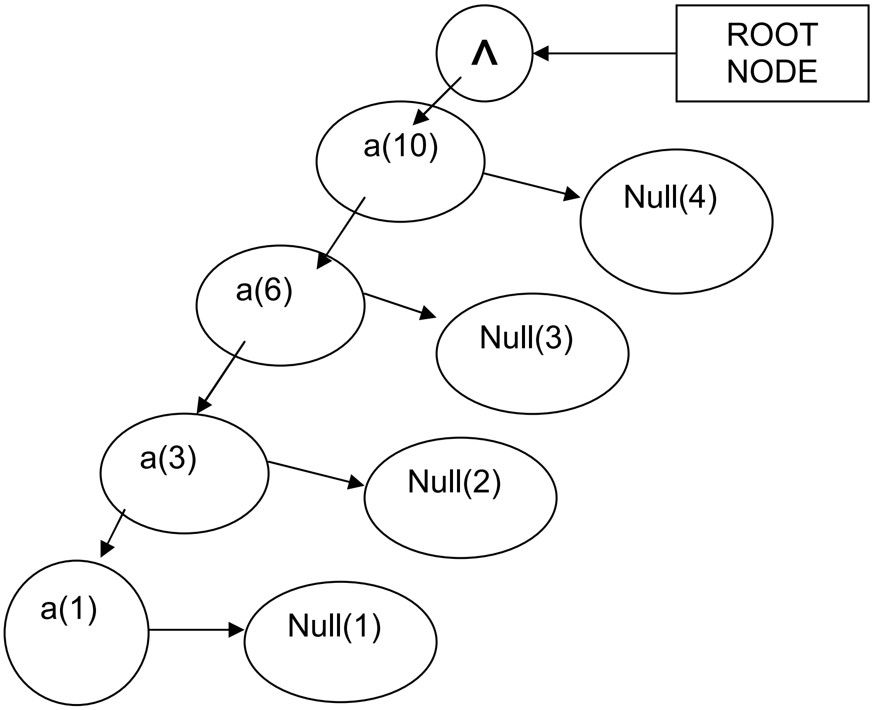
```
------------------------------------------------------------------------
Post processing
------------------------------------------------------------------------
Input : symbol r
Output: Rate
------------------------------------------------------------------------

1) Start
2) compare each Symbol Si with r
3) find the group
4) calculate the corresponding loss rate using the equation
5) find Rate best for the loss calculated
6) return Rate
```

To further illustrate the working of the algorithm let us see an example:

Let the sequence of characters be ‘aaaaaaaaaa’. The encoding phase will give the following dictionary. Dictionary D = {a, aa, aaa, aaaa} and the alphabet set V = { a }. The tree building phase will give the figure given below:

Tree T: =



The tree clearly is an unbalanced tree with only one alphabet ‘a’ . the last phrase in the dictionary is ‘aaaa’, therefore we would go down to that level and find the probability of the letter ‘a’.

The total probability will be therefore:

P(a) = 1/3 + 2/3 { 3/6+ 3/6 { 6/10 + 4/10} }

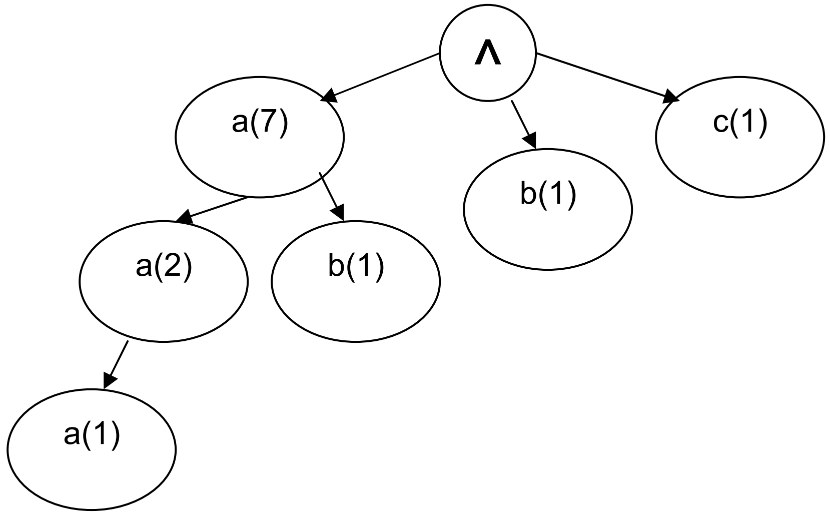
=> P(a) = 1/3 + 2/3 { ½ + ½ {1}}

=> P(a) = 1/3 + 2/3 {1}

=> P(a) = 1/3 + 2/3

=> P(a) = 1

So, the total probability of symbol ‘a’ turns out to be 1 which is the expected value. Let’s look at another example involving more than 1 symbol. Let the input array be ‘aaaabbcaaa’. The encoding stage would yield the following dictionary D = {a, aa, ab, b, c, aaa}. The alphabet set V = {a, b, c}. the tree building stage would yield the following tree T :=



Now, from the dictionary we see the last word was ‘aaa’. Therefore, we go down to the leftmost sub-tree to the second level and start calculating the prediction values of all the symbols in the alphabet set. The prediction calculations are:

P(a) = ½ + ½ { 2/7 + 4/7 (7/10)}

=> P(a) = 59/70

Similarly,

P(b) = 1/2 { 1/7 + 4/7 { 1/10} }

=> P(b) = 7/70

And,

P(c) = ½ { 4/7 { 1/10} }

=> P(c) = 2/70

From the above calculations it is evident that symbol ‘a’ has the highest probability of occurrence as the next symbol and therefore we select ‘a’.

4.3 Working of LeZiRate

LeZiRate algorithm works in two stages. It starts off by observing the signal strength reported by the hardware for a learning period of 500 milliseconds (default configuration) during that time it transmits at the 24 Mbps for the 802.11 a/g mode and 11 Mbps for the 802.11 b mode. The sample values from the window are then converted into a series of symbols using a static symbol table. The whole range of Signal-to-Noise power level reported by the hardware is divided into 10 groups. It was observed through experimental observations that the power level varied from -10 dBm to -90 dBm. The range was divided into the groups with each group comprising of 9 or 10 values. Each group was in turn mapped to a specific symbol from the English alphabet set using a static symbol table. Once it has a series of 10 symbols, it encodes this stream of symbols using the technique used in Lezi-Update [1] algorithm. Lezi-Update algorithm was devised to be a learning algorithm that could have a potential application for optimizing the paging strategy for mobile terminals. Therefore the stream is encoded into a dictionary of words to compress the total length of the stream. The size of the sampling window can be adjusted according the length of history the user wants to consider. The default configuration considers 10 samples from the past in a period of 500 milliseconds. The sampling frequency can also be adjusted according to the users wish. The default configuration has the sampling period as 50 milliseconds. The second phase is called the prediction phase which comprises of a Lezi tree being constructed using the LeZi-Update algorithm. The dictionary from the previous phase is then decoded to optimally construct the Lezi tree. The depth of the tree for a window size of 10 samples can be a maximum of 4 starting from the root node. Considering root node to be 0-order, the tree can have information till 3rd order Markov chain. This tree captures the history information and based on the most recent coded word from the sample; a path is traced back to the root calculating the probability of each node visited on the way starting from the leaf node. All the probable symbols are ranked according to the probability of occurrence. The symbol with the highest probability is chosen to be the next predicted symbol. Using this symbol the algorithm is able to predict the range in which the hardware will observe the received power in the next sampling period. This power level is used to calculate the link quality that should prevail during that period of time. For simulation sake the mapping of signal strength to loss rate is done through an exponential function constructed. This exponential function makes the following assumptions:

1) A high signal strength indicates good reception and therefore good link quality. It assumes very little loss at this level (approx 1.5% due to retransmission, collisions)

2) The low signal strength indicates just the opposite and therefore assumes high loss rate (approximately 98% due to control packets still being exchanged).

3) The loss characteristics do not exhibit uniform linear behavior but tends more towards exponential behavior.

Since this thesis deals with simulation work only, it is impossible to simulate actual channel behavior without increasing the complexity of the simulation program significantly. Once the amount of loss that this power level would experience is calculated, it is mapped to the rate that would optimize the throughput using a static table again. The new calculated rate is then set as the optimum transmission rate during the next sampling period.

4.4 Implementation

The LeZiRate algorithm was implemented as a set of C functions and not C/C++ to keep the implementation similar to the other existing ones. Some amount of pre-processing is done before the data can be fed into the algorithm. The raw data is converted to a stream of symbols using Perl code that extracts the signal-strength values from all the data that is reported by the hardware and then those signal strength values are converted into symbol stream.

The figure in the next page depicts a flowchart of the simulation program flow of control and the functions used in it.

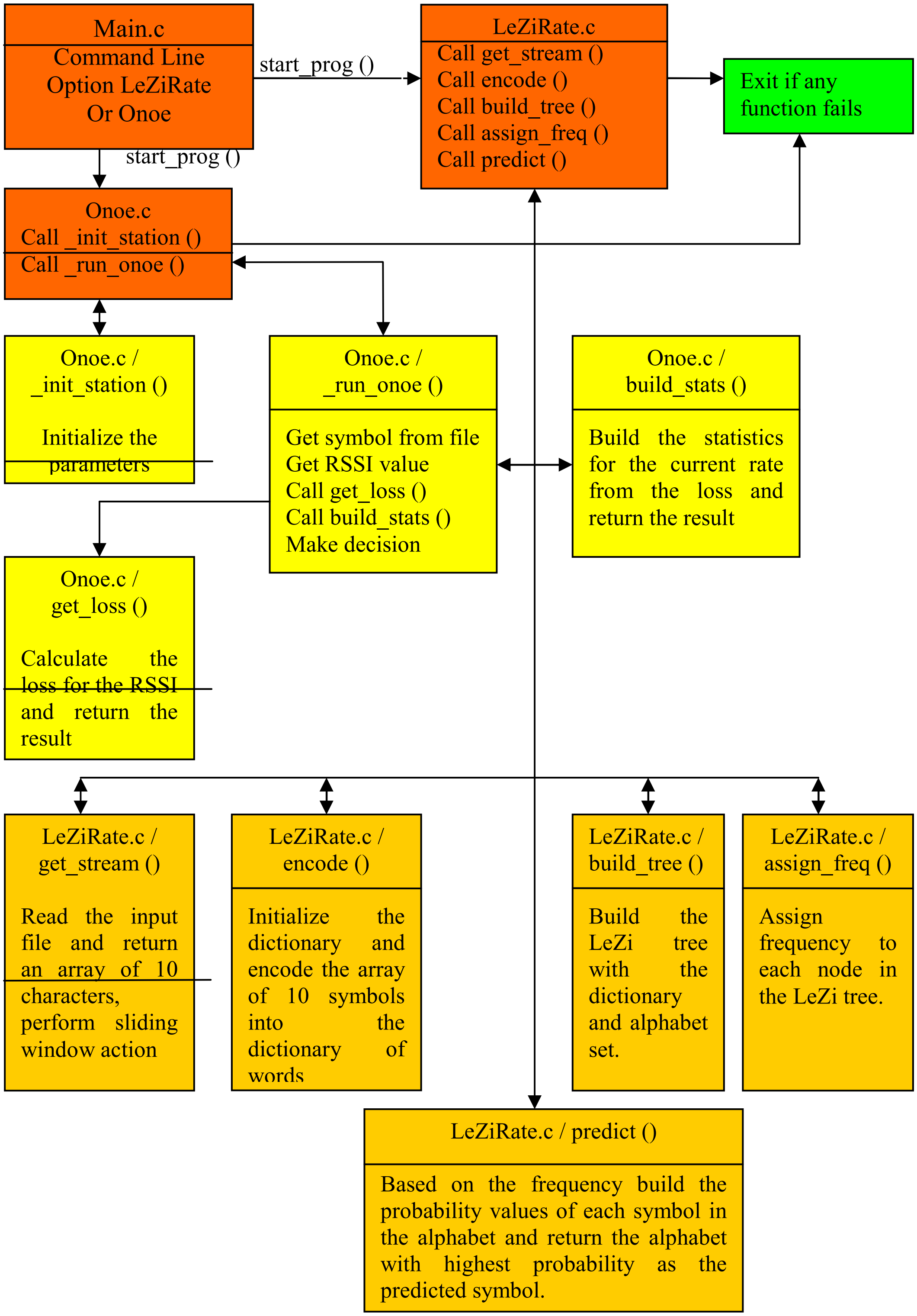

Main.c
Command Line
Option LeZiRate
Or Onoe
start_prog ()
start_prog ()
LeZiRate.c
Call get_stream ()
Call encode ()
Call build_tree ()
Call assign_freq ()
Call predict ()
Exit if any
function fails
Onoe.c
Call _init_station ()
Call _run_onoe ()
Onoe.c /
_init_station ()
Initialize the
parameters
Onoe.c /
_run_onoe ()
Get symbol from file
Get RSSI value
Call get_loss ()
Call build_stats ()
Make decision
Onoe.c /
build_stats ()
Build the statistics for the current rate from the loss and return the result
Onoe.c /
get_loss ()
Calculate the loss for the RSSI and return the result
LeZiRate.c /
get_stream ()
Read the input file and return an array of 10 characters, perform sliding window action
LeZiRate.c /
encode ()
Initialize the dictionary and encode the array of 10 symbols into the dictionary of words
LeZiRate.c /
build_tree ()
Build the LeZi tree with the dictionary and alphabet set.
LeZiRate.c /
assign_freq ()
Assign frequency to each node in the LeZi tree.
LeZiRate.c / predict ()
Based on the frequency build the probability values of each symbol in the alphabet and return the alphabet with highest probability as the predicted symbol.

Figure 4.1: Simulation program flow chart

There are 4 key functions in the algorithm implementation. The first one is: *encode( )*. This function encodes the symbol stream into words and builds the dictionary according to the Lezi-Update algorithm. The function takes the stream as the input and returns the dictionary containing the coded words along with the size of the longest word found in the dictionary. The dictionary is then passed on to the *build_tree( )* function that takes this dictionary as input and then effectively decodes the words and builds the tree taking each word encountered in the dictionary and adding nodes starting from the root node in a depth-first fashion. The root node in the tree does not have any symbol associated with it and is called the NULL prediction node. The tree is represented by an n-ary tree where n could be a maximum of 10 in the worst case scenario assuming default configuration. Each node is represented by a symbol and the path from the root node to the leaf node represents one word from the dictionary. The encoding in the previous step helps in creating a compact tree. The next key function is: *assign_frequency( )*. This function takes the tree and the dictionary as input and assigns a number representing frequency of occurrence of that symbol in the decoded dictionary. The frequency at each node represents the combined frequency of its child sub-trees. This assignment is again done in a depth first manner. The sum of frequencies of all the child sub-trees at the root level represents the sampling window size. Next comes *predict( )* function that takes the last coded word in the dictionary as the input and predicts occurrence of each of the symbols in the alphabet. The symbols not appearing in the current window are automatically assigned a value of zero. Once all the symbols have been assigned a prediction value they are sorted in descending order to find out which symbol has the highest probability of occurrence out of the whole alphabet. That symbol is the predicted symbol during the next sampling period. The appropriate loss factor is then calculated for each predicted signal-strength value and corresponding rate is calculated.

The loss function that was used is a purely exponential function and is given below with an example. This loss function was assumed and was developed using empirical knowledge of channel characteristics. It was developed assuming 98% percentage loss when the signal strength is -90dBm and only 1.5% loss when the measured power level is -10dBm. Further verification and actual channel characteristic curve will be a part of future research.

Let,

Y = Loss (in percentage %)

Base = constant = 0.9506

X = Signal Strength (dBm)

Therefore,

$Y = (\text{Base})^{X}$ or,

$\text{Loss} = (0.9506)^{(\text{signal strength})}$

Example:

Let Signal Strength = -50dBm then,

$\text{Loss} = (0.9506)^{(-50)}$

Loss = 12.592 %

The figure in the next page shows the plot of the loss function that was used.

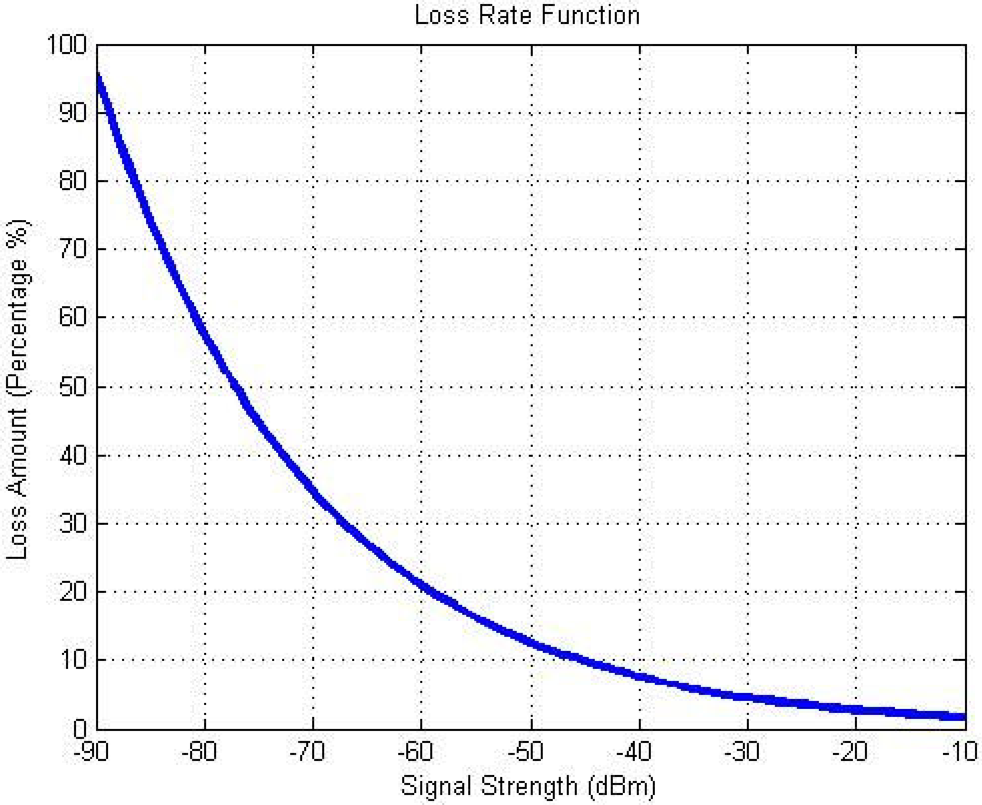


Figure 4.2: Graphical representation of loss rate function used in the simulation program

CHAPTER 5

EXPERIMENTATION AND SIMULATION

This chapter presents the details about the experiments that were conducted , an overview of the simulation program and its working. Simulating LeZiRate needed some amount of true data. It was imperative to get some idea of the behavior of actual link conditions to simulate Rate Control algorithms and compare behavior with the existing algorithms.

5.1 Experimental Setup

Experiments were conducted to collect data for signal strength variation for a mobile user. The aim of the experiment was to simulate a mobile user and also various types of channel conditions ranging from very high signal strength to a point where the terminal is on the verge of losing connectivity to the wireless network. A Centrino based Toshiba A55 laptop with 512 MB RAM (Random Access Memory) and 60 GB hard disk capacity was used for the experiment. The laptop was equipped with an inbuilt wireless card capable of connecting to both 802.11 b and g modes. Second set of experiments were conducted using the same laptop but equipped with a NetGear WAG511 802.11 a/b/g external card put in through the PCMCIA slot in the laptop. The laptop was configured with the dual boot option; the first operating system being Windows XP Service Pack 2 and the second operating system being Red Hat Linux 9.0 with an unpatched 2.4.3 kernel.

On the network side a simple 802.11 access point was used to provide outside network access. The access point was also configured not to use SSID broadcast. The cards were configured not to use RTS/CTS message exchange at all times. While using the NetGear wireless card the internal network card was disabled all the time during the experiment to make sure there was only one active interface on the terminal. A ping command was issued to an outside IP address to generate regular packets and simulate isochronous data application. All the packets were therefore unicast packets destined to a specific IP address. To simulate actual environment scenario, the access point was configured to use the default channel (Channel 11) for 802.11 b and 802.11 g mode. The terminal card was configured to report the transmission statistics by maintaining a Log file that would capture the required data. The steps for the experiment were as follows:

1) The access point was configured with the SSID "Bodhi".

2) The terminal was taken near to the access point and the wireless interface enabled and logging started.

3) The ping command was issued from the terminal window to an outside network IP address with the –t option (this would make the ping command continue without stopping till it receives an interrupt from the user).

4) The terminal was then slowly moved away from the access point along a predetermined path at a constant speed. The signal strength was constantly monitored to make sure no major interference causes a loss of connectivity.

5) The terminal was moved to a physical point where the signal strength measured was just enough to maintain connectivity but no data messages could go through.

6) The terminal was brought back to the original starting point through the same route.

7) The logging was stopped and ping command interrupted.

8) The wireless interface was disabled.

The data was then processed using Perl code to extract signal strength and time information from the log file. A total of 10 samples were collected with each experimental run conducted for approximately 30 minutes. The log file consisted of different parameters like date, Tx bytes, Rx bytes, IP address etc along with time and signal strength. This extracted data was then imported into MATLAB to plot a graph.

The figure in the next page shows the results from one such experimental run.

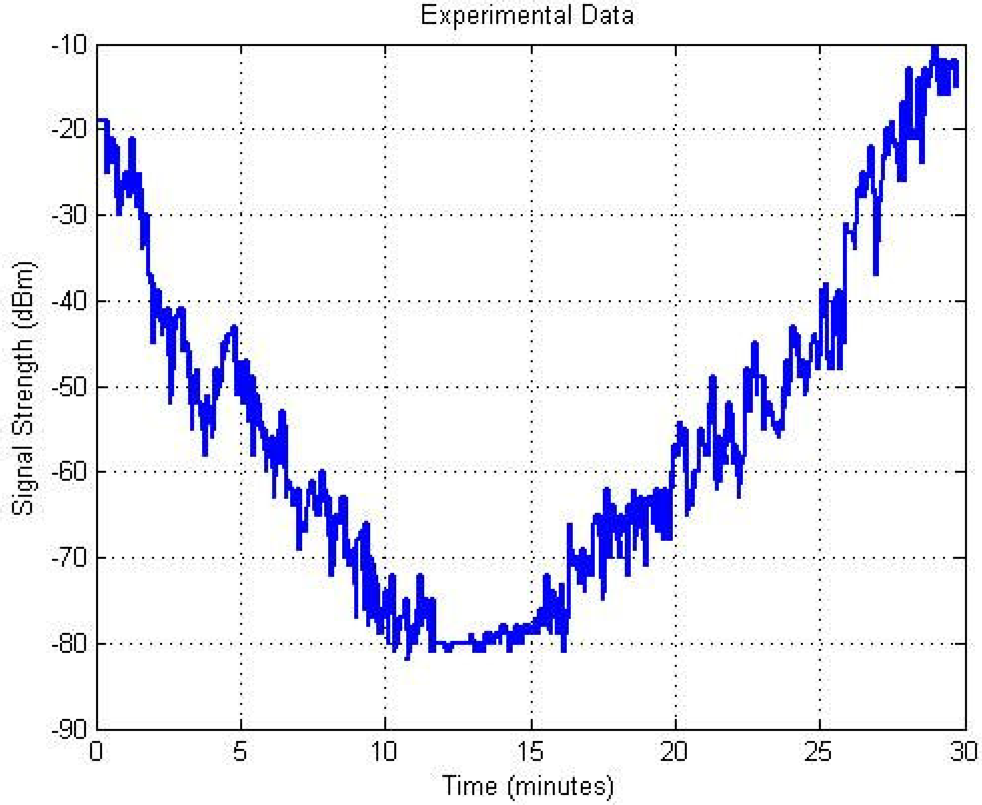


Figure 5.1: Graphical representation of experimental data

The X-axis depicts the timescale in terms of minutes and the Y-axis depicts the observed signal strength values in terms of dBm. The higher values in dBm represent good signal strength and the lower values represent low signal quality.

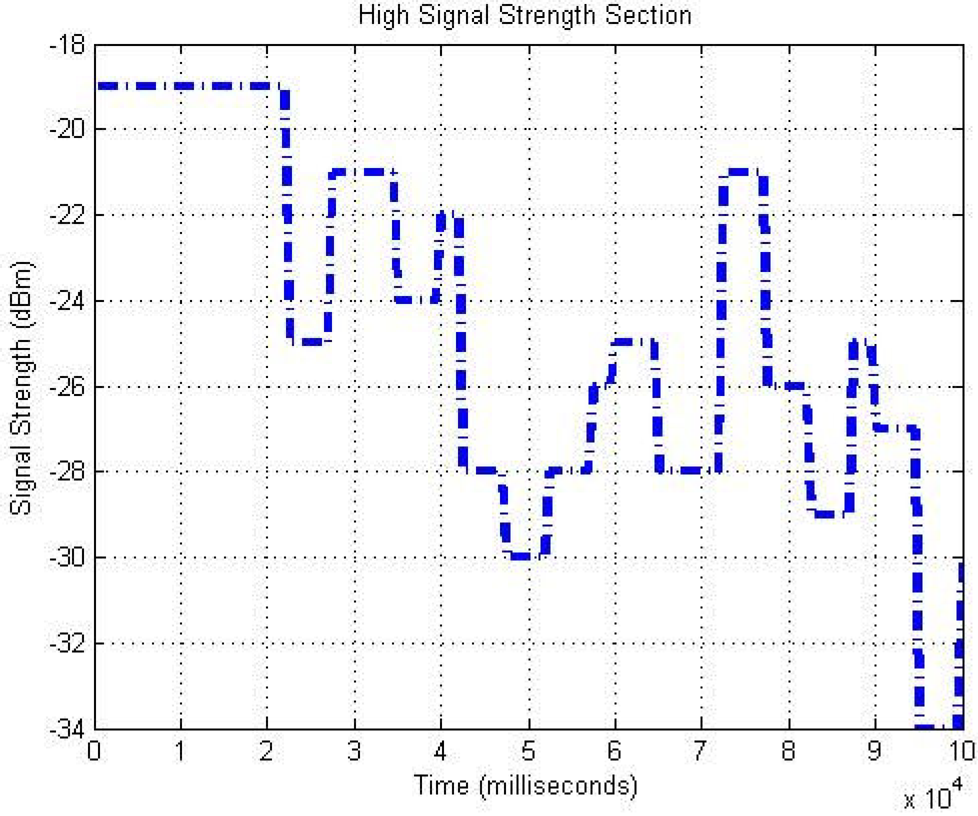


Figure 5.2: Graphical representation of high signal strength area of the experimental data

Figures 5.2 and 5.3 show snapshots from specific areas of the figure 5.1, the first one being the high signal strength section and the next one being from the low signal strength section.

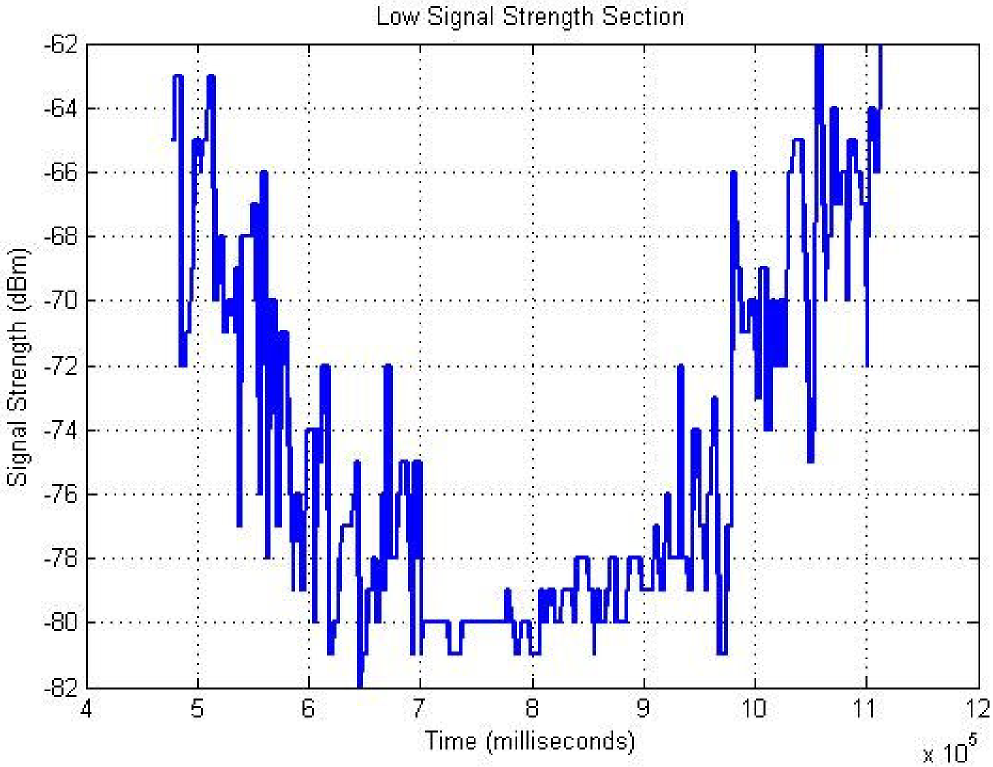


Figure 5.3: Graphical representation of low signal strength area of the experimental data

5.2 Simulation Program

The simulation program was built on server running Red Hat Enterprise Linux. The C compiler that was used for this is the GNU C compiler version 3.2.3. The program gives user the option to choose the algorithm to run for the simulation out of the two. For comparison with LeZiRate algorithm, Onoe was implemented in the simulation program. The user has an option of selecting algorithm to run by modifying a flag in the Makefile. GNU make then compiles code for appropriate algorithm and runs it against the data file that needs to be supplied at run time. Preprocessing is done to generate the data file to be used for the C program. Post processing is also done after the C program dumps out data for proper analysis. Both preprocessing and post processing is done with the help of two Perl programs. The final file dumped by the Perl program is then used for analysis using MATLAB.

CHAPTER 6

RESULTS AND CONCLUSIONS

This chapter presents the simulations results and the conclusions that can be drawn from the results. This includes data from simulation runs of both Onoe and LeZiRate Rate Control algorithms. All the comparisons are done using the same simulation environment and comparison was done using the same data set for both the algorithms.

The key points from the results of simulation are: Onoe moves in a step wise manner and therefore takes longer amount of time to stabilize, whereas LeZiRate is much faster and flexible to switching to a higher or lower rate. Onoe uses transmission statistics to gauge the link conditions and while taking time to stabilize uses network resources.

6.1 Results

This section presents the results that were observed from the simulation runs of both the algorithms. Onoe was selected for comparison for the following reasons:

1) The algorithm is documented and freely available in the open-source community.

2) The algorithm has a presence in the current implementation of the MADWiFi project.

3) Onoe is easier to implement in a simulation environment.

The figure below shows the simulation results from both the algorithms.

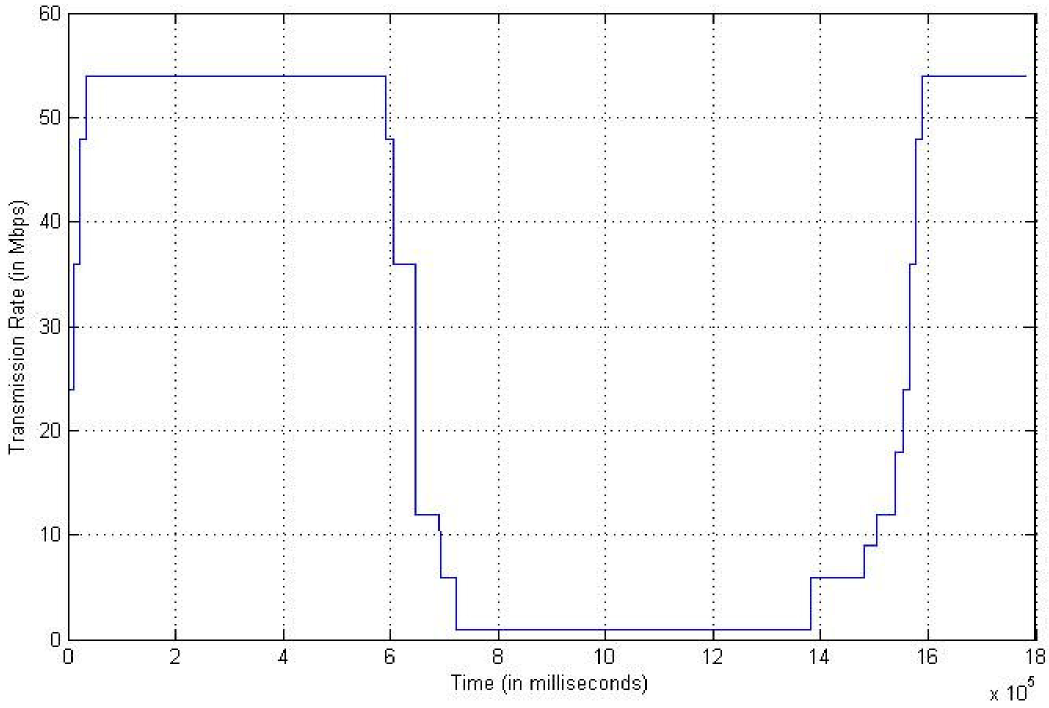


Figure 6.1: Performance of Onoe

Figure 6.1 shows the simulation run using Onoe as the Rate Control algorithm. The configuration for the timer of Onoe was selected to be the default one (1000 milliseconds). Onoe performs in a step wise manner as expected. It starts of at 24 Mbps and then keeps adjusting to the changing channel conditions. However it is conservative in nature and does not step up to higher rates fast enough, even though it could have been possible at some of the instances. It has been shown from pervious work that Onoe does not work well in low quality links [2][3].

Figure 6.2 shows the simulation results of the LeZiRate algorithm.

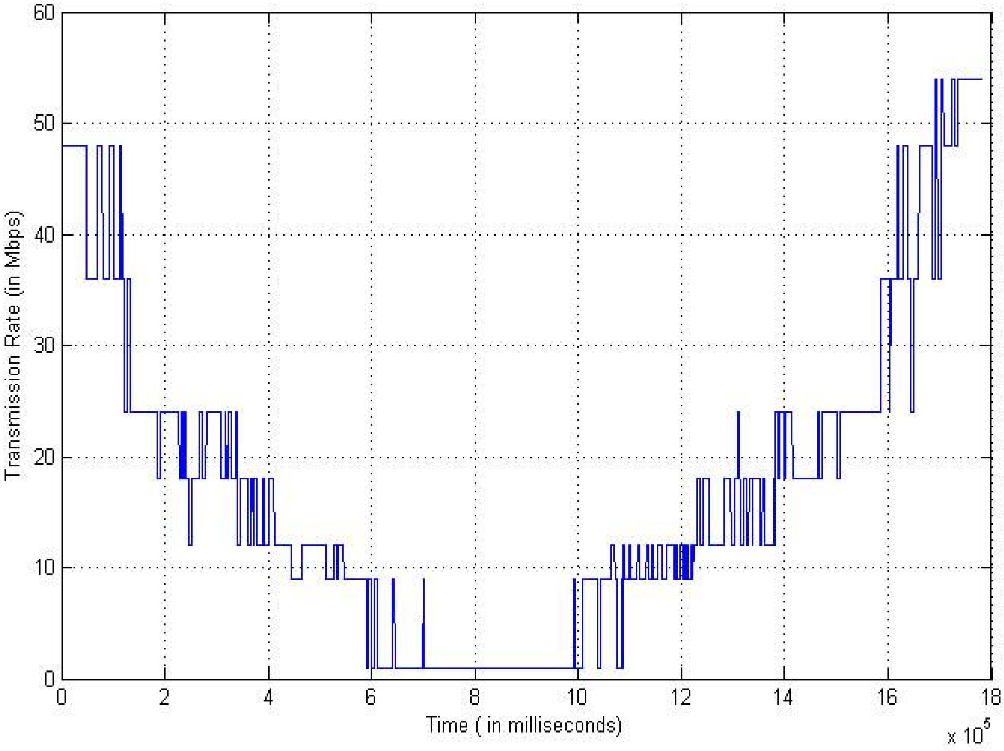


Figure 6.2: Performance of LeZiRate

The second figure shows the performance of LeZiRate algorithm. It is evident from the two graphs that LeZiRate is more sensitive to the changing channel conditions. It also does not move in a step wise fashion and is more flexible in switching from one rate to another. The figures in the next page compare the performance of Onoe with respect to LeZiRate. The first figure shows Onoe performance chart on top of LeZiRate performance chart. The second figure depicts a portion of the figure, it depicts the recovery phase of the two algorithms from the low signal strength phase. the figure also shows the average throughput of the two algorithms during that time period.

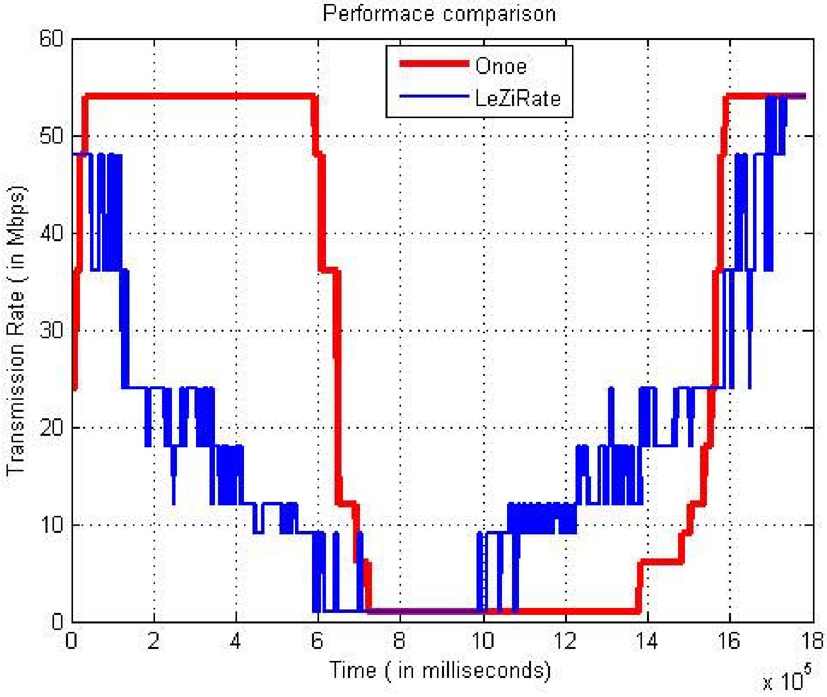


Figure 6.3: Performance of Onoe Vs LeZiRate

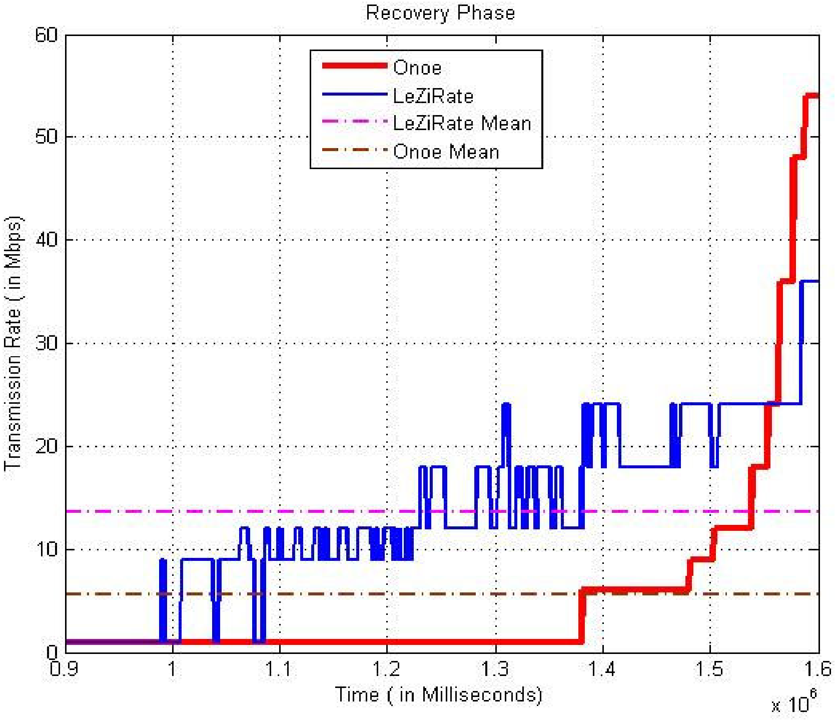


Figure 6.4: Recovery of Onoe Vs LeZiRate from low signal strength area

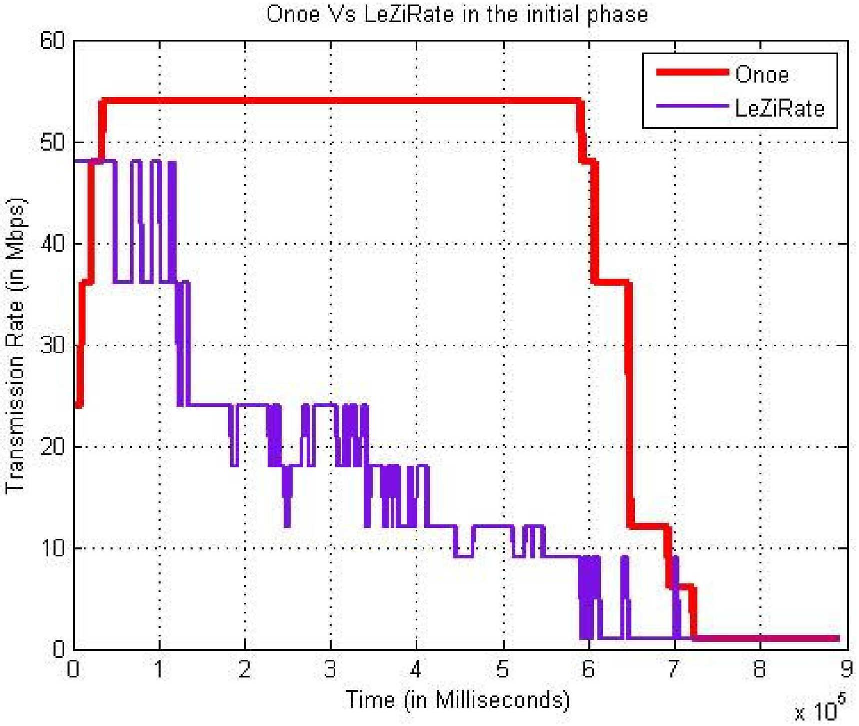


Figure 6.5: Initial Phase

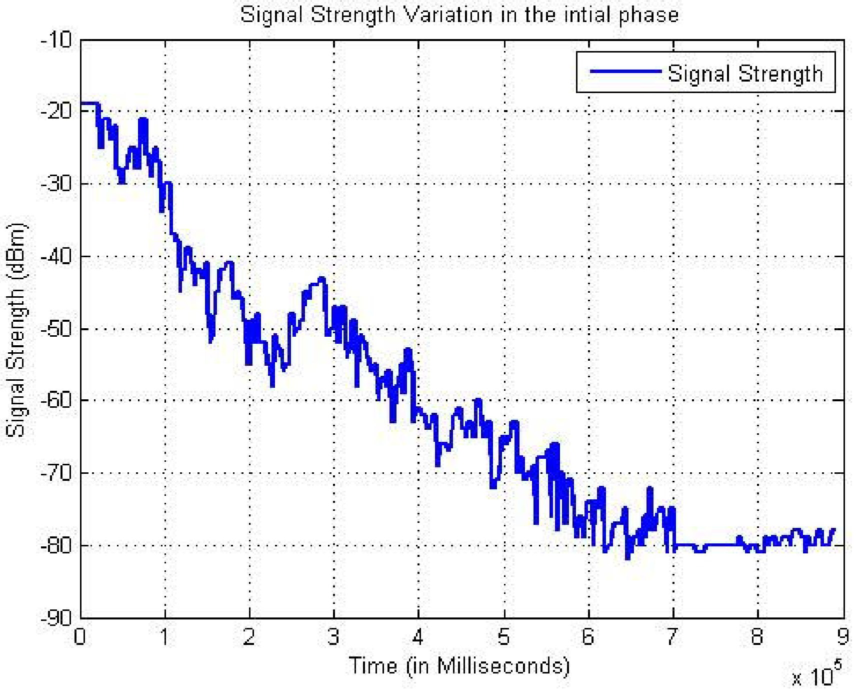


Figure 6.6: Signal Strength in the initial phase

Figure 6.6 depicts the variation in the signal strength during the initial phase and figure 6.5 depicts the performance of the two bit-selection algorithms during that phase. comparing the two figures we can see that LeZiRate follows the signal variation more closely. It adapts the rate quickly as the signal strength keeps diminishing. Onoe on the other hand does not follow it and suddenly drops the rate when the signal strength falls to a low level. By this we can predict that Onoe suffers loss during that time period, although it tries to maintain a high rate. this is observed due to the sluggish behavior of Onoe. From this we can infer that LeZiRate is more sensitive and follows the signal variation more closely and therefore able to extract more out of the channel resources with minimal network resource wastage.

Similarly, from figure 6.4 it can be seen that Onoe takes considerable amount of time to recover from the low signal strength phase and therefore keeps transmitting at 1Mbps. At this speed there is no data packet transmission and only management and control packets are exchanged. In comparison LeZiRate is quick to recover and starts transmitting at higher rates when possible. Therefore it achieves a better average throughput than Onoe during that time period.

LeZiRate also uses sliding window mechanism to get the symbols. During the initial learning phase it takes 10 symbols. After the initial phase it only takes 1 symbol every time. Therefore it slides the window 1 symbol at a time. This also helps LeZiRate in correcting the symbol if it made a mistake in prediction.

6.2 Conclusions

This thesis presented a simulation study and comparison of a new Rate Control Algorithm called the LeZiRate with one of the existing algorithms in use on the wireless network interface cards. LeZiRate uses two-phase mechanism to learn from the recent past quality of signal and predict the future quality of the signal strength. Based on the predictions it predicts the best bit-rate that would minimize the transmission loss and optimize the throughout. It does not make use of the network bandwidth at all and only passively monitors the channel condition. LeZiRate using the sliding window techniques manages to keep history information about the mobility trend of the user. However, LeZiRate will fail to exploit the history information if the channel conditions show a non-predictable and impulsive behavior where in the signal strength keeps fluctuating over a large range in a very short period of time. However it would be possible to compare the performance of the two algorithms more accurately and realistically after implementing the LeZiRate in the driver code and conducting actual experiments on real time wireless environments.

Future work on this would mainly comprise of devising algorithms that take mixed approach towards finding the link quality and not just relying on the signal strength while having the ability to maintain history information about the channel quality. Minimizing the use of network bandwidth for decision making would greatly decrease the amount of penalty imposed for taking a wrong decision.

REFERENCES

[1] Amiya Bhattacharya and Sajal Das, LeZi Update: An Information-Theoretic Framework For Personal Mobility Tracking in PCS Networks. Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking, August 1999.

[2] Sourav Pal, Sumantra R. Kundu, Kalyan Basu and Sajal Das, IEEE 802.11 Rate Control Algorithms: Experimentation and Performance Evaluation in Infrastructure Mode, 2006.

[3] J.C. Bicket, Bit Rate Selection in Wireless Networks, M.S Thesis, MIT, February 2005.

[4] MADWiFi Project, http://madwifi.sourceforg.net

[5] M. Lacage, M. Hossein and T. Turletti, IEEE 802.11 Rate Adaptation: A Practical Approach, IEEE MSWiM, October 2004.

[6] Matthew S. Ghast , 802.11 Wireless Networks, The definitive guide, O'Reilly Publications.

[7] J. Heiskala and J. Terry, OFDM Wireless LANs: A theoretical and practical guide, SAMS, 2001.

[8] Douglas S. J. De Couto, Daniel Aguayo, John Bicket, and Robert Morris, A high-throughput path metric for multi-hop wireless routing. In Proceedings of the 9th ACM International Conference on Mobile Computing and Networking (MobiCom '03), San Diego, California, September 2003.

[9] Mark Yarvis, W. Steven Conner, Lakshman Krishnamurthy, Jasmeet Chhabra, Brent Elliott, and Alan Mainwaring, Real-world experiences with an interactive ad hoc sensor network. In Proceedings of the International Workshop on Ad Hoc Networking, August 2002.

[10] M. V. Clark, K. K. Leung, B. McNair, and Z. Kostic, Outdoor IEEE 802.11 cellular networks: Radio link performance. In Proc. of IEEE ICC 2002, April 2002.

[11] D.C. Cox, Delay Doppler characteristics of multipath propagation at 910 MHz in a sub-urban mobile radio environment. In IEEE Transactions on Antennas and Propagation,AP-20(5):625-635, 1972.

[12] A. Kamerman and L. Monteban, WaveLAN-II: A high-performance wireless lan for the unlicensed band, AT&T Bell Laboratories Technical Journal, pages 118-133, 1997.

[13] Wikipedia, http://en.wikipedia.org/wiki/IEEE_802.11

[14] http://www.isoc.org

[15] http://brown.usc.edu/~nahm/research.htm

BIOGRAPHICAL INFORMATION

Bodhisatwa Chakravarty was born in Kolkata, India in November 1981. He received his Bachelors degree in Computer Science and Engineering from Delhi, India in July 2003. After completing his undergraduate studies he worked as a student software developer at Indian Institute of Technology (IIT), Delhi, India. He also worked for IBM India before coming to UT Arlington to pursue his graduate studies. He worked as an intern in the Mobile Devices Division (MDD) at Microsoft Corporation, Redmond while pursuing his graduate degree in UT Arlington. He is currently working for Texas Instruments in Dallas as a wireless software engineer. His research interests include networking, wireless networking in 802.11 networks.