This is to certify that the
thesis entitled

ON SNR AWARE ANALYSIS AND MODELING OF 802.11B
LINK-LEVEL RESIDUAL ERRORS

presented by

UTPAL M. PRABHU PARRIKAR

has been accepted towards fulfillment
of the requirements for the

M.S. degree in Electrical Engineering

Subir Mishra
Major Professor's Signature

May 10, 2006
Date

ON SNR AWARE ANALYSIS AND MODELING OF 802.11B LINK-LEVEL RESIDUAL ERRORS

By

Utpal M. Prabhu Parrikar

A THESIS

Submitted to
Michigan State University
In partial fulfillment of the requirements
for the degree of
MASTER OF SCIENCE

Department Of Electrical And Computer Engineering

2006

ABSTRACT

ON SNR AWARE ANALYSIS AND MODELING OF 802.11B LINK-LEVEL RESIDUAL ERRORS

By

Utpal M. Prabhu Parrikar

In an 802.11b cross-layer protocol, the utility of a corrupted Medium Access Control (MAC) frame is dependent on residue error patterns. Previous attempts at modeling residual errors have been oblivious to the radio-link quality (SNR oblivious). In this thesis we show that modeling techniques, which take into consideration the radio link-quality appropriately describe the error behavior across varied environments and thus perform better. We do so by demonstrating that the memory length of the error process, the average frequency of bit errors and thus the useful information content in MAC frame varies significantly with respect to SNR and hence cannot be appropriately captured by single model. Hence in this thesis we address the difficult and previously un-addressed task of characterizing the residue error performance across different environments with a single model. Firstly we show that the average bit error rate in a MAC frame has a fixed relationship with the associated SNR across varied environments. This observation is used to motivate the idea that SNR values can be used to adapt the model parameters and thus allow a single model to represent the error process in diverse setups. Therefore, we develop a non-homogenous Markov model, whose state transition probabilities can be altered as a function of SNR.

In loving memory of my mother...

ACKNOWLEDGEMENT

This thesis is the result of the inspiring and thoughtful guidance and supervision of my advisor Dr. Lalita Udpa. But for her this work would not have been possible. I would like to extend my deepest gratitude to her for supporting and guiding me from my early days at Michigan State University.

I would also like to thank my co-advisor Dr. Hayder Radha for encouraging me to pursue the problem discussed in this work, and also giving valuable inputs and directions to realize it. I am also grateful to Wireless And Video Communications lab for all the technical support provided along the way in this endeavor.

I also thank Dr. Peixin Zhong for his accommodating attitude as member of my committee. I thank him for providing me with this opportunity to present my work.

I would also like to gratefully acknowledge the support of some very special individuals. They helped me immensely by giving me encouragement and even more importantly, friendship. Shirish and Kiran mirrored back my ideas and helped me refine them. Their contribution to my overall academic experience cannot be summarized in such a short space.

Lastly, I would like to appreciate all the support and motivation provided to me by my family, and words simply cannot express my appreciation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

True computing mobility is not the pipe dream it used to be. Smaller and faster processors, and expanded Operating Systems are paving the way for a richer mobile computing experience. This increasing availability of the mobile computing devices has fueled the need for high-speed wireless networks. Once considered purview of large universities and research labs, these days' wireless networks are more and more deployed in everyday home and office environments. Recent years have seen an increased deployment of 802.11b based Wireless Local Area Networks (WLANs) through ready availability of the required wireless hardware in nearby electronic stores. Concurrent with this trend there has been an increased demand for multimedia applications. The above two synergistic growths have in turn led to an increased demand for seamless availability of multimedia content over wireless media. However, often in practical deployment, the wireless networks suffer from errors and losses in the presence of network congestion and transmission medium degradation. This decrease in throughput can adversely affect the performance of network applications, especially the multimedia applications.

Traditionally the network provides reliability by recovering the corrupted/lost packet through retransmissions, method which has degrading influence on the performance of multimedia applications due to their real-time delay-sensitive nature. However, multimedia applications show some degree of tolerance to such errors (especially those delivering streaming media)-, and as a result, some of the recent studies have advocated a cross-layer error-control strategy that can recover data even from the corrupted packets and improve throughput. The utility/feasibility of data recovery from corrupted MAC frames is a function of the residue error patterns observed at the link-level[1], and consequently they appear within (corrupted) packets at the link-layer and possibly at other higher layers. This makes it imperative to develop a thorough understanding of error and loss patterns observed over the network.

This thesis analyzes and models the behavior of the actual residue error patterns observed at the 802.11b link-level based on crucial parameters and side information that can be collected at the physical layer. This work focuses on utilizing the Full State Markov (FSM) Chains for all modeling purposes.

1.1 Motivation

Wireless channels generally exhibit complex behavior with increased sensitivity to surrounding environment as compared to wired media. It is not unnatural to presume that residue errors are observed in significant proportions only when the receiving node is not within the range of the Access Point. Wireless network is susceptible to phenomenon such as reflection, interference from nearby network and other wireless devices (cordless phones operating at 2.4 GHz are known to have caused interference in 802.11b networks). Thus a natural question to ask is whether the information content in a corrupted packet

[1] residue error represents bit errors that are not corrected by the physical layer

received at high Signal-To-Noise Ratio (SNR) is similar to that at low SNR values? Furthermore, would a cross-layer error control scheme used in low-noise environments be suitable for more severe conditions too. All such questions necessitate the need for an analysis that takes the radio-link quality into consideration.

Previous attempts (specific to 802.11b) at modeling such residue errors have been oblivious of the radio-link parameters. The stochastic behavior of an error process can be drastically different in packets received at different SNR-levels. Generating different residue error models for different SNR make the models more reliable for analysis, simulations and emulations. Later chapters of this thesis will help establish some of the following observations: (i) Residue errors are observed over a significant range of SNR values and thus the variations in behavior of the residue error process over a range of SNR values should be investigated. (ii) The second order statistics in terms of log-variance varies significantly from one environment to another and thus there is a need for a modeling technique that is not environment dependent. (iii) The average bit error rate in a corrupted MAC frame, when expressed as a function of SNR has a constant relationship across different environments. In addition it is shown that the bit error rate varies significantly as a function of SNR Thus an SNR aware modeling scheme could capture the stochastic behavior of the error process better than an SNR unaware model and could also be environment independent.

1.2 Objective

The primary objective of this research effort is developing an SNR aware non-homogeneous Markov model that alters the state transition matrix as a function of SNR. We measure the performance of the full-state Markov chains (FSM) in terms of the ability of the synthesized data to replicate the features of the actual error process. The features are defined in terms of random variables such as Inter-arrival rate (I), burst-length (B) and the frequency of errors per packet (p). The model is created by analyzing residue error at MAC layer of 802.11b network. We will evaluate the performance of the models when the training data and test data are from the same environment and also when the training data and test data are from varied environments.

We focus on bit-level analysis and modeling but while keeping track of packet boundaries, i.e. we use packet Frame Check Sequence (FCS) to determine whether the packet is good or bad, and only perform bit-level analysis/modeling on corrupted packets. This way we make full use of the FCS, which is the best indicator to determine if packet is good or bad, and restricting our model to generate error patterns only for bad packets.

1.3 Thesis Outline

The remainder of this thesis is organized as follows. Chapter 2 provides essential background information about 802.11 wireless networks. We discuss 802.11 network in infrastructure (WLAN) mode, access methodology and associated network functionality. Chapter 3 starts with the discussion about the environments in which error traces were collected, and the methodology employed to collect the traces. We perform packet- and bit-level analysis of the observed error process at 5.5 and 11 Mbps bit-rates of the 802.11b network. We compare the difference in the observed error process for these two bit-rates. Chapter 4 proceeds to the modeling aspect of this work. We discuss the generation of SNR Aware as well as SNR Unaware (conventional) Full State Markov (FSM) Model from the data collected at the two bit-rates. Performance of these models is quantified using Entropy Normalized Kullback-Leibler divergence, a standard Information Theoretic measure. Finally, Chapter 5 offers a few concluding remarks and suggests areas for future explorations.

Chapter 2

BACKGROUND

2.1 IEEE 802.11 And Wireless Local Area Networks

An 802.11 Wireless Local Area Network (WLAN) is based on cellular architecture where the system is sub-divided into cells where each cell called Basic Service Set is controlled by base station called Access Point (AP). Even though WLAN can be formed by just single BSS with one AP, most installations are formed by multiple BSS cells connected by some kind of backbone called Distribution System (DS) typically using Ethernet or in some cases Wireless itself. The whole interconnected WLAN including the different cells, their respective Access Points and Distribution System, is seen by the upper layers of OSI model, as a single 802 network, and is generally called Extended Service Set (ESS). The figure 2.1 shows an ESS with all its components.

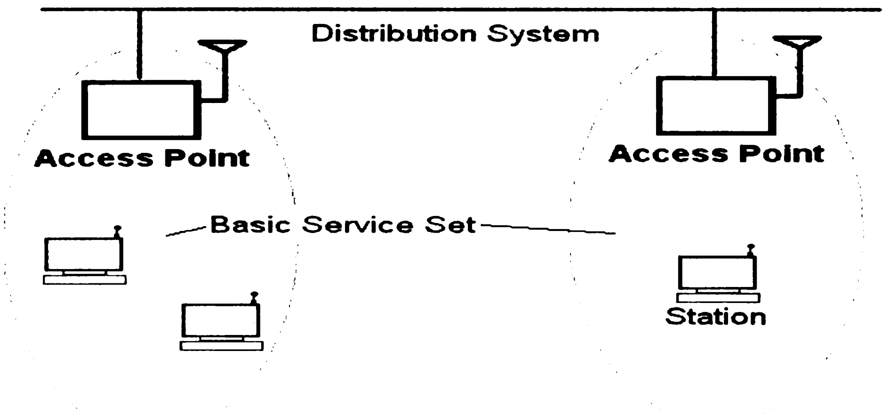


Figure 2.1 IEEE 802.11 WLAN Extended Service Set

2.1.1 802.11 Layers

The 802.11b protocol as shown in the figure 2.2 covers the Physical and Medium Access Control (MAC) layers. The Physical layer can be of three types, Frequency Hopping Spread Spectrum (FHSS) in 2.4 GHz band, Direct Sequence Spread Spectrum (DSSS) in 2.4 GHz band, and Infrared (IR). The MAC layer besides performing the standard functionality of MAC layers also performs other functions that typically relate to upper layers protocols, such as Fragmentation, Retransmission and Acknowledgments. The MAC layer defines two access methods, the Distributed Coordination Function and the Point Coordination Function.

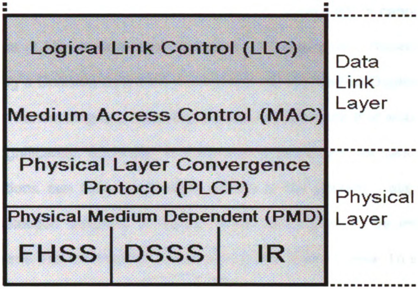


Figure 2.2 IEEE 802.11 OSI Layers

2.1.2 Medium Access Mechanism

The basic access method called Distributed Coordination Function, is basically a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) mechanism. In CSMA, the station desiring to transmit senses the medium, if the medium is busy (i.e. if some other station is transmitting) the station will defer its transmission to a later time, if the medium is sensed free the station is allowed to transmit. This is effective when the medium is not heavily loaded, but there is always a chance of stations transmitting at the same time (collision), caused by the fact that both the stations sensed the medium to be free and decided to transmit at the same time. This collision needs to be identified so the MAC layer can retransmit the packet by itself and not rely on upper layer for retransmission as this would cause significant delay. Ethernet uses Collision Detection and then goes to retransmission phase based on Exponential Random Back-off algorithm.

While the Collision Detection method is good on Wired LAN, it cannot be used on Wireless LAN environment because of two main reasons. Firstly, implementing a Collision Detection mechanism will require Full Duplex radio with capability of transmitting and receiving at once, an approach that would increase the price significantly. Secondly, in a wireless environment we cannot assume that all stations can hear each other (which is the primary . assumption of Collision Detection scheme). A station which senses the to be medium free cannot assume that the medium is free around the receiver area. To avoid these problems, the 802.11 uses Collision Avoidance (CA) mechanism together with Positive Acknowledgment scheme. A station willing to transmit senses the medium, if the medium is busy then it differs. If the medium is free for specified time called Distributed Inter Frame Space (DIFS), the station is allowed to transmit. The receiving station checks the CRC of the received frame and sends an acknowledgment packet (ACK). The receipt of the ACK at transmitting station will indicate that no collision occurred. If the sender does not get the ACK it retransmits the fragment until it gets the ACK or is thrown away after given number of retransmissions. In order to reduce the probability of two stations colliding because they cannot hear each other, the 802.11 standard defines a Virtual Carrier Sense mechanism. The station willing to transmit will first transmit a small control packet called Request To Send (RTS), which will include the source, destination and the duration of the following transmission (i.e. the packet and the respective ACK). The destination station will respond (if the medium is free) with Clear To Send (CTS) which will include the same duration information. All the stations either receiving RTS and/or CTS, will set their Virtual Carrier

Sense indicator called Network Allocation Vector (NAV) with this duration. Stations, which are not the RTS sender, will refrain from using the medium for this duration, thus reducing the probability of collision. Also, since RTS and CTS are small frames, it also reduces overhead of collisions since these are recognized faster. So the standards allow for direct transmission of smaller packets whose size is controlled by a RTS Threshold parameter set at each station. Figure 2.3 shows the transaction between two stations A and B and the NAV setting of their neighbors.

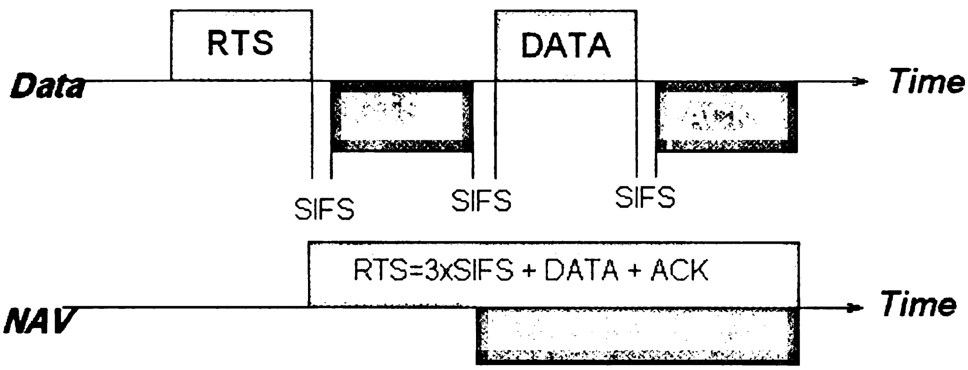


Figure 2.3 802.11 Packet Transaction

2.1.3 MAC Layer Acknowledgment, Fragmentation and Reassembly

As mentioned earlier the MAC layer expects acknowledgments from the receiving station for each transmitted frame to perform Collision Detection. Exceptions to this are frames transmitted to multiple destinations (Multicast). Besides acknowledgments the MAC layer performs fragmentation and reassembly of the packets. Typical LAN protocols use packets of several hundreds of bytes, on a WLAN environment it would be preferable to use packets of smaller size. The reasons being, that the higher bit error rate of radio link causes the probability of packet to get corrupted increase with packet size. In case of packet corruption, smaller the packet, lesser the overheads to retransmit it. In a frequency hopping system the medium is interrupted periodically for hopping, so the smaller the packet, the smaller the chance that the transmission will be postponed to after the dwell time. On the other hand it does not make sense to introduce new LAN protocol which cannot deal with existing larger size packets of the current wired LANs, so a simple Fragmentation/Reassembly scheme is introduced. A large packet is broken down into many smaller fragments and a simple send and wait algorithm is followed, where a station is not allowed to transmit a new fragment until either, it receives an ACK for the said segment, or decides that fragment was transmitted too many times and drops the whole frame. A station is allowed to transmit to a different address in between retransmissions of a fragment.

2.2 Entropy of a Random Experiment

Entropy is a measure of average information contained in a random variable *X*. If *X* is a discrete random variable on a finite set $X = \{x_1, x_2 ..., x_n\}$, with probability distribution function *p(x) = Pr(X=x)*. The entropy *H(X)* of *X* is defined as,

$$H(X) = -\sum p(x)\log(p(x)) \tag{2.1}$$

where p(x) is the probability mass function used to represent the random experiment.

The logarithm is usually taken to the base 2, in which case the entropy is measured in ``bits," or to the base e, in which case H(X) is measured in ``nats." The Entropy can be also seen as uncertainty removed after the actual outcome of X is revealed.

2.3 The Kullback-Leibler Distance (Relative Entropy)

If p and q be the two probability distribution function on the random variable X. Then the Kullback-Leibler distance[2] is the statistical measure that quantifies the difference between p and q and is defined as,

$$D(p \| q) = \sum p(x)\log\left(\frac{p(x)}{q(x)}\right) \tag{2.2}$$

Thus the Kullback-Leibler distance provides a non-negative statistical divergence measure which is zero if and only if p=q. But since $D(p \| q) \neq D(q \| p)$ that is, it is non symmetric and violates the triangle inequality, it is not a true metric. Also, it can be seen that the Kullback-Leibler distance depends on the choice of the random variable, or in other words, choice of probability distribution function, and hence a proper random variable should be selected to represent

[2] The Kullback-Leibler distance or Relative Entropy is also known as Information Divergence.

the random observation.

The Kullback-Leibler distance is one of the most important tool for this work, as it will help quantify the performance of the SNR aware, as well as, SNR unaware models developed. From a source coding viewpoint, assume p represents the probability distribution function of the actual source, and let q represent the distribution of the model approximating the source. Then the divergence $d = D(p \| q)$, quantifies the statistical divergence between the actual source and the approximating model. It also represents the overhead incurred by using an approximation instead of the actual source. Hence, Relative Entropy can be used to measure the performance of different models developed. However, since the relative entropy measures overhead in bits, it is better to weigh this measure in accordance with entropy. For example, the model having overhead of few extra bits for high entropy source might be better than the model with less overhead with low entropy source. Hence, both entropy and the Kullback-Leibler distance should be taken into consideration.

2.4 Entropy Normalized Kullback-Leibler distance (ENK)

The entropy function provides a measure of average number of bits required to represent a source. Also, as seen earlier, the Kullback-Leibler distance provides a measure of extra bits required due to use of an approximating model instead of the actual source. Both these measures can be used collectively to indicate the level of overhead incurred by a particular model. We define a new measure called Entropy Normalized Kullback-Leibler measure

(ENK), as the ratio of Kullback-Leibler distance and the entropy, i.e.,

$$ENK(p \| q) = \frac{D(p \| q)}{H(p)} \tag{2.3}$$

We will use this measure, ENK, to evaluate the performance of different models developed in this work.

2.5 Mutual Information

Let (Ω,F,μ) be a discrete probability space, and let *X* and *Y* be discrete random variables on Ω. The mutual information *I[X;Y]*, read as ``the mutual information of X and Y," is defined as,

$$I[X;Y] = \sum\sum \mu(X = x, Y = y)\log\left(\frac{\mu(X = x, Y = y)}{\mu(X = x)\bullet\mu(Y = y)}\right) \tag{2.4}$$

$$D(\mu(x,y) \| \mu(x)\mu(y)) \tag{2.5}$$

where D denotes the relative entropy.

The most obvious characteristic of mutual information is that it depends on both *X* and *Y*. There is no information in a vacuum *i.e.*--information is always about something. In this case, *I[X;Y]* is the information in *X* about *Y*. As its name suggests, mutual information is symmetric, i.e. *I[X;Y]* = *I[Y;X]*, so any information *X* carries about *Y*, *Y* also carries about *X*. The definition in terms of relative entropy gives a useful interpretation of *I[X;Y]* as a kind of ``distance" between the joint distribution μ(x,y) and the product distribution μ(x)μ(y). However, the relative entropy is not a true distance, but just a conceptual tool, and it does capture another intuitive notion of information. It is important to note that for *X*, *Y* to be independent, μ(x,y)=μ(x)μ(y). Thus the relative entropy ``distance" goes to zero, and we have *I[X;Y]=0* as one would expect for independent random variables.

2.6 Markov Chains

Markov chains are extensively employed in modeling various processes in queuing theory and statistics. In particular Markov chains are effective at state estimation and pattern recognition. Consider a stochastic process, X_n, where value of X_n is the state of the process at time n. So if the process is in state *i*, i.e. $X_n = i$, there is a fixed probability that the next state of the process is *j*, i.e. $X_{n+1} = j$. If this probability can be expressed as,

$$P\left\{X_{n+1} = j \mid X_0, X_1, \ldots\ldots, X_n\right\} = P\left\{X_{n+1} = j \mid X_n = i\right\} \tag{2.6}$$

for all states $i_0, i_1, \ldots, i_{n-1}, i, j$ and $n \geq 0$ then such a stochastic process is known as Markov Chain.

A simple way to visualize a specific type of Markov chain is through a finite state machine. If you are in state y at time n, then the probability that you will move on to state x at time n+1 does not depend on n, and only depends on the current state y that you are in. Hence at any time n, a finite Markov chain can be characterized by a matrix of probabilities, called transition probability matrix, whose (i,j) element is given by $P\left\{X_{n+1}=j \mid X_{n}=i\right\}$ and is independent of the time index n. Also, the number of states from which probabilities are defined to the next state is defined as Order of the Markov chain.

The Information Theoretic measures describe in this chapter would be used in next chapters to analyze, model and evaluate the performance of the work.

Chapter 3

DATA COLLECTION AND ANALYSIS

In this chapter we first discuss the simulation setup that we have envisaged to collect error traces over 802.11b WLAN. The 802.11b standard specifies data transmission at three bit-rates viz, 2 Mbps, 5.5 Mbps and 11 Mbps. We will focus on collecting data at 5.5 Mbps and 11 Mbps, bit rates that are more useful for multimedia applications, along with the per-packet Signal-To-Noise Ratio. The retransmission based method for corrupted/lost packet recovery requires the 802.11b standard to perform check on the 32 bit checksum of each received frame. We will use this same method to identify and store corrupted packets for residue error analysis and modeling.

3.1 Experimental Setup

The wireless trace collection setup envisaged collecting 802.11b frames in various work environments. The setup as shown in the figure 3.1, consisted of 802.11b Access Point (AP) operating in Distributed Coordination Function mode with RF output power set at 18 dBm (The output power is transmission power of Linksys WRT54G Wireless router). A station is connected to the AP using 100 Mbps Ethernet and acts like a server, while a wireless station serves as a Line Of Sight (LoS) client. It is essential for this wireless client to be Line Of Sight to avoid any packet drops, otherwise the AP can reduce the transmission bit rate. A third wireless station serves as a highly mobile sniffer machine. We used DWL 122 wireless card based on Intersil Prism 2.5 chipset with modified linux-wlan-ng-0.2.1 device driver for all sniffer machines, to avoid any fluctuations due to receiver sensitivity. The Prism based card enables operation in *monitor* mode which enables delivery of all the MAC frames, irrespective of the receiver address in the frame. With modification to the device driver we were able to receive frames even with failed Frame Check Sequence (FCS). This allowed us to capture wireless trace with inherent residue error information.

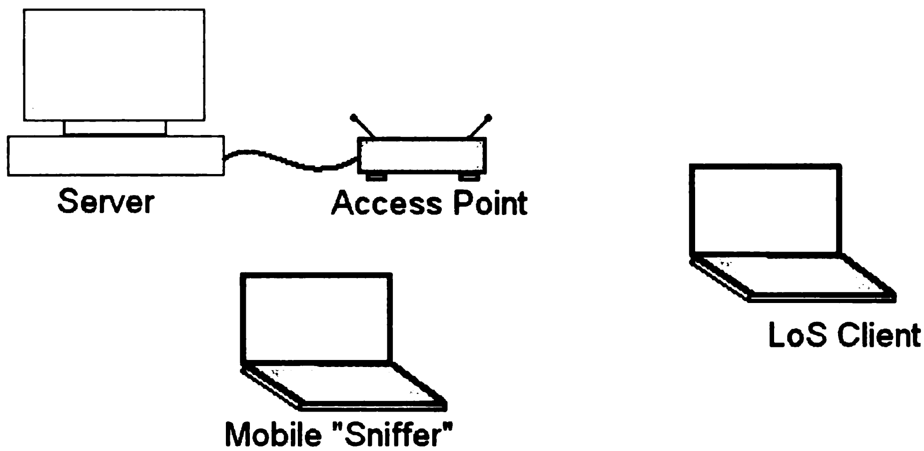


Figure 3.1 Experimental Setup showing Server, LoS Client and the Sniffer

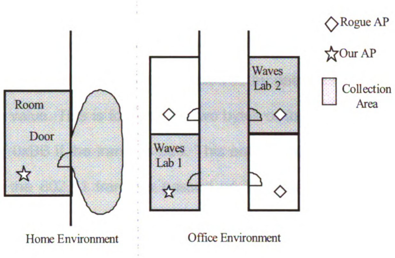


Figure 3.2 Different Trace Collection Environments

For the trace collection, the AP transmits packets over the wireless medium to the LoS client at 11 / 5.5 Mbps and the sniffer sniffs these transmissions from various locations with different link quality. The Prism2.5 device also measures received signal strength indication (RSSI) value and silence value at the antenna of the radio hardware. The RSSI is measured for 10µs while receiving the frame and provides total power observed, including signal, interference and background noise. The silence value measures total power before the start of the frame. Both these values, one byte each, are collected per-frame basis and reported as Signal-to-Noise ratio of that particular frame. In this work all references to SNR imply the RSSI to silence value ratio. Along with RSSI and silence values, the exact frame reception time provided by the hardware as four byte value is measured. Since we also collect the corrupt frames, the frame size cannot be determined just by length field which may be corrupted. As each frame is received, we add certain meta-data to it as shown in the figure 3.3 along with RSSI, silence and the time values. This helps to differentiate the frames during analysis later. The first five bytes of this meta-data correspond to ascii string UTPAL and is used for identifying frame start, the next four bytes is the frame receive time, followed by one byte RSSI value and one byte silence value. This is followed by two bytes of 0xAA if the frame is good, or two bytes of 0xBB if the frame is bad. This entire modified frame, i.e. meta-data followed by the 802.11 frame is passed up the standard network stack to be received by tools like ethereal. The figure 3.3 shows the frame as received by ethereal[3]. We then save all the frames collected by ethereal for analysis at later time.

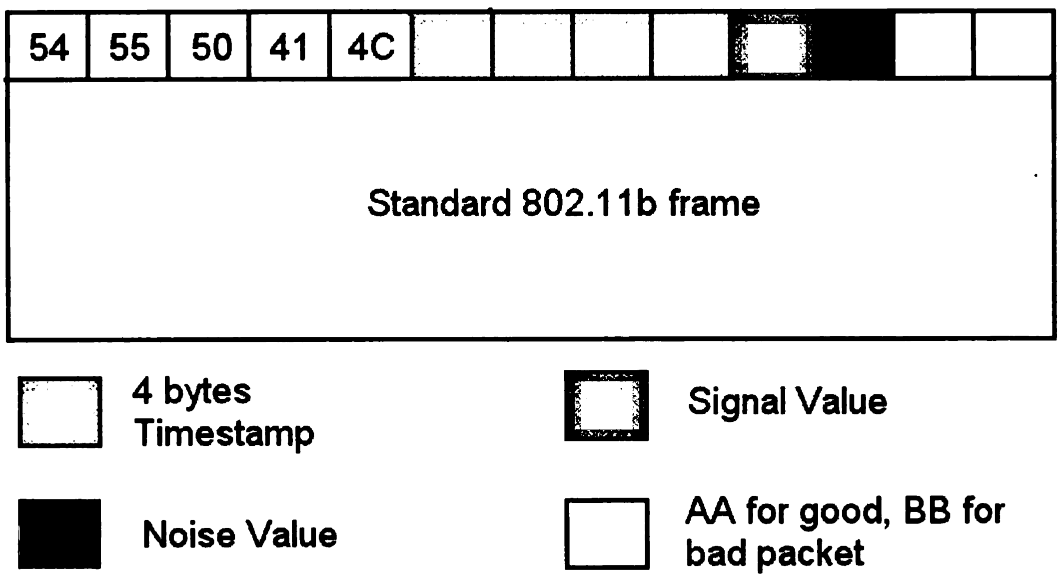


Figure 3.3 Modified Meta Frame captured by Ethereal

3 Appendix A shows 802.11 frame format in detail.

3.2 Packet- And Bit-level Analysis

To achieve our objective of developing SNR aware Full State Markov model, we first need to develop a thorough understanding of the relationship between SNR, loss and error patterns observed in the wireless traces. Good understanding of this relationship should allow us to anticipate significant improvement on account of employing a better SNR aware modeling technique. Although, we use observed bit-level error patterns to develop the Markov models, since each packet provides us with information on packet being good or bad with high degree of confidence using the FCS, we will utilize this packet-level information while generating the data from our models.

3.2.1 Packet-level Analysis (S-Curve)

Packet-loss due to either corruption, or loss when transmitting station is at a large distance, represents one of the key performance measure for any communication network. Also, since this work envisages developing a SNR aware model, it is necessary to understand the relationship between the SNR value and the packet-loss. Here, we will provide the MAC layer throughput statistics as a function of Signal-To-Noise Ratio. This corresponds to percentage of good packets reaching the link-layer at each SNR levels. Since the 802.11 MAC layer drops the packet, irrespective of the number and location of the bit-errors, this adversely affects the throughput.

The figure 3.4 shows the percentage of good packets received as a function of Signal-To-Noise Ratio (dB). The figure shows four traces, two collected at different locations from the Access Point (AP) in 11 Mbps bit-rate mode. The Trace 1 at both the bit rates belongs to Home environment, whereas Trace 2 belongs to Office Environment (See figure 3.2). The other two traces were collected with AP in 5.5 Mbps bit-rate mode. As seen in the figure, both the 11 Mbps and 5.5 Mbps traces match each other very closely. This establishes the fact that the % of good packets as a function of SNR does not change much with respect to change in the environment. Based on this figure we have divided the Signal-To-Noise Ratio span into three distinct regions, i.e. Bad (% good packets <= 10%, 9dB and below), Good (% good packet >= 90%, 15dB and above), and the Transition region (> 9dB and < 15dB).

To see how reducing the bit-rate affects this curve, we can see the curve generated by 5.5 Mbps trace. As can be clearly seen, this curve has been shifted towards left by at least 5dB. The figure 3.5 shows the corresponding shift in the three regions.

Figure 3.4 Good Packets (%) V/S Signal-To-Noise Ratio (dB)

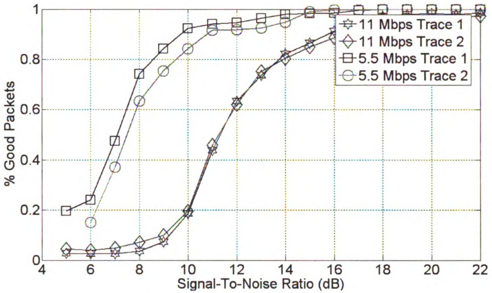


Figure 3.5 Transition Regions for 5.5 and 11 Mbps traces

3.2.2 Bit-level Analysis (U-Curve)

Since this work focuses on modeling residue error patterns, it is necessary to understand the relationship between residue error and the Signal to Noise Ratio. To establish this relationship we will consider only the packets with error, i.e. the corrupt packets. We do not consider the good packets as we have explained earlier that the best check for good packets is FCS itself. We have thus based the model in this work on those packets where FCS has failed. For each of the three traces, we analyzed the bad packets for the residue errors, and were able to establish the relationship between residue error and SNR as shown in the figure 3.6.

As seen from the figure, we can see that for all the three traces, the curve decreases as the signal level increases, i.e. the probability of the bit errors in the bad packets decrease as the signal level increases, and this is on par with the expectation. But this is only true below certain SNR level (12 dB for 5.5 Mbps and 13 dB for 11 Mbps). As the signal level increases further, it was seen that the probability of bit errors in a bad packet increases. Although this behavior is counter intuitive, there is a simple logical explanation. This is due to the fact that the bit errors are caused in the packets due to two primary reasons, first, low signal levels compared to the background noise/interference and second, packet collision when more than one station transmits at the same time. We can see that at lower SNR values, the probability of error in the bad packets is dominantly due to low signal levels and this represents the drop in probability of error with increasing signal levels. But after a certain point the bad packets are present no longer due to low signal levels, but due to the second factor i.e. collisions. The probability of error in a collided packet is generally higher, this causes the curve to turn upward as the SNR values increase.

Mathematically, this can be easily proved as follows. Consider corrupted packets due to decaying signal levels to be D, and corrupted packets due to collisions to be C, such that, N = C + D, where N is total number of corrupted packets. We can assume that the number of collisions remain constant with the change in Signal-To-Noise ratio, as the collisions are caused due to external factors such as transmissions from some other station. In this case the probability of error is given by,

$$P.E._{Conditional} = \frac{D \times P_d \times Packet_{Size} + C \times P_c \times Packet_{Size}}{(D+C) \times Packet_{Size}} \tag{3.1}$$

$$P.E._{Conditional} = \frac{D \times P_d + C \times P_c}{D + C} \tag{3.2}$$

Where P_d and P_c are probability of bit errors due to decaying signal and collisions respectively. We also know that both the D and the P_d are inversely proportional to the SNR. Hence as the SNR increases, D and P_d both decrease rapidly and beyond a certain point, no longer dominate the equation. Thus probability of error starts increasing more and more towards P_c.

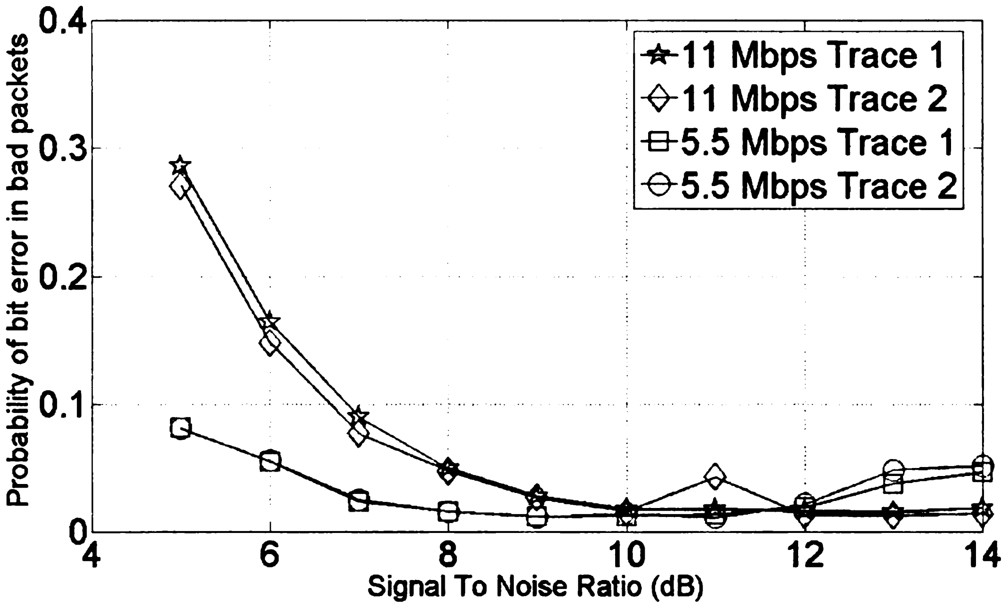


Figure 3.6 Probability of error (p) in bad packets V/S Signal-To-Noise Ratio (dB)

3.2.3 Mutual Information

As we have already seen in section 2.5, Mutual Information between two random variables X and Y, given as I[X;Y], is the information in X about Y. We can use this measure as a function of $lag(\eta)$, to quantify the memory in the residue errors of the corrupted packets. We can calculate the sample mutual information by considering sequence of errors in a packet $\{x_i\}$ and then evaluating the mean frequency $f\{x_i, x_{i+\eta}\}$ of each possible combination of $\{x_i, x_{i+\eta}\}$. The frequency $f\{x_i\}$ is nothing but the probability of error $\overline{p}$. Thus the mutual information is then calculated using the standard formula, i.e.,

$$I(\eta) = \sum f\left(x_i, x_{i+n}\right) \log_2 \left(\frac{f\left(x_i, x_{i+n}\right)}{f\left(x_i\right) \bullet f\left(x_{i+n}\right)} \right) \tag{3.3}$$

Figures 3.7 and 3.8 show Mutual Information calculated as a function of $lag(\eta)$ for 11 Mbps and 5.5 Mbps respectively. In the figures, the mutual information is calculated independently for packets belonging to different SNR values (6 dB, 8 dB and 13 dB in the figures). It can be seen that for both the traces collected at 11 and 5.5 Mbps, the mutual information decays at different rates for different SNR values. Thus ideally the length of the model should be varied as a function of SNR. This length can be optimized for different SNR values,

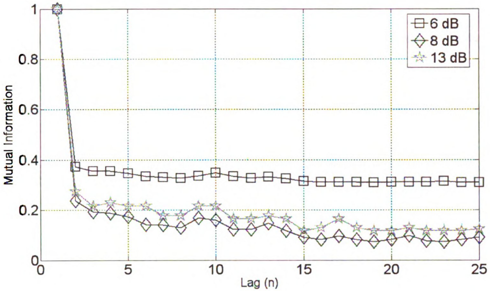


Figure 3.7 Mutual Information as function of $lag(\eta)$ for 11 Mbps trace

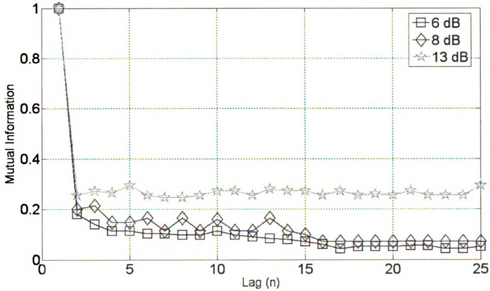


Figure 3.8 Mutual Information as a function of $lag(\eta)$ for 5.5 Mbps trace

In order to simplify the analysis and make a fluent presentation in this work, we will develop a model having equal memory length for all SNR values. We will however develop a number of models with different memory lengths. Thus unlike the traditional approach of modeling, where the memory length of the process is identified before choosing a model, we choose a specific memory length and then try to evaluate the utility of the model. Thus the utility of Markov model may vary as function of SNR.

Chapter 4

MODELING AND PERFORMANCE MEASUREMENT

The previous chapter dealt with the collection and the analysis of the 802.11b WLAN data at two different bit-rates. In this chapter, we will discuss the use of this data to build the SNR Aware and the SNR Unaware (Conventional) Full State Markov models, and generation of data from these two types of models. We will use some of the information theoretic tools described in Chapter 2 to measure the performance of both these types of models and see how SNR information helps improve the performance.

4.1 Full State Markov Models

For a memory length k, the bit errors $\bar{x} = [x_1, x_2 \cdots x_k]$ have 2^k possible error patterns. The states of the Markov model we employ here are described by the bit error pattern of the previous k bits. Since we formulate a distinct state for each possible error pattern, the markov model used in this thesis can be referred to as a full-state Markov (FSM) chain. The markov chain we employ is auto-regressive in nature, such that for each state transition $[x_1, x_2 \cdots x_k] \rightarrow [x_2, \cdots x_k, x_{k+1}]$ the output is a single error bit x_{k+1}. Thus the state transition probabilities provide us with the probability of error at a particular bit location, conditioned on the error pattern of the previous k bit locations. Such a modeling approach is particularly useful in practical setups.

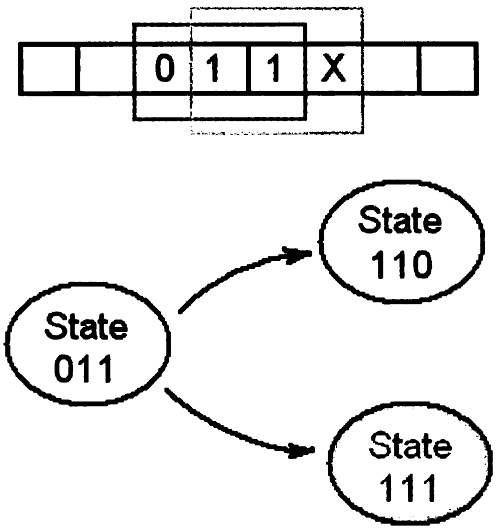


Figure 4.1 Transition possibilities for sliding window with k=3

In this work, a sliding window was used to compute the transition probability matrices. Figure 4.1 shows one such sliding window of width 3, i.e. the states of the markov model in this example are described by bit error pattern of previous 3 bits. The first window in the figure shows current state ,$(011)_2$ and as the window slides towards right, the most significant position bit will be dropped, and a new bit added at the least significant position. Since the data is binary (bit error present or not), the chain can either transit to $(110)_2$ or $(111)_2$. That is the current state can jump only to two possible next states. We thus compute the probability with which the current state will jump to either of the two possible subsequent states, and we obtain a transition probability matrix, computed for all the possible current states.

The conventional model is simply obtained by computing the transition probability matrix for all the corrupt packets taken together. In the proposed SNR Aware Markov Model, we compute different transition probability matrices for each SNR interval. For example, for SNR values going from 0 to 25, we will have 26 distinct transition probability matrices, each computed from corrupted packets having the corresponding SNR values. Hence the proposed model, is referred to as SNR Aware model, as it is not oblivious to SNR values at which each corrupt packet was received.

In this work we will generate models having memory length ranging from k=1, i.e. 2 States, to memory length k=12, i.e. 2^k States. We will then observe performance of these models using some of the Information Theory tools described in Chapter 2, as function of memory length k.

4.2 Data Generation

We first generate the models based on a subset of data from the 5.5 and 11 Mbps traces, called the *training data*. We will then use these models to predict data based on the remaining data called the *test data* In case of SNR Unaware model, we first parse through the test data and if the packet has no error, we will replicate this packet as it has no error, We are more interested in modeling the bit-level residue error patterns only. If the packet has error, then we will refer to the transition probability matrix of the model in use, and generate an approximated packet. In this way we get the approximated data using a model, instead of the actual source. In the SNR Aware case, we repeat the same steps, but for a corrupted packet, we use the transition probability matrix of the same SNR value as that of the packet. This again reflects the SNR Aware nature of the model. That is, as the SNR values changes our proposed model dynamically changes the transition probability matrix. We then use this generated data for performance evaluation of the different models used.

4.3 Performance Evaluation

We measure the performance of the full-state Markov chains (FSM) in terms of the ability of the synthesized data to replicate the *features* of the actual error process. The *features* are defined in terms of random variables such as Inter-arrival rate **I**, burst length **B** and the frequency of errors per packet ***p***. The performance of the model is quantified in terms of *Entropy Normalized Kullback-Leibler* (ENK) *Divergence* between the probability distributions of the above-mentioned random variables.

4.3.1 Inter-arrival rate I

This feature of the error process captures the distance between two error bits in terms of non-error bits. That is we measure number of times two error bits arrive with separation i. Figure 4.2 shows the ENK computed for data synthesized from 11mbps and 5.5 mbps models as a function of memory length k.

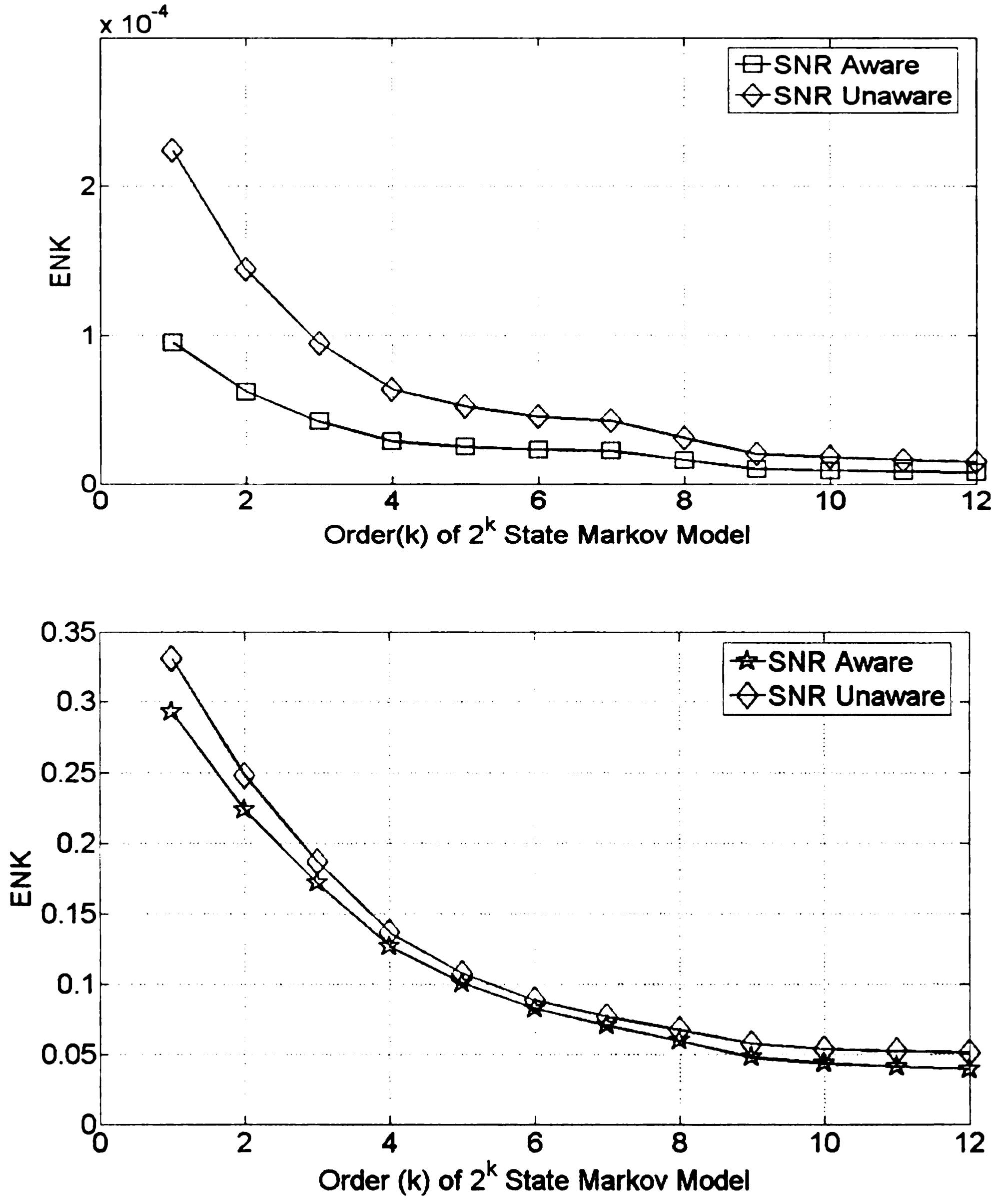


Figure 4.2 ENK for Inter-Arrival of errors for 11Mbps(Top) and 5.5 Mbps (Bottom)

The ENK in the figure measures how close the approximated data is with the *test data.* Figure 4.3 shows the ENK for the same generated data but computed with respect to error traces collected in some other (*Office*) environment.

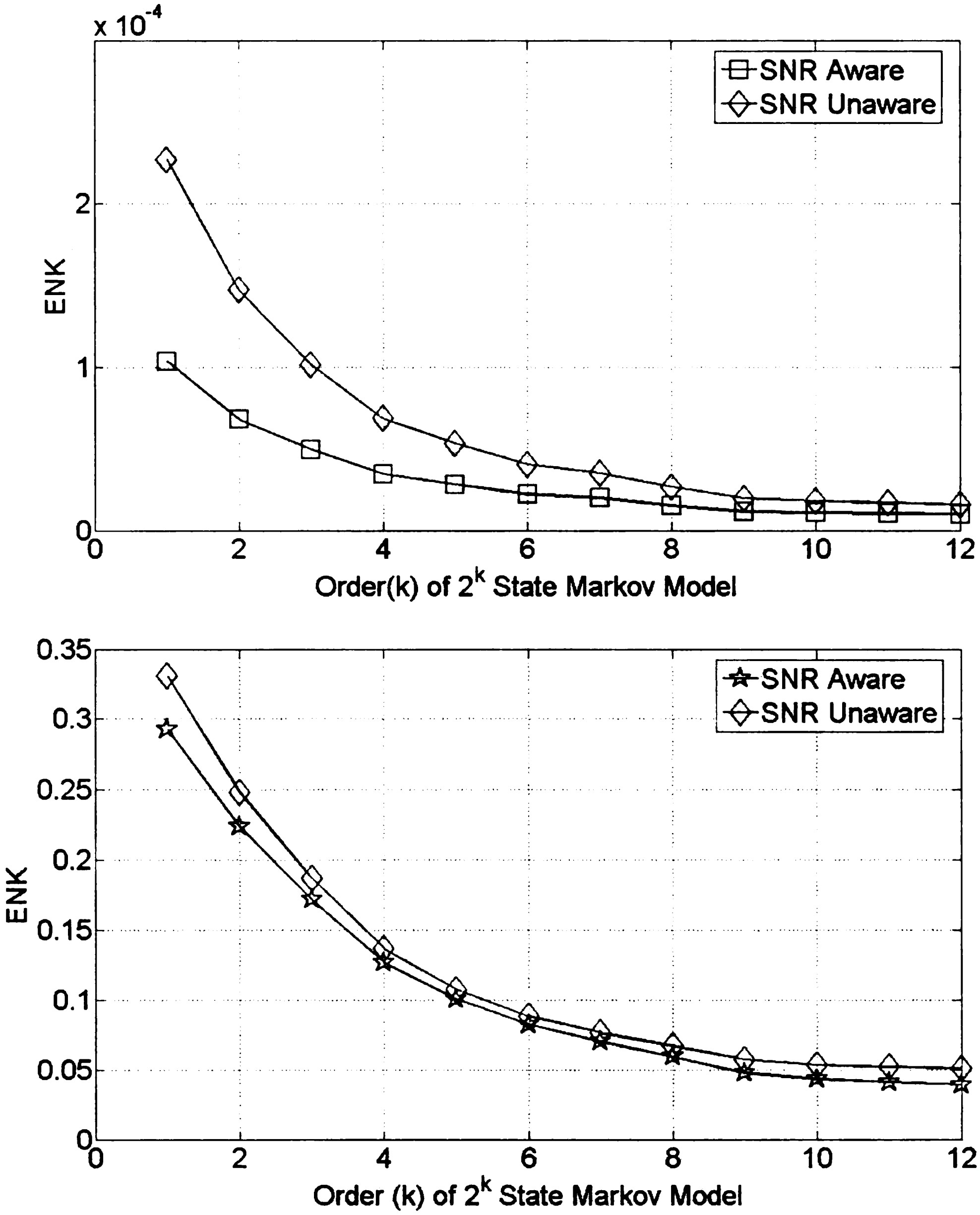


Figure 4.3 ENK for Inter-Arrival of errors for 11Mbps(Top) and 5.5 Mbps (Bottom) for Office Data

The ENK results computed for I-arrival feature, shows that the SNR Aware model performance better than the SNR Unaware model for all memory lengths.

4.3.2 Burst Length

This feature measures the number of times a sequence of error bits occurs. That is, it computes the frequency of occurrence of each burst of error bits. Figure 4.4 shows the ENK computed for data synthesized from 11mbps and 5.5 mbps models as a function of memory length k.

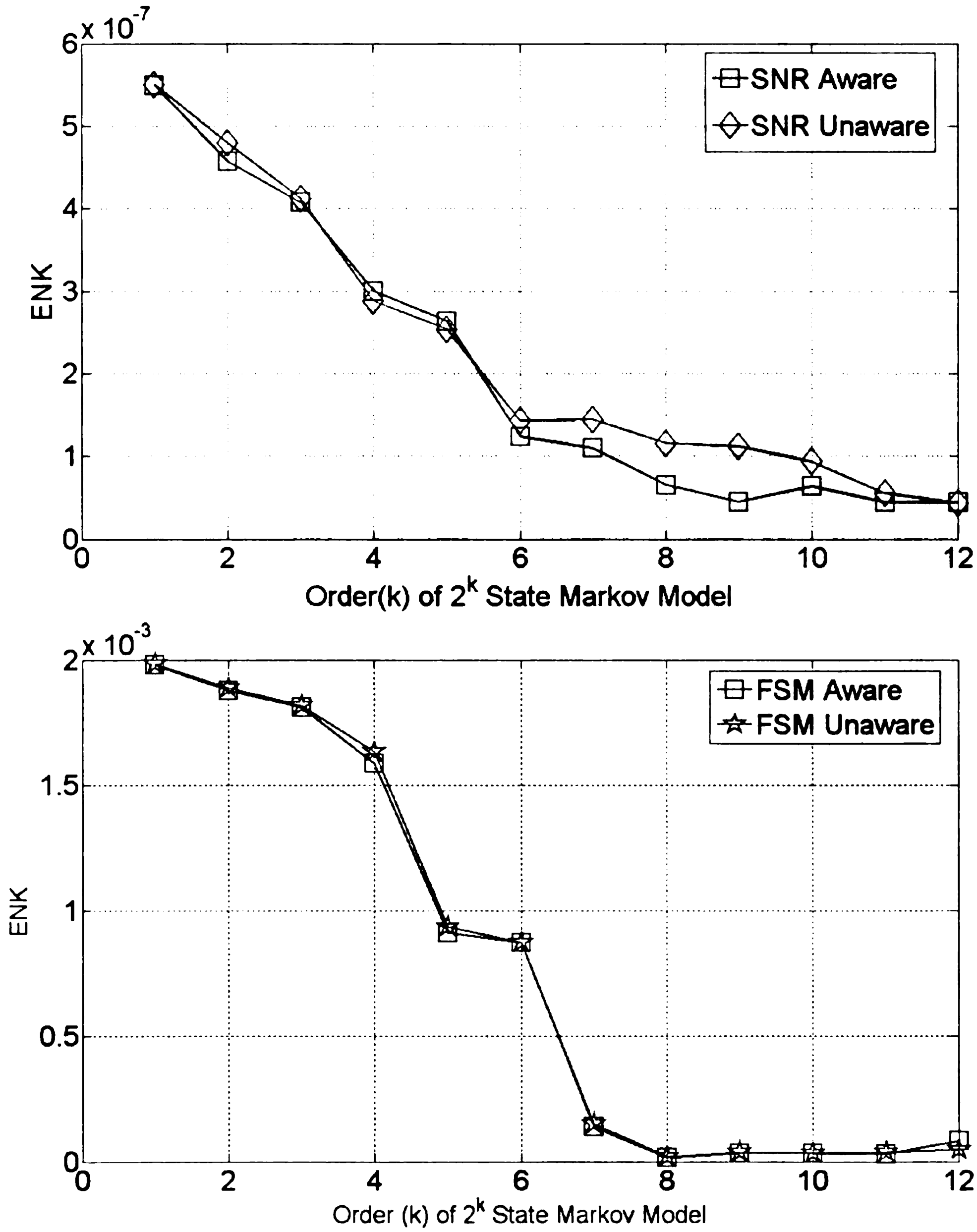


Figure 4.4 ENK for Burst Length of errors for 11Mbps(Top) and 5.5 Mbps (Bottom)

The figure 4.5 shows the ENK for the same generated data but computed with respect to error traces collected in some other (*Office*) environment.

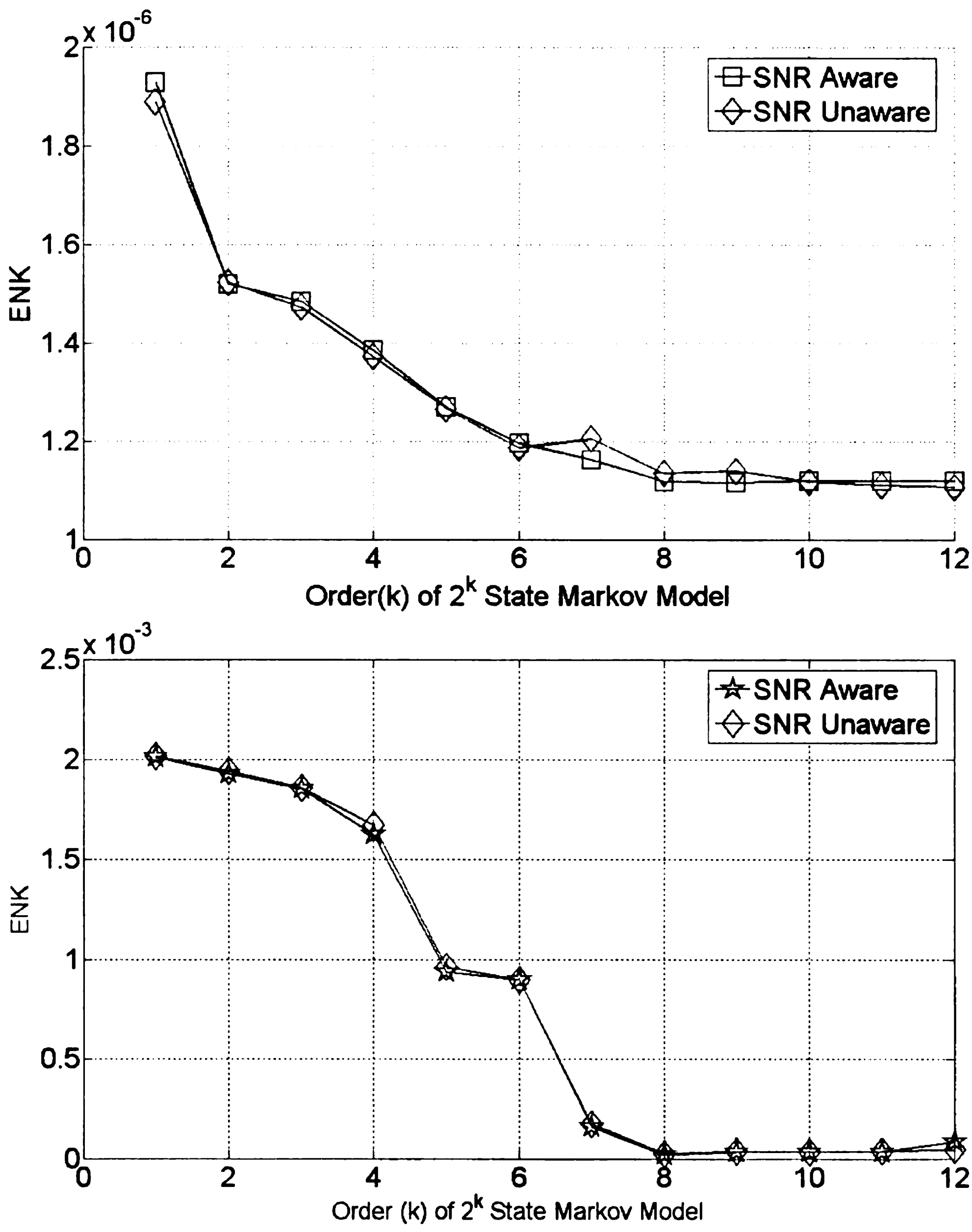


Figure 4.5 ENK for Burst Length of errors for 11Mbps(Top) and 5.5 Mbps (Bottom) for Office Data

It can be seen that both the SNR aware as well as the SNR unaware model are capable of maintaining the burst error characteristics of the source.

4.3.3 Frequency Of Errors per Packet p

This feature computes the frequency of occurrence of packet with n errors. That is, it measures the number of times a corrupt packet arrives with certain number of bit errors. Figure 4.6 shows the ENK computed for data synthesized from 11mbps and 5.5 mbps models as a function of memory length k.

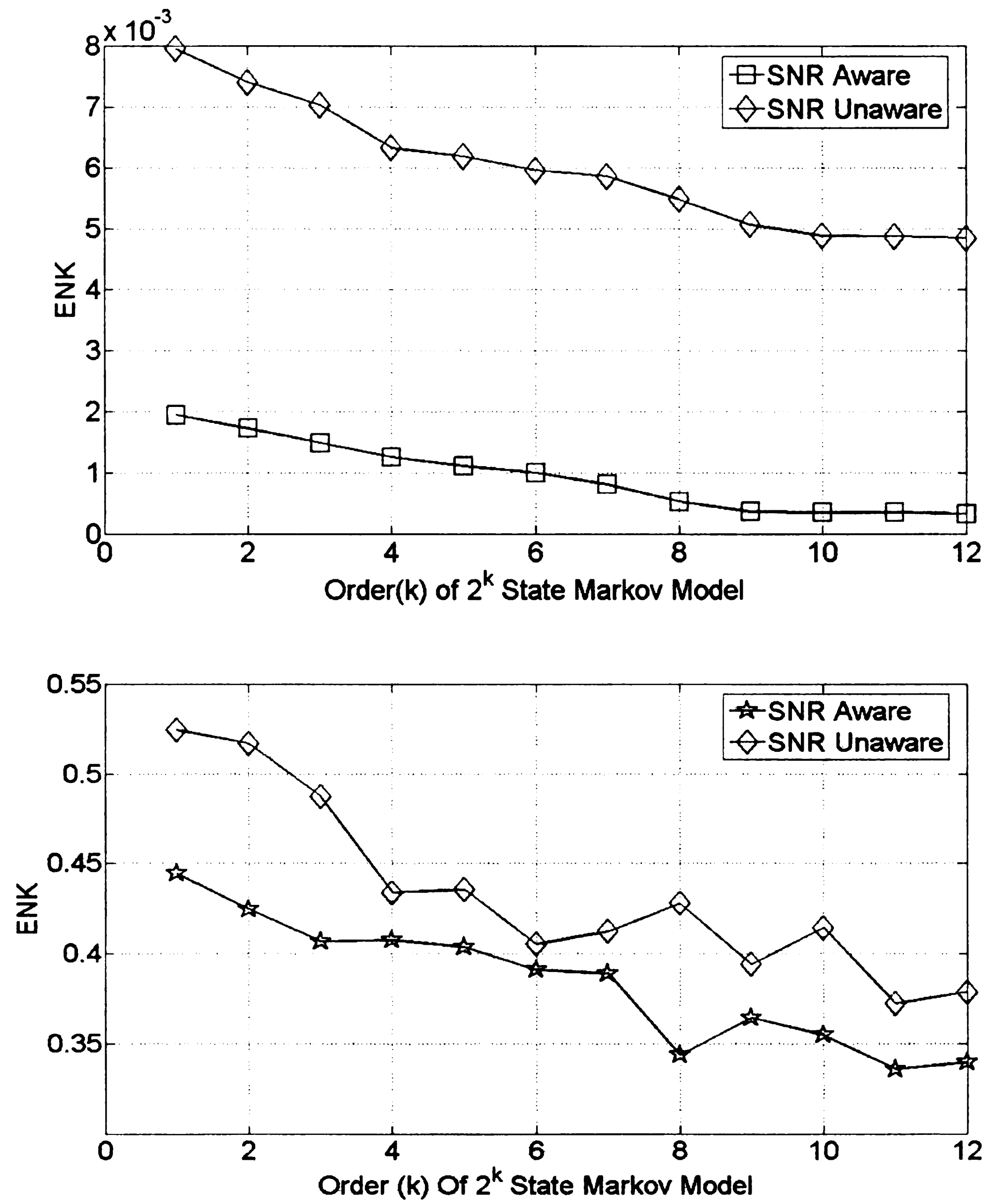
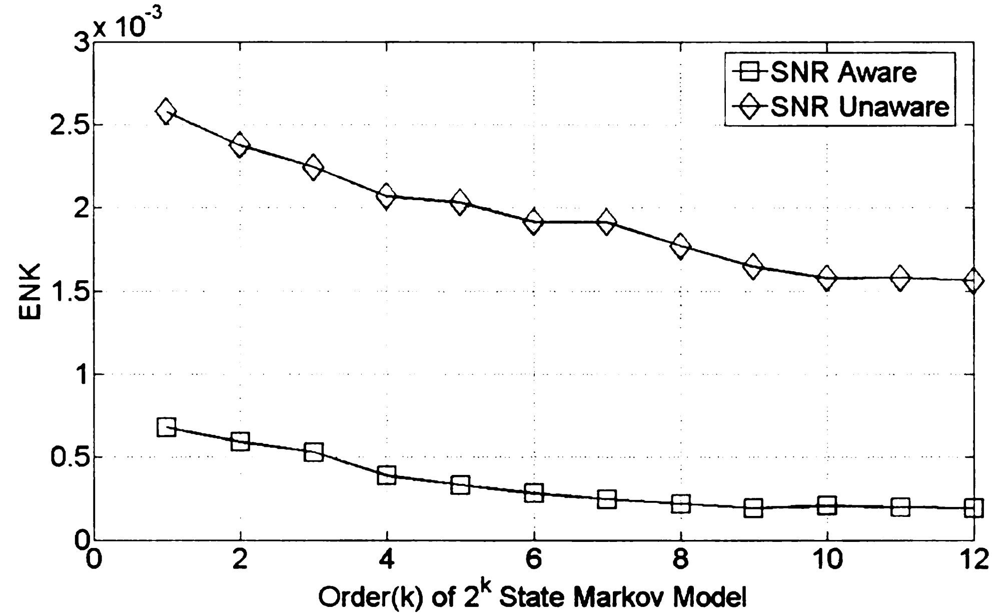


Figure 4.6 ENK for per packet errors for 11Mbps(Top) and 5.5 Mbps (Bottom)



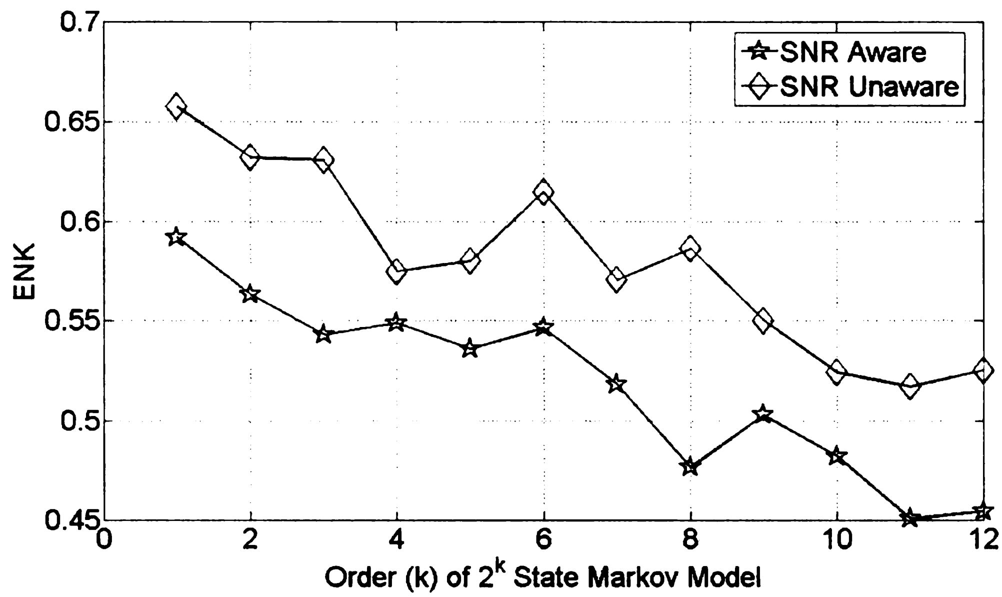


Figure 4.7 ENK per packet errors for 11Mbps(Top) and 5.5 Mbps (Bottom) for Office Data

The figure 4.7 shows the ENK for the same generated data but computed with respect to error traces collected in some other (*Office*) environment. The ENK results shows that the SNR Aware model performance much better than the SNR Unaware model for all memory lengths.

Chapter 5

CONCLUSIONS AND FUTURE WORK

In this work we provide analysis and modeling of residue bit errors of the 802.11b Wireless Local Area Network, and also propose an alternate SNR Aware Markov Modeling scheme. First we present the packet-level analysis of the data collected over 802.11b WLAN at 11 and 5.5 Mbps bit-rates. We see how the throughput varies as a function of Signal-To-Noise Ratio (SNR) and also the bit-rate. That is, reducing the bit rate from 11 Mbps to 5.5 Mbps makes the percentage of good packets (S-Curve) shift left, i.e. it registers increase in good packets at lower SNR values with decrease in the bit rate.

Primary focus of this work was bit-level analysis and modeling. We have analyzed the residue errors in a corrupted packet as a function of SNR, and displayed it using the 'U-Curve'. The probability of bit error decreases rapidly as the signal level increases, but beyond certain point, the value starts increasing as events like collisions etc., play a more dominant role in packet corruption. We also show that memory in the channel varies as function of SNR by using Information Theoretic measures such as Mutual Information.

Continuing our work at bit-level, we first employed conventional Full State Markov model. We synthesized data for 11 and 5.5 Mbps bit-rates. We then developed the proposed SNR Aware model for the same data, and synthesized data by using the model. Using information theory measures defined in Chapter 2, over different features (burst length, inter-arrival rate, frequency of errors per packet) of the synthesized and source data, we showed that the proposed SNR Aware model has significantly better performance. Thus, it has been shown that the 802.11b 11Mbps link-level can be greatly enhanced by making them SNR aware and using the ability of Markov models to characterize the residue errors. We show that the overall behavior of link-level residue errors is a function of the environment in which the wireless traces are collected. SNR aware Markov model were shown to provide excellent performance in foreign environments also; and thus should prove useful for developing future error control, simulation/emulation applications.

Based on this work we can identify some future directions, especially toward Network Simulators. In a network simulator, the SNR values can be obtained from the physical layer emulator. A simple two-state Markov model used to imitate the packet drop in a conventional setup can be used to model frame corruptions. The bit errors inside a corrupted packet can then be modeled on the basis of the SNR values obtained from the physical layer emulator and the SNR aware markov models proposed in this paper. Incorporating such a setup in some popular network simulators can significantly improve their performance.

We are also investigating use of SNR in dual/multiple antenna systems. The SNR information can be used to improve the throughput by making use of smart cross-layer strategies in these systems. Multimedia applications over wireless can benefit immensely from any such schemes. However, any work in this area is beyond the scope of this thesis, and will be part of future work.

Appendix A

802.11 FRAME FORMATS

All the 802.11 frames have the following format,

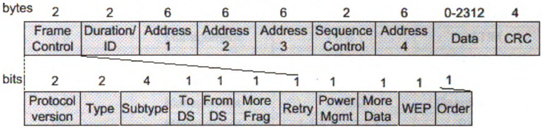


Figure A.1 802.11 Frame Format

The Frame Control field can be further divided into sub-fields as shown in the figure.

Protocol Version: This field has 2 bits, which represent the version of 802.11 Standard. All current versions have default value 0.

Type And Sub-Type: These fields together indicate type of the frame being transmitted. Following figure shows all the possible types of frames.

Type Value	Type Description	Subtype Value	Subtype Description
0 0	Management	0 0 0 0	Association request
0 0	Management	0 0 0 1	Association response
0 0	Management	0 0 1 0	Reassociation req
0 0	Management	0 0 1 1	Reassociation res
0 0	Management	0 1 0 0	Probe request
0 0	Management	0 1 0 1	Probe response
0 0	Management	0 1 1 0 – 0 1 1 1	Reserved
0 0	Management	1 0 0 0	Beacon
0 0	Management	1 0 0 1	ATIM
0 0	Management	1 0 1 0	Disassociation
0 0	Management	1 0 1 1	Authentication
0 0	Management	1 1 0 0	Deauthentication
0 0	Management	1 1 0 1 – 1 1 1 1	Reserved
0 1	Control	0 0 0 0 – 1 0 0 1	Reserved
0 1	Control	1 0 1 0	Power Save Poll
0 1	Control	1 0 1 1	RTS
0 1	Control	1 1 0 0	CTS
0 1	Control	1 1 0 1	ACK
0 1	Control	1 1 1 0	Contention Free End
0 1	Control	1 1 1 1	CF-End + CF-ACK
1 0	Data	0 0 0 0	Data
1 0	Data	0 0 0 1	Data + CF-ACK
1 0	Data	0 0 1 0	Data + CF-POLL
1 0	Data	0 0 1 1	Data + CF-Ack + CF-Poll
1 0	Data	0 1 0 0	Null Function
1 0	Data	0 1 0 1	CF-Ack
1 0	Data	0 1 1 0	CF-Poll
1 0	Data	0 1 1 1	CF-Ack+CF-Poll
1 0	Data	1 0 0 0 – 1 1 1 1	Reserved
1 1	Reserved	0 0 0 0 – 1 1 1 1	Reserved

Table A-1 Type And Sub-Type Field Values

To DS: This bit is set to 1 if the frame is addressed to AP for forwarding to Distribution System (DS).

From DS: This bit is set to 1 when frame is coming from the Distribution System.

Address Fields: Address1, Address2, Address3 and Address4 fields indicate various hardware addresses.

Following table shows the use of To DS and From DS fields and Address fields.

ToDS	FromDS	Adresse 1	Adresse 2	Adresse 3	Adresse 4
0	0	DA	SA	BSSID	N/A
0	1	DA	BSSID	SA	N/A
1	0	BSSID	SA	DA	N/A
1	1	RA	TA	DA	SA

Table A-2 ToDS And FromDS Values

More Fragments: Set to 1 when there are more fragments belonging to this same frame.

Retry: This bit set to 1 indicates fragment is the retransmission of previously attempted transmission of a segment.

Power Management: This indicates that the station would be in power management mode after transmission of this frame.

More Data: This is also used by the power management and is used by AP to indicate that there are more frames buffered for the station.

WEP: Set to 1 when WEP encryption is used.

Order: This set to 1 indicates the station is using Strictly-Ordered service set*.

Duration I/D: This field has two meaning depending on frame types. In Power-Save poll messages, this is station ID. In other frames this is the NAV duration.

Sequence Control: Used to determine sequence of different fragments of the same frame.

CRC: This field contains the 32-bit Cyclic Redundancy Check

---

* The Strictly-Ordered Service Class is defined for users who cannot accept change of ordering between Unicast frames and Multicast frames. The only known protocol to use this is DEC's LAT.

REFERENCES

[1]ISL3873: Wireless LAN Integrated Medium Access Controller with Baseband Processor Intersil Corporation, 2000. Application Note FN4868.

[2]http://www.linux-wlan.org

[3]http://www.egr.msu.edu/waves/people/shirish.htm

[4]http://www.egr.msu.edu/waves/people/Ali.htm

[5]ISO/IEC 8802-11:1999(E), "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," August 1999.

[6]IEEE Std 802.11b-1999, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz band," September 1999.

[7]L. Larzon, M. Degermark, and S. Pink, "UDP Lite for Real Time Multimedia Applications," IEEE International Conference of Communications (ICC), Vancouver, June 1999.

[8]H. Zheng and J. Boyce, "An Improved UDP Protocol for Video Transmission Over Internet-to-Wireless Networks," IEEE Trans. on Multimedia, vol. 3, no. 3, pp. 356–365, September 2001.

[9]S. A. Konrad, A. D. Joseph, "Performance Evaluation of UDP Lite for Cellular Video," NOSSDAV, 2001.

[10]S. A. Khayam, S. S. Karande, H. Radha and D. Loguinov, "Performance Analysis and Modeling of Errors and Losses over 802.11b LANS for High-Bitrate Real-Time Multimedia," Signal Processing: Image Communication, vol. 18, Aug 2003.

[11]Shirish Karande and Hayder Radha, "Does Relay of Corrupted Packets Lead to Capacity Improvement?," IEEE Wireless Communications and Networking Conference (WCNC), March 2005.

[12]Servetti, J.C. De Martin, "Link-Level Unequal Error Detection for Speech Transmission over 802.11b Networks," Proc. Special Workshop in Maui (SWIM), Maui, Hawaii, January 2004.

[13]T. M. Cover and J. A Thomas, “Elements of Information Theory,” Wiley Series in Telecommunications.

[14]S. S. Karande, S. A. Khayam, M. Krappel, and H. Radha, "Analysis and Modeling of Errors at the 802.11b Link-Layer," IEEE International Conference on Multimedia and Expo (ICME), July 2003.

[15]S. A. Khayam and H. Radha, “Markov-based Modeling of Wireless Local Area Networks," ACM MSWiM, Sept. 2003.

[16]S. A. Khayam, Selin Aviyente, and Hayder Radha, "On Long-Range Dependence in High-Bitrate Wireless Residual Channels," Conference on Information Sciences and Systems (CISS), March 2005.

[17]S. A. Khayam and Hayder Radha, "Linear-Complexity Models for Wireless MAC-to-MAC Channels," ACM/Kluwer Wireless Networks (WINET) Journal - Special Issue on Selected Papers from MSWiM'03, vol. 11, no. 5, pp. 543-555, September 2005

[18]http://www.sss-mag.com/pdf/802_11tut.pdf