LOW COMPLEXITY CHANNEL MODELS FOR APPROXIMATING FLAT RAYLEIGH FADING IN NETWORK SIMULATIONS

A Dissertation

by

JEFFREY MICHAEL MCDOUGALL

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

August 2003

Major Subject: Electrical Engineering

LOW COMPLEXITY CHANNEL MODELS FOR APPROXIMATING FLAT RAYLEIGH FADING IN NETWORK SIMULATIONS

A Dissertation

by

JEFFREY MICHAEL MCDOUGALL

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Approved as to style and content by:

Scott L. Miller (Chair of Committee)	
Costas N. Georghiades (Member)	A. L. Narasimha Reddy (Member)
Du Li (Member)	Chanan Singh (Head of Department)

August 2003

Major Subject: Electrical Engineering

ABSTRACT

Low Complexity Channel Models for Approximating

Flat Rayleigh Fading in Network Simulations. (August 2003)

Jeffrey Michael McDougall,

B.S., Texas A&M University;

M.S., Johns Hopkins University

Chair of Advisory Committee: Dr. Scott L. Miller

The intricate dependency of networking protocols upon the performance of the wireless channel motivates the investigation of network channel approximations for fading channels. Wireless networking protocols are increasingly being designed and evaluated with the assistance of networking simulators. While evaluating networking protocols such as medium access control, routing, and reliable transport, the network channel model, and its associated capacity, will drastically impact the achievable network throughput. Researcher relying upon simulation results must therefore use extreme caution to ensure the use of similar channel models when performing protocol comparisons. Some channel approximations have been created to mimic the behavior of a fading environment, however there exists little to no justification for these channel approximations.

This dissertation addresses the need for a computationally efficient fading channel approximation for use in network simulations. A rigorous flat fading channel model was developed for use in accuracy measurements of channel approximations. The

popular two-state Markov model channel approximation is analyzed and shown to perform poorly for low to moderate signal-to-noise ratios (SNR). Three novel channel approximations are derived, with multiple methods of parameter estimation. Each model is analyzed for both statistical performance and network performance. The final model is shown to achieve very accurate network throughput performance by achieving a very close matching of the frame run distributions.

This work provides a rigorous evaluation of the popular two-state Markov model, and three novel low complexity channel models in both statistical accuracy and network throughput performance. The novel models are formed through attempts to match key statistical parameters of frame error run and good frame run statistics. It is shown that only matching key parameters is insufficient to achieve an acceptable channel approximation and that it is necessary to approximate the distribution of frame error duration and good frame run duration. The final novel channel approximation, the three-state run-length model, is shown to achieve a good approximation of the desired distributions when some key statistical parameters are matched.

DEDICATION

To my wife, Mary

and

To my children, Clare, Katherine, William, David and John

ACKNOWLEDGEMENTS

All glory and honor be to God. Thank you for your guidance, wisdom, strength, grace and for the gift of my family… whose support made this journey worthwhile.

I would like to thank my advisor Dr. Miller, who was willing to take a chance on a student with a significant number of distractions, a student who was "uninterested in chasing the bit-error-rate monster". Thank you for your support and advice and willingness to guide me through the 'unfamiliar' territory of networking.

To Dr. Georghiades and Dr. Watson, I thank you for your initial encouragement and excitement upon my brief visit in February of 1999. I would like to give a special thank you to the Wireless Communication Group for providing a TxTEC fellowship for my first year and to the Engineering Technology Department for providing meaningful employment throughout the duration of my work.

A special thank you is necessary for my mother and father who provided me with a good home, love, respect, a strong faith and financial support. I count the blessings you have given me daily.

To my family, thank you for living with a distracted father and husband. The final steps in this effort were much more difficult for you than for me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

FIGURE Page

FIGURE Page

FIGURE Page

CHAPTER I

INTRODUCTION

Networks rely upon intricate interactions between layers of protocols to achieve a distributed form of communication. As networks have grown in size, complexity and variety, researchers have gravitated towards network simulations to assist in the introspection of protocol interactions. Currently, researchers use network simulation results to demonstrate the performance of new techniques and protocols. These empirical trials have been used to achieve advances in networking protocols such as medium access control (MAC), routing and transport control protocols (TCP). Recently the use of wireless local area networks and wireless ad hoc networks has produced a desire to analyze the performance of networking protocols across wireless links. This desire garnished attention for the subject of operating networking protocols across wireless links experiencing fading. While the motivation exists for observing networks operating across such 'faded' channels, the fading channel models are themselves problematic.

This dissertation will show the widely accepted two-state Markov Model generates pessimistic network performance for wireless devices operating at low to moderate signal-to-noise ratios (SNR).

The journal model is *IEEE Transactions on Automatic Control*.

Protocols evaluated through simulations using the Markov Model will reflect this pessimism with complexity thus increasing the cost of implementation and power consumption for wireless devices operating in low to moderate SNR.

Protocol development is inherently intuition based with performance established through network simulations, hence protocols established under the two-state Markov model tend to contain significant complexity, not necessarily required at low to moderate SNR. Worse, network simulations utilizing the Markov model might indicate that a protocol will not function, precluding its use in low power devices. With the addition of improved wireless network channel models, new protocols can be generated to realize modest wireless packet data transmissions in low SNR, where the pessimism of previous network simulations employing a two-state Markov model precluded such operation.

Significant research in the areas of TCP performance, data link and radio link protocols is based upon simulations employing a two-state Markov model. These bodies of research will be fundamentally altered by employing a more realistic physical layer model, possibly diminishing the impact of suggested TCP variants or ad hoc coordination schemes. Furthermore, the field of ad hoc wireless networking has largely overlooked the effects of small-scale fading. In the case of transport control protocol, the achievable wireless link throughput is based upon consecutive successful packet transmissions. Thus, physical layer models that cannot track the higher order variations in frame error will tend to misrepresent the performance of a TCP scheme. In each of

these articles, the two-state Markov model is used when observing the effects of small-scale fading.

- The subject of appropriate TCP variations for use on a wireless link is well documented in the literature [1], [2], [3], [4], [5], [6], [7], [8].
- Similarly, data link layer retransmission schemes and radio link protocols (RLP) schemes can be affected by the higher order variations in frame error. The following articles represent some recent areas of interest in data link and RLP schemes, [9], [10], [11], [12], [13], [14], [15]. In some schemes, such as MACAW [16] and some 802.11, MAC research [17], the subject of small-scale fading is overlooked entirely.

The field of ad hoc networking has experienced a significant amount of attention and most research in this area, that incorporates fading on wireless links, employs the two-state Markov model. It is believed that the areas of ad hoc routing, medium access coordination and power minimization will be drastically affected by employing a more realistic channel model.

- Most Ad hoc routing protocols have neglected small-scale fading entirely, including RDMAR [18], DREAM [19] and others [20], [21], [22], [23], [24]. The omission of fading can impact the delay, convergence speed and overhead associated with ad hoc routing protocols and thus obscuring the actual impact of each.

The optimum method for observing packet-data performance over wireless channels is to simulate the network interactions over a baseband equivalent physical layer. Most network simulations involve the transmission and reception of millions of packets each containing thousands of bits. Due to the computational complexity of detailed physical layer simulations, this approach increases the run-time of networking simulations by orders of magnitude, making them impractical at best. Therefore, a challenge exists to add realistic channel statistics to networking simulations without overburdening the simulation with complexity. Knowledge of the relationships between network performance and channel simulations would help quantify the costs of using computationally efficient channel models. This work analyzes the popular Markov model fading channel approximation and produces three alternatives with more accurate statistical and throughput performance.

In Chapter II, the reader is introduced to the background of fading channels, modulated signals and the coded modulation scheme of complementary code keying. The term fading is defined for the case of large and small temporal dispersion relating to frequency selective and flat fading respectively. The definitions are followed by a series of channel measurements that help to quantify a 'real-world' temporal dispersion. The coded-modulation technique employed by the wireless local area network (WLAN) protocol 802.11b is explained and accompanied by an AWGN performance approximation.

Chapter III describes the most common network simulation tool employed by network researchers. The network simulator 2, or ns2, is described through an

accounting of its inception and roots as well as through a functional description of its operation. Following the simulator's description, an explanation for each of the novel channel approximation modules is provided, describing how they were implemented within ns2. The justification of form and parameter estimation is left for Chapter V.

Chapter IV performs a detailed analysis of the popular two-state Markov model channel approximation. The chapter begins by explaining the theory behind using a two-state Markov model to approximate a flat fading channel. Implementation questions are addressed in this chapter including how to model heterogeneous frames using homogeneous frame statistics. The method of parameter estimation for the two-state Markov model is described. Frame statistics for the Markov model channel approximation are compared against those from a representative flat Rayleigh fading channel model. The chapter concludes with an analysis of an ns2 network simulation incorporating the two-state Markov model. The results indicate the two-state Markov model is unable to accurately approximate both the frame statistics and network throughput.

Chapter V presents three new low complexity models that are formed to better approximate key frame statistics. Each model is described and the parameter estimation procedure is explained. It is shown through the model development that better network performance is achieved through a more accurate approximation of the probability mass distributions (PMD) for both frame error durations and good frame run durations. The 'N'-state Markov model is formed as a natural extension to the two-state model and analyzed for both statistical and network performance. The N=128 version of this model

is shown to match all of the observed frame error and good frame run statistics but not was unable to match the desired PMD. As a result, this model performs very poorly in network simulations. The second model, the run-length model (RLM), is only able to match a subset of the desired statistical parameters, however it is capable of matching the tail of the frame error duration PMD. The performance of the RLM is better than that of the N-state Markov model, but is hampered by the inability to match the good frame run duration PMD. The final model presented is the three-state run-length model (3SRLM). This model does a good job of approximating the key statistical parameters and matching the tails of both the frame error duration PMD and the good frame run duration PMD.

Chapter VI presents the network performance for the channel approximations described in Chapter IV and Chapter V. It is shown that the 3SRLM's ability to match the desired PMDs of frame error duration and good frame run duration, produces a good approximation of network throughput for the flat Rayleigh fading channel. Finally conclusions are drawn in Chapter VII.

CHAPTER II

BACKGROUND

In wireless networks, a fundamental issue in performance evaluation is the impact of the channel upon packet data throughput, delay and jitter. The wireless channel is susceptible to a number of impairments including multipath, fading, shadowing, interference and noise. These impairments can cause drastically different performance results when compared to wireline networks. Of these impairments, the concept of fading channels merits discussion. This chapter provides the appropriate background concepts concerning the fading channel, the representation of modulated signals and the block-coded modulation scheme specified in 802.11b called complementary code keying (CCK).

A. The Fading Channel

The term *fading* is used to describe a rapid fluctuation of a radio signal's magnitude over a short time period. Fading is the result of interference between two or more versions of a transmitted signal arriving at a receiver. Due to the nature of electro-magnetic (EM) wave propagation, a single transmission from a wireless device will often encounter 'reflective' objects resulting in multiple versions of the transmitted waveform that are attenuated, phase shifted and delayed in time. The time difference between arrivals plays an important role in channel modeling for wireless

communications. When the temporal dispersion is small compared to the transmitted symbol period, the channel alters the received SNR but does not induce inter-symbol-interference. This type of fading is often modeled as a Rayleigh or Rician distributed random processes. Jakes' [25] original work is one of the most heavily cited references on the generation of a Rayleigh process.

When the time dispersion of the transmitted symbols is large with respect to the symbol period, the channel induces inter-symbol interference (ISI). Rapapport [26] presents some common time dispersion parameters to grossly quantify the multipath channel. The parameters include *mean excess delay, rms delay spread,* and *excess delay spread (XdB),* each of which can be derived from a power delay profile (PDP). The PDP is a measurement of received signal power over time and is a measure of a channel's impulse response.

Mean excess delay spread is the first moment of the power delay profile. This parameter is calculated from the impulse response magnitude samples a_k and their corresponding sample times t_k as

$$\bar{t} = \frac{\sum_k a_k^2 t_k}{\sum_k a_k^2}. \qquad (2.1)$$

The metric of RMS delay spread is the square root of the second central moment of the power delay profile displayed as

$$s_t = \sqrt{\bar{t}^2 - (\bar{t})^2} \qquad \text{where,} \qquad (2.2)$$

$$\bar{t}^2 = \frac{\sum_k a_k^2 t_k^2}{\sum_k a_k^2}. \qquad (2.3)$$

The maximum excess delay (XdB) is the maximum time delay during which multipath energy falls to within XdB of the maximum received power. This parameter measures the time during which 'significant' ISI is present, where the term significant is defined by the value of 'XdB'.

Each of the parameters presented above rely upon the knowledge of the received power a_k^2 over time from a specific transmitted waveform. Such knowledge can be gained through the process of channel sounding. Many techniques have been developed for observing temporal responses of a wireless channel [26]. Of these techniques, the sliding correlator provides very accurate multipath arrival times and attenuation measurements when compared to other forms of channel sounding including: direct pulse, single tone and frequency modulated continuous wave [27]. Thus by performing channel measurements with a sliding correlator, one can achieve accurate observations of the temporal dispersion in specific channels.

1. Flat Fading Channel

The term flat fading is used to describe multipath channels that contain an RMS delay spread (2.2) much smaller than the desired symbol period of a communication scheme. If a signal labeled $g_s(t)$ is transmitted in a wireless channel, the resulting signal observed at a receiver can be written as

$$g_r(t) = \sum_{i=1}^{N} A_i g_s(t - t_i)\exp(jq_i) + N(t), \tag{2.4}$$

where A_i, q_i and t_i are defined as:

$$\begin{aligned} A_i &= \text{attenuation of the } 'i^{th}' \text{ path,} \\ q_i &= \text{phase shift of the } 'i^{th}' \text{ path,} \\ t_i &= \text{time delay of the } 'i^{th}' \text{ path.} \end{aligned} \tag{2.5}$$

In general, flat fading occurs when the maximum difference between any two significant delay paths is much less than the desired symbol period. When flat fading occurs, the term $g_s(t-t_i)$ can be approximated by the term $g_s(t)$, and (2.4) becomes

$$\begin{aligned} g_r(t) &= g_s(t)\sum_{i=1}^{N} A_i \exp(jq_i) + N(t) \\ &= g_s(t)g + N(t). \end{aligned} \tag{2.6}$$

Thus flat fading will cause the transmitted signal to be altered by a multiplicative variable g and an additive noise term $N(t)$. The term random fading process refers to the variation of the multiplicative term with respect to time. It is not obvious from (2.6) that the multiplicative term should vary with time, however as the environment is altered the attenuation and phase shifts of the different mulitpaths will change. This results in a change in the multiplicative term. It should be noted that the rate of change in the environment affects the rate of change in the multiplicative term.

Some slight algebraic manipulations can be performed upon the multiplicative term in (2.6) to arrive at

$$g = x + jy = \sum_i A_i \exp(jq_i), \tag{2.7}$$

$$x = \sum_i A_i \cos(q_i), \qquad y = \sum_i A_i \sin(q_i). \tag{2.8}$$

If one assumes there are a large number of paths (i.e. N is large) and that no single path has considerably less attenuation than the rest (i.e. similar values for A), then the central limit theorem (CLT) can be applied to both the terms x and y. The CLT results in both x and y being approximated as Gaussian random variables and the multiplicative term g being approximated by a complex Gaussian random variable. It is well known that for these assumptions, the amplitude of the multiplicative term will have a Rayleigh distribution while the phase of the multiplicative term will have a uniform distribution. The term *Rayleigh fading* is commonly used when one applies the CLT in deriving the multiplicative term. So, a flat Rayleigh fading channel can be modeled as shown in Fig. 1.

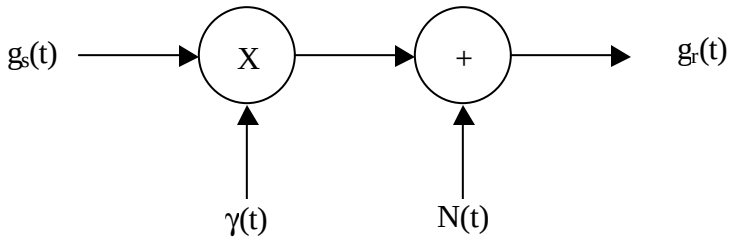

Fig. 1 Flat Rayleigh Fading Channel Model

If either the transmitter or receiver is in motion, one can argue that the fading term *g(t)* can be appropriately represented as a zero mean Gaussian process with a power spectral density (PSD) of

$$S(f) = \frac{P_R}{p\sqrt{1-(f/f_d)^2}} \quad |f| < f_d. \tag{2.9}$$

The received power is represented as P_R, and the Doppler frequency is represented as f_d. The time domain method of creating such a random process is described in chapter 3.

Flat Rayleigh fading channels are very prevalent within the literature. They contain almost no variability as the Doppler frequency is the only variable in (2.9) and it simply alters the speed of the process. Thus, the flat Rayleigh fading channel offers researchers the ability to easily reproduce experiments without the need to question assumptions in channel variables. The popularity of the flat Rayleigh fading channel and its pervasiveness in the literature is justification for its inclusion in network simulations.

2. Frequency-Selective Fading Channel

When the multipath delay spread is significant with respect to the symbol period, the channel filters the transmitted signal. Conceptually, the channel's passband bandwidth is smaller than the transmitted signal's bandwidth resulting in distortion of the transmitted signal. This distortion causes inter-symbol interference (ISI) and is commonly referred to as frequency selective fading. The term frequency selective comes from the observation that the channel exhibits different gains for different frequency components. A common rule of thumb [26] is that a channel can be considered frequency selective if the symbols period $T_s < 10\ \sigma_\tau$. In general, frequency selective fading channels are more difficult to model and reproduce than flat fading

channels. The frequency selective channel contains many variables where any one choice can be questioned as to applicability for a given scenario. In an effort to determine the channels one might experience, a study was conducted to measure the impulse response of an office environment. The results of this study are presented in the next section.

3. Channel Measurements

Ray-tracing techniques offer an analytical approximation for estimating the propagation of electro-magnetic waves in specific environments [28], [29], [30]. It was determined that these techniques would not offer the desired accuracy [31], and thus channel measurements were taken to observe the indoor environment. Channel sounding measurements were taken for an indoor office suite and the fuselage of a large commercial aircraft. The measurements were performed using a sliding correlator system provided by Motorola. Sliding correlators generate a continuous wave carrier signal that is modulated by a pseudonoise sequence (or PN sequence) of significant length. The chip rate employed by the sliding correlator will determine the granularity of time achievable while measuring a channel's impulse response. The modulated signal is then amplified and exposed to a bandpass filter. A generic sliding correlator transmitter is shown in Fig. 2. The waveform produced by the transmitter in Fig. 2 can be expressed as [27]

$$s(t) = \sum_{n=0}^{N-1} a_n p(t - nT) \cos(2\pi f_c t), \tag{2.10}$$

where a_n ={-1,+1} is the PN sequence, T is the chip period and p is the pulse shape.

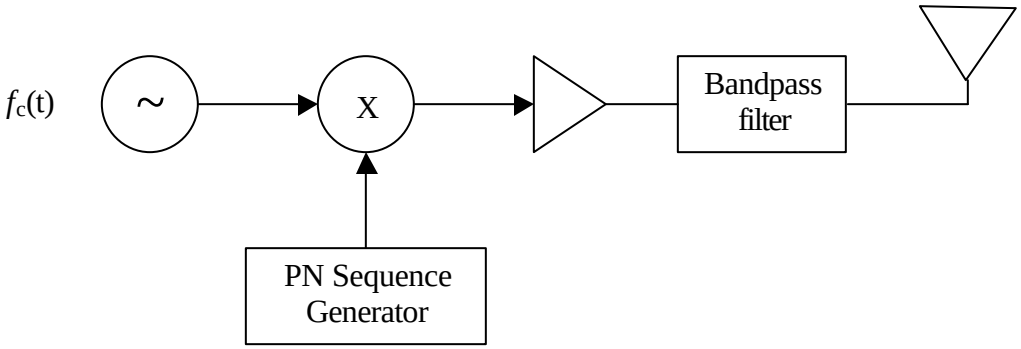

Fig. 2 Generic Sliding Correlator Transmitter

The receiver performs a correlation against the same PN sequence used in the transmitter but at a slightly slower chip rate. By using different chip rates, the local (receiver) PN sequence 'slides' past the received sequence thus performing a correlation between the two sequences over time. Due to the near self-orthogonality of the chosen PN sequence, the cross-correlation of the signals will be maximized only when the two signals are synchronized in time. At times when the two sequences are not synchronized, the resulting cross-correlation is essentially zero. A wireless channel often results in multiple paths between the transmitter and the receiver where each path is attenuated, phase shifted and delayed in time as shown in (2.4). When the received signal is correlated against the PN sequence, each path will produce a strong correlation at time t_i, or at the relative delay of the path from the LOS path. Therefore, the output of

the receiver correlator yields an estimate of the arrival time t_k and relative strength a_k parameters used in (2.1), (2.2) and (2.3).

The channel sounder utilized in this research, the ST-515, employs a sliding correlator operating at a transmission rate of 40Mcps sending a chip sequence of 16,383 chips to a receiver with a sampling rate of 120Msps. The resulting temporal resolution achieved by the ST-515 under these settings is ~8.33ns. Each measurement requires a calibration routine and synchronization procedure to ensure the receiver is observing the channel at the appropriate time. A sample power delay profile is presented for a severe case of multipath in Fig. 3. The sample was measured in the cabin of a commercial aircraft and shows a channel with a 10dB *maximum excess delay* of ~200ns.

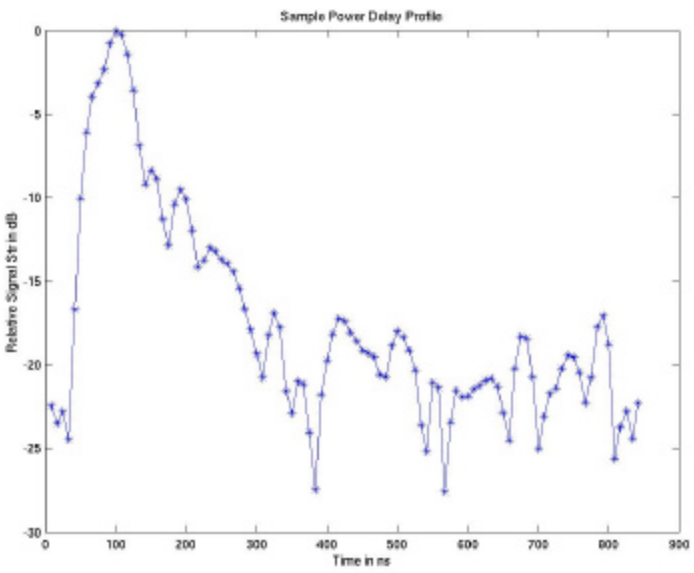

Fig. 3 Sample Power Delay Profile

Some 85 different channel measurements were taken in a small suite of 7 offices in the Wisenbaker and Engineering Research Center (WERC) at Texas A&M

University. Measurements included line-of-sight (LOS) paths, non-LOS paths and one set of extreme non-LOS paths between the adjacent hallway and a middle office. Each measurement was analyzed for *mean excess delay*, *rms delay spread*, and *maximum excess delay spread (10dB)*. The results of all 85 tests are presented in Fig. 4 to provide sample values of the temporal dispersion encountered in an office environment. Given that the symbol period of an 802.11b chip is ~91ns, the charts clearly indicate that these devices will encounter only moderate frequency selectivity.

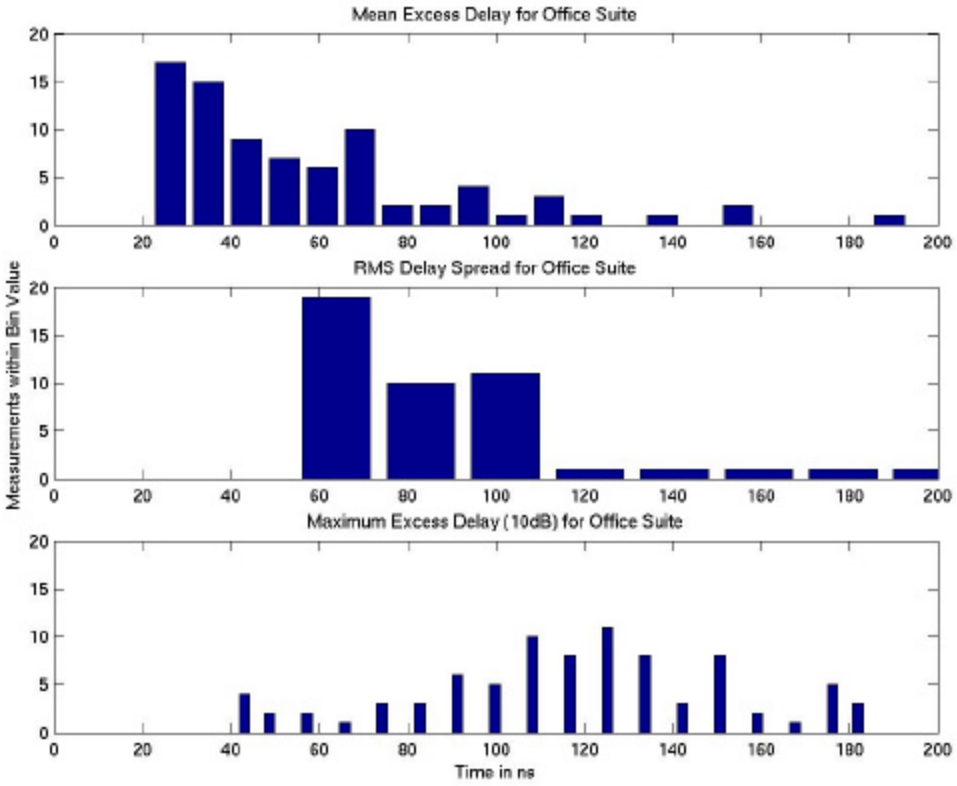

Fig. 4 Temporal Distortion Measurements of an Office Environment

Some 77 different channel measurements were taken inside the fuselage of a large commercial aircraft. Measurements included line-of-sight (LOS) paths within a

single cabin and non-LOS paths between two or more cabins. Each measurement was analyzed for *mean excess delay*, *rms delay spread*, and *maximum excess delay spread (10dB)*. The results of all 77 tests are presented in Fig. 5. The aircraft results contain even more significant temporal distortion than the office environment and might justify the use of a frequency selective channel.

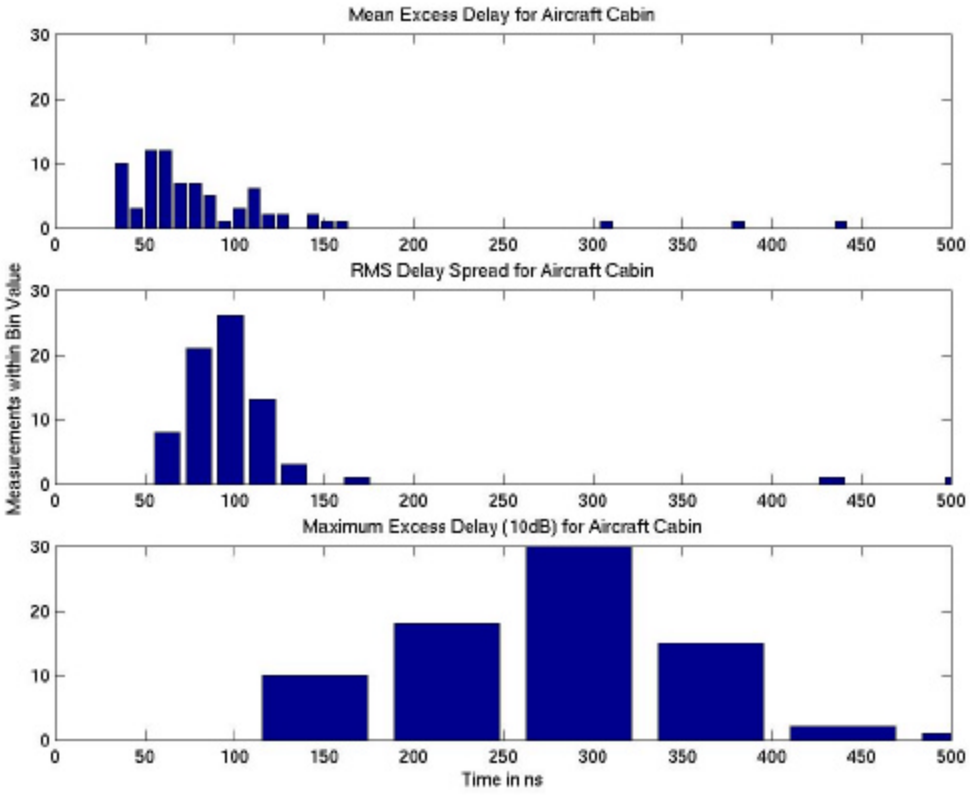

Fig. 5 Temporal Distortion Measurements of an Aircraft Fuselage

The moderate frequency selectivity of the channel sounding results combined with the simplicity and reproducibility of the flat Rayleigh fading channel model,

justifies focusing efforts on producing network channel approximations for the case of flat Rayleigh fading channels.

B. Representation of Modulated Signals

Communication requires the transfer of information between parties. In the case of wireless communications, it is convenient to alter a message by transforming it into a series of pulses whose frequency content is within the passband of the wireless channel. Carrier modulation allows a direct manipulation of a high-frequency carrier signal to obtain a desired frequency and message content. A carrier signal

$$s(t) = A(t)\cos(w(t) + f(t)), \tag{2.11}$$

is usually modulated in one or more of the three following areas: amplitude ($A(t)$), phase ($f(t)$) and frequency ($df(t)/dt$). Due to rapid fluctuations of amplitude in the wireless channel, it is often desirable to restrain modulation to phase and frequency when possible. A common alternative to the representation of (2.11) is the Cartesian-form

$$s(t) = [A(t)\cos(f(t))]\cos(w_c(t)) \quad - \quad [A(t)\sin(f(t))]\sin(w_c(t)). \tag{2.12}$$

This form does a good job of representing how most modulation actually occurs, through the use of inphase ($I(t)$) and quadrature ($Q(t)$) components. By the assignments

$$\begin{aligned} I(t) &= A(t)\cos(f(t)) \\ Q(t) &= A(t)\sin(f(t)), \end{aligned} \tag{2.13}$$

one can realize a modulator using the simple procedure outlined in Fig. 6.

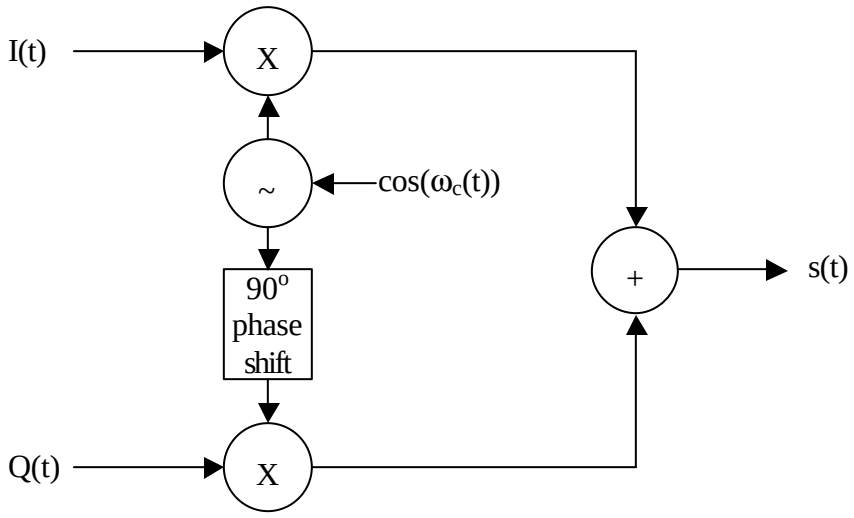

Fig. 6 I/Q Modulator Diagram

Quadrature phase shift keying (QPSK) is a well-known form of carrier modulation that is employed in the wireless local area networking protocol 802.11b. This scheme is very efficient and can be implemented by simply choosing *I(t)* and *Q(t)* to be of the set {+A, -A}. Creating a possible four combinations (symbols), each of which is capable of representing two bits of information. It is worth noting that the standard does not impose a particular pulse shape on the symbols, however there is a spectral mask, under which the transmitted PSD must remain. To comply with the mask, some sort of pulse shaping is required. The resulting waveform is

$$s(t) = \pm A\cos(w_c(t)) \quad + \quad \pm A\sin(w_c(t)). \tag{2.14}$$

It is well known that the Maximum Likelihood (ML) detector is optimum for the case of equally probable transmitted symbols [32]. For the special case of an additive

white Gaussian noise (AWGN) channel, the ML detector reduces to a minimum distance detector. A minimum distance detector simply finds the signal s_m (from the transmitter's library) that is closest in distance to the received signal r_k given a signal energy of E_s,

$$\hat{s}_k = \min_m \{\langle r, s_m \rangle\} = \max_m \left\{ 2\int_0^T r(t)s_m(t)dt - E_{s_m} \right\}. \qquad (2.15)$$

The minimum distance detector can be represented in the form of the well-known correlator detector shown in Fig. 7.

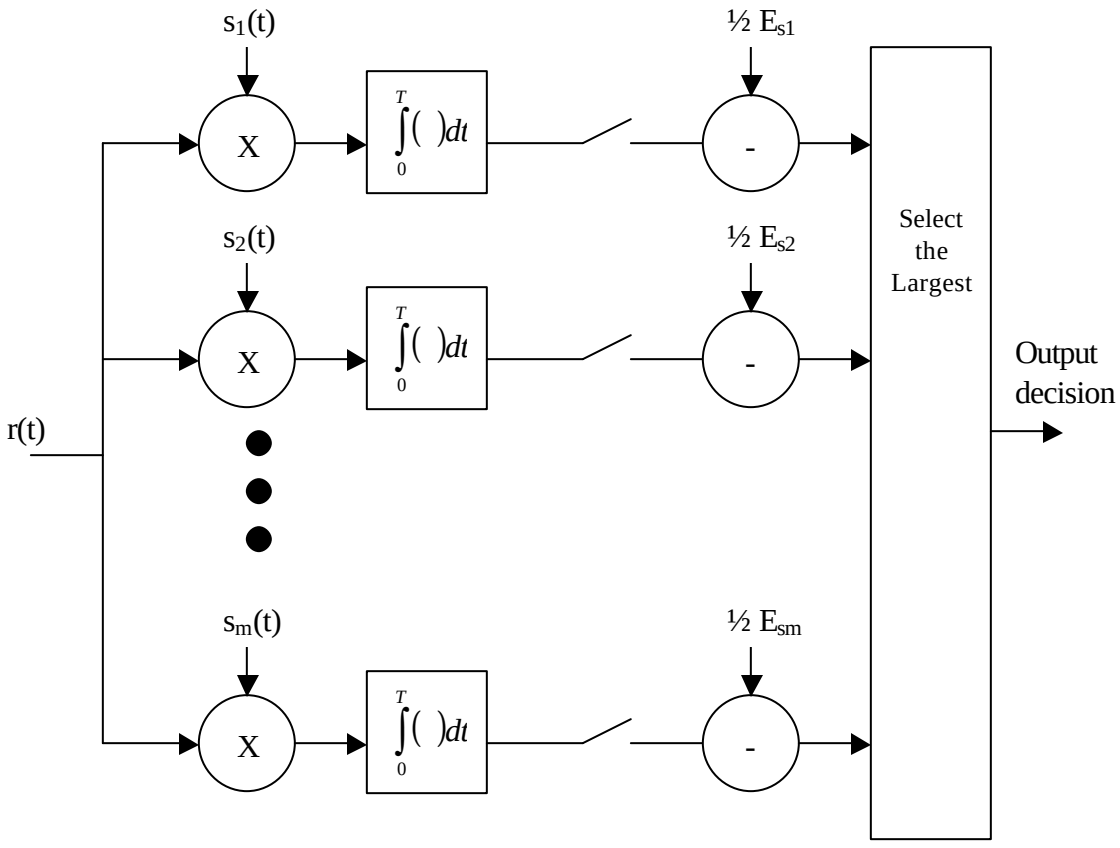

Fig. 7 Correlator Detector

As shown in Fig. 7, the correlator detector looks for similarities between the received signal and the known symbol alphabet ($s_1 \ldots s_m$). For additional details see [32]. The details of 802.11b modulation and coding are covered in the next section.

C. Complementary Code Keying

The wireless standard 802.11b utilizes a block-coded modulation scheme called complementary code keying (CCK) [33]. This scheme combines eight bits into a codeword that is then represented by eight QPSK symbols. Symbols are grouped into frames and pilot symbols are added to aid in synchronization. The complied frames are then transmitted at a rate of 11Mbps. The modulation procedure is depicted in Fig. 8.

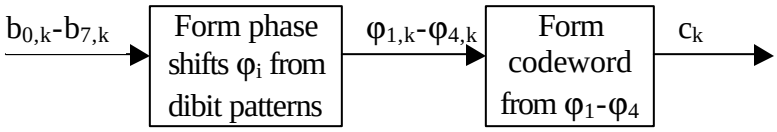

Fig. 8 Codeword Formation for 802.11b

The first step shown in Fig. 8 converts eight bits ($b_{0,k}$-$b_{7,k}$) into four phase shifts ($j_{1,k}$-$j_{4,k}$) through a form of differential encoding shown in Table I.

Table I. Dibit to Phase Conversion for 802.11b 11Mbps

Dibit pattern $[d_i,d_{i+1}]$ (d_i is first in time)	Phase
00	0
01	$\pi/2$
10	π
11	$3\pi/2$ ($-\pi/2$)

Following the formation of the phase terms, a codeword is constructed as

$$c = \left\{ e^{j(j_1,j_2,j_3,j_4)}, e^{j(j_1,j_3,j_4)}, e^{j(j_1,j_2,j_4)}, -e^{j(j_1,j_4)}, e^{j(j_1,j_2,j_3)}, e^{j(j_1,j_3)}, -e^{j(j_1,j_2)}, e^{j(j_1)} \right\}. \quad (2.16)$$

The codeword consists of eight phases, each representing a single QPSK symbol. CCK as described in this chapter, is simply a rate ½ block code with a total of 256 codewords. As such, one can form a correlator detector as shown in Fig. 7 with each s_i representing one of the 256 codewords. While this approach is inefficient for hardware architectures, it is surprisingly efficient for use in software simulations. It should be mentioned that this is a coherent receiver, and thus requires knowledge of the received signal's phase. All simulations conducted in this research assumed perfect phase knowledge for each transmitted frame. This assumption is justified as a minimum of 1056 pilot symbols are transmitted with each 802.11b frame operating at 11Mbps, and each frame has a maximum size of 20,528 symbols (including pilot symbols).

The performance of CCK in AWGN can be easily approximated by forming a basis function for each of the QPSK symbols in a CCK codeword. For a codeword of the form

$$c_i = \left\{ e^{jq_{i,0}}, e^{jq_{i,1}} \cdots e^{jq_{i,7}} \right\} \quad \text{where} \quad q_i = \left\{ 0, \frac{p}{2}, p, \frac{3p}{2} \right\} \quad i = \{1,2 \cdots 256\}, \tag{2.17}$$

one can define eight basis functions of the form

$$f_r(t) = \begin{cases} \sqrt{\frac{T_s}{8}} & \text{for} \quad r\frac{T_s}{8} \le t < (r+1)\frac{T_s}{8} \\ 0 & \text{Otherwise} \end{cases} \tag{2.18}$$

$$\text{where } r = \{0,1 \cdots 7\},$$

to represent the codewords in the form of

$$c_i(t) = \sum_{r=0}^{7} c_{i,r} f_r(t). \tag{2.19}$$

The next step in performance analysis involves computing the distance between all symbols,

$$d_{x,y}^2 = \left\| c_x(t) - c_y(t) \right\| = \left\| \sum_{r=0}^{7} c_{x,r} - c_{y,r} \right\|. \tag{2.20}$$

The minimum value of (2.20) for any two codewords will tend to dictate the performance at high signal-to-noise ratios (SNR). It can be shown that the minimum distance for CCK is $d^2_{min} = 8$. For each symbol, there are an average of 24 neighbors at this distance. Since the probability of symbol error for the AWGN case can be represented as [32]

$$\Pr\left(s_i \text{ chosen} \middle| \text{s}_\text{j} \text{ sent}\right) = Q\left(\sqrt{\frac{d_{i,j}^2}{2N_o}}\right), \tag{2.21}$$

one can approximate the union bound to only consider the nearest neighbors forming the performance equation

$$\Pr\left(symb-error\right) \cong 24Q\left(\sqrt{4\frac{E_b}{N_o}}\right), \tag{2.22}$$

Fig. 9 compares the approximation of performance in (2.22) against a simulation of CCK in AWGN.

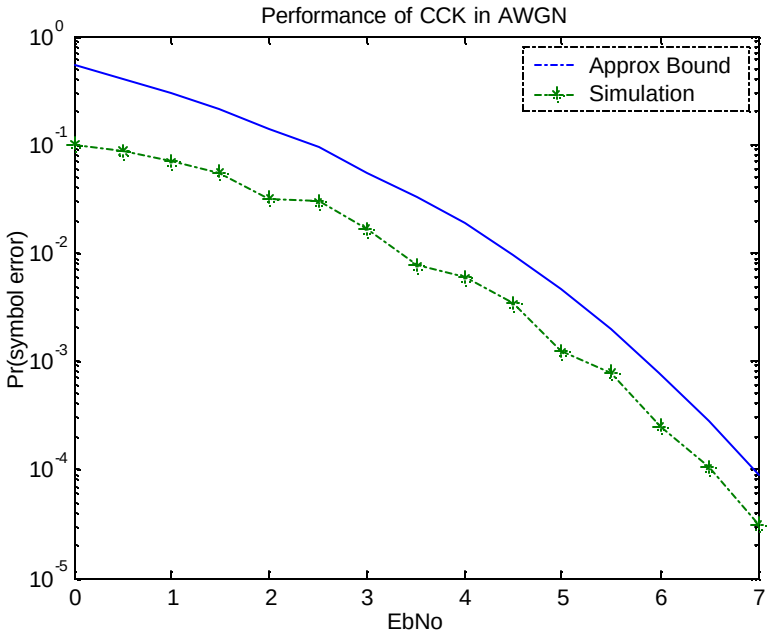

Fig. 9 Performance of CCK in AWGN

D. Summary

In this chapter, background concerning the fading channel and modulation was introduced. The concept of flat and frequency selective fading channels was reviewed. Channel sounding measurements were presented for an office suite as means of justification for considering the flat fading environment. The basic representation of carrier modulation was reviewed and an I/Q modulator presented. In closing, this chapter reviewed the modulation and detection of the 802.11b, 11Mbps modulation scheme and provided an approximated union-bound performance measure. The next chapter covers the operation of the network simulator n2 and the modifications performed to the simulator during this research.

CHAPTER III

NETWORK SIMULATOR 2

Specialized tools are prevalent and pervasive in most all disciplines. The area of networking is no different. Where carpenters rely upon plum lines and levels to ensure flush connections and sturdy constructions, networking researchers use simulations to evaluate protocols and observe obscure interactions. In early research, each investigator created unique simulations with little to no common functionality. Without a common tool, duplicating network research is problematic at best. Even more troubling is the lack of reusability in early simulators. The issues of reproducibility and reusability of unique simulations led researchers to investigate generalized tools that could be used as a common base for more advanced simulations. Network simulator 2 (ns2) is a mature network-simulation software tool that has been widely accepted by the networking community and has filled the need for a common networking tool.

A. Origins of ns2

Ns2 can trace its origin to the Network Simulation and Prototyping Tool (NEST) developed in 1988 [34]. NEST evolved into the Realistic and Large (REAL) [35] network simulator that served as the base for the development of ns. In 1995, DARPA funded the development of the Virtual InterNetwork Testbed (VINT) to "build a network simulator that will allow the study of scale and protocol interaction in the context of

current and future network protocols." The VINT project, through the collective efforts of the University of Southern California/Information Sciences Institute (USC/ISI), Xerox Palo Alto Research Center (PARC), Lawrence Berkeley National Laboratories (LBNL), and University of California Berkeley (UCB) jointly developed the network simulator known as 'ns'. The simulator was updated and reissued as Network Simulator 2. There exists a community of contributors to ns including Carnegie Mellon University (CMU), Sun Microsystems, and two new DARPA projects, SAMAN and CONSER. The 'Simulation Augmented by Measurement & Analysis of Networks' (SAMAN) project was formed to build robust networks through understanding the detection and prediction of failure conditions. CONSER, or Collaborative Simulation for Education & Research, was formed to extend the functionality of ns through network research (new module integration, existing module improvement) and network education (nam, educational scripts repository, ns-edu mailing list, ns tutorial, etc).

The development of ns2 provided researchers a tool for evaluating network protocols under varying network conditions, including the conditions presented in [36].

> "*Abstraction*: Varying simulation granularity allows a single simulator to accommodate both detailed and high-level simulations. Researchers study networking from the detail of an individual protocol to the aggregation of multiple data flows and the interaction of multiple protocols. The abstraction mechanisms in ns allow researchers to examine protocols without changing simulators and to validate abstractions by comparing detailed and abstract results.

Emulation: Most simulation experiments are confined to a single simulated world that employs only the protocols and algorithms in the simulator. In contrast, emulation allows a running simulator to interact with operational network nodes.

Scenario generation: Testing protocols under appropriate network conditions is critical to achieving valid, useful results. Automatic creation of complex traffic patterns, topologies, and dynamic events (link failures) can help generate such scenarios.

Visualization: Researchers need tools that help them understand the complex behavior in network simulation. Merely providing tables of summary performance numbers does not adequately describe a network's behavior. Visualization using the network animation tool *nam* provides a dynamic representation that allows researchers to develop better protocol intuition and aids in protocol debugging.

Extensibility: The simulator must be easy to extend if its users are to add new functionality, explore a range of scenarios, and study new protocols. Ns employs a split-programming model designed to make scripts easy to write and new protocols efficient to run."

Ns2 has proven itself a valuable tool for network researchers by providing a common simulator with sufficient complexity to allow protocol engineering to occur with insight into the operations and interactions of networking protocols. In general, ns2 is commonly utilized for three broad researching themes: selecting a mechanism or protocol variant among several options, exploring the complex behavior of networks, and investigating unforeseen multiple-protocol interactions. Researches have employed ns2 to investigate many protocols, some examples are shown below:

- TCP: selective acknowledgement, forward acknowledgement, fast restart, explicit congestion notification, rate-based pacing and asymmetric satellite links.

- Routing: robustness and convergence speed of integrated distance-vector and link-state routing protocols, multi-hop/wireless routing protocols, ad hoc routing in wireless networks, fairness of routing protocols

- Variable Traffic Sources: effects of variable traffic sources including web, ftp, telnet, constant bit rate, and stochastic sources.

- Queuing Disciplines: random early detection (RED), drop-tail, class-based queuing and others

- Multicast Transport: scalable reliable multicast (SRM) and variants, PIM variants, router support for multicast, congestion control, protocol validation and testing, reliable multicast

- Multimedia: layered video, audio and video quality-of-service, transcoding

- Wireless Networking: snoop and split-connection TCP, medium access control including evaluation of point-control function and distributed control function.

With its ever-increasing popularity, ns2 has become a worldwide benchmark by which network researchers evaluate protocols and validate innovative ideas and approaches.

B. Functional Description of ns2

Ns2 is a collection of networking modules with a well-defined method of interaction between the modules. Many modules have been designed and suggested for the specific OSI network layers utilized by TCP/IP networks, specifically: application, transport, inter-network, data-link (including medium access control), and physical layers. Ns2 allows researchers to evaluate unique techniques and networking protocols with minimal integration effort required.

Operationally, ns2 is a discrete event simulator that provides a split-level programming model where packet processing is performed in the system language of

C++, while setup is performed in the scripting language Otcl. C++ code is utilized for implementing the simulation kernel where all brute force simulations occur, thus taking advantage of the efficient computational structure of this language. The Otcl scripting language performs the definition, configuration and control of the simulation. The split-language approach cleanly separates the burden of simulator design, maintenance, extension and debugging from the user by providing the simulation programmer with an easy-to-use, re-configurable, programmable simulation environment. Ns2 is written in a modular fashion, resembling the OSI network architecture. It contains modules for simulating the application layer, transport layer, network layer, medium access control (MAC), and the physical layer. Each layer contains multiple modules from which a researcher can choose for any given simulation. That is, a researcher may choose to test a TCP-Reno transport layer across an IP network with an 802.11 defined MAC and a 2-ray fading physical layer. Assuming the appropriate C++ modules are included in the simulator, a researcher must simply alter his/her Otcl script file to choose specific modules for each layer. The networking layers are interconnected with a common interface so that a particular module can be altered easily. Each modular layer is comprised of multiple protocols. This allows for easy manipulation of a particular protocol within a network simulation.

The execution model for ns2 is best described by way of an example. Assume two nodes exist within a simulation where one node is sending data to the other. Each node has associated with it at least one module at each of the following layers,

application, transport & network, MAC and physical. The relationships between the layers are shown in Fig. 10.

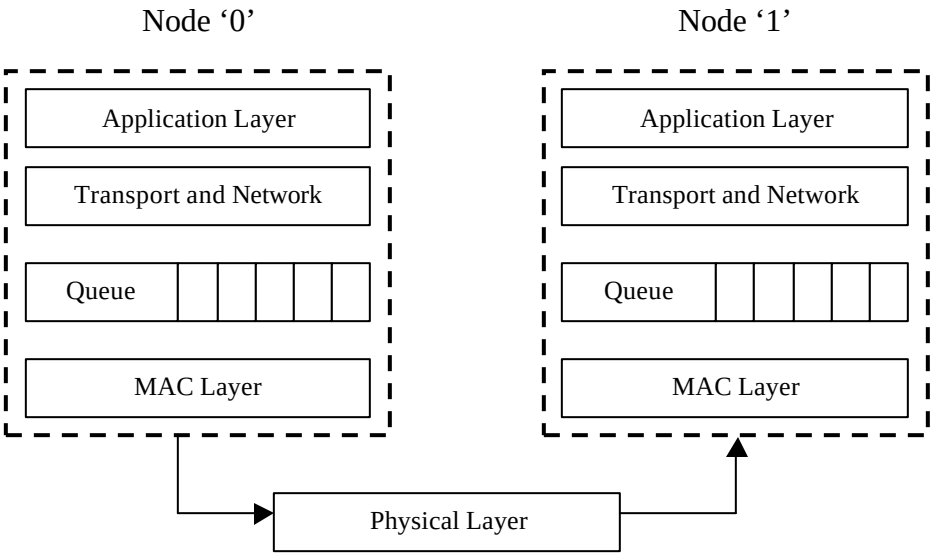

Fig. 10 Module Interaction Between Two ns2 Nodes for a Simplex Link

Modules are assigned to nodes depending upon their function. The application layer and transport & network layer can be set on a per-node basis, with any node containing multiple transport and/or multiple applications. The queue type, MAC and Physical layer are usually set to the same module type on all nodes for simple simulations and are therefore configured on a simulation basis. In this example, an application will be attached to a node thereby offering data to the transport/network module to be sent to another application. The transport/network module will contain protocols dictating the rules for packaging and sending the application's data. When packaged into a packet, the transport/network module places the packet into a queue to

exit the node. The MAC layer is then responsible for packaging a queued packet into a frame and offering the frame to the physical layer (or channel). One the other side, the receiving node is responsible for accepting the frame (MAC layer) and placing the encapsulated packet into a receiving queue for processing by the transport/network module. The transport/network module then defragments the incoming packets and performs automatic repeat requests (ARQ) as directed by the module's protocols. It should be noted that the above example contains a simplex link. In a duplex link, each node would have both a sending and receiving module for the queue and application modules.

Since each node can contain multiple applications, each requiring specific transport/network modules, ns2 provides a generic interface for attaching applications and transport agents. An "Agent" is created to handle transporting data from an application to another agent. Therefore, agents should come in pairs with both a source and sink. Agents define a transport methodology for data, allowing applications to choose the appropriate agent. This is very similar to how IP-terminals communicate, with each application having the ability to choose the associated transport layer (TCP/UDP). Data is created by an entity called an "Application", where applications are associated with agents. To summarize, a researcher would first create two network nodes and then create a source agent (like TCP/Reno) and attach it to node 0. Next, the sink agent would be created (like TCPSink) and attached to node 1. Then, the two agents (TCP/Reno and TCPSink) would be logically connected. Finally, an application is created (like FTP) and attached to node 0.

Agents require a “link” to facilitate communications. The link contains information about how nodes are connected including information about the data rate, delay and queue type (and length). So by setting up a “link” between node 0 and node 1, the two agents can communicate and the application can begin offering load to the network.

Another example is provided below to solidify the concepts of ns2 module interaction and to introduce the basics of Otcl scripting. This example is not intended nor is it sufficient to teach Otcl or ns2 scripting. Assume a simple 3-node, linear, point-to-point network like the one shown in Fig. 11 is desired. The following Otcl script will create this network and place an FTP application on one side sending data to the node on the far side. This example employs TCP Reno.

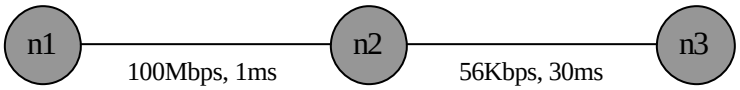

Fig. 11 Three-node Example Network

An example Otcl script follows below.

```
1       set ns [new Simulator]
```

```
2    set tf [open “trace_file_out.out” w]
3    $ns trace-all $tf
4    set n1 [$ns node]
5    set n2 [$ns node]
6    set n3 [$ns node]
7    $ns duplex-link $n1 $n2 100Mb 1ms Droptail
8    $ns duplex-link $n2 $n3 56Kb 30ms Droptail
9    set tcp_reno [new Agent/TCP/Reno]
10   $ns attach-agent $n1 $tcp_type
11   set sink_reno [new Agent/TCPSink]
12   $ns attach-agent $n3 $sink_reno
13   set ftp_n1 [new Application/FTP]
14   $ftp_n1 attach-agent $tcp_reno
15   $ns connect $tcp_reno $sink_reno
16   proc finish {} {
17   global ns tf
18   $ns flush-trace
19   close $tf
20   exit 0
21   }
22   $ns at 1.0 “$ftp_n1 start”
23   $ns at 100 “$ftp_n1 stop”
```

```
24      $ns at 120 finish
25      $ns run
```

The first line simply calls a new simulator and assigns it the handle 'ns'. The second two lines,

set tf [open "trace_file_out.out" w]

$ns trace-all $tf

open an output file called *trace_file_out.out* and assign this file to the handle *tf*. The next line tells the simulator to record all the information specified by *trace-all*. Lines 4-6 create three nodes naming them *n1*, *n2* and *n3*. The next two lines,

$ns duplex-link $n1 $n2 100Mb 1ms Droptail

$ns duplex-link $n2 $n3 56Kb 30ms Droptail

create a data link between nodes. Line 7, the first line shown above, creates a 100 Mbps link between nodes *n1* and *n2* with an average delay of 1ms. The queue for this link is specified as *Droptail*. As the second word implies, this line of code creates two links or a duplex link. At this point, the Otcl script has created the environment and placed links between nodes. The next step is to create transport agents for moving data between nodes. Lines 9 & 10, duplicated below, assign a new agent to the handle *tcp_reno*.

set tcp_reno [new Agent/TCP/Reno]

$ns attach-agent $n1 $tcp_type

The transport agent is declared to be of the sub-class *TCP*, specifically a subclass of TCP called *Reno*. After the transport agent is created and assigned a handle, the agent is

attached to node *n1*. Now, applications attached to node *n1* will be able to utilize the transport agent *tcp_reno*. The next two lines (11 & 12) create another agent, this time on node *n3*.

set sink_reno [new Agent/TCPSink]

$ns attach-agent $n3 $sink_reno

A new agent is created of the sub-class *TCPSink* and assigned the handle *sink_reno*, and then attached to node *n3*. Next, the Otcl script defines an application for node *n1*.

set ftp_n1 [new Application/FTP]

$ftp_n1 attach-agent $tcp_reno

Line 13 creates a new application of sub-class *FTP* and assigns it to the handle *ftp_n1*. The next line attaches the application to the transport agent *tcp_reno* located on node *n1*. Now the two transport agents (*tcp_reno* and *sink_reno*) must be connected. This is accomplished through line 15. The remaining code is used to set simulation parameters. For example, the *ftp* application begins sending data at time 1.0 as set by line 22 and turned off at time 100.0 by line 23. The simulation is killed at time 120, allowing sufficient time for queued messages to be delivered. In summary, the Otcl script file started a new simulation, created nodes, created and defined the links between nodes, assigned agents to nodes *n1* and *n3*, created an *ftp* application and attached it to the agent on node *n1*, linked the agents *tcp_reno* and *sink_reno*, and defined some events such as starting and stopping times of the application and simulation.

C. Modifications to ns2

1. Introduction

Point-to-point networks are of limited interest and thus ns2 was extended to support the operation of a local area network (LAN). LANs within ns2 operate by sharing a common transmission medium, very much like a bus network architecture. To facilitate this within the ns2 simulator, the data link is abstracted into three layers: link layer, medium access control layer and physical layer [37]. The physical layer contains a procedure for determining if the channel corrupts a transmitted frame. The MAC layer contains a certain set of functionalities such as: carrier sense, collision detection, collision avoidance and such. Since these functions affect both the sending and receiving of frames, they are implemented in a single MAC object. The Link layer sits above the MAC layer and contains two objects, a queue and a data link protocol.

This research focused upon altering the MAC layer *mac-802_11.cc* and creating new physical layers. In order to allow complete control over the physical channel, the MAC was altered to utilize a custom function each time the simulation attempted to send a frame from the MAC object. Information about the frame including length, sending and receiving nodes positions and transmission time are passed to the physical layer function. The physical layer function then generates an output to indicate frame integrity to the MAC layer. This approach allowed for quick and efficient modification of the physical layer without the necessity of altering the simulator for each variation. Another modification made to the MAC layer was the inclusion of the pilot symbols for

an 802.11b network. The original MAC layer did not adequately account for the 16-byte pre-amble of each frame. The modified MAC layer employed in all simulations presented in this document accounted for the pre-amble in the *mac-802_11* object and also settled pre-amble interference between frames with knowledge of the frames' received SNR. That is, when two frames collide at a node and both are in the pre-amble state, the node attempts to receive (or synchronize to) the frame at a higher SNR even if the other frame was transmitted first. Other than theses small modifications at the MAC layer, the majority of this work focused upon creating unique physical layers for the ns2 simulator. Four of these physical layer modules are included in this chapter: TAMU-PHY, MM-PHY, RLM-PHY and 3SRLM-PHY.

2. TAMU-PHY

From the outset, this research had to establish a baseline from which to compare possible channel models. This baseline had to exactly model a wireless channel and the transmission and reception of frames across it. The TAMU-PHY channel model was created to simulate the 802.11b baseband equivalent modulation, demodulation and transmission through a time selective flat Rayleigh fading process. This TAMU-PHY physical layer module operates at a symbol level for each packet transmission, providing a very thorough, and computationally intense simulation of 802.11b performance over a flat Rayleigh fading channel. In this module, the scheduler clock instance relays the frame transmission time, allowing the module to advance the fading process for lapsed time between frame transmissions.

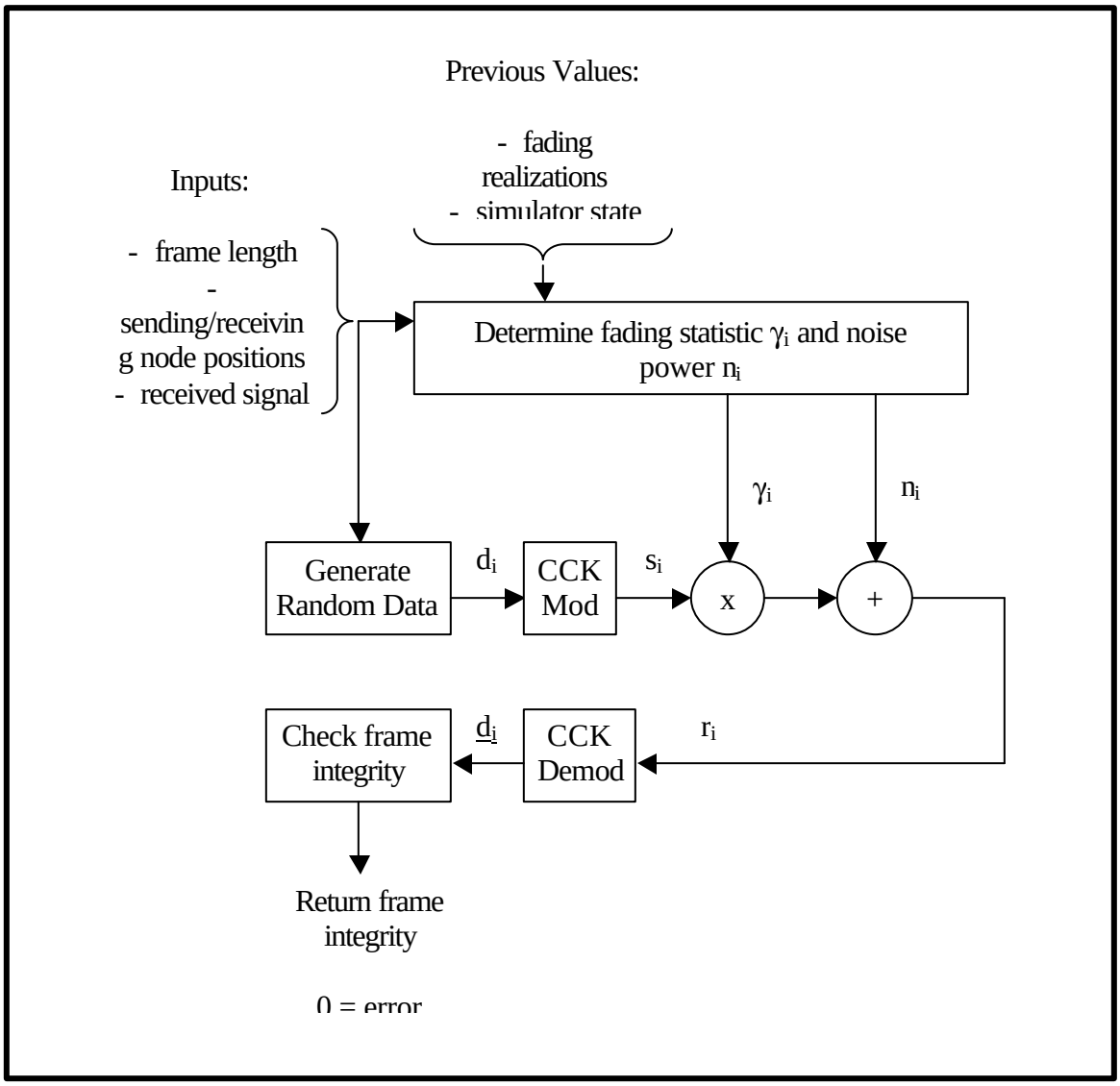

Fig. 12 Logical Operation of TAMU-PHY Module

To our knowledge, this is the first time a network simulator has incorporated a rigorous fading process that is based upon simulation time and does not require an assumption of i.i.d. fading statistics. Furthermore, modulation and demodulation of baseband symbols

is actually performed within the simulator. Fig. 12 depicts a logical operation of the TAMU-PHY module.

As shown in Fig. 12, information is gathered from the MAC module to assist in determining the fading statistic g, the noise power n and the number of bits in the transmitted frame. This information is combined with previous values of the fading statistic and the last 'simulator time' the routine was executed. Together, the information is sufficient to produce a realization of a flat fading process. Frame length information is used to generate a set of random data bits d_i that are subsequently modulated using a block code called complementary code keying (CCK). The resulting symbols s_i are subjected to the traditional flat fading channel (Fig. 2.1) including a multiplicative fading statistic g_i and an additive white Gaussian noise term n_i. Following the channel corruption, the altered symbols are received as r_i and subjected to a minimum-distance receiver that outputs an estimate of the data $\underline{d_i}$. The data estimate is compared against the randomly generated data d_i and the result is returned to the MAC layer.

Generating a Rayleigh random process within the simulator was a significant effort. The common 'time-domain method' [38] was chosen for two reasons: it provided an 'open ended' solution that did not require knowledge of the process' duration, and it required a minimal amount of memory to run. The goal of the Rayleigh random process generator is to produce a complex Gaussian random process with a prescribed power spectral density (PSD). The time-domain method produces a white Gaussian process and then passes this through a filter to achieve a colored Gaussian

process with a prescribed PSD. For the TAMU-PHY model, the desired PSD is describes as

$$S(f) = \frac{P_R}{p\sqrt{1-(f/f_d)^2}} \quad |f| < f_d. \tag{3.1}$$

The Doppler frequency of the channel is labeled f_d and the power of the fading channel (usually normalized) is P_R. The resulting autocorrelation is simply a Bessel function,

$$R(t) = P_R J_0(2pf_d t) \tag{3.2}$$

Since the filter input is a white Gaussian process, it has an autocorrelation defined by a Dirac delta function. It is well know, that for linear filters the autocorrelation of the filter output can be represented as

$$R_{OO}(t) = R_{II}(t) * h(t) * h(t), \tag{3.3}$$

where the input autocorrelation is $R_{II}(t)$, the output autocorrelation is $R_{OO}(t)$ and the filter's impulse response is $h(t)$. Since $R_{II}(t) = d(t)$, the output autocorrelation of the filter is simply $h(t)*h(t)$. Given that one desires to match the autocorrelation of (3.2) it follows that the filter should be designed to satisfy

$$|H(f)| \cong \frac{1}{\sqrt{f_d^2 - f}}. \tag{3.4}$$

Following the approach used in [38], a third-order filter was designed with the transfer function of

$$H(s) = \frac{w_o^3}{(s^2 + 2xw_o s + w_o^2)(s + w_o)}, \tag{3.5}$$

$$w_o = \frac{2pf_d}{1.2}. \tag{3.6}$$

The resulting value of ξ is 0.175 and fully describes the impulse response of a third-order filter capable of approximating the desired autocorrelation. It is often useful to map the continuous time transfer function into a discrete time filter using a bilinear transformation [39] as

$$H(z) = H(s)\Big|_{s=2f_s\left(\frac{z-1}{z+1}\right)}. \tag{3.7}$$

The solution to the transformation (3.7) can be organized into the form of

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3}}{a_0 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}}. \tag{3.8}$$

If one creates the following intermediate variables *A*, *B* and *C* defined as:

$$\begin{aligned} A &= (2f_s)^3, \\ B &= (2f_s)^2 w_o (1+2x), \\ C &= (2f_s) w_o^2 (1+2x), \end{aligned} \tag{3.9}$$

then, (3.8) can be solved with:

$$\begin{aligned} a_0 &= A + B + C + w_o^3, \\ a_1 &= -3A - B + C + 3w_o^3, \\ a_2 &= 3A - B - C + 3w_o^3, \\ a_3 &= -A + B - C + w_o^3, \\ b_0 &= w_o^3, \\ b_1 &= 3w_o^3, \\ b_2 &= 3w_o^3, \\ b_3 &= w_o^3. \end{aligned} \tag{3.10}$$

The term f_s in (3.9) is the frequency of the random process, or the rate of realizations per second. By saving the three previous inputs (I) and outputs (O) one can use the values in (3.10) to form the difference function

$$\begin{aligned} O[n] = b_0 I[n] + b_1 I[n-1] + b_2 I[n-2] + b_3 I[n-3] \\ - a_1 O[n-1] - a_2 O[n-2] - a_3 [n-3], \end{aligned} \tag{3.11}$$

and solve directly for a current output of the random process. The filter coefficients in (3.10) should be scaled to insure the filter has a gain of unity. An easy way of implementing this is by finding the filter's impulse response and normalizing the transfer function (3.8) by the root of the impulse response. It should be noted that there will be some transient responses at the beginning of the filter. To obtain acceptable results, initialize the filter by passing a sufficient number of inputs through the filter until all transients have passed. The number of inputs required will be inversely proportional to the channel's Doppler frequency. By following this procedure, the TAMU-PHY model is able to produce a Rayleigh fading process with f_s samples per second. The application of these realizations to the simulator is not trivial and required the use of sub-frames.

Network simulators produce a variety of traffic resulting in widely variable frame sizes. The difference in frame sizes produces difficulties in fitting fading process realizations with fade sizes. Ideally, the fading process would generate a single realization for every symbol transmitted, however as f_s grows the non-idealities of the filter approximation become prohibitive. Since the transmission rate of 802.11b is 11Mbps, it was desirable to have a single sample utilized for a group of bits. This approach is valid as long as the fading process is sufficiently slow. The number of bits

grouped per realization was determined by examining the number of bits in the smallest protocol packets. Each group of bits is called a sub-frame, where a single sub-frame contains 296 bits. The choice of using sub-frames makes this work extendable to block codes such as turbo-codes, interleavers or truncated maximum likelihood sequence estimators. It will be shown that other versions of the physical layer, such as the MM-PHY, require significant sub-frame sizes to appropriately set their parameters.

Another implementation detail not shown in Fig. 12 is the accounting for non-transmission time. That is, the simulator must account for the time between frame transmissions by advancing the random process a time equal to the idle period between frame transmissions. For frames transmitted at a fast rate, the idle period will be minimal and frames will experience correlated fading realizations. When frames are transmitted infrequently, the idle period between transmissions is significant and the frames will experience nearly independent fading. A more detailed accounting of the TAMU-PHY model is shown in Fig. 13.

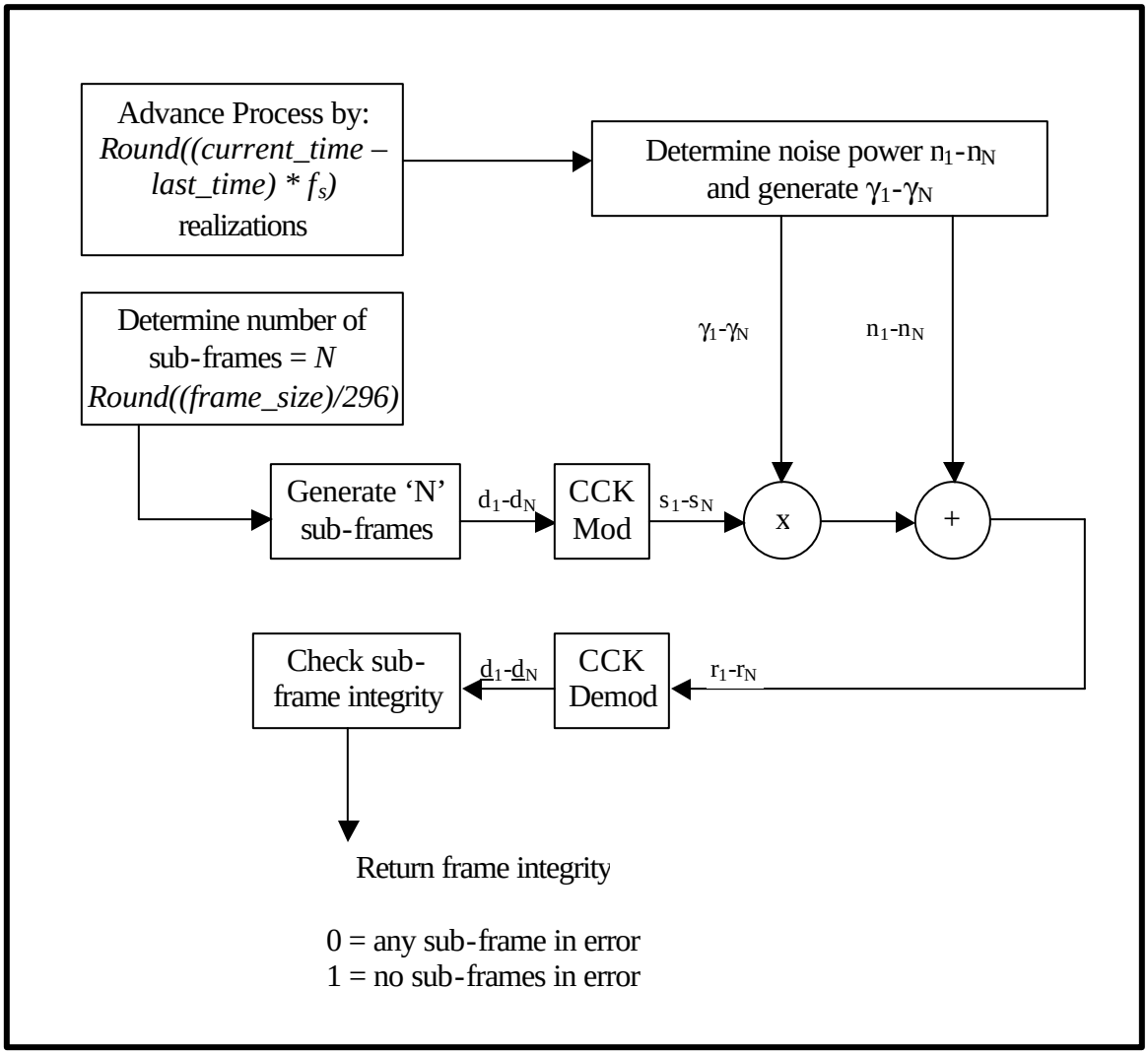

Fig. 13 Sub-frame Operation of TAMU-PHY Module

The details of the modulation and demodulation operations shown in Fig. 13 were presented in Chapter II. Most network simulations involve the transmission of

millions of packets, each containing thousands of bits thus resulting in extreme run times when using the TAMU-PHY module. While this physical layer module creates a very accurate representation of network performance across a flat Rayleigh fading channel, the run time required by the TAMU-PHY is prohibitive for most protocol investigations.

3. MM-PHY

The Markov-Model module (MM-PHY) was created to simulate the two-state Markov model channel approximation. This MM-PHY module operates by alternating between a 'good' and 'bad' state according to a set of transition probabilities. The occurrence of a 'bad' state indicates a channel error. In order to maintain the assumptions used to approximate the flat Rayleigh fading channel, the Markov model must operate with a fixed-length frame size. As network simulations involve the transmission of heterogeneous frame lengths, a single Markov model state transition cannot be used for all simulated frames. To our knowledge this issue of using a single state transition to represent heterogeneous frame sizes has not been adequately addressed in the literature. The severity of this issue becomes apparent in simulations, as small protocol packets should not progress the frame error process the same amount as larger data packets. In way of an example consider the case of a Markov model created to match a fixed-length frame size of 500 bytes, where three successive frames are in error. This model represents the occurrence of a burst error length of 1500 bytes; however, in the case of three protocol packets being sent, the actual bytes in error would be closer to

200. Thus the next packet transmitted in our example would not experience an error, even though it should.

This module utilizes a simplistic solution through the use of sub-frames. Whereby each frame is decomposed into a set of sub-frames, and each sub-frame is evaluated through a single Markov model state transition. The integrity of the actual frame will depend upon the integrity of each sub-frame. State transition probabilities are derived from observing the sub-frame error statistics of a wireless system. Therefore, the wireless system must be able to produce accurate sub-frame statistics so that the MM-PHY can be formed properly. The actual procedure for generating the transition probabilities from sub-frame statistics will be discussed later in this document. The logical operation of the MM-PHY is shown in Fig. 14. As with the TAMU-PHY, this module must also account for idle time between transmissions. Any occurrence of the 'bad' state during a sub-frame transmission time will result in an entire frame error. The MM-PHY is extremely efficient and requires minimal computational load, making it very attractive for most network simulations that desire to observe protocols in the presence of fading.

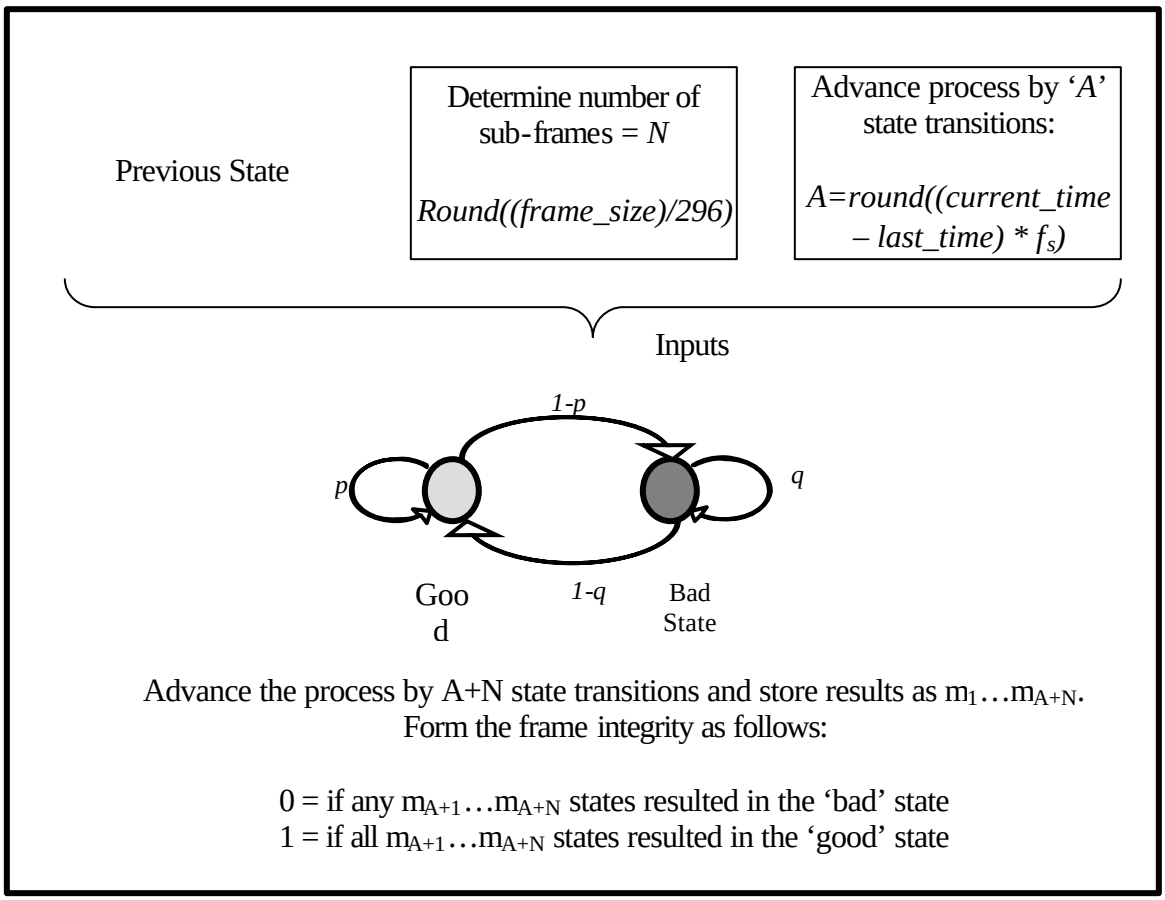

Fig. 14 Sub-frame Operation of MM-PHY Module

4. RLM-PHY

The run-length model (RLM) is very similar in form to the two-state Markov model, however the RLM differs greatly in its operation. The RLM contains two distinct states representing the occurrence of a 'fade' and the lack of a fade, or 'no-fade'. The state transition probabilities of the RLM are fixed so that the variability of this model comes from the duration of both the 'fade' and 'non-fade' states. In addition to this change, the RLM also allows for the possibility of frames transmitted during the

'non-fade' state to be in error. In the RLM-PHY, all frames transmitted during the 'non-fade' state have a fixed probability of error equal to *fer*. The logical operation of the RLM-PHY is presented in Fig. 15.

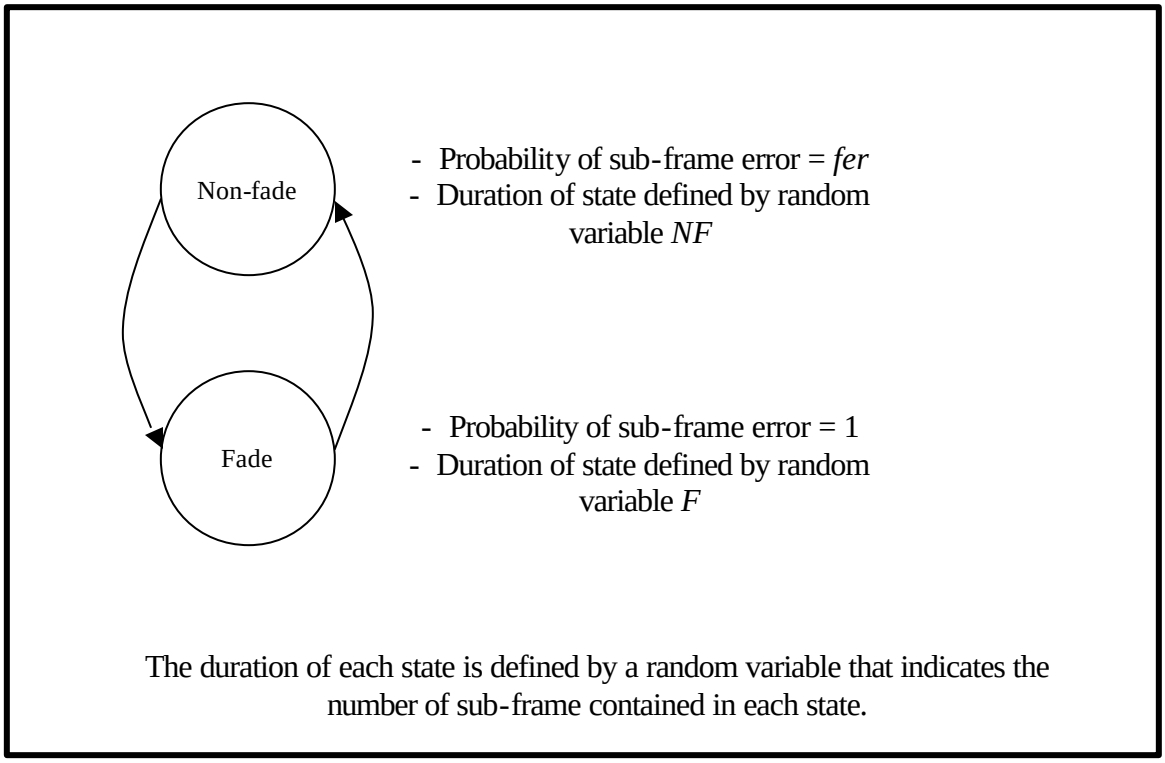

Fig. 15 Logical Operation of RLM-PHY Module

This model also accounts for idle times between frame transmissions by advancing the process an appropriate number of sub-frames according the amount of time lapsed since the last transmission. A simple counter is used to decrement the number of sub-frames remaining in any state before a transition occurs. Upon a state transition, a new number of sub-frames is found by generating a realization of either the *F* random variable (fade state) or the *NF* random variable (non-fade state). The

'expiration' of a state can occur during an idle period or during a frame transmission. Each sub-frame is evaluated for integrity according to the current state of the module, where in the 'fade' state the sub-frame is always in error and in the 'non-fade' state a sub-frame error occurs with probability = *fer*. The sub-frame operation of the RLM-PHY is described in Fig. 16.

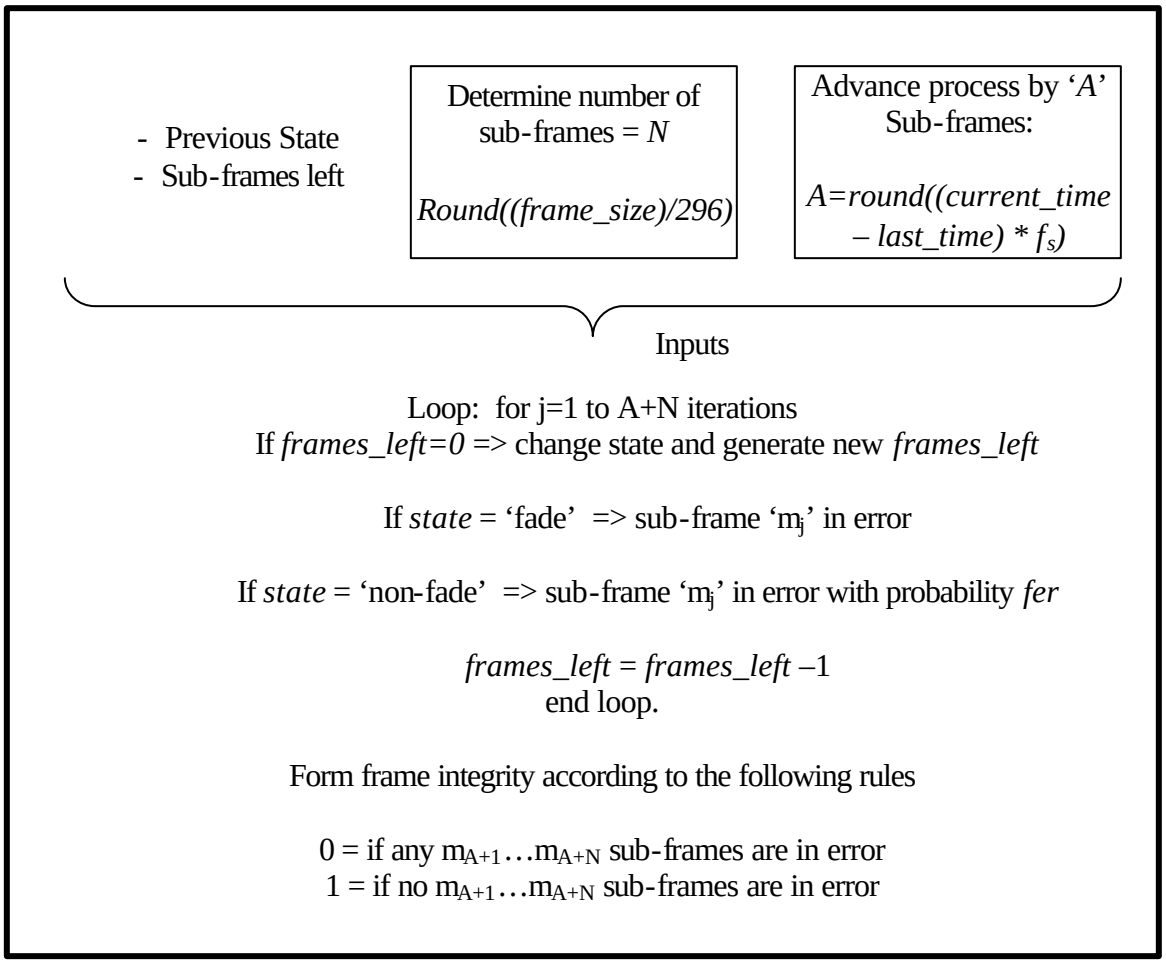

Fig. 16 Sub-frame Operation of RLM-PHY Module

5. 3SRLM-PHY

The three-state run-length model (3SRLM) is an extension of the RLM, with a new 'transition' state. The transition state better models the fading process by allowing the representation of a degraded SNR that is not necessarily a deep fade. For tractability, the transition state is modeled by a two-state Markov model. The operation of the 3SRLM-PHY is a simple combination of the MM-PHY and the RLM-PHY. As with previous models, the 3SRLM operates upon sub-frames and accounts for idle time between transmissions. The logical operation of the 3SRLM-PHY is shown in Fig. 17. The model begins in the 'non-fade' state and remains for a duration of sub-frames prescribed by a realization of the random variable *NF*. All sub-frames transmitted during the 'non-fade' state are unaltered and no errors occur. Upon expiration of the 'non-fade' state, the 3SRLM-PHY moves into the 'transition' state and remains for a duration of sub-frames according to the random variable *T*. The 'transition' state is composed of two 'sub-states' called 'good t' and 'bad t', where the relationship between the sub-states is defined by a two-state Markov model. The initial sub-state is chosen randomly according to a desired transition state probability of error (T_{fer}). Each instance of a sub-frame will cause the two-state Markov model to attempt a transition that might result in a sub-state transition between 'good t' and 'bad t'. The integrity of each sub-frame in the transition state is found exactly as in the MM-PHY with the 'bad t' sub-state resulting in a sub-frame error with probability one and the 'good t' sub-state resulting in no error with probability one. Upon expiration of the transition state, the 3SRLM enters the 'fade' state and remains for a duration of sub-frames according to the

random variable *F*. All sub-frames transmitted during the 'fade' state result in errors. Upon expiration of the fade state, the 3SRLM returns to the non-fade state and the cycle continues. The detailed sub-frame operation of the 3SRLM-PHY is shown in Fig. 18. It should be noted that the theory behind this model and its structure is presented in Chapter V, along with a method of parameter estimation to match specific frame error statistics.

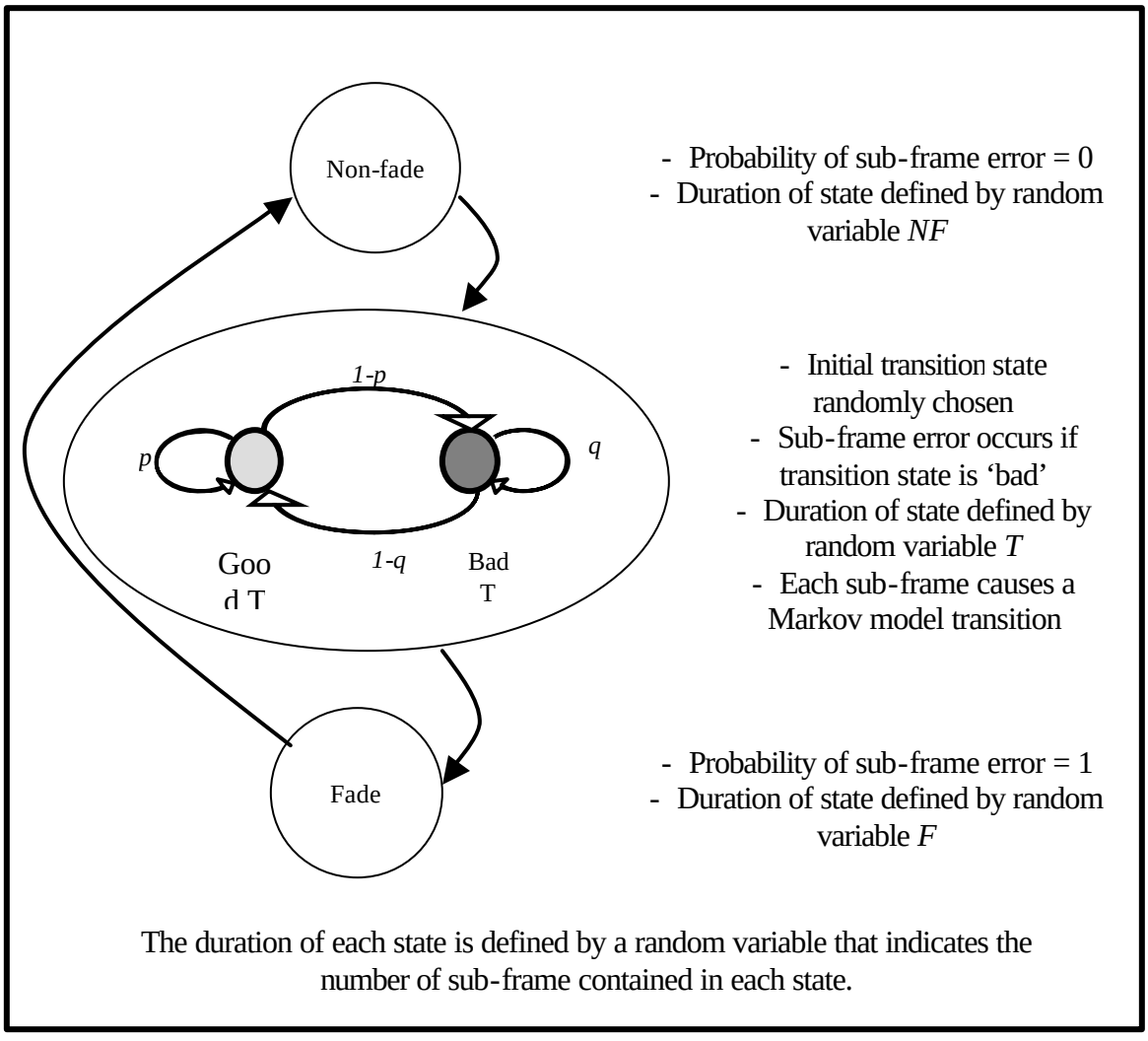

Fig. 17 Logical Operation of 3SRLM-PHY Module

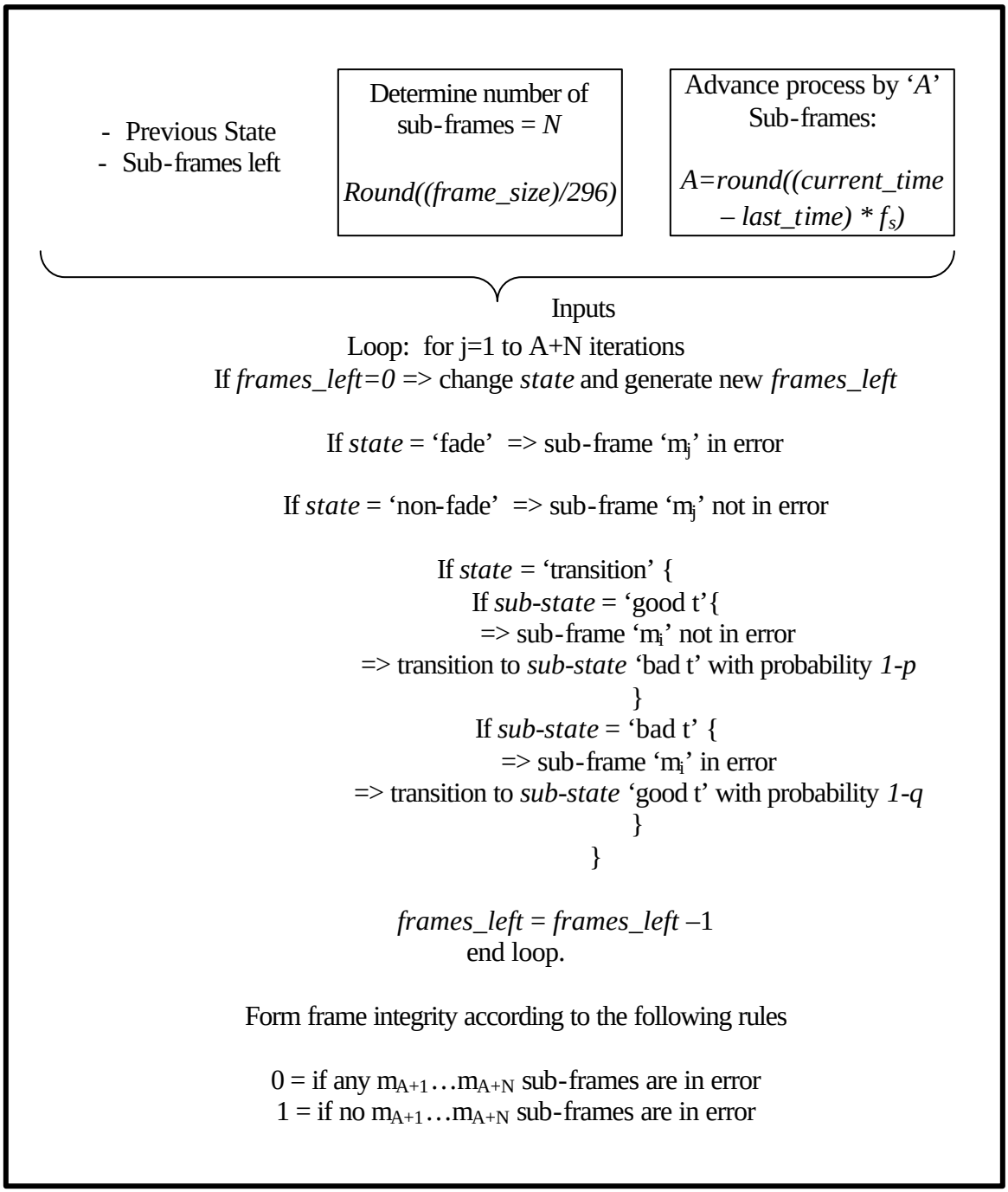

Fig. 18 Sub-frame Operation of 3SRLM-PHY Module

D. Summary

This chapter began by describing of the origins of the network simulation tool called ns2. A functional description is included in this chapter along with a brief example script. Four physical layer channel modules are presented in this chapter. Each module contains a functional description of the process employed to determine the integrity of a transmitted frame. The origins of the modules and parameter estimation for the MM-PHY, RLM-PHY and 3SRLM-PHY will be presented in Chapter V.

CHAPTER IV

MARKOV MODEL CHANNEL APPROXIMATION

Since most networking protocols perform observations at the 'packet level', there is considerable interest in focusing upon the block-error statistics of wireless channels. The intricate dependency of networking protocols upon the performance of the wireless channel motivates investigating the sensitivity of wireless network performance to higher-order block error statistics. In recent years, significant advances have been made in forming simplified Markov Models that approximate *some* key block error statistics of Rayleigh fading channels. These models are extremely efficient and require very little processing overhead to implement. However, in some fading environments, the two-state Markov Model channel approximation is inadequate and should be avoided.

The two-state Markov model channel approximation for the block error statistics of a flat Rayleigh fading channel has been well documented in the literature [14], [40], [41], [42], [43]. Wang [40], [41] presents an argument for using a Markov model to simulate a Rayleigh envelope random process. This argument was advanced through the publications of Zorzi, Rao, and Milstein [14], [42], [43]. Currently, there exists a set of work exploring the use of a computationally efficient channel model, which exhibits representative block error statistics of a single path Rayleigh fading channel.

The two-state Markov approach to modeling the frame error process of a wireless physical layer seems to enjoy wide acceptance among the networking community. This is evidenced by the large number of recent publications that have used a Markov model

channel approximation to study various network issues, e.g. [1], [2], [3], [4], [5], [9], [10], [11], [12], [13], [15], [44], [45], [46], [47], [48], [49], [50], [51], [52].

As summarized by Rao [53], network simulations are sensitive to the higher order statistics of error processes, such that the effects of fading at the physical layer have repercussions at progressively higher layers of the networking protocol stack. Due to the rapid acceptance of this channel approximation, there is a need to identify the appropriate, and more importantly inappropriate, conditions under which the two-state Markov Model should be employed.

First, this chapter covers the justifications for using a two-state Markov model to approximate the frame error process of a communication scheme operating over a flat Rayleigh fading channel. Next, results are presented as to the proper implementation of the two-state Markov model including arguments for simulating a random process using sub-frames. The statistics of the approximate channel are then compared to those gathered from the actual channel. Finally, the chapter concludes by quantifying the impact of employing a two-state Markov model through the use of network simulations.

A. Markov Approximation of Flay Rayleigh Fading

Much work has been conducted on the area of wireless network channel modeling. Recently, flat fading channel approximations using two-state Markov models have commanded the majority of the attention in this area. The term *fading* is used to describe a rapid fluctuation of a radio signal's amplitude over a short time period.

Fading is the result of interference between two or more versions of a transmitted signal arriving at a receiver. The time difference between arrivals plays an important role in channel modeling for wireless communications. When the temporal dispersion is small compared to the transmitted symbol period, the channel is said to exhibit flat fading. Flat fading alters the received symbol but will not induce inter-symbol-interference. These fluctuations are often modeled as a Rayleigh or Rician distributed random processes. Jakes' [25] original work is one of the most heavily cited references on the generation of a Rayleigh fading process.

The study of the finite-state Markov channel originates from the two-state (Markov) channel known as the Gilbert-Elliott channel [54]. In this channel model, the states correspond to specific channel conditions. In general, the states can be viewed as binary symmetric channels with given crossover probabilities of '0' and '0.5'. Therefore, the Gilbert-Elliott channel represents either a perfect or corruptive channel environment. Owing to its simplicity, the Gilbert-Elliott channel was utilized by early network simulations without proof of its appropriateness.

Wang [40] advanced the notion of a flat fading process being appropriately modeled as a Markov process through the use of mutual information between successive fading samples. The flat fading process is often modeled as a multiplicative complex function, $a(t)$, with unit energy and a covariance function, defined as

$$C_{aa}(t) = E[(a(t+t)-m)^{*}(a(t)-m)], \tag{4.1}$$

where $m=E[a(t)]$. The special case of $m=0$, produces a Rayleigh distribution for the envelope of the fading statistic $a(t)$. In the literature, it is widely acceptable to model the random process $a(t)$ with a band-limited, non-rational spectrum defined as [26]

$$S_{aa}(f)=\begin{cases} S_{aa}(0)\left[1-\left(\frac{f}{f_D}\right)^2\right]^{-1/2} & for\,|f|<f_d. \\ 0 & otherwise \end{cases} \tag{4.2}$$

This spectrum corresponds to the covariance function

$$C_{aa}(t)=J_0(2pf_D|t|), \tag{4.3}$$

where $J_0(*)$ is a zero-order modified Bessel function of the first kind. It can be noted, that the correlation properties of a random process, thus defined, will greatly depend upon the normalized Doppler frequency $f_D t$. As the mobile experiences significant normalized Doppler shifts, the correlation between successive samples becomes less and the mobile is said to experience "fast" fading. The alternative, "slow" fading, is a product of insignificant normalized Doppler shifts owing from a slowly varying channel that produces significant correlation in the fading process. Wang [40] develops an approach to evaluating the relative *value* contained in past samples of a Rayleigh fading process by examining the mutual information between samples. Specifically, Wang suggests the mutual information in sample R_i provided by the sample R_{i-2} with full knowledge of R_{i-1}, or $I(R_i;R_{i-2}|R_{i-1})$, is insignificant when compared to the mutual information of R_i and R_{i-1}, or $I(R_i;R_{i-1})$. A metric is formed and defined as

$$V_R=\frac{I(R_i;R_{i-2}|R_{i-1})}{I(R_i;R_{i-1})}, \tag{4.4}$$

where z_R is a measure of the value in knowing the previous sample with respect to the value of knowing the current sample in forming the future realization. In fact, a small z_R indicates that knowledge of a single past sample (R_{i-2}) is insignificant when compared to knowledge of the current state (R_{i-1}) when forming the next realization (R_i). The mutual information argument is the foundation for justifying the use of Markov models to approximate the performance of flat Rayleigh fading channels.

In [41], Wang stresses the importance of Markov state-transition probabilities upon the simulated channel's capacity and cautions that the use of Markov channel models with incorrect transition probabilities will cause significant errors in the channel capacity and thus cause erroneous results. To better match realistic values to Markov model channels, a two-state model was evaluated and parameters defined in [42]. Zorzi, Rao and Milstein provide a method of choosing the transition probabilities for a simple 'first-order Markov Model' to approximate the frame-error statistics of a flat Rayleigh fading channel. Since network protocols operate on the success or failure of frames, not bits, it follows that frame error statistics are of more importance than bit error statistics when evaluating the performance of networking protocols. To this extent, a new random process, $b(t)$, is generated as a function of the Rayleigh fading process $a(t)$, as

$$b(t) = j\,(a(t)). \tag{4.5}$$

In the scheme presented in [42], the block b_1 is successfully received if $|a_1|^2 > 1/F$, where F is called the fading margin. The random process $b(t)$ can be argued to be approximately Markov through evaluations of mutual information contained in previous

samples. The quantity V_b was defined and evaluated in [42] as a measure of the Markov model approximation's "goodness".

$$V_b = \frac{I\left(b_i ; b_{i-2} \middle| b_{i-1}\right)}{I\left(b_i ; b_{i-1}\right)} \tag{4.6}$$

The literature shows that V_b is in fact <<1 for "slow" fading, thus suggesting the first-order Markov approximation is adequate for representing the block error process of a flat Rayleigh fading channel. One significant issue in this approach is the determination of the fading margin '*F*', as it depends upon the communication scheme. An approach, suggested in [43], disregards the fading margin value and forms the transition probabilities of a two-state Markov model from block error statistics generated through simulating a communication scheme.

B. Proper Modeling of a Two-State Markov Model

This section provides some detail on the two-state Markov model channel approximation and how its transition matrix is defined. Start by defining a memoryless, homogeneous, irreducible and recurrent non-null two-state Markov process defined by the transition probabilities (see Fig. 19)

$$\Pi_j = \begin{pmatrix} p & 1-p \\ 1-q & q \end{pmatrix}, \tag{4.7}$$

where p is the probability of a successfully transmitting a frame given the previous frame was successfully transmitted, and 1-q is the probability of successfully transmitting a frame given the previous frame was in error.

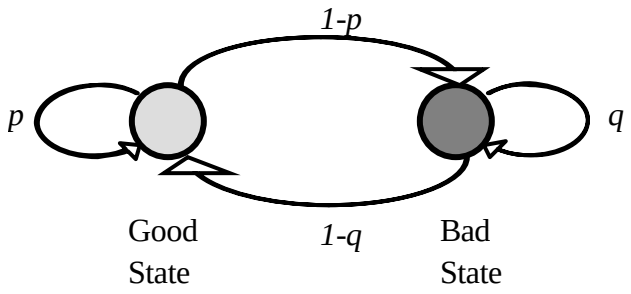

Fig. 19 Two-state Markov Model

Minor evaluation of the above transition probabilities results in the steady-state error rate (e) of this model,

$$e = \frac{1-p}{2-p-q}. \tag{4.8}$$

The quantities of e and average burst error length can be computed from any arbitrary communication scheme by observing the block error statistics produced while operating over a flat Rayleigh fading channel. Where e is set to be the average frame error rate produced by the simulation and 1/(1-q) is set equal to the average burst error length. With the knowledge of e and 1/(1-q), the calculation of the remaining transition probability p is trivial,

$$p = 1 - \frac{(1-q)e}{1-e}, \tag{4.9}$$

and therefore fully defines the two-state Markov model channel approximation. One should note, the frame statistics will be dependent upon frame size – and thus the transition probabilities will also depend upon frame size.

1. Independent vs. Correlated Fading Processes

Flat fading will produce time-correlated losses of signal such that multiple successive samples will be affected. Some network simulations assume that traffic is periodic and encounters nearly independent fading samples. This greatly simplifies the modeling of fading by ignoring any dependence upon time; however the two-state Markov model generates a correlated random process and thus does not perform independent fading. Care should be taken to avoid improperly modeling independent fading with a Markov model. The magnitude of a flat Rayleigh fading process is presented in Fig. 20 to illustrate the temporal correlation of fading samples.

As motivation for pursuing the use of modeling fading as a correlated random process instead of using a white process, results are presented from network simulations using both techniques. A simulation was conducted with one node sending data to a second node employing an 802.11b communication scheme across two channels. The first channel performs independent fading upon every transmitted frame. The second channel performs a fading process whereby successive transmissions will tend to experience a similar fading environment.

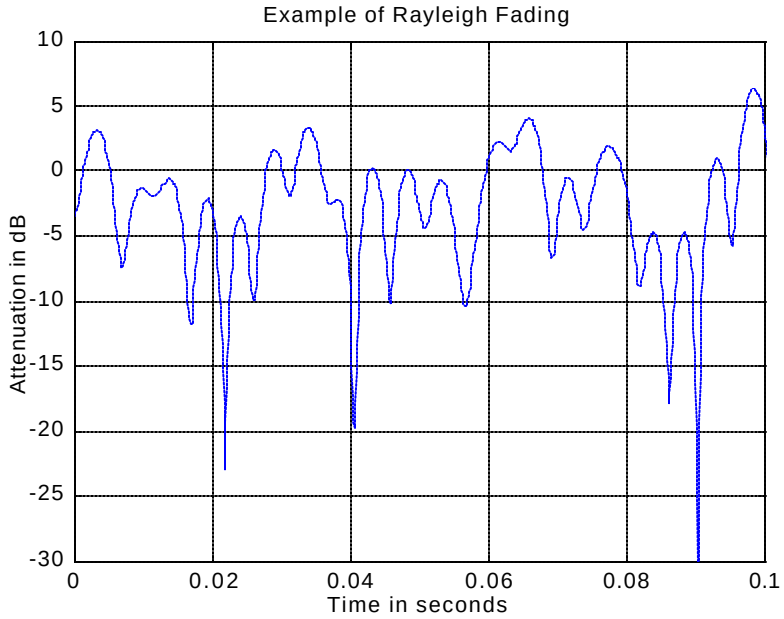

Fig. 20 Example Flat Rayleigh Fading Process

Fig. 21 and Fig. 22 quantify the effects of ignoring fading correlations upon throughput in a two-node network operating over a wireless channel. As depicted in the Fig. 21, the independent fading assumption will tend to overestimate the throughput of a network simulation employing the user data gram (UDP) transport protocol. Furthermore, a simple fixed frame error rate (FER) was matched to the independent fading case and produced similar results. A similar simulation was performed for the case of transport control protocol (TCP) and the results are presented in Fig. 22.

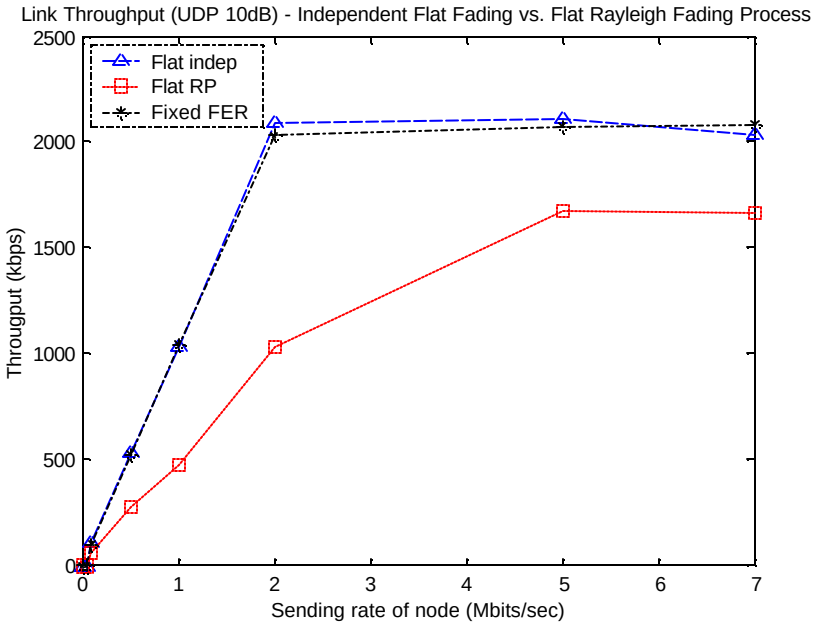

Fig. 21 UDP – Independent vs. Correlated Process Fading

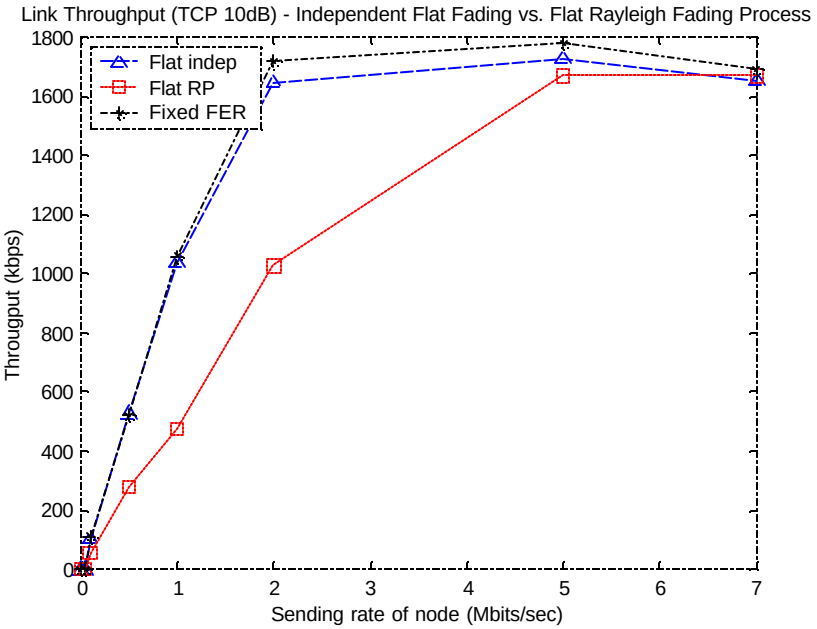

Fig. 22 TCP – Independent vs. Correlated Process Fading

The impact of improperly modeling fading as an independent process is to overestimate the network throughput, motivating the use of fading processes within network simulations. The following section will explore the proper method of implementing the two-state Markov model as a random process within network simulations.

2. Sub-Frame Analysis

Each state of the two-state Markov model represents the channel results from the transmission of a single fixed-length frame operating at a constant signal-to-noise ratio (SNR) across a specific communication scheme. The desired 'representative' Markov model will, at best, approximate the frame error process of fixed-length frames, where each state transition represents the elapse of a fixed amount of time. Network simulations involve variable frame sizes and must account for the proper temporal progression of the representative frame error process.

Another issue involving the fixed length error process is the progression of time between the transmissions of frames. Network simulations should account for idle time between transmissions, such that successive frames transmitted with a significant time gap between them should experience roughly independent fading. To account for the inequities of a fixed-length frame, we propose to utilize sub-frames. Our approach models the performance of small 'sub-frames' to define the representative two-state Markov model. Therefore, the transitions within our Markov model represent the progression of a very small amount of time. Consequently, the performance of frames

generated by network simulations can be analyzed through observing the appropriate number of sub-frames. Time gaps between frame transmissions are similarly accounted for by advancing the process an appropriate number of sub-frames. The use of sub-frames allows network simulations to appropriately utilize the random process generated by a representative two-state Markov model.

C. Statistical Analysis of the Two-State Markov Approximation

The two-state Markov channel model can be designed to match the frame error rate (FER) and average burst error length (ABEL) of a communication scheme. Both statistics have a decisive impact upon network performance, however it is not obvious that matching FER and ABEL is sufficient for describing the performance of networking protocols across a flat Rayleigh fading channel. In this section, we examine this issue of sufficiency through a comparison of frame error statistics generated first from a communication scheme operating over a flat Rayleigh fading channel and second from the representative two-state Markov model.

The wireless local area networking (WLAN) protocol 802.11b was chosen for this comparison due to the intense interest within the networking community. Although 802.11b was chosen for our simulations the same approach readily applies for other communication schemes of interest. The 802.11b physical layer communication scheme, as described by [43], is used to send fixed-length frames across a simulated flat Rayleigh fading channel. Frame successes and failures from the simulation are recorded

and analyzed. The FER and ABEL of the recorded data provide sufficient parameters, as discussed in section two, to produce the 'representative' two-state Markov channel approximation.

The first step of this analysis involves generating a representative two-state Markov model for our chosen communication scheme of 802.11b. As suggested in [42], fixed-length frames (matching the sub-frame length of 296 bits) containing random data are generated, modulated and subjected to a flat Rayleigh multiplicative fading process with additive white Gaussian noise. The Rayleigh fading process was created through the well-accepted time-domain technique described by [25], where each sample represents the fading encountered throughout the entire duration of the fixed-length frame. Received frames are demodulated using a minimum distance detector, and the results from the received frames are recorded as a frame error process.

The approximate Markov model was designed by simulating the communication scheme employing a fixed frame size of 296 bits sent across a flat Rayleigh fading link operating at 11Mbps with various SNR. The simulated fading processes were generated using Doppler frequencies of 100Hz and 10Hz. The relative fading speeds represented by the above frequency shifts (for a device operating at 2.45GHz) are ~12.24 and 1.23m/s.

Next, the frame error processes, from the above simulations, are analyzed for the frame error rate e and the average block error length ($1/r$). The transition probability p is then solved, and a two-state Markov model is formed to approximate the frame error. The representative two-state Markov model is simulated and its frame error process is

analyzed for higher order statistics. The results are compared against those of the communication scheme and are presented below in parts A and B.

1. Results of FER and ABEL

The analysis focuses upon the accuracy of the Markov model statistics for the Doppler frequencies of 100Hz and 10Hz with a signal to noise ratio of 5-20dB and a fixed-length sub-frame size of 296 bits. As designed, the Markov model almost exactly replicates the FER and ABEL of the 801.11b communication scheme.

While the Markov model channel matches some key frame error statistics, there remains doubt as to the sufficiency of matching FER and ABEL for evaluating networking protocols. For example, the transport control protocol (TCP) will seek to adjust the rate of transmission by monitoring packet loss and roundtrip travel time (RTT). Therefore, frame errors that are masked by the Medium Access Control (MAC) of the 802.11 protocol stack would not cause packet errors observable by TCP. This provides motivation for understanding the block error length statistics of any 'approximate' channel model.

2. Variance of Burst Error Length

The length of a burst error will often determine to what extent networking protocols are affected. A short duration channel burst error can often be overcome by retransmission at layer two of the OSI model. Longer duration burst errors will often results in a dropped packet causing perceptible errors in the transport layer. It is

therefore desirable to match the burst error length statistics for channel approximations. The model developed in section two is designed to exactly match the average burst error length and frame error rate for a given communication scheme, however it does not necessarily match the higher order statistics. But, is the variance in burst error length an important parameter in determining performance of networking protocols?

Consider the following hypothetical channel in which the first N-1 burst error lengths contain only a single frame error while the final burst error length contains N*9 burst errors. This hypothetical burst error channel will yield an average burst error length of ~10 frames. Given that the MAC/DLL can overcome a single frame error, this hypothetical scheme would allow TCP to achieve and sustain a very high transmission rate until the final burst error. The Markov model that matches the FER and ABEL of the hypothetical channel will contain an ABEL of ~10 frames. If the variance in burst error length of the representative two-state Markov channel is modest, burst errors will tend to contain multiple frame errors. Multiple frame errors will often trigger packet errors, thus reducing the achievable transmission rates of TCP. In this hypothetical channel scenario the Markov channel will tend to underestimate TCP performance. It is obvious that our hypothetical, and entirely unrealistic, channel would not be well approximated by the two-state Markov model; however, its relevance is simply to emphasize the need for accurately modeling the variance of burst error length.

In certain realizations of 802.11b operating in a flat Rayleigh fading channel, the variance of burst error length is quite high. When modeled by a representative two-state Markov process, we have discovered that the variance of burst error length is very

modest and thus results in a more evenly distributed burst error length. In fact for low SNR, the VBEL due to a flat Rayleigh fading channel tends to exceed the VBEL for its representative two-state Markov model. This effect is exaggerated for slow fading. Fig. 23 and Fig. 24 reveal the accuracy of the two-state Markov channel model in matching the VBEL of 802.11b operating in a flat Rayleigh fading channel for the Doppler frequencies of 100Hz and 10Hz.

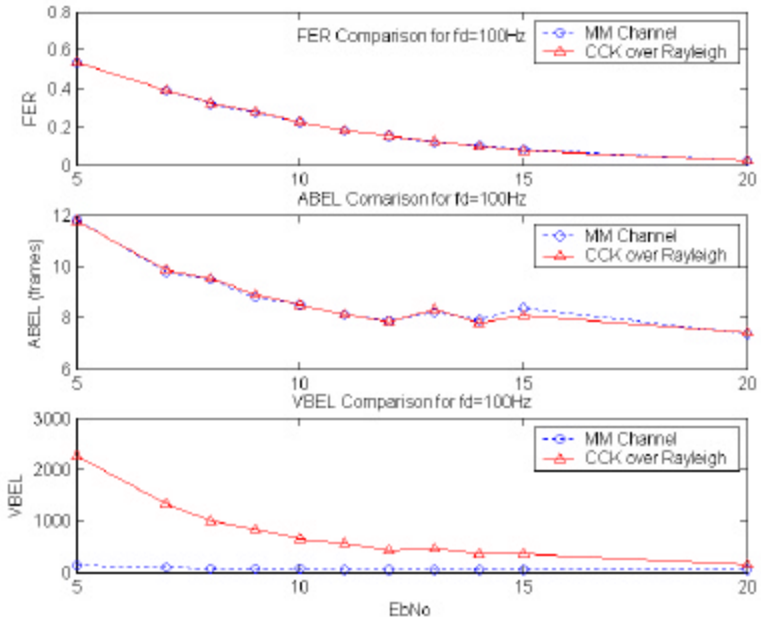

Fig. 23 Sub-Frame Error Statistics Comparison for fd=100Hz

It is obvious from the graphs above that a two-state Markov model cannot produce a frame error process that tracks the variance in burst error length of the actual frame error process. However, it is not obvious that a difference in VBEL will alter

network performance. In the next section, the impact of improperly modeling the VBEL of the frame error statistics will be explored.

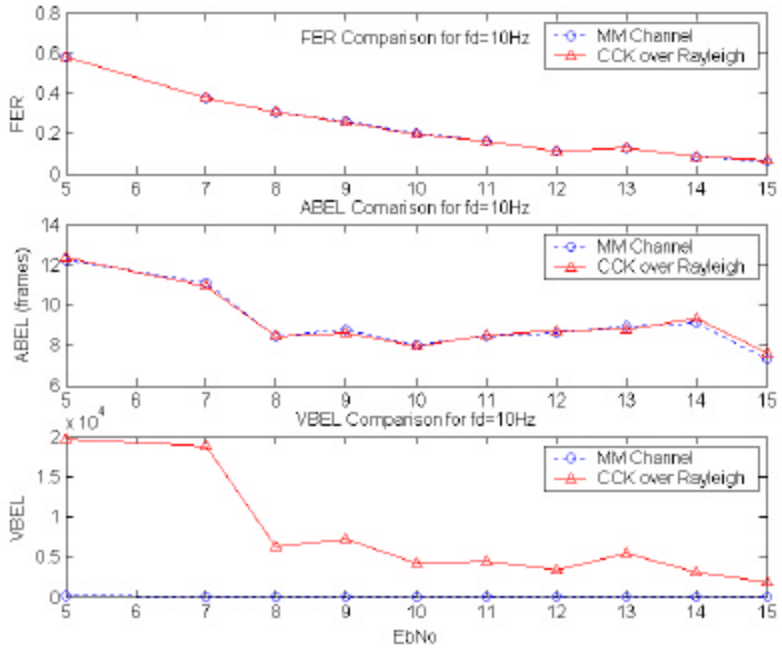

Fig. 24 Sub-Frame Error Statistics Comparison for fd=10Hz

D. Network Simulations

In an attempt to quantify the sensitivity of network simulation performance to variations in VBEL, we consider the performance of a wireless link transmitting data at various rates across a flat Rayleigh fading channel. The network simulator ns2 was altered to simulate the physical layer at a symbol level for each packet transmission. A new module was created to simulate the 802.11b baseband equivalent modulation, demodulation and transmission through a time sensitive flat Rayleigh fading process.

The scheduler clock instance relays simulation time to the module which then advances the fading process accordingly. Through this effort, we obtained a very thorough, and computationally intense simulation of 802.11b performance over a flat Rayleigh fading channel. Simulations were conducted for the Doppler frequency of 100Hz and SNR of 7 and 15dB, relating to the cases where VBEL deviated and converged for sub-frame results of section four. The simulations were conducted using the 802.11b MAC protocol and both TCP Reno and UDP transport protocols.

Additionally, a module was generated to simulate the two-state Markov model channel approximation described in section three. The Markov transition probabilities were set to match the FER and ABEL obtained in part B (of section four) for the normalized Doppler frequency and SNR stated above. The results indicate that for the case of similar VBEL, the Markov model is a reasonable approximation.

A simple two-node network is employed with one node sending application data packets of 512 bytes to the other through an 802.11b link. The wireless link is modeled using both the flat Rayleigh fading module and the representative Markov model module. The sending rate was altered between 100kbps to 5Mpbs, and each simulation was analyzed for throughput and the results are presented as throughput vs. offered load. Fig. 25 depicts performance differences for the case of SNR = 7dB and f_d=100Hz, quantifying the impact of improperly modeling the VBEL.

As shown, there is a noticeable performance difference indicating that the Markov model is an inadequate approximation for this particular scenario. The

performance differences for the case of SNR = 15dB and f_d=100Hz is presented in Fig. 26; where the VBEL was similar in the sub-frame analysis.

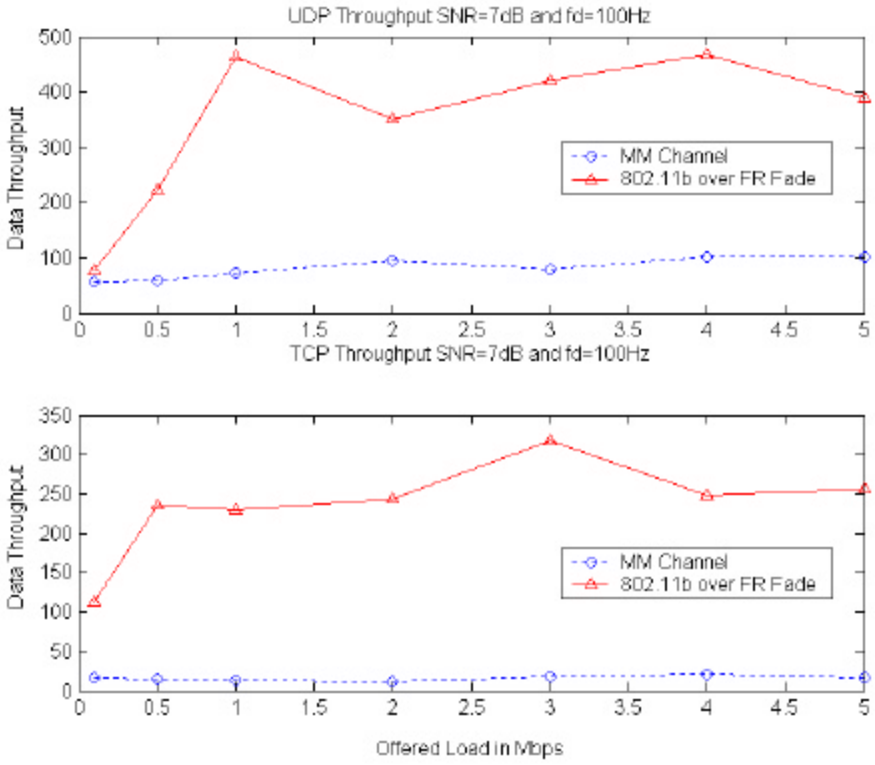

Fig. 25 Throughput Difference at SNR=7dB for Doppler fd=100Hz

The results indicate that the appropriateness of using a two-state Markov model to simulate flat Rayleigh fading is directly linked to the model's ability to adequately match the higher-order block error statistic of VBEL.

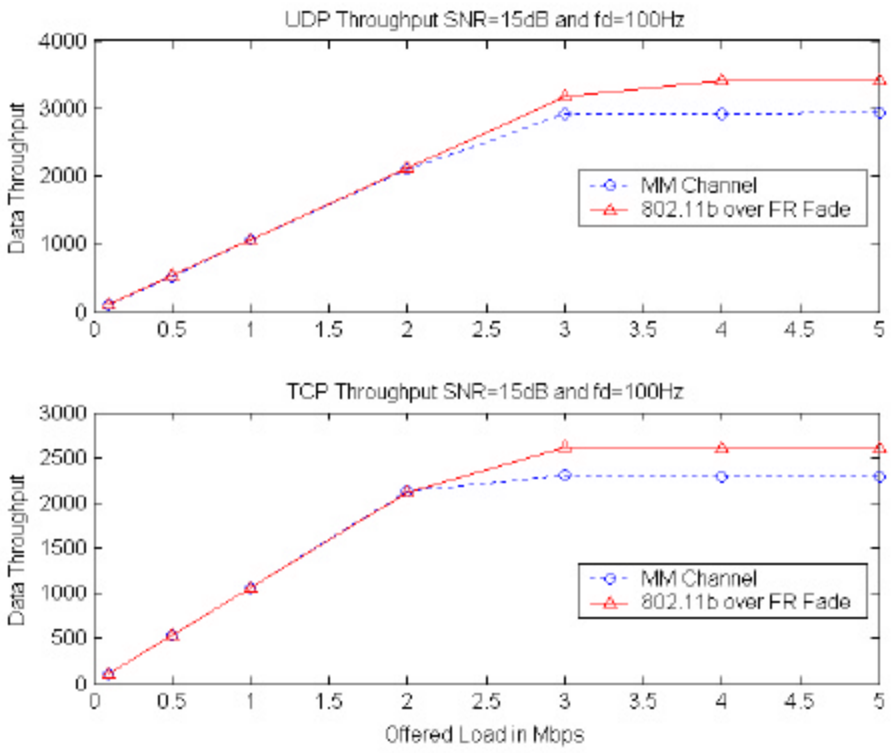

Fig. 26 Throughput Difference at SNR=15dB for Doppler fd=100Hz

E. Summary

This chapter reviewed the process of generating a representative two-state Markov channel approximation for a flat Rayleigh fading channel. The model was applied to the communication scheme of the popular 802.11b wireless LAN standard. We then analyzed the resulting block error statistics from both the communication scheme and the approximated channel model. The sufficiency of FER and ABEL for defining a communication scheme operating in flat Rayleigh fading was challenged and a new statistic was suggested as offering insight into the appropriateness of the two-state

channel model. The statistic, variance of block error length, was shown to be drastically different between the two schemes in the case of moderate SNR. Furthermore, network simulations were conducted to quantify the effects of employing a two-state Markov model for the WLAN standard 802.11b.

This chapter demonstrates that at low to moderate SNR, the Markov Model does not adequately portray the VBEL of a flat Rayleigh fading channel. Furthermore, it quantified the importance of this key block error statistic through detailed wireless network simulations. This chapter advances the understanding of the appropriate environments under which the two-state Markov Model is a suitable approximation of a flat Rayleigh fading channel and provides insight into a key block error statistic unacknowledged in previous works.

CHAPTER V

INNOVATIVE LOW COMPLEXITY CHANNEL APPROXIMATIONS

A. Overview

In this chapter, new low complexity channel models are developed to approximate a flat Rayleigh fading channel for wireless packet data networks. The new models are analyzed for their appropriateness in approximating the frame error statistics generated from a CCK modulated signal operating over a flat Rayleigh fading channel. Three new models are presented each containing multiple variations.

The first model is an extension of the current two-state Markov model to 'N' states. By including additional states with a very limited transition matrix, some higher order statistics can be matched through the use of an N-state Markov model. It is discovered that the additional computational complexity of the N-state Markov model outweighs the improvements achieved in matching frame error statistics.

The second low complexity model is a unique deviation from the traditional two-state Markov model. The 'Run Length Model' (RLM), as it's called, alternates between a fade and non-fade state, but remains in each state for a variable length of time. When the duration of a state expires, the model always transitions to the opposite state. Additional degrees of freedom are achieved in the RLM by assigning durations according to a random variable with predetermined statistics. Variations are presented in

this chapter for assigning values to the RLM parameters. Of the variations, the most accurate requires knowledge of the frame error statistics.

The final low complexity model presented is a combination of the RLM and the two-state Markov model called the 'Three State Run Length Model' (3SRLM). This model adds a transition state between the faded and non-faded states of the RLM. The 3SRLM contains a two-state Markov model for the transition state. The addition of the third state, allows the 3SRLM to match both frame error statistics and good-run length statistics. The statistical performance of this model is outstanding when compared to the previous models.

This chapter will explain the operation, parameter estimation and frame error statistical performance of: the N-state Markov model approximation, the Run Length Model channel approximation and the 3SRLM channel approximation.

B. 'N'-State Markov Model Channel Approximation

The N-State Markov model is a natural extension of the popular 2-state Markov model. Through the addition of extra states, the new N-state model is more capable of emulating higher order frame error statistics. In this model, each state represents a series of past frame realizations as well as the current frame realization. The binary output of 'good' or 'bad' frame realizations results in a very sparse state transition matrix for the N-state Markov model channel approximation. Transition probabilities for this model are set by using the sample transition matrix of a representative frame error process. The

N-state Markov model is presented in this section with example performance given for N=4, 16, 64 and 128.

1. 'N'-State Markov Model Architecture

In this model, each state represents the current frame realization as well as $\log_2(N)$-1 past frame realizations. The frame realization is a binary random variable with '0' representing a successful frame transmission and a '1' representing a frame failure in the wireless link. By increasing the number of states, this model introduces additional transition probabilities and should better model the higher order statistics encountered in a flat Rayleigh fading channel. As a means of explanation, the 4-state Markov model is presented in this section.

The 4-state Markov model's state represents the current frame realization as well as a single past frame realization as

$$state = \begin{pmatrix} current\ \ state\ realization \\ previous\ state\ realization \end{pmatrix} \tag{5.1}$$

Thus, the four possible states are

$$s1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix},\ s2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix},\ s3 = \begin{pmatrix} 0 \\ 1 \end{pmatrix},\ s4 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \tag{5.2}$$

and have a transition matrix of

$$\Pi_{4-state} = \begin{pmatrix} p_{1,1} & 0 & p_{1,3} & 0 \\ p_{2,1} & 0 & p_{2,3} & 0 \\ 0 & p_{3,2} & 0 & p_{3,4} \\ 0 & p_{4,2} & 0 & p_{4,4} \end{pmatrix}. \tag{5.3}$$

The term $p_{i,j}$ represents the probability of transitioning from state '*i*' to state '*j*'. It is obvious that some transitions are impossible, for example transitioning between states two and one in a single step. Since each state contains a single new binary frame realization, there is only two possible transitions from any state. Therefore the transition matrix will contain only 2N non-zero transition probabilities.

The N-state Markov model, as described, is completely defined by the transition matrix and thus requires the estimation of 2N parameters to define the model. One approach to estimating these parameters is to form a probability mass distribution (PMD) for state transitions from a representative error process, that is an error process generated from sending frames across a flat Rayleigh fading channel.

Once all state transitions are known, the model operates by choosing an initial value (by state probability or from a static choice) and then transitioning between states according to the state transition matrix. Each new state will contain the current value of a frame realization and thus indicate the success or failure of a frame.

2. 'N'-State Markov Model Parameter Estimation

As stated in the previous section, 2N transition probabilities are required to fully characterize the N-state Markov model, as presented in this chapter. As with the two-state Markov model, the choice of these transition probabilities is based upon a representative frame error process. Simulating the baseband modulation, transmission and detection of frames across a channel with a flat Rayleigh process can generate such a frame error process. This approach is time consuming and requires intimate knowledge

of the physical characteristics of modulation and detection, which is not necessarily prevalent in the networking community.

Once a representative frame error process is acquired, the N-state model parameter estimation will identify all the '$\log_2(N)$' frame states in the process. Each sample frame after the $(\log_2(N)-1)^{th}$ will represent a single state. Thus a frame error process of length 'M' will contain $(M - \log_2(N) +1)$ states. After the state process has been formed, the transition PMD can be formed and discrete transition probabilities assigned. This process is represented pictorially in Fig. 27 for an example of the 4-state Markov model.

Frame error process: 000111101010001110 (18 realizations)

States: -11344423232113442 (17 states)

State Transition Matrix from the above set

$$\Pi_{4-state} = \begin{pmatrix} 2/16 & 0 & 2/16 & 0 \\ 1/16 & 0 & 2/16 & 0 \\ 0 & 2/16 & 0 & 2/16 \\ 0 & 2/16 & 0 & 3/16 \end{pmatrix}$$

Fig. 27 Example PMD Calculation for a Frame Error Process

3. 'N'-State Markov Model Performance

A frame error process, generated by CCK modulated frames traversing a flat Rayleigh fading wireless link, was used to determine the PMD of state transitions for the

N-state Markov model channels with N=2, 4, 16, 64 and 128. Each N-state Markov model was then used to generate a frame error process. The models' resulting frame error processes were analyzed for frame error rate (FER), average burst error length (ABEL) and variance in burst error length (VBEL). As shown by Fig. 28 – Fig. 31, the higher order statistic of VBEL was better matched by increasing the number of states.

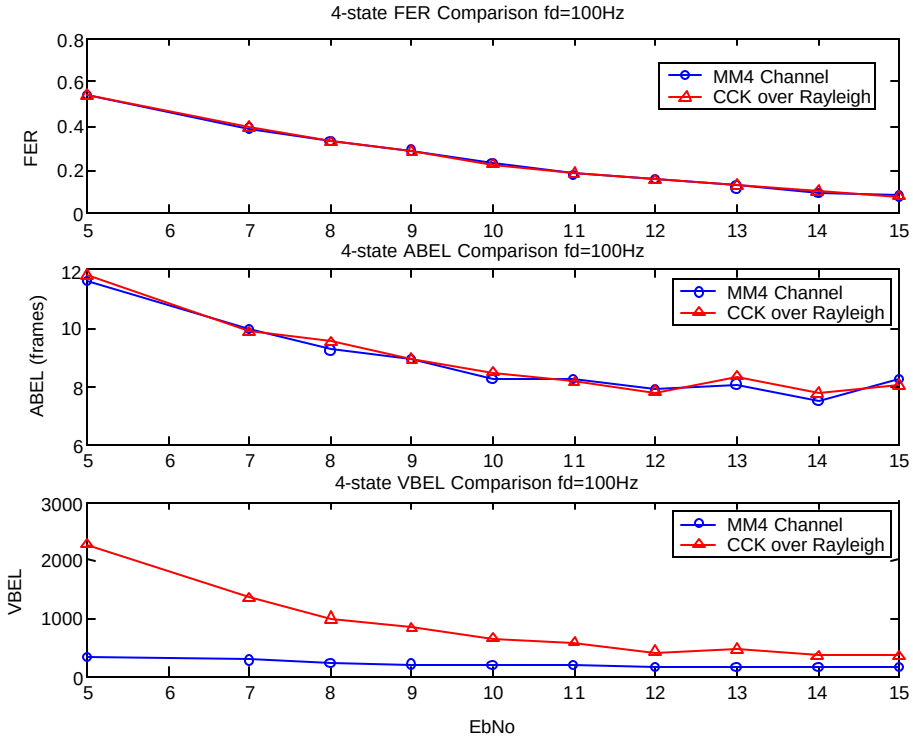

Fig. 28 Frame Error Statistic Comparison for the 4-state Markov Model

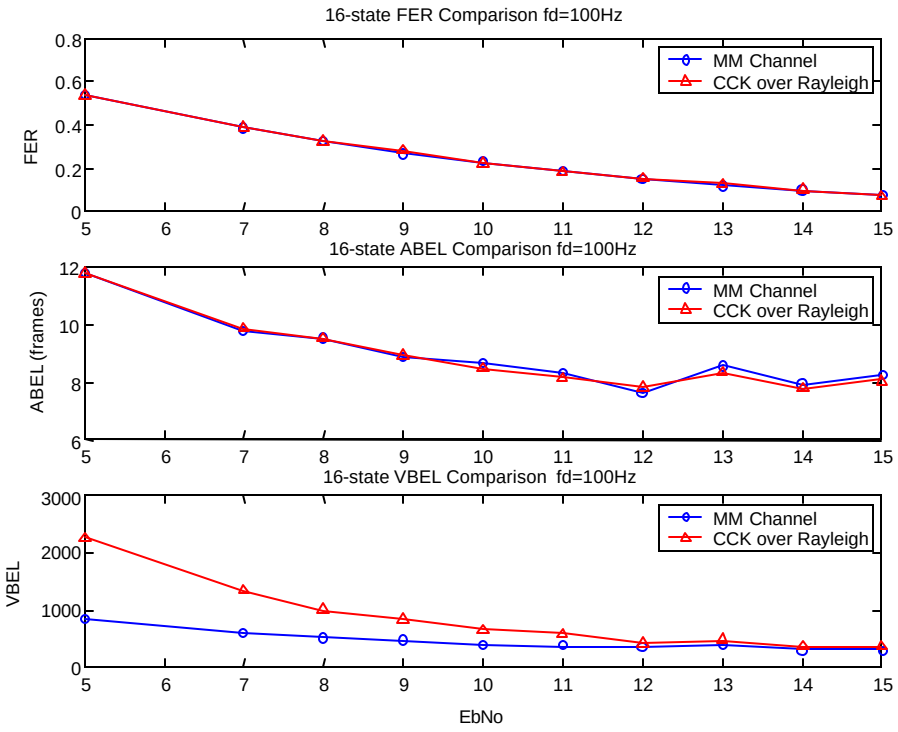

Fig. 29 Frame Error Statistic Comparison for the 16-state Markov Model

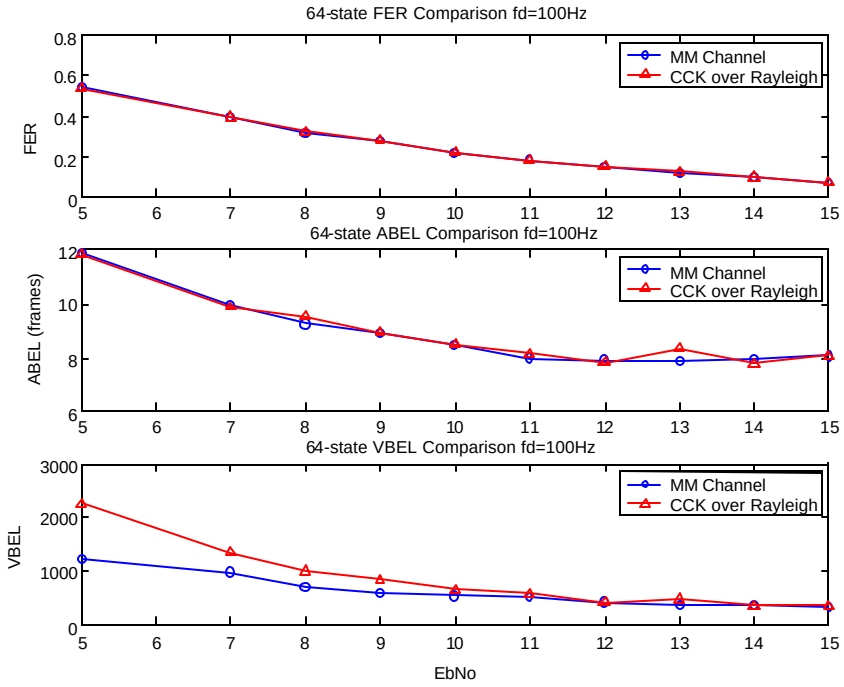

Fig. 30 Frame Error Statistic Comparison for the 64-state Markov Model

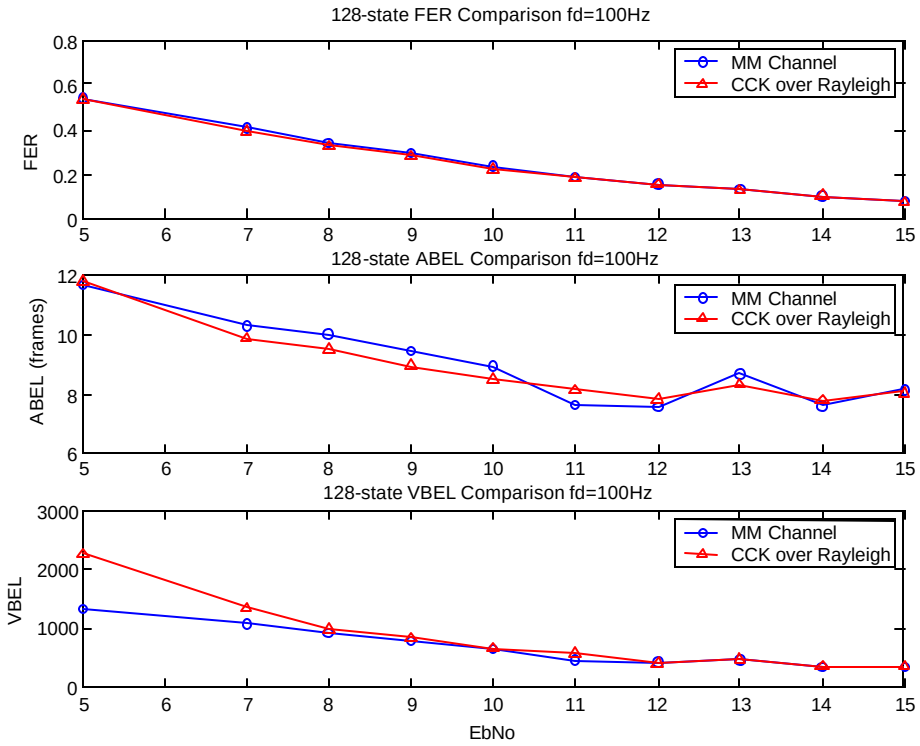

Fig. 31 Frame Error Statistic Comparison for the 128-state Markov Model

The case of the N=2-state Markov model produces frame transition probabilities of p=.9960 and q = .8823, exactly matching the values obtained when using (4.8) and (4.9) for e =.2240 and 1/(1-q)=8.4995, relating to the values of frame error rate and average FEL of the approximated CCK performance. Thus the N=2-state Markov model produces an estimate of the values derived in chapter 4 for the 2-state Markov model.

The case of the 16-state Markov model, shown in Fig. 29, has better matching of VBEL at high SNR but is incapable of matching the desired VBEL at lower SNR. The 64-state Markov model shows drastic improvement over the 2-state Markov model by matching the VBEL for SNR>~10dB as shown in Fig. 30. The 128-state Markov

model matched the VBEL for SNR>~8dB, but required significant computational complexity and long run-times to generate. This simulation, however, did not match the lower order statistics as accurately as the 64-state model as shown in Fig. 31.

Increasing the number of states in the Markov model allows the low complexity N-state Markov model channel approximation to better match higher order statistic of VBEL than the traditional two-state Markov model. While the VBEL is an indicator of matching higher order statistics, a more insightful comparison can be made through examining the probability mass distributions (PMD) of frame error duration generated by the N-state Markov model and those of CCK modulated frames transmitted over a flat Rayleigh channel. As shown in Fig. 31, the N=128-state Markov model provides a close approximation of the frame error statistics of FER, ABEL and VBEL, however it remains a Markov model and will follow an exponential distribution for frame error durations greater than 'N'.

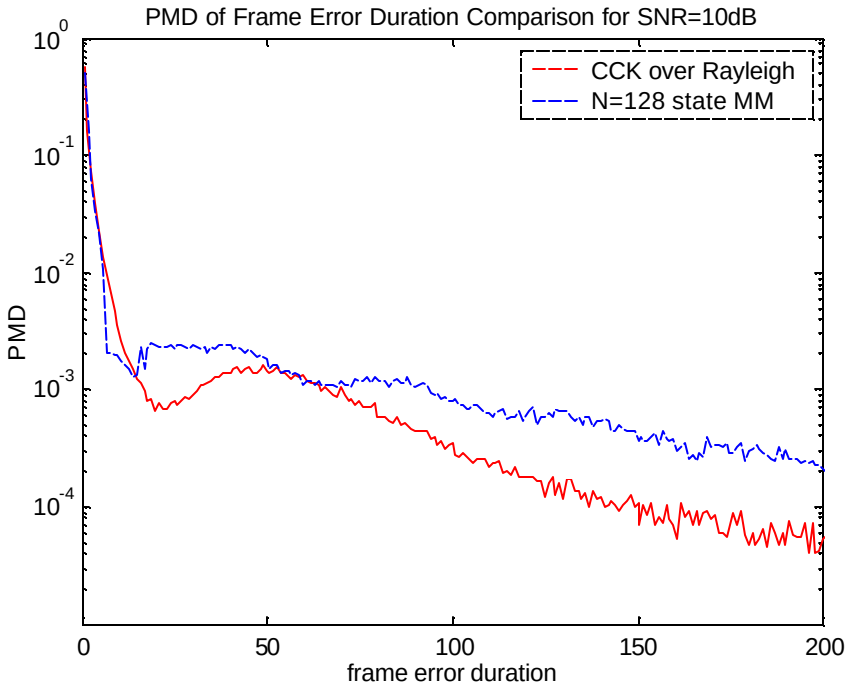

Fig. 32 Comparison of Frame Error Duration PMD for CCK vs. N=128-state MM

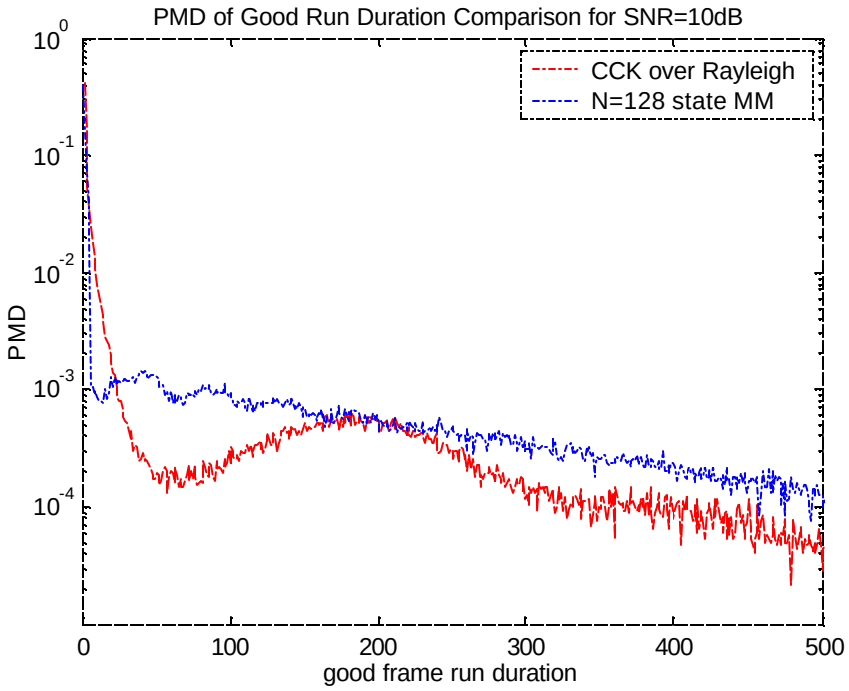

Fig. 33 Comparison of Good Frame Run Duration PMD for CCK vs. N=128-state MM

The PMD shown in Fig. 32 clearly indicates the N=128 state Markov model distribution contains an excessively heavy tail. The PMDs in Fig. 32 were formed with over 450,000 frame error durations equating to ~3.8 million frames for a SNR=10dB. Fig. 33 shows the comparison of good frame run duration or the amount of frames received in succession without error. Both probability mass distributions were formed from over 250,000 good frame runs. Even at 128 states, the N-state Markov model channel approximation does not accurately represent the PMD of frame error duration or good frame run duration. Other drawbacks of the N-state Markov model include:

- Ever increasing memory/processor utilization with larger values of 'N', as the transition matrix grows by N^2.
- The method describes in this chapter does not provide a method for choosing an appropriate value of N.
- After the N^{th} state, the model must contain an exponential distribution for frame error duration, which may or may not match the PMD of a generic frame error process.

Due to these drawbacks and the uncertainty of choosing the correct value of 'N' for an arbitrary scheme, the N-state Markov model channel approximation is not considered a substantial replacement for the two-state Markov model's inadequacies. One very attractive characteristic of the N-state Markov model is its generic structure. The network performance of the N=128 state Markov model channel approximation is included in Chapter 6.

C. Run-Length Model (RLM)

Frames traversing a wireless link are received either with or without errors. The popular two-state Markov model assumes that frame errors only occur when the channel is in the faded state. This assumption does not hold true for low signal-to-noise ratios, where errors frequently occur in the un-faded state. Another issue with the two-state Markov model is found in the lack of variability in the duration of fades. In fact, the duration of fade lengths is an exponential distribution for the two-state Markov model. Rice [55] developed a method for approximating the distribution of fading time intervals for a Rayleigh process. The conditional PDF that a Rayleigh process *R(t)* crosses a certain level r for the first time in the interval (t_1+t, t_1+t+dt) with positive slope, given a crossing downward through r at time $t=t_1$ is represented as $f(t\,;\,r)$. Rice's approximation to this density was later articulated in [56] (for the case of $r<<1$) as

$$f(t;r)=\frac{2\pi u^2 e^{-u}}{T_R(r)}\left[I_0(u)-\left(1+\frac{1}{u}\right)I_1(u)\right]. \tag{5.4}$$

Where $T_R(r)$ is the average time duration a Rayleigh process remains below a threshold r, and $I_n(\cdot)$ denotes the nth-order modified Bessel function of the first kind and $u=2[T_R(r)/t]$. Here, the average fade duration can be calculated [26] from the threshold r and Doppler frequency f_d as

$$T_R(r)=\frac{e^{r^2}-1}{\sqrt{2\pi}\, r f_d}. \tag{5.5}$$

It is informative to compare the approximation (5.4) against a sample PMD of $f(t\ ;\ r)$. As shown in Fig. 34, the analytically derived approximation contains a significantly heaver tail than the sample PMD.

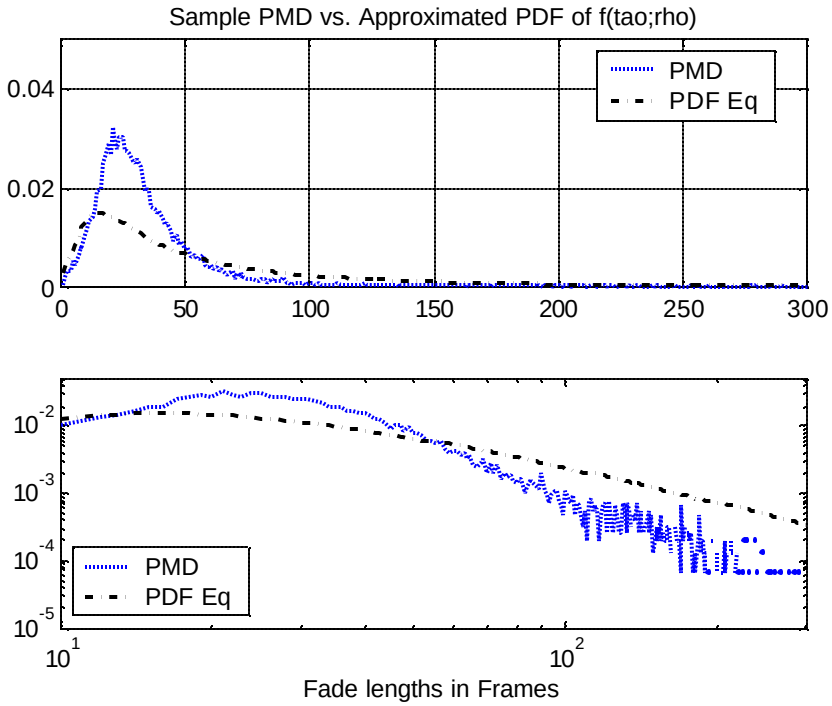

Fig. 34 PMD of $f(t\ ;\ r)$ vs. the Approximation (5.4)

The disparity in fade-length distributions between the approximated PDF (5.4) and the sample PMD motivates the use of sample statistics in future models. The drawbacks associated with the N-state Markov model channel approximation and the disparity in (5.4) motivate the creation of a new channel model that can more accurately predict the higher order statistics of the fading channel using sample distributions. One such model is the run-length model.

1. Run-Length Model Architecture

The run-length model overcomes the previously stated deficiencies of the N-state Markov model by allowing frame errors to occur in the non-faded state, and by attempting to match some of the statistical parameters of the fade-length distribution for a Rayleigh fading process. This model, Fig. 35, makes a fundamental departure from the Markov model by removing all variability from state transitions and instead varying the time duration spent in states. An arbitrary threshold r is chosen such that below this magnitude, a Rayleigh process is said to be in a faded state and above this threshold it is said to be in a non-faded state. With this definition, the Rayleigh fading process will transition between the two states with probability one, meaning at the conclusion of a non-fade state the Rayleigh process will enter into a fade state with probability one and vice versa.

Fig. 35 RLM State Diagram and Transition Probabilities

While in the non-fade state, the run-length model will allow independent frame errors to occur with a probability of *fer*. The fade state will produce frame errors with a probability of one. The duration of each state will have a probability density function to

match some of the attributes of a Rayleigh process. Although simple in theory, the run-length model adds significant variability unachievable through the two-state Markov model. The RLM, as shown in this chapter, can be directly tied to specific physical layer properties such as modulation, coding and receiver techniques.

2. Run-Length Model Parameter Estimation and Performance with Respect to Physical Layer Properties

Parameter estimation is a challenge for the run-length model, as many different values are required to specify the operation of this model. The first, and most obvious, parameter is the non-fade probability of frame error, *fer*. Next, the duration of both the non-fade and fade states must be defined to approximate, in some sense, the duration of state lengths observed on an actual Rayleigh process. This variation of the RLM sets parameters according to the modulation, coding, demodulation and frame size employed across the wireless channel. One goal of parameter estimation for the run-length model is to define a method of choosing parameters given only the performance of the physical layer and coding scheme in AWGN, and the Doppler frequency of the flat Rayleigh fading channel.

One early attempt at parameter estimation sets the *fer* and threshold r according to the physical layer bit-error rate (ber) and the frame length L_f in bits. This approach offers a method to directly calculate the *fer* as

$$fer = 1 - (1 - ber)^{L_f}. \tag{5.6}$$

Once the *fer* is known, the threshold defining the faded and non-faded states r must be calculated. This early approach set a confidence metric *P(succ | f)* to measure the probability of a faded frame being received correctly. Calculating r from the confidence metric requires knowledge of the physical layer performance across an AWGN channel. Assuming this invertible relationship is known and defined as

$$P_e = z\left(SNR\right), \tag{5.7}$$

one can form the threshold r with the assumption that the Rayleigh process has a unit energy through:

$$ber = 1 - \left(P\left(succ \mid f\right)\right)^{1/L_f}, \tag{5.8}$$

$$SNR_r = z^{-1}\left(ber\right), \tag{5.9}$$

$$r = SNR_r\left(SNR\right)^{-1}, \tag{5.10}$$

where the term *SNR* is the non-faded signal-to-noise ratio of the system.

By defining these two parameters, one can simulate a Rayleigh process and observe the probability mass distribution (PMD) for the duration of fade lengths and the duration of non-fade lengths. This is accomplished through simulating a Rayleigh process, where the faded state occurs whenever the magnitude of the fading process drops below the threshold r. Once the PMD of state durations is known, random variables must be constructed to approximate them. Since the analytically derived distribution of fade lengths (5.4) contained excessively heavy tails (Fig. 34), an alterative approach was devised. Well-known distributions were 'fitted' to a sample PMD (of fade lengths for a Rayleigh process) by matching the mean and variance of fade lengths. The

gamma distribution fit well as shown in Fig. 36 and Fig. 37 for the cases of SNR of 15dB and 10dB respectively.

The density of run-lengths is approximated as a gamma distribution defined by (5.11) – (5.14). In order to approximate the PMD of a Rayleigh process' run-lengths, one should first define the faded state *f* and non-faded *nf* state's mean and variance as:

$$m_f = \text{mean of faded-state duration}, \tag{5.11}$$

$$m_{nf} = \text{mean of non-faded-state duration}, \tag{5.12}$$

$$s_f = \text{variance of faded-state duration}, \tag{5.13}$$

$$s_{nf} = \text{variance of non-faded-state duration}, \tag{5.14}$$

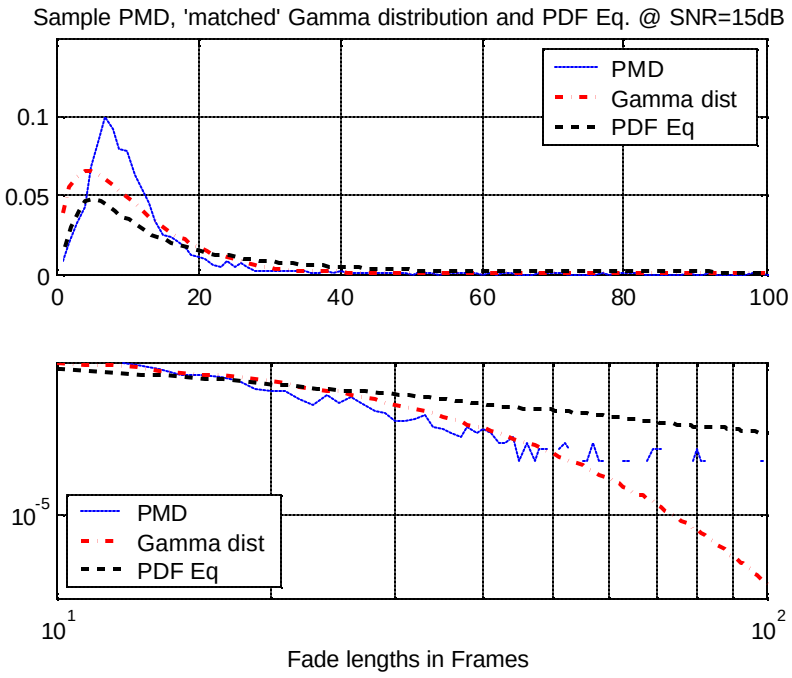

Fig. 36 PMD of *f(t ; r)*, Matched Gamma Distribution and the Approximation (5.4) for the case of SNR=15dB

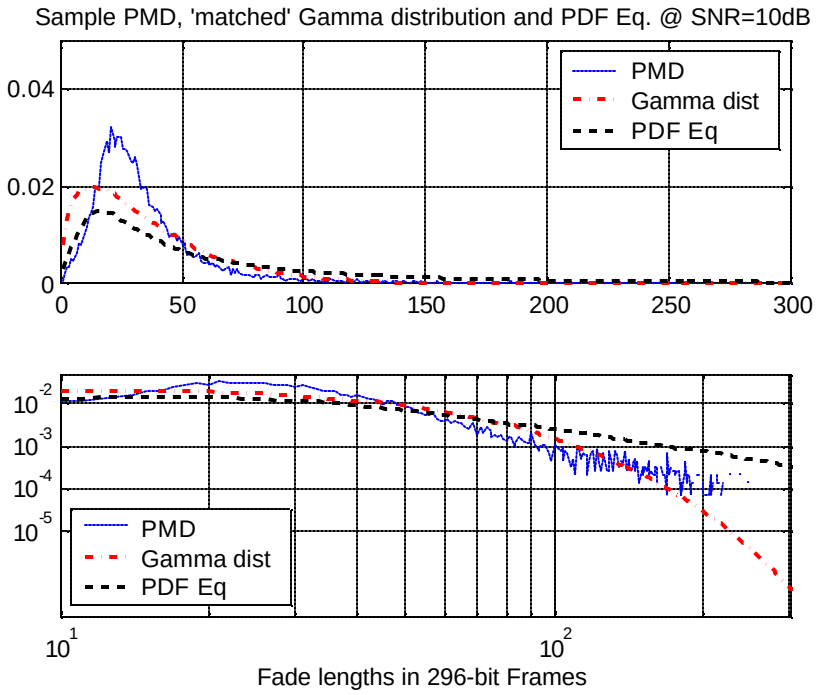

Fig. 37 PMD of $f(t\ ;\ r)$, Matched Gamma Distribution and the Approximation (5.4) for the case of SNR=10dB

The gamma distribution parameters l and a are completely defined with knowledge of the distribution's mean and variance.

a. RLM Parameter Estimation Using Nominal SNR and Confidence Metric

The parameter estimation method presented in this section operates by using a confidence metric and the CCK modulation format to define a threshold below which a Rayleigh fading process is considered to be in a fade. The threshold is used to experimentally set the RLM parameters of fer, m_f, m_{nf}, s_f and s_{nf}.

The fist step in this method is to define a confidence metric according to the probability of successfully transmitting a frame given the model is in the faded state or *P(succ|f)*. This technique is illustrated by means of and example. For the case of CCK modulation in the presence of flat Rayleigh fading, set the confidence metric to *P(succ|f)*=.001 for the case of SNR = [5..20dB] given the sub-frame size of 296 bits. For each value of SNR, the following procedure is employed to achieve *fer* and r:

$$ber = 1 - (.001)^{1/296} = .02276, \tag{5.15}$$

$$ber = z_{cck}\left(\frac{E_b}{N_o}\right) \cong 24Q\left(\sqrt{4\frac{E_b}{N_o}}\right), \tag{5.16}$$

$$SNR_{threshold} = \frac{1}{4}\left(Q^{-1}\left(\frac{ber}{12}\right)\right)^2 = 2.4056, \tag{5.17}$$

$$r = 2.4056(SNR)^{-1}. \tag{5.18}$$

The value r is presented in the Fig. 38 for values of SNR = [5-20dB].

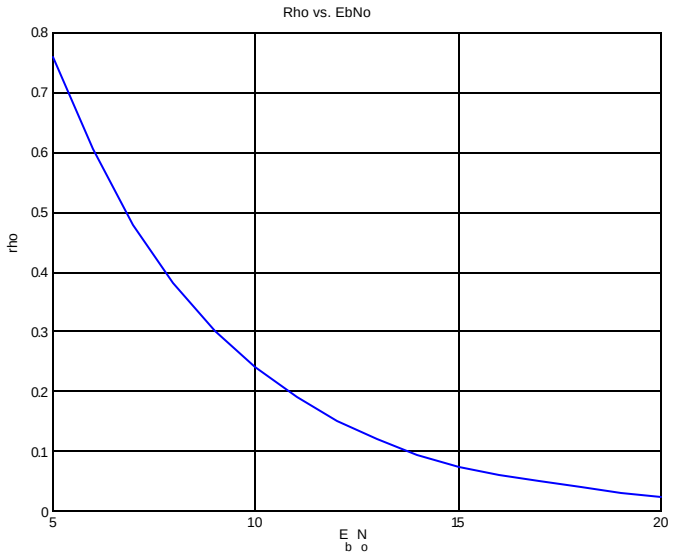

Fig. 38 Threshold Value r vs. SNR

In this method of parameter estimation, *fer* is calculated directly from the modulation scheme operating over an AWGN channel at the nominal SNR. For the case of CCK, the *fer* can be calculated directly from *ber*, approximated in (2.22), as

$$ber = z_{cck}\left(\frac{E_b}{N_o}\right) = 24Q\left(\sqrt{4\frac{E_b}{N_o}}\right) \text{ for } \frac{E_b}{N_o} = [5,6...20]dB, \tag{5.19}$$

$$fer = 1-(1-ber)^{296}. \tag{5.20}$$

The values of *fer* are presented in Fig. 39 for this example.

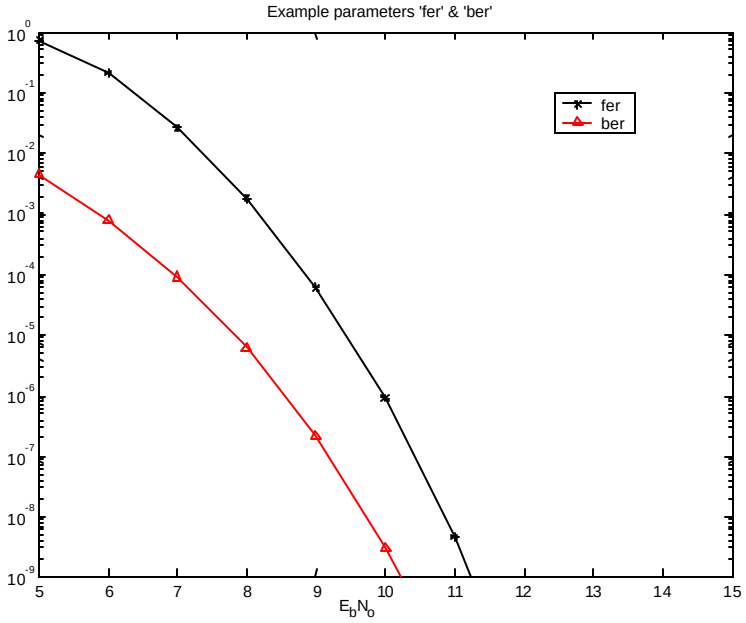

Fig. 39 Approximated Values of *fer* and *ber* for CCK in AWGN with 296bits/frame

The next step in parameter estimation for this variant is to determine the mean and variance of both the faded and non-faded states. Again the Rayleigh fading process is simulated and analyzed for the duration spent above (non-fade) and below (fade) the

threshold r. The mean and variance of the durations spent in both the fade and non-fade states are computed and used to form the Gamma distribution parameters of l and a as

$$l_x = \frac{m_x}{s_x} \qquad x = \{f, nf\}, \tag{5.21}$$

$$a_x = \frac{m_x^2}{s_x} \qquad x = \{f, nf\}. \tag{5.22}$$

The low complexity RLMv1, with parameters set as previously described, was used to generate a frame error process that is compared against the original CCK over flat Rayleigh fading frame error process in Fig. 40. The two frame error processes are compared with respect to frame error rate (FER), average burst error length (ABEL) and variance of burst error length (VBEL). The performance of a representative two-state Markov model's frame error process is included for completeness. As can be seen in Fig. 40, the observed frame error statistics (FES), ABEL and VBEL, of the RLMv1 are an extremely poor representation of the actual CCK FES. The PMD of frame error durations and the PMD of good frame run durations for the RLMv1 are compared against those of the approximated CCK over Rayleigh channel in Fig. 41. The comparison of PMD shown in Fig. 41 indicates that while the RLMv1 can track the slope of the frame error duration it cannot accurately represent the PMD of good frame run duration.

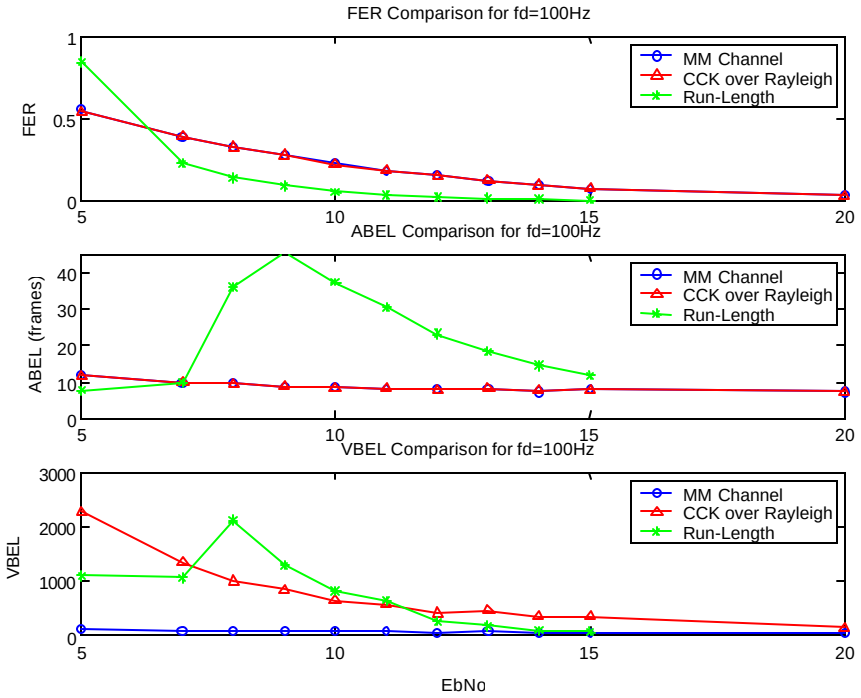

Fig. 40 Comparison of RLMv1, CCK and MM Frame Error Statistics

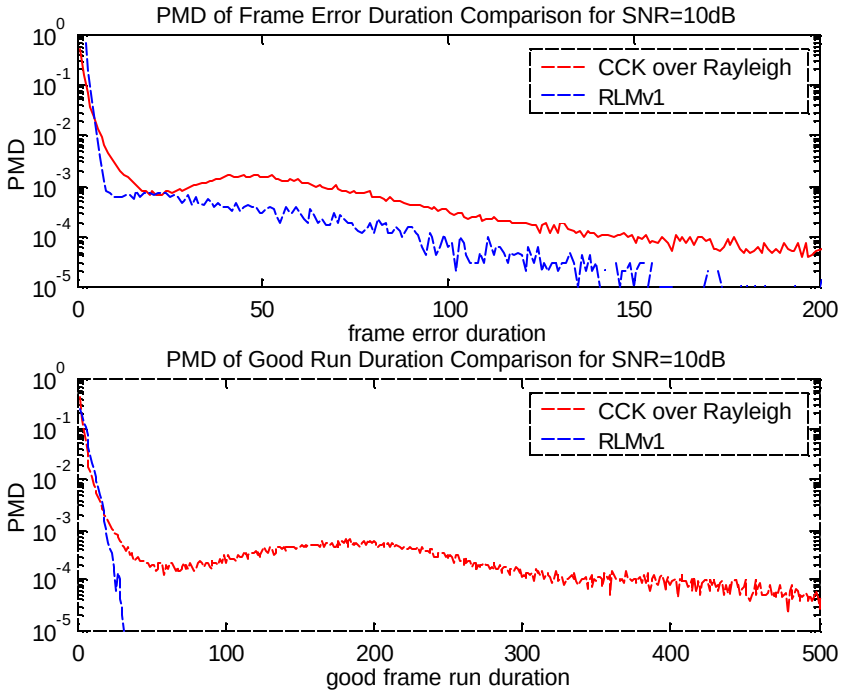

Fig. 41 PMD Comparison of RLMv1 and CCK over Rayleigh for SNR=10dB

The inaccuracies of the RLMv1 parameter estimation can be contributed in part to the *fer* parameter. In the case of CCK modulation over a flat Rayleigh fading channel, the frame error rate while the Rayleigh process was above r is dominated by the occurrences of low SNR at or directly above the threshold r. The Rayleigh process will not be constant in the non-faded case and thus the model must account for the variability of the fading process when determining the *fer* parameter. In the next version of parameter estimation for the RLM, *fer* is set through simulation of the physical layer in the non-faded state while operating across a flat Rayleigh fading channel.

b. RLM Parameter Estimation Using Confidence Metric and Simulated *fer*

This version of parameter estimation, RLMv2, operates exactly as before with the exception of the *fer* parameter. In RLMv2, the parameter *fer* is set to the sample mean of the frame error rate for CCK operating across a flat Rayleigh fading channel, for the sub-set of samples that occur while the Rayleigh process is greater than the threshold r. The resulting value is an approximation of the actual frame error rate for the non-fade state. Applying the new *fer* parameters along with the r's and gamma distributions from RLMv1, one can form a new run-length model.

The low complexity RLMv2, with parameters set as previously described in version two, was used to generate a frame error process that is compared against the original CCK over flat Rayleigh fading frame error process in Fig. 42. The two frame error processes are compared with respect to frame error rate (FER), average burst error length (ABEL) and variance of burst error length (VBEL). The performance of a

representative two-state Markov model's frame error process is included for comparison. In addition to the observed metrics of FER, ABEL and VBEL, the PMD of frame error duration and the PMD of good frame run duration is presented in Fig. 43 for the RLMv2 and the approximated CCK over Rayleigh.

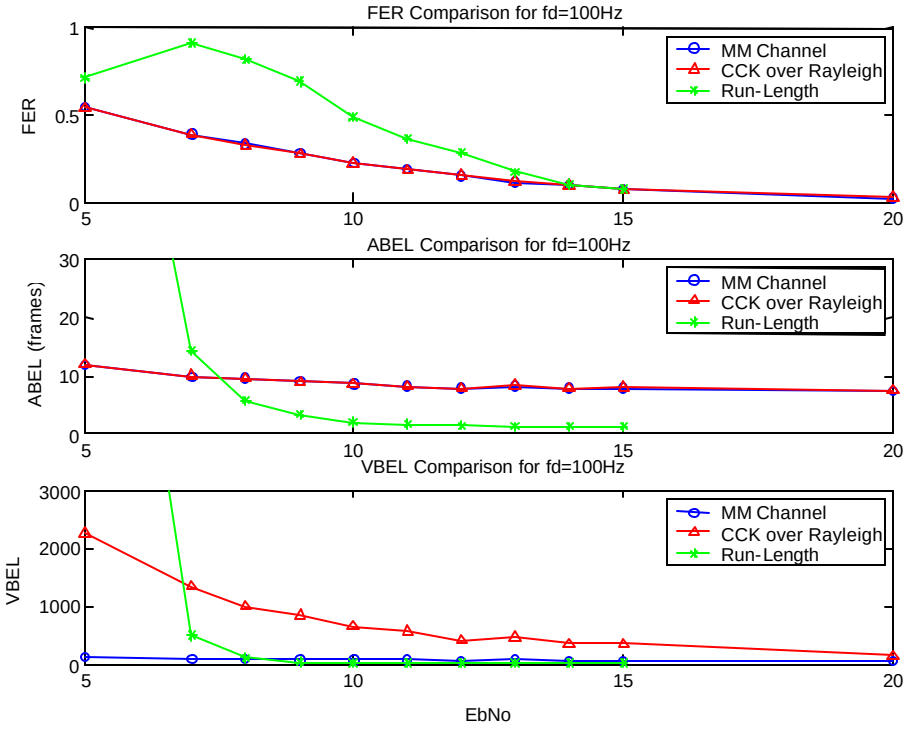

Fig. 42 Comparison of RLMv2, CCK and MM Frame Error Statistics

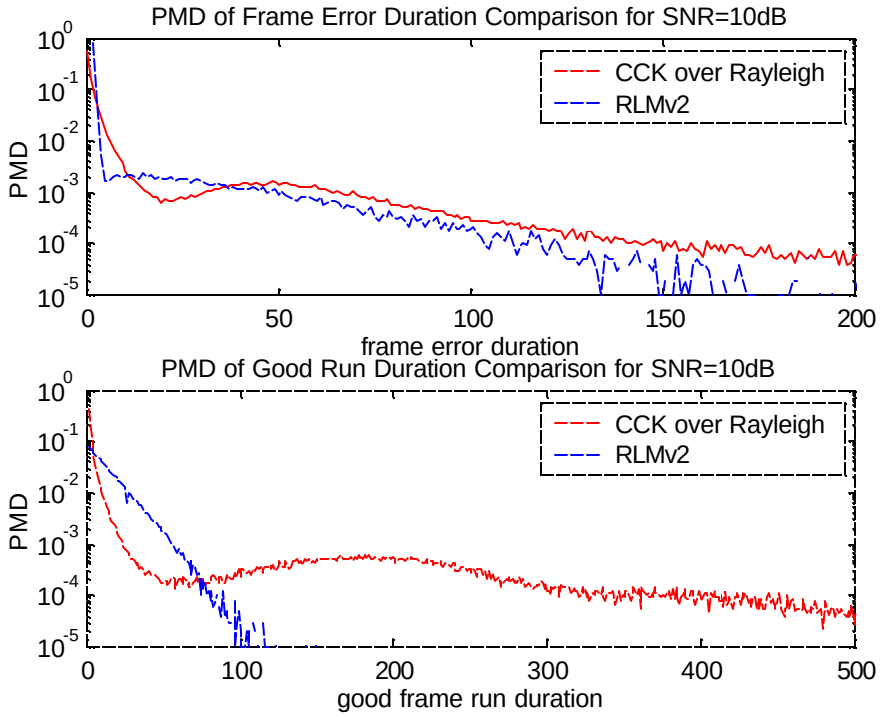

Fig. 43 PMD Comparison of RLMv2 and CCK over Rayleigh for SNR=10dB

Simulation results (Fig. 42) show that by altering the parameter of *fer*, the run-length model produces better estimations of model frame error rate (FER) at a cost of poorer performance in ABEL and VBEL. The PMD results of Fig. 43 indicate the RLMv2 is able to better approximate the frame error durations and the good frame run durations. These results indicate that although the 2-state Markov model is incapable of matching higher order frame error statistics, it might be preferable to the RLM. This comparison is a bit misleading, because the Markov model requires knowledge of the frame error process statistics of FER and ABEL. If we allow the RLM access to these parameters, the model can more accurately match the statistics of a frame error process.

3. RLM Parameter Estimation and Performance with Respect to Frame-Error Process Statistics

The run-length model is analyzed for frame error rate (FER) and average burst error length (ABEL) in this section. Knowledge of these relationships allows us to estimate the model parameters, resulting in greater accuracy in matching a frame error process's statistics. In this scenario, five unknown parameters describing the run-length model will be estimated with knowledge of the desired FER and ABEL. Some run-length parameters require the assumption that the frame error process was generated by a packet data network operating over a flat Rayleigh fading channel. The analysis begins with the derivation of FEL and ABEL for the RLM.

The RLM's FEL is the overall average of frame errors and is comprised of the rate of errors in each state, scaled by the time spent in each state as

$$FER = \Pr(fe) = \Pr(fe \mid nf)\Pr(nf) + \Pr(fe \mid f)\Pr(f). \tag{5.23}$$

The term *fe* is frame error, *nf* indicates the non-faded state and *f* indicates the faded state. The amount of time spent in both the faded and non-faded states can be determined from the duration random variables as

$$\Pr(f) = \frac{m_f}{m_{nf} + m_f}, \tag{5.24}$$

$$\Pr(nf) = \frac{m_{nf}}{m_{nf} + m_f}. \tag{5.25}$$

Given that frame errors occur in the *nf* state with probability *fer*, and in the *f* state with a probability of one, the FER can be restated as

$$FER = \frac{fer(m_{nf}) + m_f}{m_{nf} + m_f} = \frac{(fer)R + 1}{R + 1}, \tag{5.26}$$

where the term R is the ratio of m_{nf} to m_f.

The ABEL can be restated as the average duration of a burst error. In our analysis, all errors are considered 'bursts' and thus ABEL is a measure of the average duration of an 'error', where a single 'error' is defined as a group of consecutive, bad frames. The ABEL can be solved as,

$$ABEL = \frac{\left(\frac{1}{1-fer}\right)\left(\frac{fer}{fer+1}\right)m_{nf}\left(1-fer^2\right)}{\left(\frac{fer}{fer+1}\right)m_{nf}\left(1-fer^2\right)+1} + m_f \frac{1}{\left(\frac{fer}{fer+1}\right)m_f\left(1-fer^2\right)+1}. \tag{5.27}$$

Appendix A contains the derivation of (5.27).

This section has developed equations for the ABEL (5.27) and FER (5.26) for the run-length model with respect to three of the five model parameters: m_f, m_{nf} and fer. An additional relationship can be formed between some of the parameters with an assumption that the frame error process was generated through a flat Rayleigh fading process. Some common statistics of a flat Rayleigh fading process include average fade duration $T_R(r)$ and average number of fades per second $N_R(r)$ [26]. The average time spent in the fade and non-fade states can be related using $N_R(r)$ and the time period of a single frame as

$$m_{nf} = \frac{1}{N_R(r)} - m_f. \tag{5.28}$$

where

$$N_R(r) = \sqrt{2p}\, f_d\, r \exp\left(-r^2\right) \tag{5.29}$$

defines $N_R(r)$ and f_d is the Doppler shift of the fading process.

The following iterative method was successfully employed to solve this system of equations. First, an arbitrary value of r is used to produce the average fade duration m_f according to $T_R(r)$, where it is assumed that the Doppler frequency f_d of the fading channel is known. Next, the FER and ABEL equations are combined to form

$$fer^2\left[-ABEL\right] + fer\left[\left(ABEL-1\right) + \frac{m_f - ABEL}{m_f\left(1-AFER\right)}\right] - \frac{AFER\left(m_f - ABEL\right)}{m_f\left(1-AFER\right)} = 0. \tag{5.30}$$

Since ABEL and FER are provided and m_f has been calculated from an arbitrary threshold, this results in a simple quadratic equation for *fer*. Finally, the metric used in this iterative approach is

$$metric = \left|m_{nf}\left(fer\right) - m_{nf}\left(r\right)\right| = \left|\left(\frac{1-AFER}{AFER-fer}\, m_f\right) - \left(\frac{1}{N_R(r)} - m_f\right)\right|, \tag{5.31}$$

or the difference between the values of m_{nf} produced from the *fer* calculation and m_{nf} produced from the threshold equation. Many iterative approaches are possible, however a straightforward iteration of r about the operational SNR was performed. Following this optimization routine, the RLM parameters of m_f, m_{nf} and *fer* are computed, leaving only the variances of the fade s_f and non-fade s_{nf} states to be computed.

The variances were set according to a sample variance of 'X', a function of a Rayleigh process. For the case of s_f, the function X_f is defined as

$$X_f = \text{duration of } \left|\text{Rayleigh process}\right| < r, \tag{5.32}$$

for the time period beginning at a negative r-crossing and ending at a positive r-crossing. For the case of s_{nf}, the function X_{nf} is defined as

$$X_{nf} = \text{duration of } \left|\text{Rayleigh process}\right| > r, \tag{5.33}$$

for the time period beginning at a positive r-crossing and ending at a negative r-crossing. And the RLM parameters are assigned according to

$$s_r = \text{sample variance } (X_r) \quad \text{for r} = \{f, nf\}. \tag{5.34}$$

This version of the RLM, with parameters set though matching frame error statistics, was used to generate a frame error process that is compared against the original CCK over flat Rayleigh fading frame error process in Fig. 44. The two frame error processes are compared with respect to frame error rate (FER), average burst error length (ABEL) and variance of burst error length (VBEL). The performance of a representative two-state Markov model's frame error process is included for completeness.

Fig. 44 presents a fair comparison of the RLM and the two-state Markov model (MM) as both models were created with the knowledge of the CCK's frame error statistics of FER and ABEL. As shown in Fig. 44, both the RLM and the two-state MM are capable of matching the given statistics of FER and ABEL, however only the RLM is able to represent the higher order statistic of Variance in Burst Error Length (VBEL). No justification has been presented to the use of VBEL as a metric for comparing channel approximations, however VBEL is a good indicator of a model's ability to match the PMD of frame error durations. In fact, it will be shown that not only is VBEL important, but also the variance in good run lengths (VGRL), or the variance in the

duration of successfully transmitted frames. The frame error processes generated by the RLM were analyzed for the average good run length (AGRL) and the VGRL and the results are presented in Fig. 45.

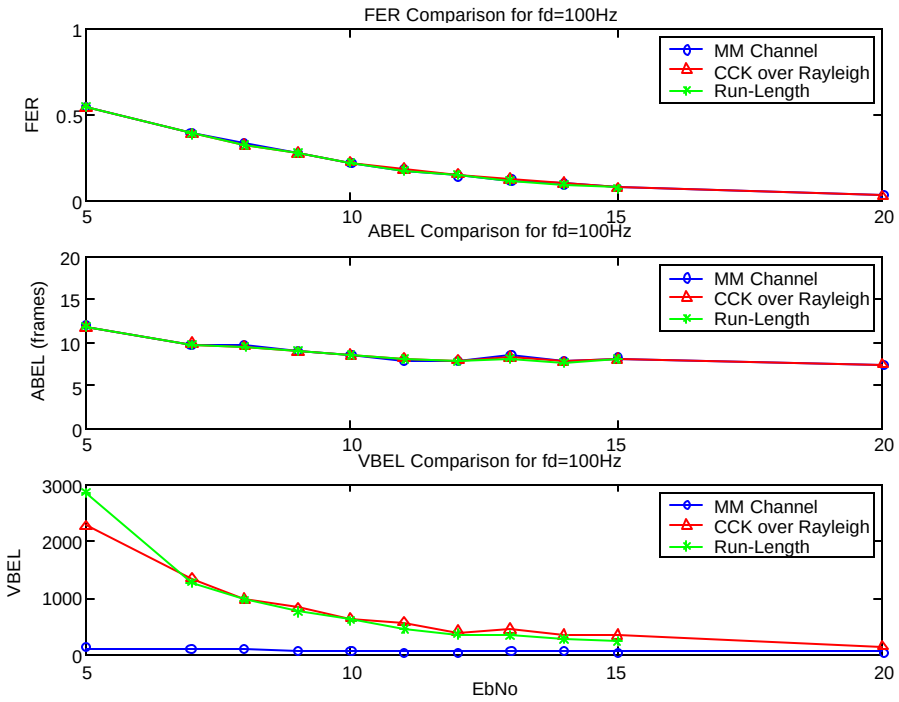

Fig. 44 Comparison of RLM, CCK and MM Frame Error Statistics

The PMD of frame error duration and the PMD of good frame run duration of the RLM is presented in Fig. 46 for the case of SNR=10dB. By matching the statistics of FER, ABEL and VBEL the RLM sacrifices the ability to match frame error durations of moderate length in order to better match the longer duration frame errors.

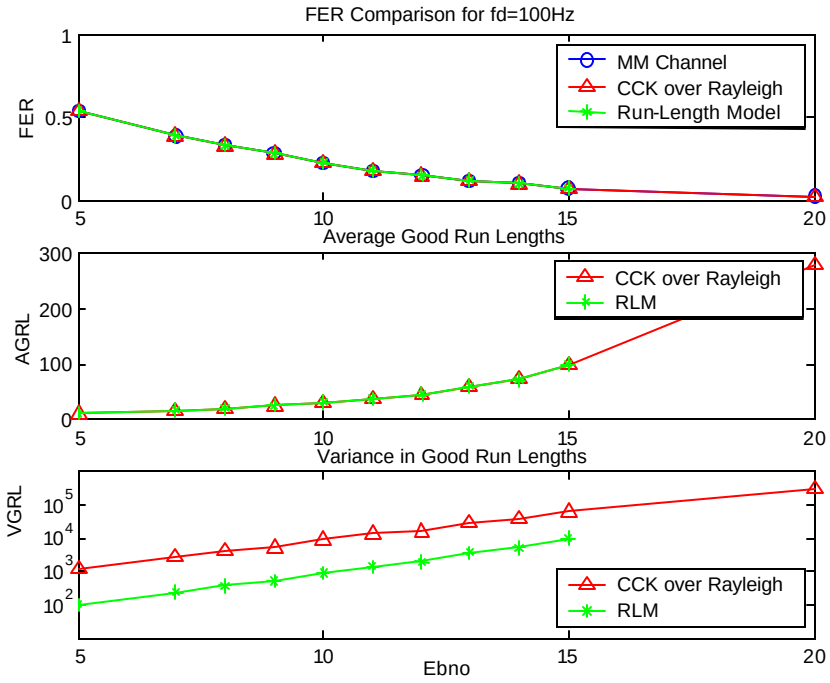

Fig. 45 Comparison of RLM, CCK and MM Good Run Length Statistics

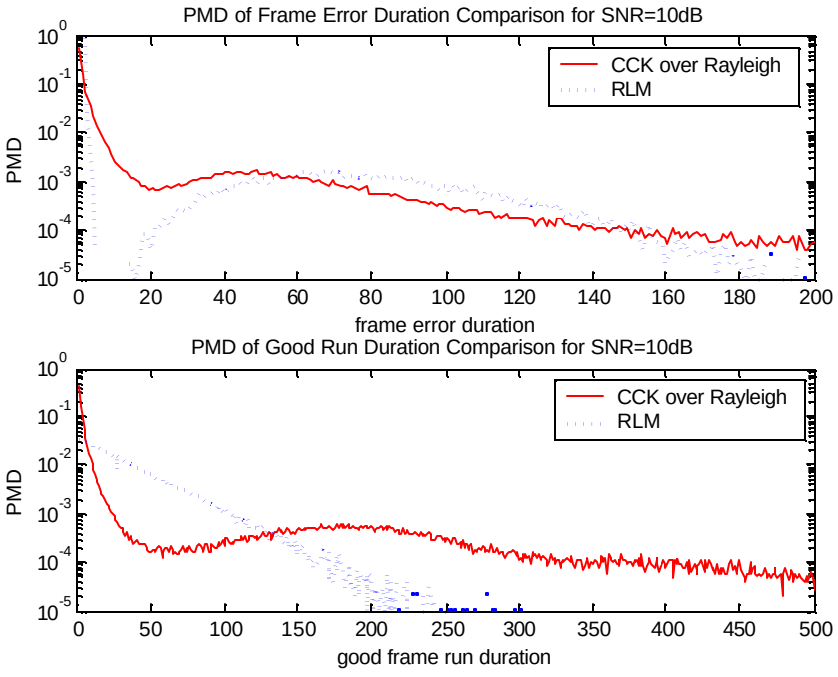

Fig. 46 PMD Comparison of RLMv2 and CCK over Rayleigh for SNR=10dB

By employing a static probability of frame error in the non-fade case, the RLM contains an exponential distribution for good frame run lengths. Unfortunately, the good frame fun length PMD being approximated in Fig. 46 have a much heaver tail than the resulting RLM's PMD. The inability of the RLM to match the good frame run length statistics led to the development of a new model, one with sufficient variability to approximate both the PMD of good frame run duration and the PMD of frame error duration. The new model should attempt to match at a minimum the frame statistics of FER, ABEL, VBEL, AGRL and VGRL. One such model is presented in the next section.

D. Three-State Run-Length Model (3SRLM)

The three-state run-length model (3SRLM) is a natural extension of the RLM to include a transition state between the non-fade and fade states of the RLM. Through the addition of a third state, the 3SRLM contains sufficient variability to attempt matching the frame performance statistics of FER, ABEL, VBEL and VGRL. Many approaches were attempted to define the operation of the transition state with the two most successful being a fixed frame error rate, and a two-state Markov model. It was found that the fixed frame error rate approach could be solved to exactly match the desired frame statistics but in some cases it required unacceptable values for the parameters of m_f (5.11), m_{nf}, (5.12) or m_t,

$$m_t = \text{mean of transitio n - state duration .} \tag{5.35}$$

By assigning the transition state to be a two-state Markov model, the 3SRLM was capable of matching all the desired frame statistics while using acceptable values for all model parameters. The general architecture for the 3SRLM was presented in chapter 3, but is repeated here for the reader's convenience. Following the architecture is an explanation for how the parameters of the 3SRLM are set and the resulting frame statistics.

1. Architecture of the 3SRLM

The three-state run-length model (3SRLM) is an extension of the RLM, with a new 'transition' state. The transition state better models the fading process by allowing the representation of a degraded SNR that is not necessarily a deep fade. For tractability, the transition state is modeled by a two-state Markov model. The operation of the 3SRLM-PHY is a simple combination of the MM-PHY and the RLM-PHY. The logical operation of the 3SRLM-PHY is shown in Fig. 3.8. The model begins in the 'non-fade' state and remains for a duration of sub-frames prescribed by a realization of the random variable *NF*. All sub-frames transmitted during the 'non-fade' state are unaltered and no errors occur. Upon expiration of the 'non-fade' state, the 3SRLM-PHY moves into the 'transition' state and remains for a duration of sub-frames according to the random variable *T*. The 'transition' state is composed of two 'sub-states' called 'good t' and 'bad t', where the relationship between the sub-states is defined by a two-state Markov model. The initial sub-state is chosen randomly according to a desired transition state probability of error (T_{fer}). Each instance of a sub-frame will cause the two-state Markov

model to attempt a transition that might result in a sub-state transition between ‘good t’ and ‘bad t’. The integrity of each sub-frame in the transition state is found exactly as in the MM-PHY with the ‘bad t’ sub-state resulting in a sub-frame error with probability one and the ‘good t’ sub-state resulting in no error with probability one. Upon expiration of the transition state, the 3SRLM enters the ‘fade’ state and remains for a duration of sub-frames according to the random variable *F*. All sub-frames transmitted during the ‘fade’ state result in errors. Upon expiration of the fade state, the 3SRLM returns to the non-fade state and the cycle continues. The ordered transitions between the three states are deterministic to reduce the complexity in matching frame statistics. The transition from the fade state bypasses the transition state to increase the duration of the transition state following the non-fade state. This is desireable to justify assumptions made in parameter estimation, mainly in setting the expected frame error duration for the transition state.

2. 3SRLM Parameter Estimation and Performance with Respect to Frame Error Statistics

In this section, the three-state run-length model is analyzed for frame error rate (FER), average burst error length (ABEL), average good run length (AGRL), variance of burst error length (VBEL) and variance of good run length (VGRL). Knowledge of these relationships allows the model parameters to be determined given a representative frame error process to approximate. After the explanation of the parameter estimation,

the 3SRLM is formed to approximate CCK operating over a flat Rayleigh fading channel and the resulting frame statistics are presented.

The 3SRLM's FEL is the overall average of frame errors, which can be computed by summing the probability of an error in a state scaled by the time spent in each state as

$$FER = \Pr(fe) = \Pr(fe|nf)\Pr(nf) + \Pr(fe|t)\Pr(t) + \Pr(fe|f)\Pr(f), \tag{5.36}$$

where t represents the transition state. Note that for this discussion, the term 'error' refers to a group of frame errors or a burst error. All terms except t are as defined in (5.23). The probability of each state can be determined be examining the average time spent in each state as

$$\Pr(r) = \frac{m_r}{m_{nf} + m_t + m_f} \qquad r = \{m_{nf}, m_t, m_f\}. \tag{5.37}$$

Given that frame errors always occur in the f state, never occur in the nf state and occur with a probability of fer in the t state, the FER can be restated as

$$FER = \frac{m_f + fer(m_t)}{m_{nf} + m_t + m_f}. \tag{5.38}$$

The ABEL of the 3SRLM can be approximated as the sum of the expected error lengths in a cycle divided by the expected number of errors in a cycle, or

$$ABEL = \frac{\sum E[BEL]}{E[\#\,BEL]} \quad per\ cycle, \tag{5.39}$$

where an error length is represented in the equations as BEL. The fade state will only have a single error while the non-fade state will contain no errors. If one makes the

definition of $E[BEL|t]$ for the expected error length in the t state, then ABEL can be restated as

$$ABEL = \frac{m_f + (ENRL - 1)E(BEL|t)}{ENRL}, \tag{5.40}$$

where $ENRL$ stands for the expected number of errors in a single cycle.

As can be expected, the average number of errors in a cycle should equal the average number of good runs in a cycle. This is evident as each good run is followed by an error and vice versa. Following this logic and the assignment of $E[GRL|t]$ to be the expected good run length in state t, the average good run length (AGRL) of the 3SRLM can be formed as

$$AGRL = \frac{m_{nf} + (ENRL - 1)E[GRL|t]}{ENRL}. \tag{5.41}$$

The term $ENRL$ can be solved by making the observation that all errors, except for one, occur in the transition state as

$$ENRL = E[\#\, BEL|t] + 1. \tag{5.42}$$

The expected number of errors in t can be computed from the ratio of expected number of frame errors in t to the expected error length in t, as

$$E[\#\, BEL|t] = \frac{E[\#\ frame\ errors|t]}{E[E_L|t]} = \frac{m_t(fer)}{E[E_L|t]}. \tag{5.43}$$

If the 3SRLM uses the same parameter names as shown in Fig. 3.5 for the two-state Markov model, it can be shown that the resulting frame error rate of the model is

$$fer = \frac{1-p}{2-p-q}. \tag{5.44}$$

The denominator of (5.43) is a Geometric random variable indicating the number of independent Bernoulli trials until a successful departure from the 'bad t' state. Using this information (5.42) can be restated as

$$ENRL = \frac{m_t(1-q)(1-p)}{2-p-q}. \tag{5.45}$$

Now the mean of the transition state can be solved with respect to m_f by reworking (5.38) into

$$m_{nf} = \frac{m_f(1-FER)}{FER} + \frac{m_t(fer-FER)}{FER}, \tag{5.46}$$

and (5.41) into

$$m_{nf} = ENRL\left(AGRL - \frac{1}{1-p}\right) + \frac{1}{1-p}, \tag{5.47}$$

then substituting (5.45) into (5.47) to produce

$$m_t = \frac{\left(AGRL - \frac{m_f(1-FER)}{FER}\right)}{\left(\frac{fer-FER}{FER}\right) - \left(\frac{(1-p)(1-q)}{(2-p-q)}\left(AGRL - \frac{1}{1-p}\right)\right)}. \tag{5.48}$$

If one assumes that m_f, p and q can be approximated, then m_{nf} can be solved by (5.46), *fer* by (5.44) leaving only to solve for s_f and s_{nf}.

The variance of the non-fade and fade states can be found through analysis of the VBEL and VGRL of the 3SRLM. VGRL can be stated as

$$VGRL = E\left[(GRL - E[GRL])^2\right] = E\left[GRL^2\right] - AGRL^2, \tag{5.50}$$

where *E[GRL]* is the AGRL and *GRL* is simply good run length. If one assumes the good run lengths due to transitions between states are negligible, then the 3SRLM distribution of GRLs is simply a weighted sum of the GRL distributions of the non-fade and transition states as

$$f_{GRL}(g) = \frac{E[\#GRL|t]\Pr(t)}{E[\#GRL]} f_{GRL\_t}(g) + \frac{E[\#GRL|nf]\Pr(nf)}{E[\#GRL]} f_{GRL\_nf}(g). \quad (5.51)$$

Since the expected value of GRL^2 can be written as

$$E[GRL^2] = \int_{-\infty}^{\infty} GRL^2 f_{GRL}(g)dg, \quad (5.52)$$

one can substitute (5.51) for $f_{GRL}(g)$ and simplify to form

$$E[GRL^2] = \frac{1}{ENRL}\left[(ENRL-1)\int_{-\infty}^{\infty} GRL^2 f_{GRL\_t}(g) + 1\int_{-\infty}^{\infty} GRL^2 f_{GRL\_nf}(g)\right], \quad (5.53)$$

and further simplified into the form

$$E[GRL^2] = \frac{1}{ENRL}\left((ENRL-1)E[GRL\_t^2] + E[GRL\_nf^2]\right). \quad (5.54)$$

The term $E[GRL\_nf^2]$ can be formed similar to (5.50) and altered to form

$$E[GRL\_nf^2] = \text{var}(GRL\_nf) + E[GRL\_nf]^2 = s_{nf} + AGRL^2. \quad (5.55)$$

Since *GRL_nf* is a gamma random variable, the variance is simply the ratio of a_{nf} and l_{nf}^2 as

$$E[GRL\_nf^2] = \frac{a_{nf}}{l_{nf}^2} + m_{nf}^2. \quad (5.56)$$

For ease of computation and lack of a realistic metric, the random variable determining the duration of the transition state was removed and instead a constant duration of m_t is

used for the duration of the transition state. Since the transition state is a two-state Markov model, the value of GRL is a geometric random variable with a mean of

$$E[GRL\_t] = \frac{1}{1-p}, \tag{5.57}$$

and variance of

$$\mathrm{var}(GRL\_t) = \frac{p}{(1-p)^2}. \tag{5.58}$$

Following the logic of (5.55), one can form

$$E[GRL\_t^2] = \frac{p}{(1-p)^2} + \left(\frac{1}{1-p}\right)^2 = \frac{1+p}{(1-p)^2}, \tag{5.59}$$

for the expected value of GRL_t^2. Combining (5.54), (5.56) and (5.59) forms

$$E[GRL^2] = \left(1 - \frac{1}{ENRL}\right)\left(\frac{1}{1-p}\right)^2 (1+p) + \frac{1}{ENRL}\left(\frac{m_{nf}^2}{a_{nf}} + m_{nf}^2\right), \tag{5.60}$$

that can be solved for a_{nf} using (5.50) as follows:

$$a_{nf} = \left[\left[\left(VGRL - AGRL^2\right) - \left(1 - \frac{1}{ENRL}\right)\left(\frac{1}{1-p}\right)^2 (1+p)\right]\frac{ENRL}{m_{nf}^2} - 1\right]^{-1}. \tag{5.61}$$

Due to the symmetry of the two-state Markov model definition and the use of a gamma function for the fade state, the derivation of a_f follows the same logic as a_{nf}, resulting in

$$a_f = \left[\left[\left(VBEL - ABEL^2\right) - \left(1 - \frac{1}{ENRL}\right)\left(\frac{1}{1-q}\right)^2 (1+q)\right]\frac{ENRL}{m_f^2} - 1\right]^{-1}. \tag{5.62}$$

The remaining term l of a gamma distribution X,

$$f_X(x) = \frac{\lambda(\lambda x)^{\alpha-1} e^{-\lambda x}}{\Gamma(\alpha)} \quad \text{for } x > 0 \text{ and } \alpha > 0, \lambda > 0, \tag{5.63}$$

is solved by manipulating the mean

$$E[X] = \frac{\alpha_x}{\lambda_x}, \tag{5.64}$$

and variance

$$\text{var}[X] = \frac{\alpha_x}{\lambda_x^2}, \tag{5.65}$$

to arrive at the following solution:

$$\lambda_x = \frac{\alpha_x}{E[X]} = \frac{\alpha_x}{m_x} \quad \text{for } x = \{f, nf\}. \tag{5.66}$$

At this point, the 3SRLM parameters of m_t, m_{nf}, fer, α_{nf}, α_f, λ_{nf} and λ_f have all been defined with respect to m_f, p and q. The average duration spent below a threshold for a Rayleigh process is well known and is presented in (5.5). This equation is used to approximate the value of m_f, using the threshold r as defined in (5.18). With the value of m_f approximated, the only values remaining are p and q. It turns out that multiple values of p and q will result in acceptable values for 3SRLM parameters. Due to the previous assumption that state transitions can be ignored when forming the expected value of run lengths, it is desirable to make m_t as long as possible. A three-dimensional graph of m_t, p and q is presented in Fig. 47 for the case of SNR=5dB, Fig. 48 for the case of SNR=10dB and in Fig. 49 for the case of SNR=15dB. In each case there exists values of p and q for which the value of m_t is negative or close to zero. It was discovered through experimentation that the for the case of $p = 0.8$ and $0.3 < q < 0.8$ the value of m_t

was acceptable across the range of SNR={5 – 12dB}. For SNR > 12dB, the value of m_t becomes smaller than $E[GRL|t]$ and the equations governing parameter estimation change dramatically.

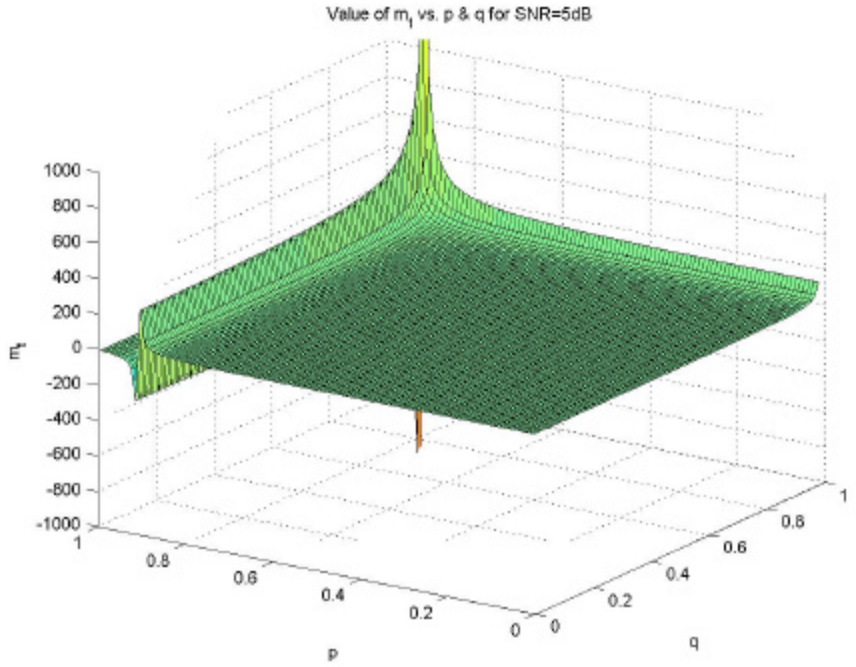

Fig. 47 Values of m_t vs. p and q for SNR=5dB

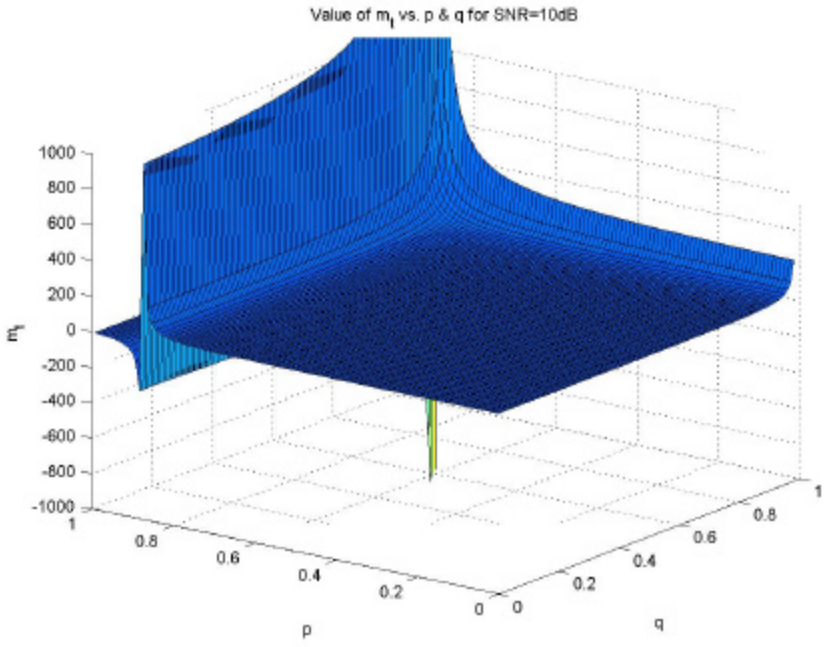

Fig. 48 Values of m_t vs. p and q for SNR=10dB

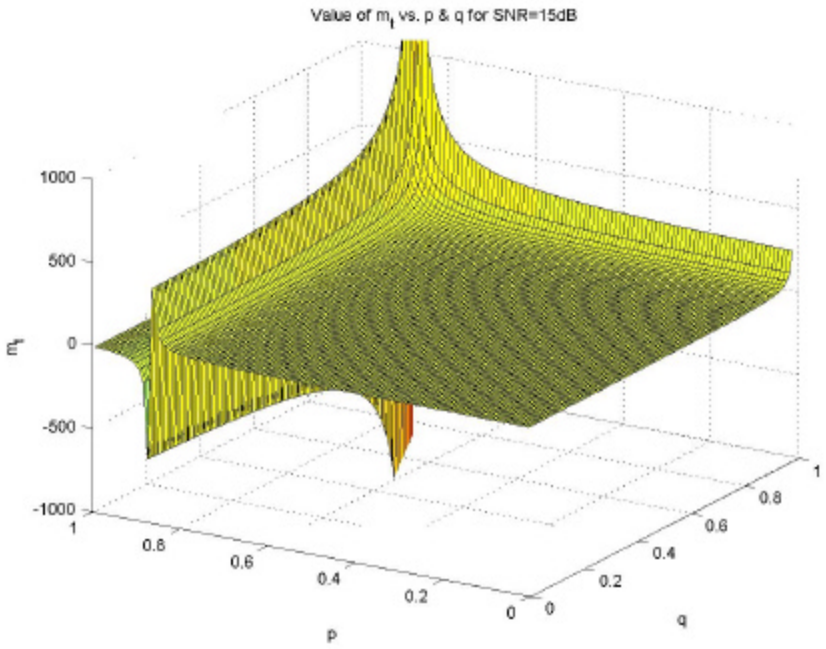

Fig. 49 Values of m_t vs. p and q for SNR=15dB

For the special case of $m_t < \max\{E[GRL|t], E[BEL|t]\}$, parameter estimation must be altered. Since the expected value of the run length (frame errors or good runs) is longer than the expected value of the transition period for this case, it reasonable to assume that all the frames in the transition state (on average) will either be good or in error. Under this assumption, the transition region is meaningless as the good frames are simply a continuation of the non-fade state and a series of errors are a continuation of the fade state. It is therefore reasonable under this special case to assume m_t=0. This assumption leads to a two state model with all good run statistics equal to the non-fade state statistics and all the frame error statistics equal to the fade state statistics. So, for the special case of $m_t < \max\{E[GRL|t], E[BEL|t]\}$, the 3SRLM estimates parameters as:

$$m_{nf\_sc} = AGRL, \tag{5.67}$$

$$m_{f\_sc} = ABEL, \tag{5.68}$$

$$m_{t\_sc} = 0, \tag{5.69}$$

$$a_{nf\_sc} = \frac{AGRL^2}{VGRL}, \tag{5.70}$$

$$l_{nf\_sc} = \frac{AGRL}{VGRL}, \tag{5.71}$$

$$a_{f\_sc} = \frac{ABEL^2}{VBEL}, \tag{5.72}$$

$$l_{f\_sc} = \frac{ABEL}{VBEL}. \tag{5.73}$$

Employing a special case for parameter estimation has very little impact upon the end users as parameter estimation will be preformed infrequently and published for network researchers.

By allowing variability in the values p and q, the 3SRLM has the possibility of matching future frame statistics (at low SNR) that are found to be important. For the simulations presented in this work, $p = 0.8$ and $q = 0.6$.

The 3SRLM was used to generate a frame error process that is compared against the original CCK over flat Rayleigh fading frame error process in Fig. 50 and Fig. 51. The two frame error processes are compared with respect to FER, ABEL and VBEL in Fig. 50 and FER, AGRL and VGRL in Fig. 51. The performance of a representative two-state Markov model's frame statistics is included for completeness in Fig. 50. One can observe the exact matching of ABEL, VBEL, AGRL and VGRL when the 3SRLM is in the special case. Both Fig. 50 and Fig. 51 present the 3SRLM frame statistics when a perfect gamma random number generator is used.

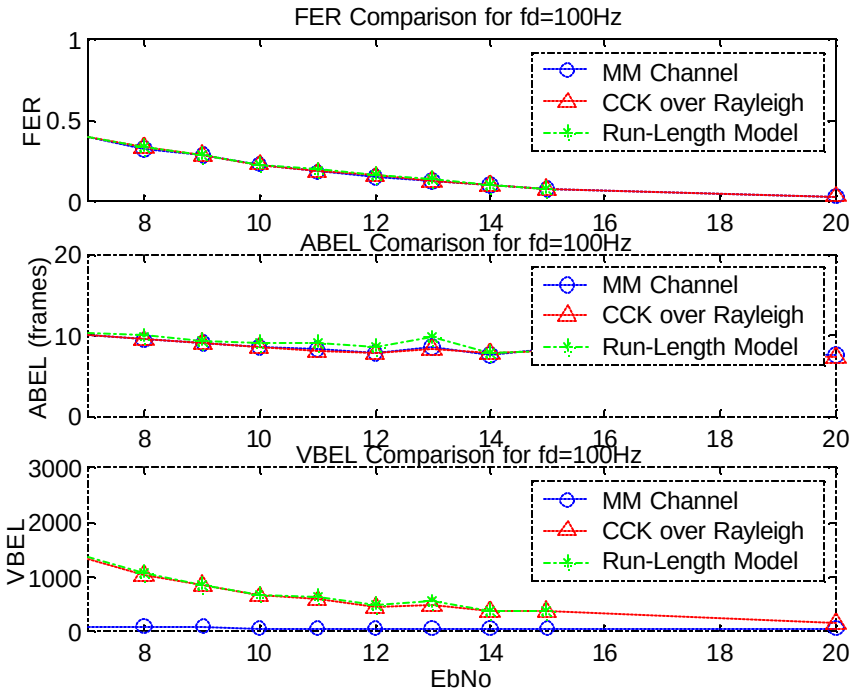

Fig. 50 Comparison of 3SRLM, CCK and MM Frame Error Statistics

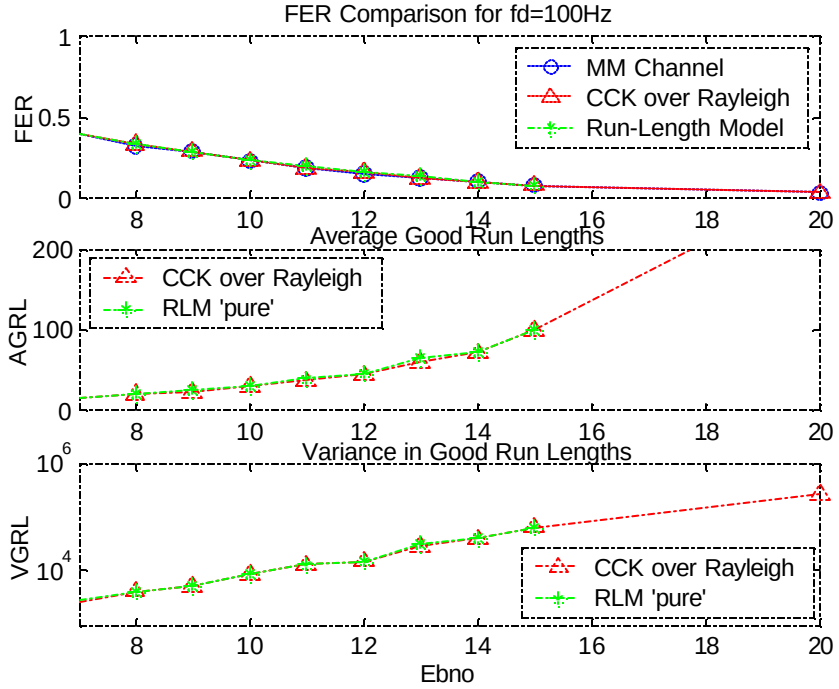

Fig. 51 Comparison of 3SRLM, CCK and MM Good Run Statistics

For the case of high SNR, the terms a_{nf} and a_f become very small and the gamma distribution becomes quite large near the origin. Using the rejection method for random number generation, it becomes difficult to form adequate cover sequences that have a defined pdf for the case of a gamma distribution with a <1. By taking some liberties in the assignment of the cover sequence, one can form an approximate random number generator that performs well for most values of a <1. One such random number generator was employed in the 3SRLM to achieve the frame statistics shown in Fig. 52 and Fig. 53.

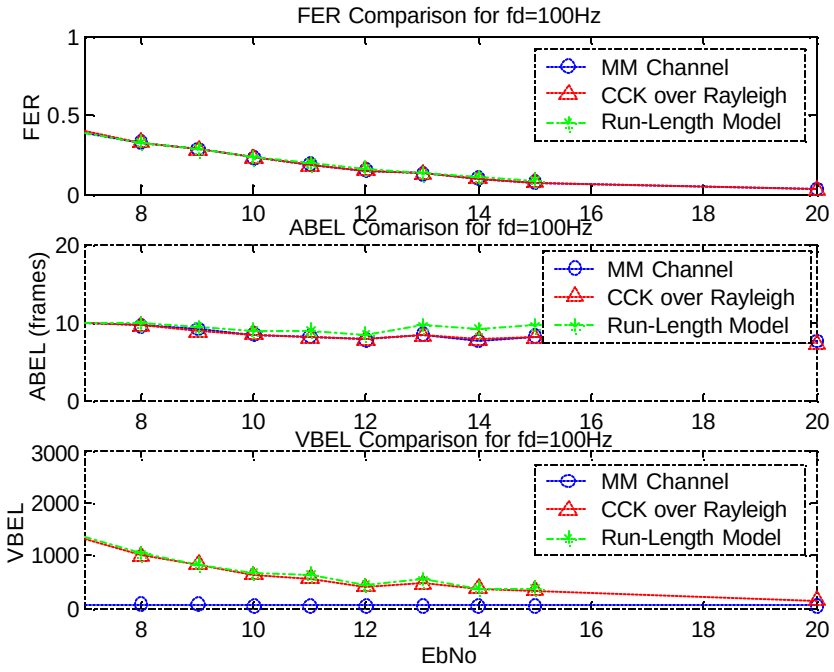

Fig. 52 Comparison of 3SRLM, CCK and MM Frame Error Statistics for Sub-optimal Gamma Random Number Generator

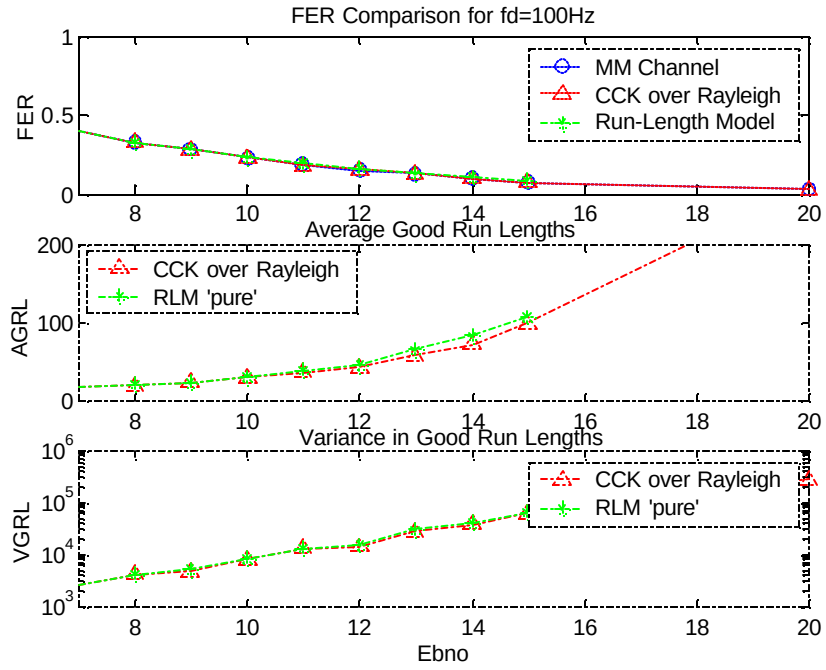

Fig. 53 Comparison of 3SRLM, CCK and MM Good Run Statistics for Sub-optimal Gamma Random Number Generator

The focus of this research is to produce low complexity channel models that more accurately reflect the network performance of a simulated flat Rayleigh fading channel. In line with this goal the sub-optimal gamma random number generator is used in the 3SRLM-PHY module presented in chapter 2 and also in forming the PMD presented in Fig. 54.

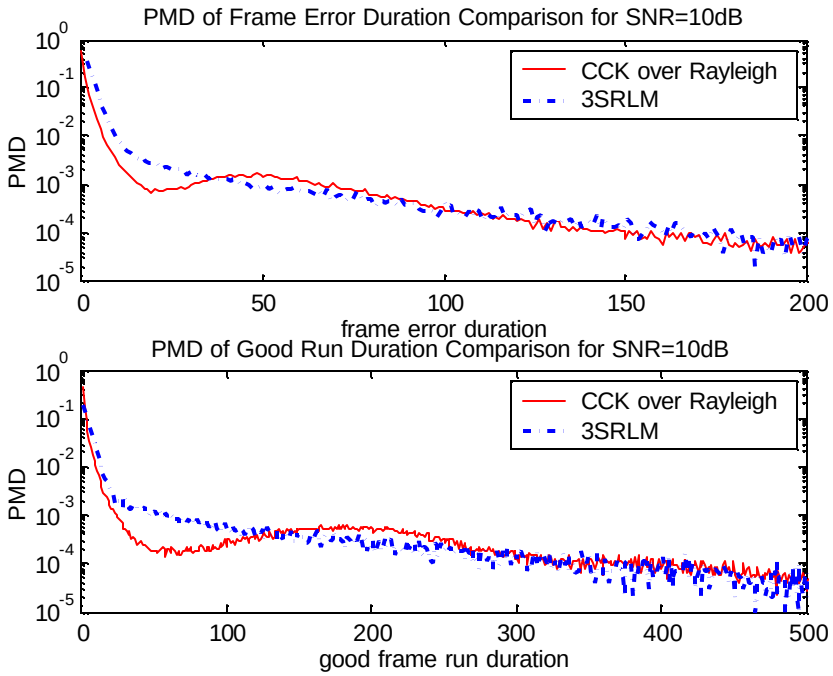

Fig. 54 PMD Comparison of 3SRLM and CCK over Rayleigh for SNR=10dB

As shown in Fig. 52 and Fig. 53, the 3SRLM, is capable of matching not only the frame error statistics of FER, ABEL and VBEL, but is also capable of matching the good run statistics of AGRL and VGRL. When a sub-optimal gamma random number generator is used, the 3SRLM is able to operate with minimal computational complexity while only sacrificing a negligible amount of accuracy. The 3SRLM achieves a good approximation for the observed frame statistics,and in doing so it also achieves a fair approximation of both the PMD of frame error durations and the PMD of good frame run durations, as shown in Fig. 54. This result suggests that matching the frame statistics of FER, ABEL, VBEL, AGRL and VGRL can produce a good approximation

of the PMD's for CCK modulated frames transmitted across a flay Rayleigh fading channel.

E. Summary

This chapter presented alternatives to the two-state Markov model for modeling a flat Rayleigh fading channel in network simulations. An 'N'-state Markov model was presented and frame error results were obtained for the cases of N= 2, 4, 16, 64 and 128. Poor ABEL and VBEL results were obtained for N < 64, furthermore the required number of states (N) for matching a given frame error process was ambiguous at best. Next, the run-length model (RLM) was introduced with three methods for estimating model parameters. The first two methods attempted to assign parameters according the physical layer performance of a modulation scheme and blind to the desired frame error statistics. Results from the first two methods performed very poorly with respect to FER, ABEL and VBEL but were able to approximate the tail performance of the PMD of frame error duration. The third method for estimating RLM parameters was based upon matching the frame error statistics of FER, ABEL and VBEL. This model performed very well with respect to producing the observed frame error statistics, however the model was unable to match the good frame run duration statistic VGRL. Finally, the 3SRLM was presented and parameter estimation derived for matching the frame statistics of FER, ABEL, VBEL, AGRL and VGRL. Performance of the 3SRLM was superb with the resulting model producing similar frame statistics to the CCK

operating over a flat Rayleigh fading channel. Thus far, this chapter has not quantified the effects of matching the frame statistics. It will be shown in chapter 6 that by matching frame statistics a low complexity channel model increases a simulations ability to emulate the network performance obtained when simulating a modulation scheme operating over actual flat Rayleigh fading channel.

CHAPTER VI

NETWORK PERFORMANCE

While matching frame error statistics is impressive, it has not been shown to be a sufficient metric for determining the worth of a channel approximation. This chapter again examines the sufficiency of low complexity channel approximation, but in contrast to section 5, the metric of comparison is network performance instead of frame error statistics. In this chapter, the physical layer channel modules presented in Chapter III are formed through parameter estimations, as shown in Chapter V, placed into the network simulator ns2 and examined for the network performance metric of throughput. It was shown in Chapter IV that the popular two-state Markov model (MM) channel approximation was unable to accurately represent the throughput of the TAMU-PHY model. This chapter will present the performance of the run-length model (RLM) and the three-state run-length model (3SRLM) and compare them against the performance of the two-state MM. It is shown using over 300 simulations, that the RLM is able to achieve substantial improvements over the MM, but the 3SRLM is capable of faithfully approximating the network performance of a flat Rayleigh fading channel.

A. Network Setup

Network simulations were conducted for the specific purpose of analyzing the performance of low complexity channel approximations. The impact of accurately

modeling the flat fading channel should be very interesting, and produce fruitful discussions however this document focuses specifically upon the performance of channel approximations. Therefore, a simple two-node network was created with one node sending application data packets to the other. The nodes are separated by a single symmetric flat fading channel represented by the modules TAMU-PHY, MM-PHY, RLM-PHY and 3SRLM-PHY. An additional module was included for the N=128 state Markov model and called 'N=128MM'. This module is a simple revision of the MM-PHY to include N^2 states and otherwise operates exactly as stated in the MM-PHY description. The inclusion of the 'N=128MM' module is not intended for analysis purposes, as this approximation technique was deemed unsuitable, and is only included for completeness.

Each simulation was conducted for a fixed signal-to-noise ratio (SNR) allowing for meaningful comparisons between modules. A symmetric channel was chosen to reduce the amount of variability in the simulations. If unique channels were employed for both the 'forward' and 'reverse' links, the correlation between the two channel realizations would become important for relatively short duration simulations. The transport layer datagram size of 512 bytes was chosen to fit completely within a single packet. By doing so across a single link, the measurements of goodput and throughput become synonymous.

The channel approximations were formed for the Doppler frequency of 100Hz and SNR {5dB-15dB}. The ns2 simulations were conducted using the 802.11b MAC protocol, as modified in Chapter II to accept custom physical layer channels. Both TCP

Reno and UDP transport protocols are utilized. All ns2 network simulations employed the distributed coordination function (DCF) at the MAC layer and incorporated a queue length of 50 packets. Those simulations using TCP Reno were initiated with a window length of 100. It is worth noting that the choice of packet length and window size do impact network performance, however they have limited impact on link throughput and the values chosen for this work are extremely conservative.

B. Parameter Estimation

As explained in Chapter III, the TAMU-PHY module performs symbol level modulation and demodulation across a simulated flat Rayleigh fading channel. This module contains significant computational complexity and is not feasible for use in large network simulations. The physical layer modules of MM-PHY, RLM-PHY and 3SRLM-PHY are formed to approximate the performance obtained in TAMU-PHY with only a fraction of the computational complexity. The results of network throughput are presented in this section.

The MM-PHY contains only two parameters; p and q that are calculated from the frame error statistics of e and r in (4.8) and (4.9). The procedure defined in Chapter IV was used to create the values in Table II to form the two-state MM channel approximation for SNR={5-15dB}.

Table II. Two-State Markov Model Parameters

	MM Parameters			
SNR	ε	*avg BEL*	*p*	*q*
5	0.5381	11.775	0.9011	0.9151
7	0.3898	9.8563	0.9352	0.8985
8	0.3256	9.527	0.9493	0.8950
9	0.2797	8.9337	0.9565	0.8881
10	0.224	8.4995	0.9660	0.8823
11	0.1809	8.1269	0.9728	0.8770
12	0.1515	7.8329	0.9772	0.8723
13	0.1232	8.3289	0.9831	0.8799
14	0.0978	7.7794	0.9861	0.8715
15	0.076	8.0765	0.9898	0.8762

Sub-frame	296 bits
fd	100 Hz

The RLM contains two states each with a duration random variable modeled as a gamma distribution. The non-fade state of the RLM also contains a frame error rate defined by *fer*. Chapter V presents multiple methods for parameter estimation of the RLM, the third method defined in equations (5.23) through (5.35) was used for the network simulations. RLM parameters are presented in Table III. The gamma distribution terms of a and l are solved through (5.21) and (5.22).

The 3SRLM contains a third state called a 'transition' state that is defined by a static duration of m_t. The parameter estimation was presented in Chapter V following the RLM explanation. Table IV fully defines the 3SRLM for approximating the performance of a flat Rayleigh fading channel.

Table III. Run-Length Model Parameters

RLM Parameters

SNR	m_nf	m_f	fer	var_nf	var_f
5	178.012	171.62	0.0927	12040	17010
7	222.54	120.25	0.0600	19220	4490
8	248.56	102.9	0.0465	31680	2480
9	2.72.71	90.9	0.0397	39360	1329
10	310.13	77.27	0.0306	54490	881
11	349.65	66.87	0.0242	71760	584
12	385.64	59.68	0.0201	93430	265
13	427.97	53.04	0.0145	115920	162
14	484.93	46.19	0.0118	158500	118
15	550.35	40.28	0.0083	190300	51

Sub-frame 296 bits
fd 100 Hz

Table IV. Three State Run-Length Model Parameters

3SRLM Parameters

SNR	m_nf	m_f	m_t	var_nf	var_f	p	q
5	52.84	89.37	62.77	9398	14420	0.80	0.60
7	75.12	51.95	42.96	12967	6905	0.80	0.60
8	88.23	42.21	34.93	17098	4351	0.80	0.60
9	84.04	30.74	25.50	17066	3039	0.80	0.60
10	98.98	25.56	21.35	25250	2105	0.80	0.60
11	108.54	20.82	16.44	33685	1647	0.80	0.60
12	104.01	16.08	11.61	31137	941	0.80	0.60
13	100.03	12.70	5.64	45044	783	0.80	0.60
14	71.65	7.78	0.00	37268	367	0.80	0.60
15	98.18	8.08	0.00	60954	358	0.80	0.60

Sub-frame 296 bits
fd 100 Hz

C. Network Performance

Simulations were conducted for an offered load of 1100kbps, 3300kbps, 6600kbps and 9900kbps representing a data link load of 10%, 30%, 60% and 90% respectively of the 11Mbps link. Each network simulation was conducted for a simulation time of 10 minutes, representing a total of ~22.3 million simulated sub-frames in the channel models. A trace file was formed for each simulation and later analyzed for throughput. In addition to the trace file, the physical layer modules recorded the frame integrity results and frame size each time the mac-802_11.cc file called a *-PHY module. The following figures show the percent-throughput of each link measured as a ratio of the data delivered (in bytes) across the link to the data offered to the link (in bytes).

For the case of 10% load in Fig. 55, the 3SRLM almost exactly matches the performance of the TAMU-PHY module which it is built to approximate. At high SNR all three channel approximations are very close in performance, however at low SNR both the RLM and the MM do not provide an adequate approximation to the flat Rayleigh fading channel, as simulated in TAMU-PHY. Fig. 56, Fig. 57 and Fig. 58 present similar results.

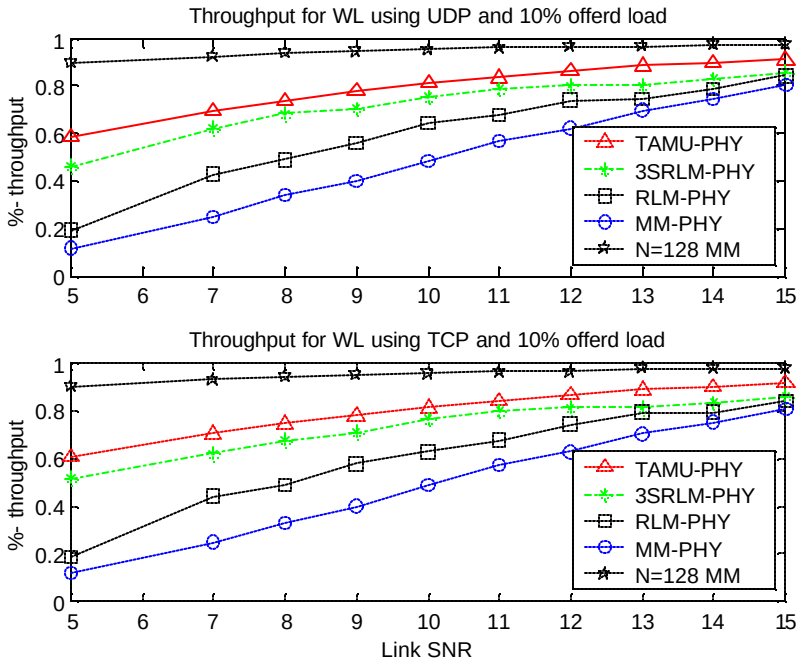

Fig. 55 Percent-throughput Comparison for Offered Load = 10%

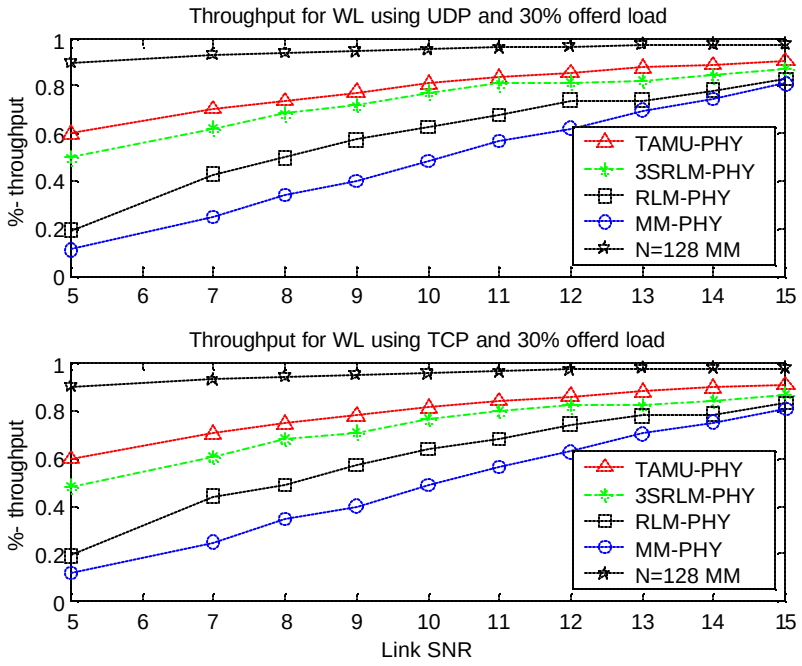

Fig. 56 Percent-throughput Comparison for Offered Load = 30%

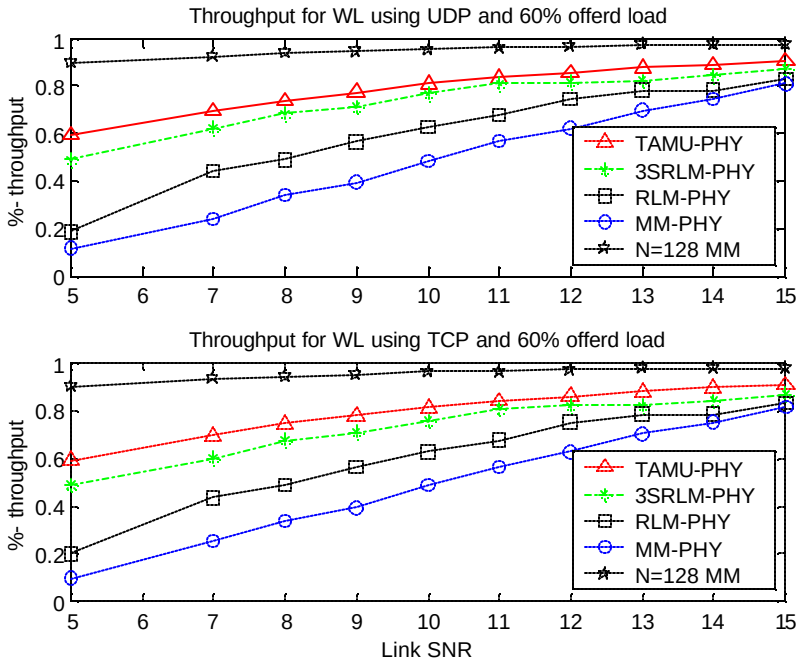

Fig. 57 Percent-throughput Comparison for Offered Load = 60%

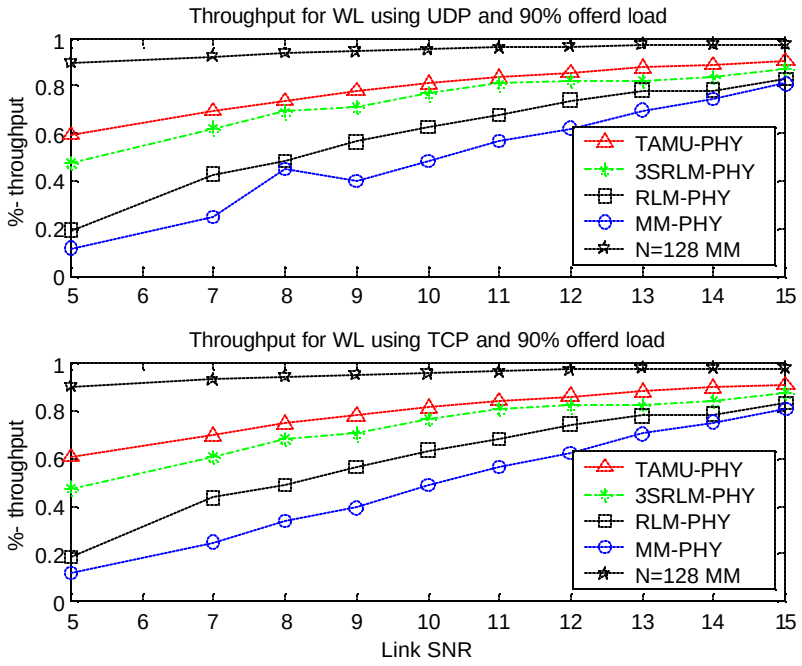

Fig. 58 Percent-throughput Comparison for Offered Load = 90%

The use of percent-throughput as a metric can be deceptive as the MAC and transport layers may attempt to send data when a 'good' channel is perceived. The actual throughput for each module is presented in Fig. 59 for the case of 30% offered load, and again in Fig. 60 in a log scale.

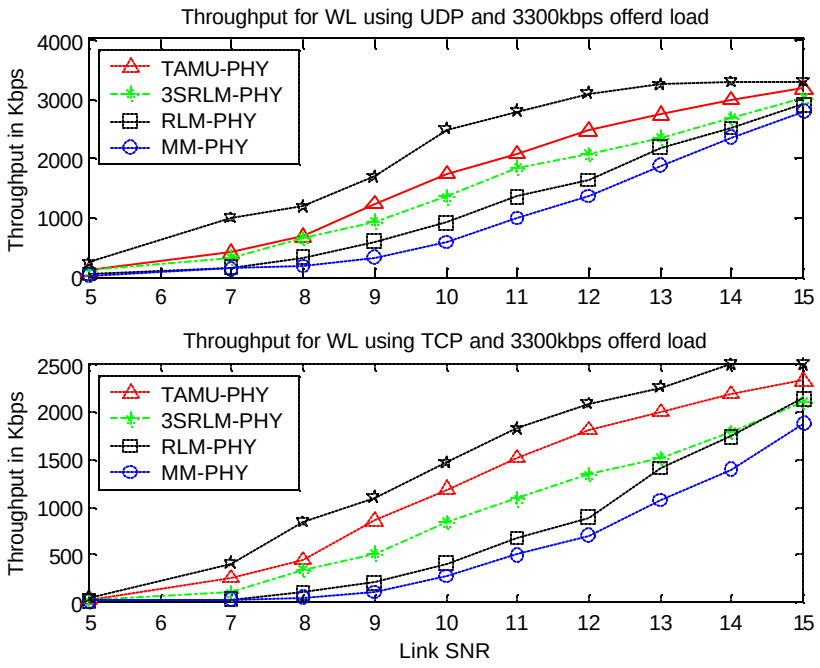

Fig. 59 Absolute-throughput Comparison for Offered Load = 30%

Yet another way to represent the performance of the channel approximations is through a percent-error calculation against the TAMU-PHY. Fig. 61 clearly represents the percent-error of the channel approximations' percent-throughput.

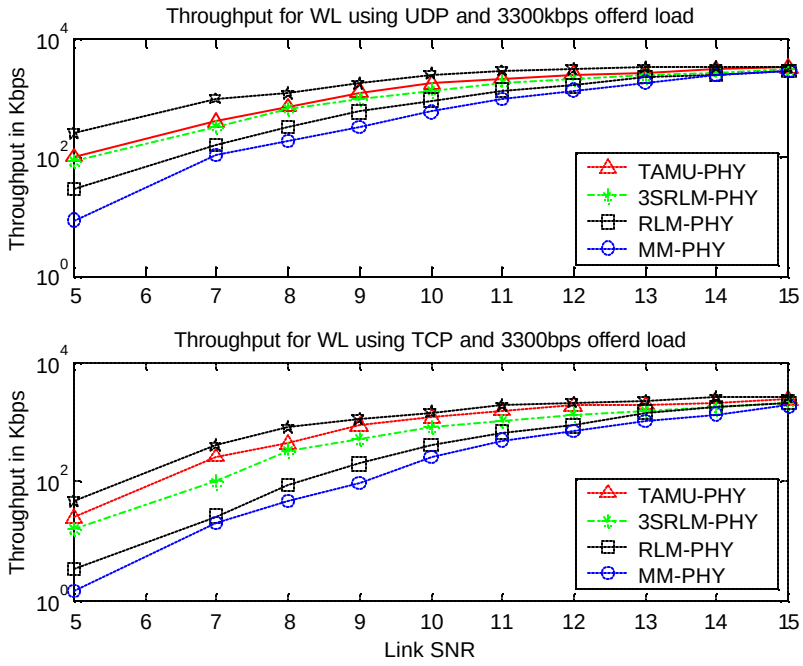

Fig. 60 Log Absolute-throughput Comparison for Offered Load = 30%

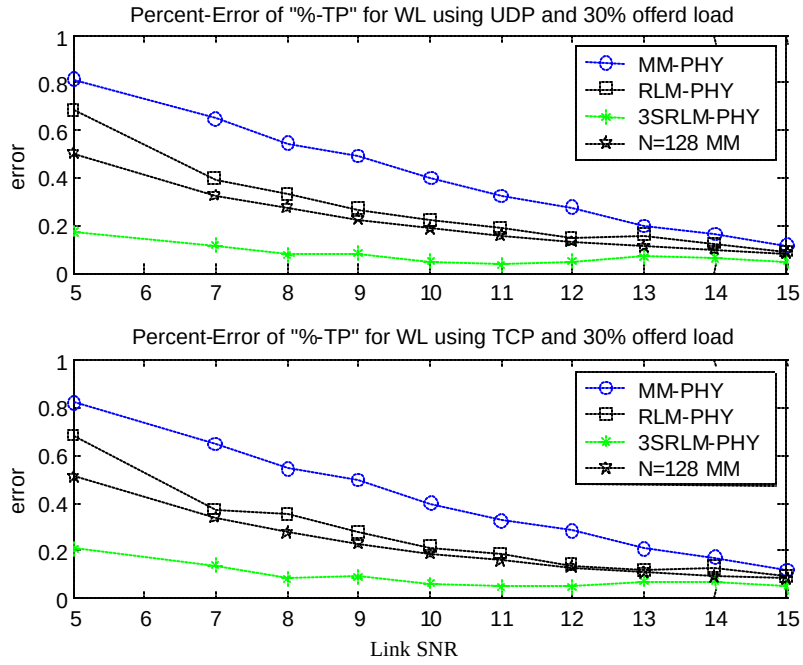

Fig. 61 Percent-error for Offered Load = 30%

It is evident that the 3SRLM is a vastly superior channel approximation for flat Rayleigh fading channels as it is designed to approximate higher order frame statistics as well as FER and ABEL. The two-state Markov model is shown to contain as much as 40% error at an SRN of 10dB, an up to 80% error at 5dB SNR. The 3SRLM's gain in performance over the MM comes at the cost of greater complexity in parameter estimation and operation. The resulting 3SRLM-PHY takes ~3 times longer to run than the MM-PHY. As a data point, the MM-PHY takes 68 seconds to execute a 600 second simulation with an offered load of 90% using UDP on a dual 1800MP+ processor computer with 2GB of DDR RAM. The RLM takes 67 seconds to run the same simulation, while the 3SRLM takes 186 seconds. The TAMU-PHY module requires ~68,000 seconds to perform the same simulation. If one assumes the majority of the simulation time is due to channel approximations, then the delay comparison can be extended to networks with more than two nodes. Since every node is required to attempt detection of every transmitted frame in an 802.11b network, the channel delay will be N*(N-1)-1 times greater than the two node case for networks with N nodes. Fig. 62 represents the approximate channel delay of the MM-PHY, 3SRLM-PHY and TAMU-PHY, with respect to the MM-PHY, for simulations with up to 100 nodes.

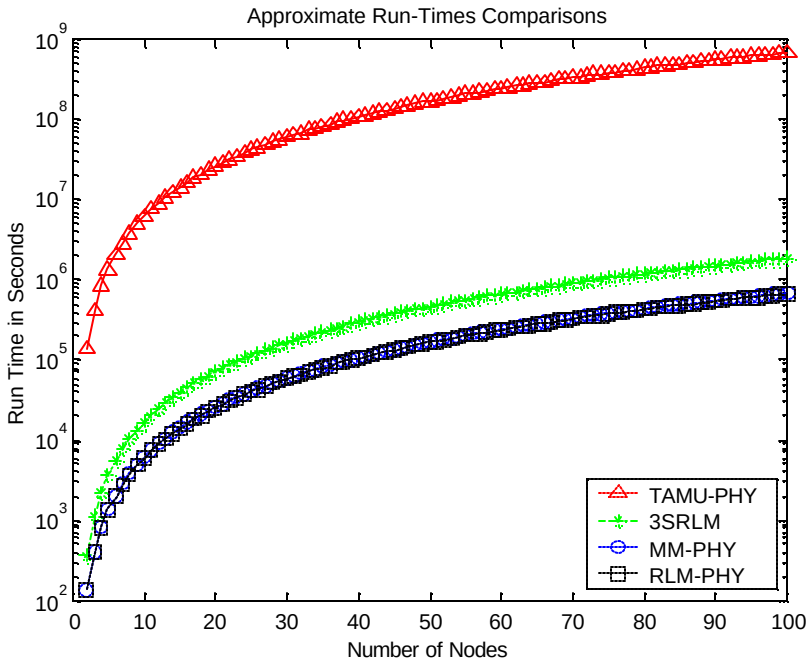

Fig. 62 Run Time Comparison of Channel Modules

In way of an example, if one simulated a 100 node network using the MM-PHY channel module, as previously described, it would take ~7.8 days using the workstation mentioned earlier. The 3SRLM-PHY would require ~21.3 days and the TAMU-PHY module would require ~21.3 years. The 3SRLM provides very accurate results at a slight cost in computational complexity. More over, the 3SRLM allows simulations to incorporate the effects of flat Rayleigh fading channels with multiple nodes, where a direct implementation would require unreasonably long run times.

D. Summary

This chapter identified a network setup capable of comparing the physical layer channel approximations in a network simulator. Furthermore, the parameter estimations used in the physical layer modules of MM-PHY, RLM-PHY and 3SRLM-PHY were presented. Network simulation results are presented for each PHY module developed in Chapter III. The throughput performance results clearly signify the 3SRLM as a superior channel approximation to the widely accepted Markov model for approximating the performance of flat Rayleigh fading channels. Additionally, the 3SRLM is able to remain within 20% of the actual throughput at low SNR where the MM can achieve higher than 80% error. These results also indicate a strong correlation between approximating network performance and matching the PMD of frame error duration and the PMD of good frame run durations of a representative frame error process.

The RLM exactly matched the flat Rayleigh fading channel's frame statistics of FER, ABEL, VBEL and AGRL but was unable to match the variance in good run length (VGRL). It was shown in Chapter V that the RLM produces a PMD of good fame run durations with an exponential distribution that was unable to match the approximated, heavy tail PMD. This is due to the parameter estimation requirement of exactly matching the FER, ABEL, VBEL and AGRL. The RLM's throughput performance was better than the MM, but not sufficient to acceptably approximate the flat fading channel's network performance. To overcome the restrictions of the RLM, additional degrees of freedom were added to form the 3SRLM.

The 3SRLM produces a PMD of frame error duration and PMD of good frame run duration that very closely approximates the PMD of the CCK operating across a flat Rayleigh channel. This is achieved through the model structure and the parameter estimation. By including two states (fade, non-fade) with a controlled random distribution, the 3SRLM is able to match frame-duration PMDs with tails heaver than the transition state's exponential distribution. In addition to its extremely flexible frame-duration structure, the 3SRLM can be formed to almost exactly match the flat Rayleigh fading channel's frame statistics of FER, ABEL, VBEL, AGRL and VGRL. Thus, the 3SRLM is capable of approximating the network performance exceptionally well. The additional complexity of the 3SRLM-PHY was shown produce run times of only three times longer than the MM-PHY. In summary, the results presented in this chapter validate the approach of designing channel approximation to match high-order frame statistics, and validate the 3SRLM as an appropriate, low complexity, flat Rayleigh fading channel approximation.

CHAPTER VII

CONCLUSION

The work contained in this dissertation addressed the design and analysis of low complexity channel approximations for network simulations. First, the popular two-state Markov model channel approximation was explained and shown to inadequately approximate the frame statistics and network performance of the TAMU-PHY flat Rayleigh fading channel model. Next, novel low complexity channel models were described and analyzed for statistical and network performance. Finally, it was shown through statistical and network analysis that the performance of a channel approximation is directly related to its ability to mimic the probability mass distribution (PMD) of frame error duration and good frame run duration.

Detailed background information was provided on the fading environment. Real-world channel measurements were taken and results provided. This information was used to justify the need for analyzing the flat fading channel for high-speed wireless links.

The network simulator ns2 was introduced along with a detailed description for each channel module used in acquiring network simulation results. Implementation issues such as heterogeneous frame sizes and temporal correlations were described along with their solutions.

A well accepted two-state Markov model channel approximation was rigorously analyzed for statistical accuracy and network performance and found to provide

inappropriately conservative throughput results for low to moderate SNR. It was hypothesized that matching the statistical parameters of VBEL and VGRL along with FER, ABEL and AGRL would achieve accurate network performance results; however the first novel channel model (N=128 state Markov model) proved this to be false.

The Markov model was extended to include multiple states through the first new channel approximation: the N-state Markov Model. This model was created to match the frame statistics of FER, ABEL, AGRL, VBEL and VGRL. While the model successfully approximated all of the desired frame statistics, it failed to match the PMD of frame error duration and good frame run duration. The most obvious departure between the desired and actual PMDs was the tail region, with the N-state Markov model containing overly heavy tails. The network performance of the N-state Markov model was overly-optimistic, containing nearly 90% throughput across all SNR. It was desired to find a model that could more accurately approximate the tails of a PMD.

The run-length model (RLM) was introduced and analyzed for three different versions. The first two, RLMv1 and RLMv2, derived the parameter estimation through physical layer performance equations without the use of frame error statistics. In both cases, the RLM contained a frame error duration PMD that better approximated the desired PMD than the N-state Markov model. The RLMv1 and RLMv2's inability to match the basic frame statistics of FER, ABEL and VBEL overshadowed the closer tail PMD performance. A third approach was presented and simply called RLM, where frame error statistics are used to estimate the model parameters. This approach resulted in an exact matching of the frame statistics of FER, ABEL, AGRL, and VBEL but was

unable to accurately model the good frame run duration statistic of VGRL. More importantly, the RLM did not accurately match the distributions of good frame run durations. This model was shown to improve upon the accuracy of two-state Markov model and still provide a conservative estimate of the network performance.

The three-state run-length model (3SRLM) was the final novel, low complexity channel model presented in this work. Parameter estimation for the 3SRLM was derived through knowledge of the desired frame statistics of FER, ABEL, AGRL, VBEL, and VGRL. It was shown through the N-state Markov model that matching these frame statistics is insufficient to achieve the desired PMD for frame error duration and good frame run duration, however the 3SRLM contained a structure that allowed purely random good frame run and frame error states. The 3SRLM's structure allowed sufficient variability in the tails to achieve the desired probability mass distributions through the matching of the aforementioned frame statistics. Network simulations were performed upon the 3SRLM and showed a very accurate approximation of throughput.

In conclusion, this work provided motivation for the development and use of flat fading channel approximations. Furthermore, it analyzed the most popular existing channel approximation and found it lacking. New models were created and, through their analysis, it was determined that by matching the PMD of frame error duration and the PMD of good frame run duration, a channel approximation achieves a more accurate network performance in low to moderate SNR. One of the models, the 3SRLM, is shown to achieve a good approximation of both the desired probability mass distributions with only knowledge of the key frame statistics of FER, ABEL, AGRL,

VBEL and VGRL. The cost of this new model is described in terms of relative computational complexity through run-time observations and is shown to be only three times more complex than the two-state Markov model.

REFERENCES

[1] A. Chockalingam, M. Zorzi and R. R. Rao, "Performance of TCP on wireless fading links with memory," in *Proc. IEEE Int. Conf. on Communications*, vol. 1, pp. 595-600, Jun 1998.

[2] D. Huang and J. J. Shi, "TCP over packet radio," in *Proc. IEEE Emerging Technologies Symposium: Broadband, Wireless Internet Access*, 2000.

[3] D. Huang and J. J. Shi, "TCP over packet radio link with adaptive channel coding," in *Proc. IEEE Vehicular Technology Conf.*, vol. 2, pp. 790-794, Spring 2000.

[4] A. Kumar, "Comparative performance analysis of version of TCP in a local network with a lossy link," *IEEE/AMC Trans. on Networking*, vol. 6, pp. 485-498, Aug. 1998.

[5] J. L. Mineweaser, J. S. Stadler, S. Tsao and M. Flanagan, "Improving TCP/IP performance for the land mobile satellite channel," in *Proc. IEEE Military Communications Conf.*, vol. 1, pp. 711-718, 2001.

[6] M. Zorzi, A. Chockalingam and R. R. Rao, "Throughput analysis of TCP on channels with memory," *IEEE Journal on Select. Areas in Communications*, vol. 18, pp. 1289-1300, July 2000.

[7] F. Anjum and L. Tassiulas, "An analytical model for the various TCP Algorithms operating over a wireless channel," in *Proc. IEEE Wireless Communications and Networking Conf.*, vol. 2, pp. 943-947, 1999.

[8] M. Zorzi and R. R. Rao, "On the statistics of block errors in bursty channels," *IEEE Trans. on Communications*, vol. 45, pp. 660-667, June 1997.

[9] N. M. Chaskar, T. V. Lakshman and U. Madhow, "TCP over wireless with link level error control: analysis and design methodology," *IEEE/AMC Trans. on Networking*, vol. 7, pp. 605-615, Oct. 1999.

[10] C. F. Chiasserini and M. Meo, "Improving TCP over wireless through adaptive link layer settings," in *Proc. IEEE Global Telecommunications Conf.*, vol. 3, pp. 1766-1770, 2001.

[11] A. Chockalingam, W. Zu, M. Zorzi and L. M. Milstein, "Throughput-delay analysis of multichannel wireless access protocol," *IEEE Trans. on Vehicular Technology*, vol. 49, no. 2, pp. 661-671, 2000.

[12] H. K. Shiu, Y. H. Chang, T. C. Hou and C. S. Wu, "Performance analysis of TCP over wireless link with dedicated buffers and link level error control," in *Proc. IEEE Int. Conf. on Communications*, vol. 10, pp. 3211-3216, 2001.

[13] H. L. Vu and S. Hanly, "A study of TCP performance and buffer occupancy over a fading wireless link," in *Proc. IEEE Global Telecommunications Conf.*, vol. 6, pp. 3478-3482, 2001.

[14] M. Zorzi, R. R. Rao and L. B. Milstein, "A Markov model for block errors on fading channels," in *Proc. IEEE PIMRC '96.*, vol. 3, pp. 1074-1078, 1996.

[15] M. Zorzi, R. R. Rao and L. B. Milstein, "ARQ error control for fading mobile radio channels," *IEEE Trans. on Vehicular Technology*, vol. 46, no.2, pp. 445-455, Feb. 1997.

[16] V. Bharghavan, A. Demers, S. Shenker, S. Shenker, and L. Zhang, "MACAW: A media access protocol for wireless LANs," in *Proc. AMC SIGCOMM*, pp. 212-225, August 1994.

[17] B. Bensaou, Y. Wang and C. C. Ko, "Fair medium access in 802.11 based wireless ad-hoc networks," in *Proc. Workshop on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pp. 99-106, Aug. 2000.

[18] G. Aggelou and R. Tafazolli, "RDMAR: A bandwidth-efficient routing protocol for mobile ad hoc networks," in *Proc. AMC Int. Workshop on Wireless Mobile Multimedia (WoWMoM)*, August 1999.

[19] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward, "A distance routing effect algorithm for mobility (DREAM)," in *Proc. ACM/IEEE Int. Conf. on Mobile Computing and Networking*, pp. 76-84, Oct. 1998.

[20] S. H. Bai, S. J. Lee, W. Su and M. Gerla, "The design, implementation and performance evaluation of the on-demand multicast routing protocol in multihop wireless networks," *IEEE Networks*, vol. 14, pp. 70-77, Jan. 2000.

[21] P. Bhagwat and A. Segall, "A routing vector method (RVM) for routing in bluetooth scatternets," in *Proc. IEEE Int. Workshop on Mobile Multimedia Communications (MOMUC)*, pp. 375-379, Nov. 1999.

[22] J. Broch, D. A. Maltz, D. B. Johnson, Y. Hu and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proc. ACM/IEEE Int. Conf. on Mobile Computing and Networking*, pp. 85-97, Oct. 1998.

[23] J. Chang and L. Tassiulas, "Energy conserving routing in wireless ad hoc networks," in *Proc. IEEE INFOCOM*, pp. 22-31, March 2000.

[24] R. Dube, C. D. Rais, K. Wang and S. K. Tripathi, "Signal stability based adaptive routing (SSA) for ad hoc mobile networks," *IEEE Personal Communications*, vol. 4, pp. 36-45, Feb. 1997.

[25] W.C. Jakes, Jr., *Microwave Mobile Communications*, New York, NY: John Wiley & Sons, 1974.

[26] T. S. Rappaport, *Wireless Communications – Principles and Practice*, Upper Saddle River, NJ: Prentice Hall, 1996.

[27] P.G. Flikkema and S.G. Johnson, "A comparison of time-and-frequency-domain wireless channel sounding techniques," in *Proc. IEEE Southeastcon '96*, pp. 488-491, April 1996.

[28] J. W. McKown and R. L. Hamilton, Jr., "Ray tracing as a design tool for radio networks," *IEEE Network Mag.*, pp. 27-30, Nov. 1991.

[29] R. A. Valenzuela, "Ray tracing prediction of indoor radio propagation," in *Proc. 5th IEEE Int. Symp. Personal, Indoor, Mobile Radio Communications.*, vol. 34, no. 22, pp. 140-144, Sept. 1994.

[30] S. C. Kim, B. J. Guarino Jr., T.M. Willis III, V. Erceg, S. J. Fortune, R. A. Valenzuela, L. W. Thomas, J. Ling, and J. D. Moore, "Radio propagation measurements and prediction using three-dimensional ray tracing in urban environments at 908 MHz and 1.9 GHz," *IEEE Trans. Vehicular Technology*, vol. 48, pp. 931-946, May 1999.

[31] L. Piazzi and H. L. Bertoni, "Achievable accuracy of site-specific path-loss predictions in residential environments," *IEEE Trans. Vehicular Technology*, vol. 48, pp. 922-930, May 1999.

[32] J. G. Proakis, *Digital Communications*, New York, NY: McGraw-Hill, 1995.

[33] ISO/IEC 8802-11; ANSI/IEEE Std 802.11, 1999 edition, Aug. 1999

[34] D.F. Bacon, J. Schwartz and Y. Yemini, "Nest: A network simulation and prototyping tool," in *Proc. USENIX Conf. '99,* pp. 71-78, 1999.

[35] S. Keshav, "REAL: A network simulator," tech. Report 88/472, Univ. California, Berkeley, 1988.

[36] L. Breslau, D. Estrin, K. Fall, S. Floyd, J. Heidemann, A. Helmy, P. Huang, S. McCanne, K. Varadhan, Ya Xu and Yu Haobo, "Advances in network simulation," *Computer*, vol. 33, no. 5, pp. 59-67, May 2000.

[37] K. Fall, K. Varadhan, "The *ns* Manual (formerly *ns* Notes and Documentation)," Available at http://www.isi.edu/nsnam/ns/doc/ [Accessed Jan. 2003].

[38] S. Miller, "Unit3B," Class notes, ELEN 689, Dept. Electrical Engineering, Texas A&M University, College Station, TX, Spring 1999.

[39] A. Ambardar, *Analog and Digital Signal Processing*, Boston, MA: International Thomson Publishing, 1995.

[40] H.S. Wang "On verifying the first-order Markovian assumption for a Rayleigh fading channel model," in *Proc. ICUPC '94,* pp. 160-164, Sep. 1994.

[41] H.S. Wang, "Finite-state Markov channel – a useful model for radio communication channels," *IEEE Trans. Vehicular Technology.*, vol. 44, no. 1, pp. 163-171, Feb. 1995.

[42] M. Zorzi, R.R. Rao and L.B. Milstein, "Error statistics in data transmission over fading channels," *IEEE Trans. Communications*, vol. 46, pp. 1468-1477, Nov 1998.

[43] M. Zorzi, R. Rao, and L. Milstein, "On the accuracy of a first-order Markov model for data transmission on fading channels," in *Proc. UPC '95*, pp. 211-215, Nov. 1995.

[44] A. A. Abouzeid, S. Roy and M. Azizoglu, "Stochastic modeling of TCP over lossy links," *Nineteenth Annual Joint Conf. of IEEE Computer and Communications Societies*, vol. 3, pp. 1724-1733, Mar. 2000.

[45] A. Chockalingam and B. Gang, "Performance of TCP/RLP protocol stack on correlated fading DS-CDMA wireless links," *IEEE Trans. on Vehicular Technology*, vol. 49, pp. 28-33, Jan. 2000.

[46] A. Chockalingam, M. Zorzi, L. B. Milstein and P. Venkataram, "Performance of a wireless access protocol on correlated Rayleigh-fading channels with capture," *IEEE Trans. on Communications*, vol. 46, no. 5, May 1998.

[47] S. Choi and K. G. Shin, "A unified wireless LAN architecture for real-time and non-real-time communication services," *IEEE/AMC Transactions on Networking*, vol. 8, no. 1, Feb. 2000.

[48] E. Cianca, M. Ruggieri and R. Prasad, "Improving TCP/IP performance over CDMA wireless links: A physical layer approach," in *Proc. IEEE Int. Symp. On Personal, Indoor and Mobile Radio Communications*, vol. 1, pp. A-83 – A-87, Sep. 2001.

[49] F. H. P. Fitzek and M. Reisslein, "A prefetching protocol for continuous media streaming in wireless environments," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 10, pp. 2015-2028, Oct. 2001.

[50] A. S. Joshi, M. N. Umesh, A. Kumar, T. Mukhopadhyay, K. Natesh, S. Sen and A. Arunachalam, "Performance evaluation of TCP over radio link protocol in TIE/EIA/IS-99 environment," in *Proc. IEEE Int. Conf. on Personal Wireless Communications*, pp. 216-220, 1999.

[51] M. Liu, M. T. Liu, M. E. Muller and J. He, "A MAC protocol supporting TCP in DS-CDMA PCNs," in *Proc. IEEE Performance, Computing and Communications Conf.*, pp. 8-14, Feb. 2000.

[52] G. L. Wu and J. W. Mark, "Computational methods for performance evaluation of a statistical multiplexer supporting bursty traffic," *IEEE/AMC Trans. on Networking*, vol. 4, pp. 386-397, June 1996.

[53] R.R. Rao, "Perspectives on the impact of fading on protocols for wireless networks", in *Proc. IEEE ICPWC '97*, pp. 489-493, Dec. 1997.

[54] A. Gilbert, "Capacity of a burst-noise channel," *Bell System Tech. Journal*, vol.39, pp. 1253-1266, Sept. 1960.

[55] S.O. Rice “Distribution of the duration of fades in radio transmission: Gaussian noise model,” *Bell System Tech. Journal*, vol38, pp.581-635, May 1958.

[56] M. Patzold and F. Laue “Statistical properties of Jakes’ fading channel simulator,” *1998 Vehicular Technology Conf.*, vol. 2, pp. 712-718, May 1998.

APPENDIX A

DERIVATION OF AVERAGE BURST ERROR LENGTH (ABEL) FOR RLM

This appendix contains the derivation of the ABEL (5.27) for the Run Length Model.

The ABEL can be restated as the average duration of a burst error. In our analysis, all errors are considered 'bursts' and thus ABEL is a measure of the average duration of an 'error', where a single 'error' is defined as a group of consecutive, bad frames. One can define ABEL as the expected duration of an error given an error occurs, through:

$$ABEL = E[E_L \mid e], \tag{A.1}$$

$$ABEL = \sum_i E[E_L \mid s_i]\Pr[s_i \mid e], \tag{A.2}$$

$$ABEL = E[E_L \mid nf]\Pr[nf \mid e] + E[E_L \mid f]\Pr[f \mid e]. \tag{A.3}$$

Each of the terms in (A.3) can be found with respect to the RLM model parameters. The first term is the expected value of error duration in the non-faded state $E[E_L|nf]$. Each frame in the non-faded state has an independent probability of failure of *fer*, making the value of error duration a version of a Geometric random variable. So, this term is actually the number of independent Bernoulli trials until the first success.

$$E[E_L \mid nf] = \frac{1}{p} = \frac{1}{1 - fer} \tag{A.4}$$

The second term is the expected value of the error duration in the faded state $E[E_L|f]$. This is simply the mean of the faded state duration.

$$E[E_L \mid f] = m_f \tag{A.5}$$

The next term is a bit more complex and involves finding the probability of being in the non-faded state given that an error has occurred *P[nf|e]*. Using Bayes theorem, one can form,

$$\Pr[nf \mid e] = \frac{\Pr[e \mid nf_{att}]\Pr[nf_{att}]}{\Pr[e]}. \tag{A.6}$$

where the probability of an error in the non-faded case can be considered as the rate of errors in the non-faded case. Note, this is not the same as the rate of frame errors in the non-faded case, as an 'error' may contain multiple frame errors. The term nf_{att} refers to the number of times an error can be initiated when in the non-faded case. So, Pr(nf_{att}) will take into account that an error may contain more than a single frame error and thus will be fundamentally different than Pr(*nf*).

The probability of an error given an attempt was made in the non-faded state $P[e|nf_{att}]$ can be expressed as a ratio of the expected number of errors in the non-faded case relative to the expected total attempts in the non-faded case,

$$\Pr[e \mid nf_{att}] = \frac{E[e_{nf}]}{E[att]} = \frac{E[e_{nf}]}{E[e_{nf}] + E[no - e_{nf}]} \tag{A.7}$$

and simplified to,

$$\Pr[e \mid nf_{att}] = \frac{fer}{fer + 1} \tag{A.8}$$

through

$$E[e_{nf}] = \frac{fer}{E[E_L \mid nf]} = fer(1 - fer), \tag{A.9}$$

$$E[no - e_{nf}] = 1 - fer. \tag{A.10}$$

The only remaining unknown term in the probability of being in the non-faded state given an error *Pr[nf|error]*, is the probability of an error occurring

$$\begin{aligned} \Pr[e] &= \sum_i \Pr(e \mid s_{i_{att}}) \Pr(s_{i\,att}) \\ &= \Pr(e \mid nf_{att}) \Pr(nf_{att}) + \Pr(e \mid f_{att}) \Pr(f_{att}). \end{aligned} \tag{A.11}$$

As shown in (A.11), this term can be easily found with the knowledge of the probability of a non-faded attempt,

$$\Pr[nf_{att}] = \frac{E[nf_{att}]}{E[att]} = \frac{E[nf_{att} \mid 1\_cycle]}{E[att \mid 1\_cycle]}. \tag{A.12}$$

The probability of a non-faded attempt will be the ratio of the expected attempts in the non-faded case to the expected total attempts.

$$\Pr[nf_{att}] = \frac{E[nf_{att} \mid 1\_cycle]}{E[nf_{att} \mid 1\_cycle] + 1} \tag{A.13}$$

The expected total attempts in the non-faded case will be equal to the sum of the expected successful attempts and the expected error attempts.

$$E[nf_{att} \mid 1\_cycle] = E[good_{att}] + E[e_{att}] \tag{A.14}$$

It is known that the successful attempts will result in a single good frame, while the error attempts will result in a series of '*E[error_durr|nf]*' frame errors. Since the expected total frames in the non-faded case is the known model parameter m_{nf}, one can form the expected value of the error lengths given the non-fade state through:

$$m_{nf} = E[good_{att} \mid 1\_cycle] + E[e_{att} \mid 1\_cycle] E[E_L \mid nf], \tag{A.15}$$

$$E[good_{att}] = (1 - fer) m_{nf}, \tag{A.16}$$

$$E[E_L \mid nf] = \frac{1}{1 - fer}. \tag{A.17}$$

One can form

$$E[nf_{att} \mid 1\_cycle] = (1 - fer^2) m_{nf}, \tag{A.18}$$

for the expected value of the number of non-faded attempts in one cycle. Therefore, (A.13) can be simplified to

$$\Pr[nf_{att}] = \frac{m_{nf}(1 - fer^2)}{m_{nf}(1 - fer^2) + 1}. \tag{A.19}$$

Since each case of the 'faded' state will result in a single error and the probability of a faded attempt is just *1-Pr[nf_{att}]* one can form the probability of an error as

$$\Pr[error] = \left(\frac{fer}{1 + fer}\right)\frac{m_{nf}(1 - fer^2)}{m_{nf}(1 - fer^2) + 1} + \frac{1}{m_{nf}(1 - fer^2) + 1}. \tag{A.20}$$

The term *Pr[error]* can be used to solve for the probability of a non-faded state give an error occurred, as

$$\Pr[nf \mid e] = \frac{\left(\frac{fer}{fer + 1}\right) m_{nf}(1 - fer^2)}{\left(\frac{fer}{fer + 1}\right) m_{nf}(1 - fer^2) + 1}. \tag{A.21}$$

The term *Pr[nf|e]* is one of the four original terms in(A.3). Since the probability of the faded state given a burst error is simply *1-Pr[nf|e]*, the ABEL can be solved as

$$ABEL = \frac{\left(\frac{1}{1 - fer}\right)\left(\frac{fer}{fer + 1}\right) m_{nf}(1 - fer^2)}{\left(\frac{fer}{fer + 1}\right) m_{nf}(1 - fer^2) + 1} + m_f \frac{1}{\left(\frac{fer}{fer + 1}\right) m_f (1 - fer^2) + 1}. \tag{A.22}$$

VITA

Jeffrey Michael McDougall, born in San Antonio, TX on August 29, 1975, received his B.S. degree in electrical engineering from Texas A&M University, College Station in May 1997 and his M.S. degree in electrical engineering from Johns Hopkins University, Baltimore in May 1999. Between 1997 and 1999, he worked as an associate professional staff member at the Applied Physics Laboratory of Johns Hopkins University in the Strategic Systems Department. Jeff began the Ph.D. program at Texas A&M University, College Station in September 1999. He has since worked as a lecturer in both the Electrical Engineering Department and the Engineering Technology and Industrial Distribution Department. During his studies, Jeff consulted for the Southwest Research Institute, San Antonio and for the wireless internet service provider Invisix, Fort Worth. He received his Ph.D. in August 2003. Jeff McDougall may be contacted at: 1006 Val Verde College Station, TX or through mcdougall@tamu.edu.