## Contents

1 Introduction 2

2 Simulation Environment 2
   2.1 PHY - Physical Propagation Model 3
   2.2 MAC - Media Access Control 4
   2.3 IFQ - Interface Queue 5
   2.4 LL - Link Layer 5
   2.5 ARP - Address Resolution Protocol 5
   2.6 Upper Layers: Agent and Routing 6

3 Proposed Cross-Layer Enhancement 6
   3.1 Prediction Scheme 7
   3.2 MAC Operation 8

4 Markov Model for Rayleigh Fading Channels 10

5 Markov Model for IEEE 802.11 DCF 13
   5.1 Analysis of Standard 802.11 MAC Under Fading 14
      5.1.1 Probability of Packet Transmission and of Receiving an Erroneous Packet 15
      5.1.2 Channel Throughput 15
      5.1.3 Transmission Processing Rate 15
      5.1.4 Packet Loss 16
      5.1.5 Delay 16
   5.2 Analysis of Cross-Layer Enhancement Under Fading 17
      5.2.1 Probability of Packet Transmission and of Receiving an Erroneous Packet 17
      5.2.2 Channel Throughput 17
      5.2.3 Transmission Processing Rate 17
      5.2.4 Packet Loss 18
      5.2.5 Delay 18

6 Simulation 18
   6.1 Prediction Accuracy 19
   6.2 Two Node Scenario 19
   6.3 50 Node - UDP 21
   6.4 50 Node - TCP 23
   6.5 Verification of Model 23

7 Conclusion 24

A Errata 27
List of Figures

1 Mobile Node in ns-2 ................................................. 3
2 Access Mechanisms for 802.11 DCF ................................ 9
3 State Model of Fading Process .................................. 11
4 Markov Model of IEEE 802.11 DCF Backoff Mechanism ....... 14
5 Prediction Accuracy for Various Packet Rates .................. 20
6 Performance Measurements vs Various Initial Positions ....... 21
7 Performance Metrics for 50 Node UDP ................................ 22
8 Performance Metrics for 50 Node TCP ............................... 23
9 Verification of Model ............................................ 24
Abstract

In this report we will examine and reproduce some of the results of the proposed cross-layer enhancement from [1]. The specific protocol utilizes knowledge of the underlying fading process of the wireless channel, to correctly schedule packet transmission resulting in a performance improvement in the network. In this report we will also present some of the drawbacks of the scheme, draw conclusions on the results, and in addition introduce several modifications to the proposed scheme.
1 Introduction

Wireless networks are currently found in a wide range of environments. From University campuses, to businesses and residential homes. Wireless networks have also been utilized and extended to wireless sensor networks [2]. Commercial wireless networks are often composed of a base station communicating with individual mobile nodes. Ad hoc networks provide us with a infrastructure-free method of configuration. In addition, the IEEE 802.11 MAC protocol defines the Distributed Coordination Function (DCF) protocol which provides the channel access mechanism for such contention-based wireless networks. Moreover, Quality of Service (QoS) requirements for the access mechanism has been studied in great detail [3]. The IEEE 802.11e MAC protocol provides QoS functionality for beyond the IEEE 802.11 standard MAC.

Up until recently, work on improvement schemes for ad hoc networks has looked at MAC and physical enhancements independent of the other. Recently, there have been significant efforts made to work on cross-layer mechanisms [4], [5] to improve the performance of ad hoc networks. Such mechanisms take advantage of channel state knowledge to improve performance of the network by improvement of the MAC protocol.

The authors of [1] propose one such enhancement in the presence of multi-path fading. Such channel modelling is often used to model wireless communications in urban areas, which is often prone to multiple reflections [6].

In this report we will analyze and compare the results of the protocol in [1]. The rest of this report is organized in the following manner. In Section 2 we will provide a background of the simulation environment used for measurement. In Section 3 we will introduce the cross-layer scheme as proposed in [1], which utilizes the predictability of Rayleigh fading channels. Sections 4 and 5 will discuss the Markov models discussed in [1] for Rayleigh fading channels and IEEE 802.11 DCF protocol respectively. We will then present the simulation results in Section 6 and will draw conclusions on this work in Section 7.

2 Simulation Environment

The Network Simulator (or ns-2) [7] is an open-source discrete events simulator for simulation of networks. Unlike traditional network simulation tools, its versatility and robustness provide the ideal simulation environment. It has become a common research tool as it promotes ease of integration for custom protocols. It allows researchers to easily monitor the performance of an implemented protocol.
The software was originally designed at UC Berkeley [7]. CMU Monarch later developed wireless extensions to the software [8]. Currently, the software is a collaboration of a number of institutes and collaborators with the continued addition of new network protocols and network devices.

This report utilizes the mobile components of ns-2. In this report we will briefly talk discuss how ns-2 models mobile nodes. The structure of a node in ns-2 in terms of object can be seen in fig. 1. Since each object is modelled as a different class, this allows the user to implement a custom layer without modifying any other aspects of the simulator. We will briefly discuss the components of a mobile node.

2.1 PHY - Physical Propagation Model

The physical layer is of ns-2 is modelled as simply an attenuation by the receiving nodes. In ns-2, when a packet is sent down from the MAC to PHY, the simulation scheduler schedules reception at other nodes that are attached to the channel as defined by the simulation, following the properly calculated
propagation delay. It is the responsibility of all nodes in the network to make the decision on whether a particular packet is received. There are 3 default propagation models that are included in the standard ns-2 implementation:

- **Free Space**: This propagation model assumes that line of sight transmission and only measures path loss.
- **Two-Ray Ground**: This model assumes both line of sight transmission as well as reflection from the ground.
- **Shadowing**: This model assumes that there are obstructions similar to an in-door environment

When the packet is scheduled to be received by a node, the corresponding propagation model (based on simulation parameters) is used to calculate the received signal level at that particular node. There are two thresholds used to model the two possible receive states of packets.

- **Carrier-Sense Threshold (CSThresh)**: This parameter is used to model whether or not a given node detects a packet. If the received signal is greater than CSThresh, then the packet is detected by the node.

- **Receive Threshold (RXThresh)**: This parameter is used to determine whether or not a packet was received successfully and can be decoded by the MAC. If the propagation model determines that the packet falls between the two thresholds, this models a packet detected, but suffers from bit errors for which it cannot recover.

In this report, we utilize an extension to the simulator that implements the Ricean $K$ model from [9], where $K = 0$ for Rayleigh fading. The model from [9] accurately models the time-correlation of the received envelope in a discrete-time simulation environment and as such provides an accurate model of the fading process.

### 2.2 MAC - Media Access Control

There are many MAC protocols that are found in the default implementation of ns-2. In this report we utilize **MAC-802.11** which is the IEEE 802.11 Distributed Coordination Function (DCF). This protocol controls channel access and has some of the following features:
• **Carrier-Sense:** The 802.11 DCF defines a carrier-sense mechanism that will sense if the channel is free prior to transmission. The specifications also define a backoff function when there is contention.

• **Virtual Carrier-Sense:** The specifications also define a virtual carrier-sense mechanism where each packet that is transmitted contains a duration field. This field notifies all other nodes not involved in the transmission transaction of the duration for which the channel will be in use.

• **Collision Avoidance:** The 802.11 DCF also defines an optional collision avoidance mechanism. This mechanism uses a Request-To-Send/Clear-To-Send (RTS/CTS) packet exchange before data transmission. The underlying concept is based on the fact that the collision time for a short packet (RTS and CTS), is significantly less than a data fragment.

The overall operation of the 802.11 MAC in ns-2 is similar to the implementable protocol. However it should be noted that the occurrence of a collision is determined in the 802.11 MAC when receiving a packet.

### 2.3 IFQ - Interface Queue

The Interface Queue sub-layer in ns-2 is provides the finite length queue of the link layer to buffer packets to be processed by the MAC. When the MAC is ready to accept a new packet for transmission, it will signal the link layer to advance the buffer. The queue has priority capabilities, and gives priority to routing protocol packets.

### 2.4 LL - Link Layer

The link layer sub-layer in ns-2 manages traffic flow of the medium. It also provides translation capabilities to allow for the interface of various MAC protocols with standard upper layers. The wireless model in ns-2 utilizes a similar link layer as wired nodes, however there is an ARP component (Address Resolution Protocol). The link layer contains a query interface to ARP.

### 2.5 ARP - Address Resolution Protocol

The ARP module in ns-2 simulates the ARP procedure. The purpose of ARP is to resolve the MAC address of a target node. The procedure involves sending an ARP request broadcast to all nodes in the network. The destination node will respond to the ARP request with its corresponding MAC address.
2.6 Upper Layers: Agent and Routing

Upper layers of the mobile node (as in fig.1) are independent of the mobile implementation. The routing agent controls the routing mechanism in the network. This is not specific to mobile nodes, there are however several mobile routing protocols built in such as:

- DSR - Dynamic Source Routing
- DSDV - Destination-Sequence Distance Vector
- AODV - Ad hoc On-Demand Distance Vector
- TORA - Temporally-Ordered Routing Algorithm
- DumbAgent - Which disables multihop routing

The agent layer is where all traffic generation occurs. In terms of ns-2, a transport source agent (such as UDP or TCP) is attached the traffic generating node, while a sink agent is attached to the receiving node (null agent for UDP as no acknowledgements are required). A traffic agent is attached to the transport agent as well. In ns-2 there are many included traffic generation agents (such as exponential, constant bit rate and FTP). All parameters associated with the traffic (packet size, data rate and so on) are configurable in the simulation parameters.

3 Proposed Cross-Layer Enhancement

The cross-layer mechanism proposed in [1] is built on providing the MAC layer of a mobile node with information regarding the current state of the channel. This information can assist the access control protocol in determining whether a transmission should proceed. As such, we define two possible channel states:

- **Good**: The effects of multipath fading are not substantial enough to affect the reception of a given packet; and

- **Bad**: The effects of fading result in the receiving node being unable to receive the packet during this time.
The ideal protocol in [1] would result in the MAC layer knowing the time and duration of each channel state. In this way a decision can be made on whether to proceed with the transmission. The resulting scheme should minimize the power wasted during transmission of a packet by delaying a transmission that would normally lose a packet due to being in a bad state. In addition, in a multi-node scenario, the global knowledge of the delayed transmission would allow other contending nodes to utilize the now ideal medium.

3.1 Prediction Scheme

In a mobile scenario the receiving node can maintain a record of prior received signal levels. As such the authors utilize a prediction algorithm previously discussed in [10]. The algorithm make use of knowledge regarding the channel process. Assuming non-frequency selective fading (flat fading), the autocorrelation is a function of only the time difference between two samples and doppler shift. This autocorrelation function of the channel is well-known as:

$$\rho_\beta = \sigma^2 J_0(2\pi f_D \tau)$$  \hspace{1cm} (1)$$

In which $\rho_\beta$ is the value of the autocorrelation function, $\sigma^2$ is the power contained in the signal, $f_D$ is the maximum doppler shift and $\tau$ is the time-difference between 2 samples. The prediction provides an estimate of the next received signal level based on the linear combination of the last $V$ channel samples.

$$\beta_{\text{est}}(i) = \sum_{k=0}^{V-1} a_k(i) \beta_{\text{meas}}(i - k\delta - D)$$ \hspace{1cm} (2)$$

Where $a_k(i)$ for $k = 0, 1, \ldots, V - 1$ are the derived linear predication coefficients, $\delta$ is the time spacing in seconds between samples, $\beta_{\text{meas}}(i - k\delta - D)$ for $k = 0, 1, \ldots, V - 1$ are the previously sampled received signal levels and $D$ is period of time (in seconds) in the future for which the prediction is made ($\beta_{\text{est}}(i)$).

The derivation of $a_k(i)$ is critical on the accuracy of the algorithm. The authors utilize the MMSE criteria discussed in [10] to find an expression for the linear coefficients where $a(i) = [a_0, a_1, \ldots, a_{V-1}]^T$. This method produces $a(i)$ as:

$$a(i) = R^{-1}(i) \theta(i)$$ \hspace{1cm} (3)$$

Where $R(i)$ is the VxV autocorrelation matrix.
\[ R(i) = \begin{bmatrix} r_{0,0} & r_{0,1} & \cdots & r_{0,V-1} \\ r_{1,0} & r_{1,1} & & \vdots \\ & \vdots & \ddots & \vdots \\ \cdots & \cdots & \cdots & r_{V-1,V-1} \end{bmatrix} \]

With \( r_{m,n} \) given as \( E[\beta_{meas}(i - D - m\delta)] \beta_{meas}^*(i - D - n\delta)] \) for \( m, n = 0, 1, \ldots, V - 1 \).

\( \theta(i) \) is the autocorrelation vector of previous samples with the future predicted sample. This can be computed as only time difference must be considered as in (1). Therefore we define this vector as:

\[ \theta(i) = E[\beta_{meas}(i) \beta_{meas}^*(i - D - m\delta)] \text{ where } m = 0, 1, \ldots, V - 1 \] (4)

There are several key points to be made regarding (2) and the limitations on (3).

- The time spacing, \( \delta \), between two sequential received samples is assumed as fixed. This criterion cannot be guaranteed in a mobile network. Using various received samples data can be fit to this criterion, however with a loss of accuracy.

- (3) is an ill-posed problem [10]. Therefore time spacing \( \delta \) will have significant impact on the performance on the prediction.

- The assumption of a fixed spacing for a given prediction results in a Toeplitz autocorrelation matrix \( R(i) \). Recursive algorithms such as Levinson-Durbin [11] can be used to provide a more stable solution to (3).

In this report, the coefficients are pre-computed using Levinson-Durbin recursion. Coefficients for a variety of time spacings (\( \delta \)) and various doppler shifts (\( f_D \)) are stored in a file and loaded based on these parameters at runtime. The authors suggest that this approach is fast and energy efficient [1].

3.2 MAC Operation

The IEEE 802.11 DCF protocol offers 2 channel access mechanisms:

- Basic CSMA (Carrier-Sense, Multiple Access) (see Fig. 2(a)), and

- CSMA/CA that utilizes a RTS/CTS packet exchange (see Fig. 2(b))
Utilizing the estimated future received signal sample found in (2), the receiver can predict whether the next received packet will be received correctly based on its receive threshold. In conjunction with the mechanisms above, the receiving station can relay this information back to the transmitting station in the header of the acknowledgement packet. In the case of using the DCF collision avoidance scheme, this information can also be relayed in the CTS packet.

In addition to the whether or not a fade will occur, the receiving station also relays the expected duration of the fade to the transmitting station. This allows the transmitting station to reschedule the packet transmission when the channel returns to a good state. For a rayleigh fading channel it is widely known that for a given threshold \( c \), and doppler spread \( f_D \), the expected fade duration is:

\[
AFD(c, f_D) = \frac{\exp \left( \frac{c}{\beta_{RMS}} \right)^2 - 1}{\sqrt{2\pi f_D \frac{c}{\beta_{RMS}}}}
\]  

(5)

Where \( \beta_{RMS} \) is the RMS value of the received signal.

For the implementation in this report, the CTS and ACK packet headers contained an additional 4 bytes. In the case of an estimated fade, the average fade duration (as in (5)) in microseconds is stored in this field. In the case of no estimated fade, this field contains 0’s.

As discussed in Section 2, the ns-2 simulation environment is a discrete event simulator that provides a approximate model of real network operation. In (5) it can be that the average fade duration is found using \( \beta_{RMS} \). The limitations in the simulation environment is derived from the biased information regarding the computation of the RMS value of the signal of the envelope. In discrete time, \( \beta_{RMS} \) can be computed
for \( L \) samples as:

\[
\hat{\beta}_{RMS} = \sqrt{\frac{1}{L} \sum_{i=0}^{L-1} \beta_i^2}
\] (6)

In a simulation environment (such as ns-2), received signal measurements can only consist of packets detected by a given node above the receive threshold. The result is that no received signal measurements occur below the receive threshold \( c \) and this presents a overestimation of the RMS value of the signal. The average fade duration is inaccurately estimated. To compensate for the lack of accuracy in the estimation of \( AFD \), this report utilizes an enhancement to the protocol that the authors in [1] proposed.

- Modification: In [1], the average fade duration is computed as in (5) and relayed to the transmitting node. Following a return from the fade back-off, the transmitting node resumes transmission as described in IEEE 802.11 DCF. In this report, we modify the behavior of this protocol following the back-off. After back-off, the transmitting node will transmit either an RTS packet (in the case of collision avoidance), or a DATA packet. If the packet times out, the transmitting node assumes the receiver is till in a fade and will again enter a fade back-off. The resultant scheme will provide compensation for the under-biased estimate of the fade duration. However in this scheme we provide no method for communicating the secondary fade back-off to other nodes in the network.

4 Markov Model for Rayleigh Fading Channels

We will now present the analytical model used by the authors for modelling the fading channel as a two-state Markov process [1]. This model is widely known in the literature as the Gilbert-Elliot model [12]. The purpose of such a model is to be able to determine the probabilities of which the fading signal is in a good or bad state. First off the PDF of the rayleigh distribution is:

\[
p(r) = \begin{cases} 
\frac{r}{\sigma^2} e^{-\frac{r^2}{2\sigma^2}} & (0 \leq r \leq \infty) \\
0 & (r < 0)
\end{cases}
\] (7)
with $\sigma$ as the RMS value of envelope before detection (as in [6]). We can also define the RMS value\(^1\) of the envelope as in [6]:

$$
\beta_{RMS} = \sqrt{E[r^2]} = \sqrt{\int_{-\infty}^{+\infty} r^2 p(r) dr} = \sqrt{\int_{0}^{+\infty} r^2 \frac{r}{\sigma^2} \exp\left(-\frac{r^2}{2\sigma^2}\right) dr} = \sqrt{2\sigma}
$$

(8)

In order to determine the probability of being in either a good or bad state, we need to first determine number of times the signal crosses the given threshold, $c$, in a upward direction. This is defined as the level crossing rate, and for a rayleigh fading process, is widely-known as:

$$
N_r(c, f_D) = \sqrt{2\pi f_D} \left(\frac{c}{\beta_{RMS}}\right) \exp\left(\left(\frac{c}{\beta_{RMS}}\right)^2\right)
$$

(9)

Or if we define $\rho$ as $\frac{c}{\beta_{RMS}}$, then the equation becomes:

$$
N_r(\rho, f_D) = \sqrt{2\pi f_D} \rho \exp\left(\rho^2\right)
$$

(10)

\(^1\)In a discrete event simulator such as ns-2, there are no suitable methods to determine this value in continuous time. As such it is approximated using the discrete-time RMS value of a signal as in (6)
We can then differentiate between the states based on our threshold, $c$. We define $S_1$ as the *good* state, where $r$ is above the threshold $c$ and $S_0$ as the *bad* state, where $r$ is below the threshold $c$. First, the CDF of a Rayleigh distribution can be found as [6]:

$$P_r(r \leq R) = \int_0^R p(r) \, dr = 1 - exp\left(-\frac{R^2}{2\sigma^2}\right)$$  \hspace{1cm} (11)$$

Using this result, we can compute the probability of the signal being below the threshold (we will call this $\pi_0 = Pr[S_0]$):

$$\pi_0 = Pr(0 \leq r \leq c)$$

$$= Pr(r \leq c)$$

$$= 1 - exp\left(-\frac{c^2}{2\sigma^2}\right)$$

$$= 1 - exp\left(-\left(\frac{c}{\sqrt{2}\sigma}\right)^2\right)$$

$$= 1 - exp(-\rho^2)$$  \hspace{1cm} (12)$$

And similarly $Pr[S_1]$ or $\pi_1$:

$$\pi_1 = Pr(r > c)$$

$$= 1 - Pr(r \leq c)$$

$$= 1 - (1 - exp(-\rho^2))$$

$$= exp(-\rho^2)$$  \hspace{1cm} (13)$$

We will then find the transition probabilities (as in [1]), $t_{i,j}$, where $i, j \in \{0, 1\}$. As stated before, the level crossing rate, $N_r$, defines how many times the signal crosses the threshold during a given time frame. This means that there is $N_r^{-1}$ seconds between each crossing (on average). We also found that the average time the signal is below the threshold from (5). Using these, $t_{1,0}$ (probability of transition from *good* to
bad state) is roughly:

\[
t_{1,0} = \frac{A FD}{N_r^{-1}} = A FD \times N_r = 1 - \exp(-\rho^2)
\]  

(14)

Using similar arguments:

\[
t_{0,1} = \frac{N_r^{-1} - A FD}{N_r^{-1}} = 1 - N_r \times A FD = 1 - t_{1,0}
\]  

(15)

From this it is quite clear that:

\[
\pi_0 t_{0,1} = \pi_1 t_{1,0}
\]  

(16)

Which the authors called the state equilibrium equation [1]. Effectively this means that the 2-state Markov model for the fading channel can be used.

5 Markov Model for IEEE 802.11 DCF

The IEEE 802.11 Distributive Coordination Function (DCF) defines the back-off procedure of a transmitting station. The back-off procedure is often modelled as a finite state Markov chain for analysis [13], [14]. In fig. 5 we reproduce this model to discuss the analysis in [1]. We will look at the following performance parameters for both the standard 802.11 MAC under fading and the proposed cross-layer enhancement:

- Probability of packet being transmitted
- Probability of receiving an erroneous packet
- Channel throughput
- Packet processing rate
- Packet loss
From fig. 5, it can be seen that the probability of a packet being transmitted as:

\[
\tau = \sum_{i=0}^{m} (i, 0) = \frac{(0, 0)(1 - p^{m+1})}{1 - p}
\]  

(17)

Where \( p \) is probability of a packet not being received correctly, \( m \) is the maximum number of back-offs and \((i,j)\) is a tuple random process that defines the 802.11 DCF back-off states.

5.1 Analysis of Standard 802.11 MAC Under Fading

We will now look at the analytical performance of the standard 802.11 MAC protocol for the listed performance parameters as presented by the authors in [1].
5.1.1 Probability of Packet Transmission and of Receiving an Erroneous Packet

Without the cross-layer improvement, a packet could be erroneous if either a collision occurs (more than one node transmitting at the same time), or due to low power.

\[ p = 1 - (1 - \tau)^{n-1} + \pi_0(1 - \tau)^{n-1} \]  \hspace{1cm} (18)

Where \( \pi_0 \) is the probability of being in a bad state as discussed in Section 4. We can also present this formulation in terms of the probability for packet transmission:

\[ \tau = 1 - \left( \frac{1 - p}{1 - \pi_0} \right)^{\frac{1}{n-1}} \]  \hspace{1cm} (19)

Using (19), along with (17) we can solve for both quantities (\( p \) and \( \tau \)).

5.1.2 Channel Throughput

The channel throughput for \( n \) nodes can be found as:

\[ U = \frac{n\tau(1 - \tau)^{n-1}P_L}{\bar{\sigma}} \]  \hspace{1cm} (20)

Where \( \bar{\sigma} \) is the average channel time slot (which is discussed in much detail in [14] and \( P_L \) is the length of a packet.

5.1.3 Transmission Processing Rate

The authors in [1] assume an exponential arrival rate \( \lambda \), which is often made in networking analysis. The assumption of fixed packet size is also made. Using these assumptions, a theoretical expression of the processing rate (rate at which packets are processed by the MAC from the queue and transmitted). Using the probability of packet transmission, this rate can be found using:

\[ \mu = \frac{\tau(1 - \tau)^{n-1}}{\bar{\sigma}} \]  \hspace{1cm} (21)

Where \( \tau(1 - \tau)^{n-1} \) is the probability of 1 node transmitting and \( n - 1 \) nodes not transmitting: no collision.
5.1.4 Packet Loss

The packet loss probability has two parts. The first is the losses due to contention in the network. The second will be due to queue overflow. We will first find the probability of contention loss as in [1]:

\[ P_C = p^m(0,0) \]  

(22)

To determine the packet loss due to queue overflow, we need to take out the the losses due to contention. Therefore the arrival rate of packets taking out the effect of contention losses becomes:

\[ \lambda_e = (1 - P_C)\lambda \]  

(23)

We will also define the arrival to service ratio:

\[ \rho = \frac{\lambda_e}{\mu} \]  

(24)

The service processing rate as in (21) is general (not poisson or deterministic). As \( \tau \) and \( \pi_1 \) are based on conditions of the system. The authors however assumed that \( \mu \) was also exponential therefore reducing the M/G/1/K analysis to M/M/1/K. Using this assumption, we can find the packet loss due to overflow (blocking probability) from [15] as:

\[ P_O = \frac{(1 - \rho)\rho^K}{1 - \rho^{K+1}} \]  

(25)

The sum of packet loss probabilities finds the total packet loss probability as:

\[ P_{LOSS} = P_C + P_O \]  

(26)

\[ P_{LOSS} = p^m(0,0) + \left( \frac{(1 - \rho)\rho^K}{1 - \rho^{K+1}} \right) \]  

(27)

5.1.5 Delay

The average queue length can also be found from M/M/1/K theory [15] as:

\[ L = \frac{\rho(1 - (K + 1)\rho^K + K\rho^{K+1})}{(1 - \rho)(1 - \rho^{K+1})} \]  

(28)
Resulting in an overall packet delay using (23) and (28) of:

\[ D = \frac{L}{\lambda_e} \]  

(29)

5.2 Analysis of Cross-Layer Enhancement Under Fading

We will now look at the performance of the proposed cross-layer enhancement [1].

5.2.1 Probability of Packet Transmission and of Receiving an Erroneous Packet

With the cross-layer improvement, ideally a packet can only be erroneous if a collision occurs:

\[ p = 1 - (1 - \tau)^{n-1} \]  

(30)

Or solving for the packet transmission probability:

\[ \tau = 1 - (1 - p)^{\frac{1}{n-1}} \]  

(31)

Like before, using (31), along with (17) we can solve for both quantities \( p \) and \( \tau \).

5.2.2 Channel Throughput

The channel throughput for \( n \) nodes can be found in a similar manner as:

\[ U = \frac{n\tau(1-\tau)^{n-1}P_L}{\bar{\sigma}} \pi_1 \]  

(32)

Where \( \bar{\sigma} \) is the average channel time slot from [14] and \( P_L \) is the length of a packet as before, and \( \pi_1 \) is the probability of channel being in a good state.

5.2.3 Transmission Processing Rate

As before, the transmission processing rate, \( \mu \) is:

\[ \mu = \frac{\tau(1-\tau)^{n-1}}{\bar{\sigma}} \]  

(33)
5.2.4 Packet Loss

As before, the packet loss is the sum of contention and overflow losses:

\[ P_C = p^n(0,0)\pi_1 \] (34)

Using similar arguments as in (25), the probability of overflow is:

\[ P_O = \pi_1(1 - \rho)\rho^K \] (35)

Resulting in a total packet loss probability of:

\[ P_{LOSS} = P_C + P_O \] (36)

\[ P_{LOSS} = \pi_1 \left(p^n(0,0) + \frac{(1 - \rho)\rho^K}{1 - \rho^{K+1}}\right) \] (37)

5.2.5 Delay

As before we can find the delay using the queue length and service rate [15] as:

\[ L = \frac{\rho(1 - (K + 1)\rho^K + K\rho^{K+1})}{(1 - \rho)(1 - \rho^{K+1})} \] (38)

Resulting in an overall packet delay using (23) and (38) of:

\[ D = \frac{L}{\lambda_c} \] (39)

6 Simulation

We will now present the simulation results as found by our implementation in ns-2. Simulation parameters were chosen to coincide with those used in [1]. The simulations utilized the RTS/CTS scheme and utilized a channel datarate of 1Mb/s. As ns-2 does not have native support for Rayleigh fading, the fading model from [9] was added. Other parameters used were 0.6W power consumption for packet transmission and 0.35W for reception.

This section is divided into several parts that parallel [1]. First in 6.1 we will take a look at the
accuracy of the prediction mechanism. We will then examine the performance improvement for a two node scenario in 6.2. In 6.3 and 6.4 we will look at the multi-node scenario for UDP and TCP connection respectively. We will also verify the analytical model with 4 nodes in 6.5. This simulations are performed in a similar manner to those in [1].

6.1 Prediction Accuracy

The purpose of this section is to determine the accuracy of the prediction. Bounds must be placed on where the estimation is valid. From an implementation point of view, this will help make a determination on where the prediction scheme should be used. We utilized similar parameters to the authors in [1], including a consistent packet size of 512 bytes. In this scenario, we have two nodes: A and B. Node A and node B start at a distance of 120 metres. Node A is to transmit at a constant packet rate while node B moves away from A at a rate of 1 metre/second (m/s). We present the results of the prediction for several different packet rates. We can see from the fig. 5(a)-5(c) the performance of the prediction algorithm. We can see from that at a packet rate of 5 packets/sec, this scheme is no longer valid. This result is also confirmed by the authors in [1]. The estimate at this rate is much lower than the actual value of the received signal. This result will present itself later in 6.3. In addition, since the speed of the node is directly proportional to the doppler shift as:

\[ f_D = \frac{v}{\lambda} = \frac{vf_C}{c} \]  \hspace{1cm} (40)

Where \( c \) is the speed of light in m/s, \( f_C \) is the carrier frequency of the signal and \( \lambda \) is the wavelength. The result is a change in velocity of a node has a similar effect on the prediction as modifying the packet rate. The result is an increase in node speed by a factor \( x \) has the same effect as a decrease in packet rate by the same factor \( x \). This means that both the mobile velocity and packet rate should be taken into account when determining whether to make a channel prediction.

6.2 Two Node Scenario

We will now examine the two node scenario as presented in [1]. The scenario configuration is as follows. We have two nodes, node A which will be transmitting to node B at a constant bitrate of 93440bps. Node B will also be moving away from node A at a velocity of 1m/s. We will analyze several metrics of the
simulation as a function of the initial distance between the nodes.

In fig. 6(a), we can see the comparison of throughput versus initial distance. As we can see the average throughput over the duration of the simulation is higher using the cross-layer enhancement. This is due to the fact that when returning from a fade, the back-off timer is less than that of it would have been with no fade back-off. This is evident when we examine the packet drops due to retry in fig. 6(c). We can also see the improvements in terms of reducing control packets and lowering energy consumption in fig. 6(b) and 6(d) respectively.
We will now look at the scenario with multiple nodes and transmissions using UDP\(^2\). The simulation parameters will be similar to those presented in [1]

- AODV routing protocol
- Constant packet size of 512 bytes
- Average of 20 connections

\(^2\)Random mobility of nodes and traffic generation in ns-2 is used in 6.3 and 6.4. It is important to note that there can be a wide range of results in terms of node position, movement and connectivity. As such we will draw comparisons of the protocol on the trends compared to [1]
The scenario will be simulated for packet arrival rates of 5, 10, 15, 20 and 25 packets/second. Looking at figs. 7(a)-7(c) we can see the performance improvements with the cross-layer enhancements in terms of throughput, energy consumption and drops due to retries. An important aspect of the protocol can be seen in fig. 7(a). As discussed in 6.1, we saw that an arrival rate of 5 packets/sec resulted in an underestimation of the received signal. This result causes unneeded back-offs resulting in lower throughput using this arrival rate.
6.4 50 Node - TCP

Now we will examine the performance of the protocol under TCP. The performance will be compared for a variety of connections (from 5 to 25). The simulations using TCP were consistent with previous results

![Figure 8: Performance Metrics for 50 Node TCP](image)

(a) Packet Drops due to Retries vs. Number of Connections  
(b) Control Packets vs. Number of Connections

in terms of most performance metrics (from figs. 8(a) and 8(b)). However during simulations of TCP, we found that the inconsistency of the packet rate resulted in a large variation in terms of throughput. The throughput comparison for TCP had large bearing on the position of nodes. The result of which was instability in throughput of our protocol under TCP with the given conditions.

6.5 Verification of Model

We will also present the verification as presented in [1]. The scenario involves four nodes (2 pair). They are placed in a circular formation where the circle has a radius of 50m. The nodes are \( \frac{\pi}{2} \) radians away from each other. A given node transmits to the node directly opposed. Each node moves away from the center of the circle at 0.5m/s. Packet size is 1000 bytes and the simulation will be ran for 50 second. We will compare the average delay and network throughput for this simulation versus the packet arrival rate. The theoretical results are calculated using our analytical model and results from [14]. Examining the results in figs 9(a) and 9(b), we can see the accuracy of the analytical model. The throughput of the channel offers a nearly perfect modelling of the simulation results, however the delay is underestimated. The results in fig. 9(b) can be a result of the approximations made with regards to the queueing model.
In this report we have examined the proposed MAC layer enhancement for the IEEE 802.11 DCF protocol under fading conditions. We have shown that the proposed enhancement can offer some improvements if channel state information is shared with the MAC layer. This particular implementation poses several implementation issues that were not addressed in [1]. We have presented these issues throughout this report by will summarize the issues:

- The estimation of the average fade duration (AFD) relies on the ability to accurately measure the RMS value of the received envelope over a period of time. However in a packet network, for a receiving node to determine if the received signal is from the station that is sending it a transmission, the packet must be received correctly. The result is that only samples can be taken above the receiver signal threshold and this will provide a biased fade estimate.

- The assumption of a fixed packet rate in a contention based network is not reasonably valid. However with some accuracy received samples can be made to fit the required data-spacing.

In summary we can show that cross-layer protocols offer promising improvement for network performance.
References


A Errata

In this section we will briefly summarize some of the errors found in [1] for the purpose of assisting individuals planning to review [1].

1. In equation (9) of [1] the level crossing rate is given should be given by:

\[ N_r = \sqrt{2\pi f_D \rho} \exp(\rho^2) \]

2. In the derivation of queue length, the blocking probability \( P_{qlc}/P_{qlc} \) is already taken into account in the expression for average queue length (\( L \)).

3. For calculation of delay, the queueing model uses was \( M/M/1/Q_l \), however the service rate \( \mu_{nc}/\mu_c \) is not exponential.

4. For an \( M/M/1 \) analysis, the expression for \( L_c \) should be:

\[ L_c = \frac{\eta_c(1 - (Q_l + 1)\eta_c^{Q_l} + Q_l\eta_c^{Q_l+1})}{(1 - \eta_c)(1 - \eta_c^{Q_l+1})} \]

A similar expression can be found for \( L_{nc} \).