

# Packet Length Adaptation in WLANs with Hidden Nodes and Time-Varying Channels

Michael N. Krishnan, Ehsan Haghani, and Avideh Zakhor

Department of EECS, U.C. Berkeley

Email: {mkrishna, haghani, avz}@eecs.berkeley.edu

**Abstract**—In a wireless local area network (LAN), packets can be lost due to a multitude of reasons. It is possible to reduce the probability of occurrence of some of these loss mechanisms by reducing packet length at the medium access control (MAC) layer. However, there is an inherent tradeoff in that shorter packets decrease efficiency with respect to overhead. In current packet length adaptation literature, simplified or incomplete packet loss models are used, neglecting channel fading or collisions due to hidden nodes. In this paper, we apply a more complete packet loss model and propose a local packet length adaptation algorithm whereby each node dynamically adjusts its packet length based on estimates of the probabilities of each significant type of packet loss. In our technique, the access point periodically broadcasts channel occupancy information which each node uses in conjunction with its own local observations in order to estimate current network conditions. These are used to estimate the derivative of throughput with respect to packet length at each node under the current network conditions and to adapt the packet lengths accordingly. We demonstrate throughput gains of up to 20% via NS-2 simulations.

## I. INTRODUCTION AND RELATED WORK

802.11 wireless LANs were originally designed for small networks with limited traffic, and are thus not optimized for high traffic situations. However, as wireless LANs become increasingly ubiquitous, the design limitations become greatly stressed. One often neglected tunable parameter is MAC layer packet length. While packet length can be variable in the 802.11 standard, it is most often simply set to the maximum value to reduce the impact of overhead. This is indeed the optimal setting for a scenario with a single pair of nodes with a strong channel; however, in scenarios with hidden nodes and weaker channels, shorter packets may be preferable due to their lower susceptibility to loss.

A hidden node for a given node A transmitting to a node B in a wireless network is one which is capable of interfering with the reception of the packet at B, while being unable to sense the transmission of A. This can lead to *staggered collisions*, which occur when node A's packet is interrupted by the hidden node, causing reception to fail. The risk of this type of loss increases with packet length because as length increases, the hidden node is required to remain silent for a longer period of time. Additionally, longer packets are more susceptible to loss due to channel errors because they require the successful decoding of more symbols. Packet length adaptation seeks to address the tradeoff between lower overhead for long packet lengths, and lower probability of loss for shorter packet lengths.

There has been a significant amount of research on packet length adaptation. In current packet length adaptation literature, a simple packet loss model is typically used, assuming the

channel to have a constant bit-error rate (BER), and neglecting staggered collisions [1, 2]. This assumes that most packet losses occur due to random bit errors in the packet payload. However, it has experimentally been shown that for lower modulation rates in 802.11a, most packet losses occur due to failure to synchronize to the packet preamble [3]. This type of loss cannot be accounted for using a constant BER model, as it requires a model of channel fading. In [4], Zheng and Nelson use a fading model, but only assume channel coherence over symbols rather than packets and omit staggered collisions. Staggered collisions have also been studied in the absence of channel errors [5–7].

One barrier to designing a more sophisticated adaptation algorithm is that in a network with multiple causes of packet loss, a node must be able to determine the the proportion of each type of packet loss it experiences in order to choose the optimal packet length. In [7] and [8], algorithms are proposed based on loss statistics in which nodes attempt to use different packet lengths to test whether performance increases or decreases. By avoiding modeling loss mechanisms, they sacrifice convergence speed. In [9], optimal packet length for a wireless sensor network is chosen based on a priori knowledge of network conditions, which cannot adapt to changing network conditions.

In [10], we propose a method by which nodes in an 802.11 network with hidden terminals can locally estimate their collision probabilities based on shared information about channel occupancy by the access points (APs). In this paper, we use a packet loss model which includes random channel fading across packets as well as direct and staggered collisions to analyze the impact of MAC layer packet length on throughput. We propose a local packet length adaptation algorithm whereby each node estimates its current probability of each type of packet loss, based on [10], in order to compute the derivative of throughput with respect to packet length, and to adapt accordingly.

The remainder of the paper is organized as follows: Section II describes the packet loss model; mathematical analysis of throughput and its derivative with respect to packet length is included in Section III; the adaptation algorithm is described in Section IV; simulation results are presented in Section V; the paper is concluded in Section VI.

## II. PACKET LOSS MODEL

Losses in Wireless LANs can be broadly classified into two types: collisions, which are the result of unfavorable traffic conditions, and channel errors, which are the result of unfa-

avorable channel conditions. A collision occurs when a node's packet overlaps in time with that of another node which is spatially close enough to the destination to interfere. A channel error occurs when the SNR of a received packet is low due to a large path loss or a deep multipath fade. The total packet loss probability  $P_L$  can be computed as

$$P_L = 1 - (1 - P_C)(1 - P_e) \quad (1)$$

where  $P_C$  is the probability of collision, and  $P_e$  is the probability of channel error, which is assumed to be independent of  $P_C$ . In this analysis, we assume that all collided packets are lost, not captured, and that the probability of ACK loss is negligible compared to other losses.

There are three primary types of collisions. These include direct collisions and two types of staggered collisions. A *staggered collision of type 1 (SC1)* for a given node is one in which the node under consideration transmits first, and is then interrupted by another node. A *staggered collision of type 2 (SC2)* for a given node is one in which the node under consideration interrupts the transmission of a hidden node. This distinction is necessary because each of these two types of staggered collisions have a different cause, and as a result they must be estimated and adapted to in different ways.

For channel errors, we consider a sequence of  $k$  bits being sent at a constant modulation rate  $R$  over a channel with SNR  $\sigma$ . The bit-error rate, as a function of  $R$  and  $\sigma$ , is denoted by  $BER_R(\sigma)$ , and the probability of successfully transmitting all  $k$  bits is given by  $(1 - BER_R(\sigma))^k$ . Since an 802.11 packet consists of a preamble and PLCP header sent at low modulation rate, and a payload possibly sent at a higher rate, the probability of channel error for a single packet experiencing an SNR of  $\sigma$  can be computed as

$$\begin{aligned} P_e^p(\sigma) &= 1 - (1 - P_{e,h}^p(\sigma))(1 - P_{e,p}^p(\sigma)) \\ &= 1 - (1 - BER_{R_h}(\sigma))^{L_h} (1 - BER_{R_p}(\sigma))^L \end{aligned} \quad (2)$$

where  $P_{e,h}^p(\sigma)$  and  $P_{e,p}^p(\sigma)$  are the header and payload error probabilities respectively,  $L_h$  and  $L$  are the lengths of the header and payload respectively,  $R_h$  and  $R_p$  are the modulation rates of the header and payload respectively, and  $BER_R(\cdot)$  is assumed to be a known function, which depends on the the signal constellation for each rate. In this paper, we fix  $L_h$ ,  $R_h$ , and  $R_p$ , and adapt  $L$ .

Since the SNR is unknown and varies between packets, it is modeled as a random variable. Thus the probability of packet error over all packets,  $P_e$  is the expectation of the expression in Equation (2) taken over the distribution of SNR, denoted by  $f_\sigma(s)$ :

$$P_e = 1 - \int (1 - BER_{R_h}(s))^{L_h} (1 - BER_{R_p}(s))^L f_\sigma(s) ds \quad (3)$$

For the simulations in this paper, we assume SNR to have a log-normal distribution, where the mean is dependent on the path loss, and the variance is dependent on the type of environment [11].

It is important to discuss the impact of using this channel

error model compared to the commonly used constant BER model. A constant BER model tends to overestimate the impact of packet length on loss probability because it ignores the impact of the distribution of SNR. To demonstrate this, we plot theoretical throughput as a function of packet length for the two different loss models with the same average SNR of 9dB in Figure 1. Figure 1(a) assumes a constant BER, and Figure 1(b) assumes SNR to have a log-normal distribution. The shapes of the curves are noticeably different. This is because in the case where SNR is modeled probabilistically, the actual value of SNR has a much higher impact on an individual packet's successful transmission than the packet's length. While a constant BER model might suggest using a packet length of only 400 bytes, a more accurate model including SNR distribution shows that maximum packet length would be superior.

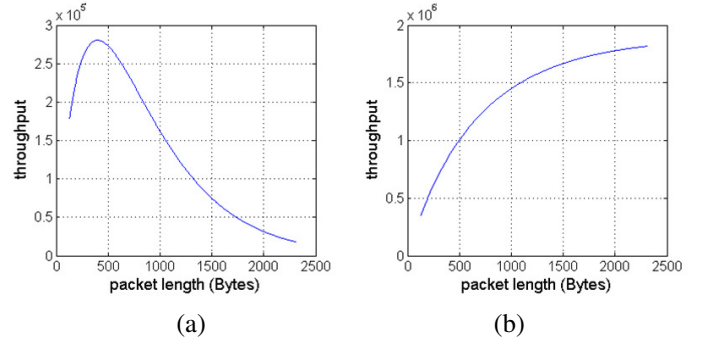


Fig. 1. Throughput vs packet length assuming no collisions for (a) SNR fixed at 9dB, and (b) SNR with mean of 9dB with standard deviation of 3dB.

### III. ANALYTICAL THROUGHPUT COMPUTATION

The throughput,  $TP$ , of a wireless node can be modeled as

$$TP = C \cdot L \cdot sendFreq \cdot (1 - P_L) \quad (4)$$

where  $C$  is a constant,  $L$  is the length of the packet payload in bits,  $sendFreq$  is the number of packets sent per second, and  $P_L$  is the overall packet loss probability.  $sendFreq$  can be expanded into

$$sendFreq = \frac{1}{\overline{W}(P_L) + \overline{T}_B + T_{ov} + L/R} \quad (5)$$

where  $\overline{W}(P_L)$  is the average amount of time spent in exponential backoff per packet, which depends on  $P_L$ ;  $\overline{T}_B$  is the average amount of time the channel is busy at the station;  $T_{ov}$  is the amount of per-packet overhead including headers, interframe spacing, and ACK time, and  $R$  is the modulation rate.  $(1 - P_L)$  can be expanded into

$$(1 - P_L) = (1 - P_{SC2}) \cdot (1 - P_{DC}) \cdot (1 - P_{SC1}) \cdot (1 - P_e) \quad (6)$$

where  $P_{SC2}$ ,  $P_{DC}$ ,  $P_{SC1}$ , and  $P_e$  are the probabilities of staggered collision of type 2, direct collision given no staggered collision of type 2, staggered collision of type 1 given no staggered collision of type 2 or direct collision, and channel error given no collision, respectively.

Since we are only adapting the packet length, the terms for  $P_{SC2}$  and  $P_{DC}$  are unimportant, and can be lumped into the

constant in Equation (4), because they do not depend on packet length to first order. Specifically, they only depend on the probability the channel is busy prior to or at the same time as a transmission start, regardless of the length of the transmission. We can thus reduce Equation (4) to the product of five factors, four of which depend on  $L$ :

$$TP = C' \cdot L \cdot \text{sendFreq} \cdot (1 - P_{SC1}) \cdot (1 - P_e) \quad (7)$$

where  $\text{sendFreq}$  is computed as in Equation (5), and  $P_e$  is computed as in Equation (3). Many of the variables in these expressions are unknown and vary with time, particularly as other nodes adapt their packet lengths. Thus it is difficult to determine a single value of  $L$  to maximize this function analytically. However, it is possible to estimate the derivative of  $TP$  with respect to  $L$  for the current network conditions given the same quantities computed in [10] for estimating collision probabilities.

For a function  $F(x) = \prod_{i=1}^n f_i(x)$ , the derivative with respect to  $x$  can be computed as:

$$F'(x) = \sum_{i=1}^n \left( f_i'(x) \prod_{j \neq i} f_j(x) \right) = F(x) \sum_{i=1}^n \frac{f_i'(x)}{f_i(x)} \quad (8)$$

Therefore, to compute the derivative of the throughput with respect to  $L$ , we must compute the derivatives of each of the four terms in the product in Equation (7), and substitute them into the following expression:

$$\frac{\partial}{\partial L} TP = TP \cdot \left( \frac{1}{L} + \frac{\frac{\partial}{\partial L} \text{sendFreq}}{\text{sendFreq}} + \frac{\frac{\partial}{\partial L} (1 - P_{SC1})}{1 - P_{SC1}} + \frac{\frac{\partial}{\partial L} (1 - P_e)}{1 - P_e} \right). \quad (9)$$

The first term is easily computed because  $L$  is a known quantity, and its derivative is simply 1. In the second term,  $\text{sendFreq}$  can be estimated empirically at each node by counting the number of packets transmitted over a fixed period of time. Its derivative can be computed as

$$\begin{aligned} \frac{\partial}{\partial L} \text{sendFreq} &= -\frac{\frac{\partial}{\partial L} \overline{W}(P_L) + \frac{1}{R}}{(\overline{W}(P_L) + T_B + T_{ov} + L/R)^2} \\ &= -\text{sendFreq}^2 \cdot \left( \frac{\partial}{\partial L} \overline{W}(P_L) + \frac{1}{R} \right). \end{aligned} \quad (10)$$

where  $\frac{\partial}{\partial L} \overline{W}(P_L)$  can be computed from the chain rule as

$$\frac{\partial}{\partial L} \overline{W}(P_L) = \frac{\partial}{\partial P_L} \overline{W} \cdot \frac{\partial}{\partial L} P_L. \quad (11)$$

$\overline{W}$  is a deterministic function of the number of attempts, which is in turn a function of  $P_L$ . It is straightforward to empirically estimate this function  $\overline{W}(P_L)$  as well as its derivative  $\frac{\partial}{\partial P_L} \overline{W}$ , which we call  $p(P_L)$ . An empirical count of  $P_L$  can be plugged into  $p(P_L)$  to obtain an estimate of  $\frac{\partial}{\partial P_L} \overline{W}$ .

For the second factor in the right-hand side of Equation (11), we also have

$$\begin{aligned} \frac{\partial}{\partial L} P_L &= -\frac{\partial}{\partial L} (1 - P_L) \\ &= -(1 - P_L) \cdot \left( \frac{\frac{\partial}{\partial L} (1 - P_{SC1})}{1 - P_{SC1}} + \frac{\frac{\partial}{\partial L} (1 - P_e)}{1 - P_e} \right) \end{aligned} \quad (12)$$

where the second equality comes from applying Equation (8) to Equation (6) and noting that  $P_{SC2}$  and  $P_{DC}$  are independent of  $L$ . The two terms in the sum are the same as the last two terms in Equation (9), so it is possible to substitute Equations (10), (11), and (12) into (9) and re-arrange to obtain

$$\begin{aligned} \frac{\partial}{\partial L} TP &= TP \cdot \left[ \frac{1}{L} + \frac{\text{sendFreq}}{R} \right. \\ &\quad \left. + (\text{sendFreq} \cdot p(P_L) \cdot (1 - P_L) + 1) \right. \\ &\quad \left. \cdot \left( \frac{\frac{\partial}{\partial L} (1 - P_{SC1})}{1 - P_{SC1}} + \frac{\frac{\partial}{\partial L} (1 - P_e)}{1 - P_e} \right) \right] \end{aligned} \quad (13)$$

Now all that remains is to estimate the last two terms. The second to last term can also be computed using the chain rule:

$$\frac{\partial}{\partial L} (1 - P_{SC1}) = \frac{\partial}{\partial l} (1 - P_{SC1}) \cdot \frac{\partial}{\partial L} l \quad (14)$$

where  $l = L/R$  is the length of the packet in seconds. It is shown in [10] that  $\frac{\partial}{\partial l} (1 - P_{SC1})$  can be approximated by a quantity denoted by  $m_2$  in [10], and which can be estimated as part of the collision probability estimation process. Thus Equation (14) can be reduced to

$$\frac{\partial}{\partial L} (1 - P_{SC1}) = -\frac{m_2}{R}. \quad (15)$$

To obtain the final term in Equation (13), we must take the derivative of Equation (3) with respect to  $L$ . This results in:

$$\begin{aligned} \frac{\partial}{\partial L} (1 - P_e) &= \int (1 - P_{e,h}^p(s)) \frac{\partial}{\partial L} (1 - \text{BER}_{R_p}(s))^L f_\sigma(s) ds \\ &= \int (1 - P_e^p(s)) \ln(1 - \text{BER}_{R_p}(s)) f_\sigma(s) ds \\ &\approx -\int (1 - P_e^p(s)) \text{BER}_{R_p}(s) f_\sigma(s) ds. \end{aligned} \quad (16)$$

The first equality comes from switching the derivative into the integral and noting that  $P_{e,h}^p(s)$  does not depend on  $L$ ; the second equality comes from taking the derivative of the exponential term; the final approximation comes from a Taylor series expansion of  $\ln(1 - x)$  for  $x$  close to zero. This can be justified for large  $s$ , because BER is low for high SNR. Even though the Taylor approximation becomes less accurate as  $s$  decreases, the terms in the integral for smaller  $s$  have lower weight because as  $s$  decreases,  $(1 - P_e^p(s)) \rightarrow 0$ . Therefore the approximation in Equation (16) is reasonable for all values of SNR. The final expression of Equation (16) depends only on the distribution of SNR. The node can thus estimate  $\frac{\partial}{\partial L} (1 - P_e)$  using only an estimate of the current SNR distribution.

If the SNR distribution  $f_\sigma(s)$  is a single-parameter distribution such as Rayleigh or a log-normal with known variance, then it can be estimated by each node, based on the estimate of  $P_e$ , as described in [10], and via a lookup table based on Equation (3). If it is a two-parameter distribution, it requires observation of another variable dependent on the SNR distribution, such as a received signal strength indication (RSSI). In the simulations of this paper we assume a log-normal distribution with a known variance. This variance depends on the type of environment and is assumed to be either pre-set by the deployer or learned from the variation in signal strength of beacon packets.

$P_e$  is estimated via Equation (1) where  $P_L$  is estimated via empirical counting and  $P_C$  is estimated via the technique in [10]. A lookup table generated from Equation (3) is then used to estimate  $f_\sigma(s)$ , which is used to estimate  $\frac{\partial}{\partial L}(1 - P_e)$  via Equation (16).

Substituting Equation (15) into Equation (13), we get a final expression:

$$\begin{aligned} \frac{\partial}{\partial L}TP &= TP \cdot \left[ \frac{1}{L} + \frac{sendFreq}{R} \right. \\ &\quad \left. + (sendFreq \cdot p(P_L) \cdot (1 - P_L) + 1) \right. \\ &\quad \left. \cdot \left( -\frac{m_2}{R \cdot (1 - P_{SC1})} + \frac{\frac{\partial}{\partial L}(1 - P_e)}{1 - P_e} \right) \right] \end{aligned} \quad (17)$$

where  $L$  is known,  $sendFreq$  and  $P_L$  are estimated via empirical counting,  $m_2$  is estimated via the technique in [10],  $P_e$  is computed from Equation (1) where  $P_C$  is estimated via the technique in [10], and  $\frac{\partial}{\partial L}(1 - P_e)$  is computed via lookup table based on  $P_e$ .

#### IV. ADAPTATION ALGORITHM

We assume an 802.11 network in infrastructure mode with multiple APs, each with several associated nodes, similar to the topology in [10]. As in [10], APs periodically broadcast medium occupancy statistics to all associated nodes, which the nodes use to estimate their probabilities of each type of loss and adapt their packet lengths. Specifically, nodes transmit at a constant packet length  $L$  for 5 seconds, and over this period the nodes and APs collect their medium occupancy statistics. The AP broadcasts its statistics to the nodes, which then use this information along with local observations to estimate  $sendFreq$ ,  $P_L$ ,  $P_C$ ,  $P_e$  and  $\frac{\partial}{\partial L}TP$ . The nodes then choose the new packet length  $L_{next}$  to be used for the next 5 second period to be

$$L_{next} = L + \alpha \cdot \frac{\partial}{\partial L}TP \quad (18)$$

where  $\alpha$  is a step size factor updated according to the following schedule:

- if  $sign(\frac{\partial}{\partial L}TP)$  changed, decrease  $\alpha$  by factor of  $\lambda$
- if  $sign(\frac{\partial}{\partial L}TP)$  not changed, increase  $\alpha$  by factor of  $\gamma$

By using this schedule, nodes use a large coefficient to quickly adjust packet length into the appropriate neighborhood, and then decrease the step size to converge to a precise final packet length. Appropriate choices of  $\lambda$  and  $\gamma$  can increase convergence rate and avoid oscillation. There is an additional caveat that if  $L$  is equal to  $L_{min}$  or  $L_{max}$ ,  $\alpha$  should not change. This is because, for example, when  $L = L_{max}$  and the  $\frac{\partial}{\partial L}TP$  is positive, the node cannot increase  $L$  to the point where the derivative's sign changes. Thus, as long as no other nodes change their behavior,  $\frac{\partial}{\partial L}TP$  continues to be positive at every iteration, causing  $\alpha$  to grow exponentially. With such a large value of  $\alpha$ , the node would no longer be able to appropriately adapt to future changes in network conditions; in particular, if the optimal packet length were to change, the node would have excessive oscillations until  $\alpha$  decreased to a reasonable value. We additionally set a maximum step size  $max\_step$ , and require  $L$  to remain between  $L_{min}$  and  $L_{max}$ .

---

#### Algorithm 1 The packet length adaptation algorithm

---

```

1: loop
2:   Transmit at current rate for 5 seconds
3:   Observe sendFreq and  $P_L$  over last 5 seconds
4:   Compute estimates of  $P_C$  and  $m_2$  via [10]
5:   Estimate  $P_e$  from  $P_L$  and  $P_C$  via Equation (1)
6:   Compute estimate of  $\frac{\partial}{\partial L}TP$  via Equation (17)
7:   if  $sign(\frac{\partial}{\partial L}TP) \neq last\_dir$  then
8:      $\alpha \leftarrow \alpha / \lambda$ 
9:      $last\_dir \leftarrow sign(\frac{\partial}{\partial L}TP)$ 
10:  else if  $L \in (L_{min}, L_{max})$  then
11:     $\alpha \leftarrow \alpha \cdot \gamma$ 
12:  end if
13:   $\Delta L \leftarrow sign(\frac{\partial}{\partial L}TP) \cdot \min(|\frac{\partial}{\partial L}TP| \cdot \alpha, max\_step)$ 
14:   $L \leftarrow median(L_{min}, L + \Delta L, L_{max})$ 
15: end loop

```

---

#### V. SIMULATION RESULTS

To test the throughput gains achievable using our length adaptation algorithm, we use the NS-2 simulation package. We have made modifications to allow all the nodes to compute their collision probabilities as in [10] as well as to execute the adaptation algorithm. The results in this paper are for 802.11b, but they can easily be extended to 802.11a or g, or to any other carrier-sense multiple access MAC with multiple available modulation rates. The test topology consists of 7 APs arranged to cover hexagonal cells, and 50 nodes placed at random over the area by a spatial poisson process sending saturated traffic to the nearest AP. SNR standard deviation is assumed to be 7dB, which is standard for an office environment with hard partitions [12].

To verify the accuracy of our derivative estimation, we fix the packet length of all but one node in the network, and vary the packet length of the single node in 2000 bit increments for the single node and measure total throughput over 4 minutes of simulation time for each packet length, thus obtaining a throughput vs. packet length curve. We repeat this for 16 different nodes in each of 10 different topologies. A plot for a representative node is shown in Figure 2 as the blue curve. From our simulation data, we also compute the  $\frac{\partial}{\partial L}TP$  estimates for each scenario and include it in the same figure. The estimates are shown as the red curve, while the green curve is computed as the average slope over three consecutive data points in the blue curve. The important feature to note in this figure is the point at which the curves for  $\frac{\partial}{\partial L}TP$  cross zero, because the algorithm converges to this packet length. Even though the estimate of the derivative is inaccurate for small packet lengths, it is quite accurate around the optimal throughput. In this example, the maximum packet length is not optimal. However, there are many cases in which maximum packet length is optimal. In particular, for nodes closer to their APs at lower noise levels, probabilities of channel errors and staggered collisions are both low, making longer packet lengths preferable. In all cases where throughput strictly increases with length, the algorithm converges to maximum packet length.

Having verified that the algorithm works for single-node

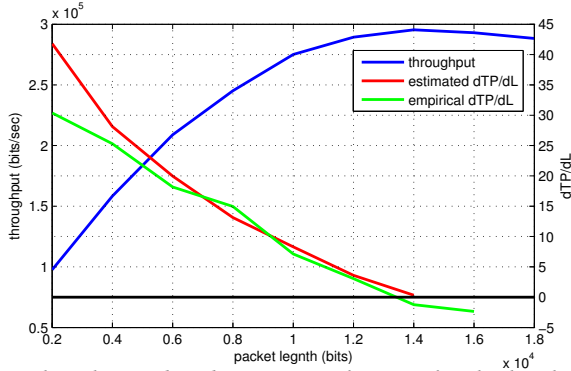


Fig. 2. Throughput and its derivative as a function of packet length, for a single node with all other nodes sending at constant packet length.

adaptation, we now allow all nodes to adapt simultaneously. Here the dynamics of our algorithm come into play. We have empirically found that initializing  $\alpha$  to 2000, and setting  $\lambda = 2$ ,  $\gamma = 1.25$ , and  $max\_step = 4000$  allows nodes to converge rapidly. We run the algorithm on all nodes in 7 different random topologies, each at 5 different level of ambient noise for a total of 35 simulations.

Figure 3 shows the results of each of the 35 simulations. Each symbol represents one topology at one noise level. The shape and color represent the noise level. The  $x$  position represents the number of nodes selecting non-maximal packet lengths for that scenario, and the  $y$  position represents the total throughput gain of the network compared to the situation where all nodes send at maximum packet length. It can be seen that as ambient noise increases, the number of nodes choosing shorter packet length increases. This is consistent with intuition, as higher noise leads to higher loss probability, which can be combatted by choosing shorter packet lengths. It is also observed that as noise power increases, throughput gain increases. This is because as noise and the packet loss rate increase, maximum packet length becomes increasingly sub-optimal for an increasing number of nodes. At lower noise powers, our algorithm results in most nodes choosing maximum packet length, which is optimal. Thus the total throughput is very close to when packet length is fixed at maximum for all nodes.

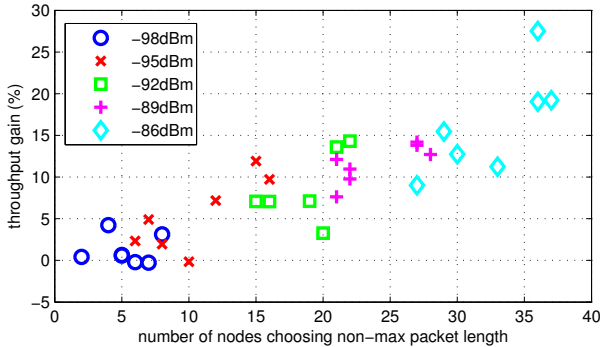


Fig. 3. Throughput gain and number of nodes choosing non-max packet length for 7 different topologies, each at 5 different noise powers.

Figure 4 shows the converged packet lengths of various nodes for one scenario with a noise power of  $-95\text{dBm}$ . The black squares represent the location of the APs, and the circles

represent the locations of the nodes. The color of the center of the circle indicates the final converged packet length after 5 minutes of simulation time. Dark red is maximum packet length, while blue and green represent shorter packet lengths. The border of the circle represents the average packet length over the simulation time. Significant difference between this color and the color of the center of the circle, this indicates slower convergence, i.e. 10-15 iterations. Depending on their specific locations, some nodes quickly converge to maximum packet length while other nodes converge to other packet lengths. As seen, 68% of the nodes converge to maximum packet length in 1-2 iterations. The nodes which converge to shorter packet lengths tend to be closer to the borders of the topology. This is because these nodes suffer the most from the hidden node problem as well as weak signal to the AP, and hence have higher packet loss rates, thus making it more beneficial to select shorter packet lengths.

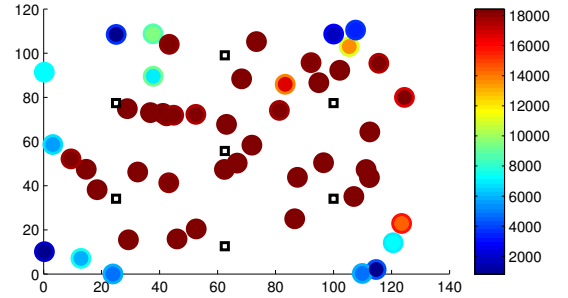


Fig. 4. Spatial plot of adapted packet length for each node in a scenario with  $-95\text{dBm}$  noise.

Figure 5(a) shows the converged packet lengths for the same topology for a higher noise power of  $-89\text{dBm}$ . As seen, more nodes choose non-maximal packet lengths in the higher noise scenario, with only 46% choosing near-maximum packet length. This is because at the higher noise power, packet loss becomes a greater concern than overhead for a greater proportion of the nodes. Figure 5(b) shows the percent throughput gain compared to the scenario using fixed maximum packet length for each node. Black squares represent the locations of the APs. The center of each circle corresponds to the location of each node, which is the same as in Figure 5(a). A node is colored green if it gains throughput and red if it loses. The size of the circle is proportional to the percent throughput change. As seen, some of the nodes choosing short packet lengths experience tremendous throughput improvements, while others have more moderate gains or even losses. Nodes which select maximum packet lengths are also affected by the changed packet lengths of their neighboring nodes, because they acquire the channel more frequently and experience fewer collisions with the nodes sending shorter packets. This coupling behavior between neighboring nodes could potentially explain the fact that gains and losses tend to be localized to certain areas of the network. For example, the nodes in the lower-right of this topology experience large gains, while the nodes in the middle experience moderate losses. Figure 5(c) shows the absolute throughput of each node in the network. There are two squares plotted for each node.

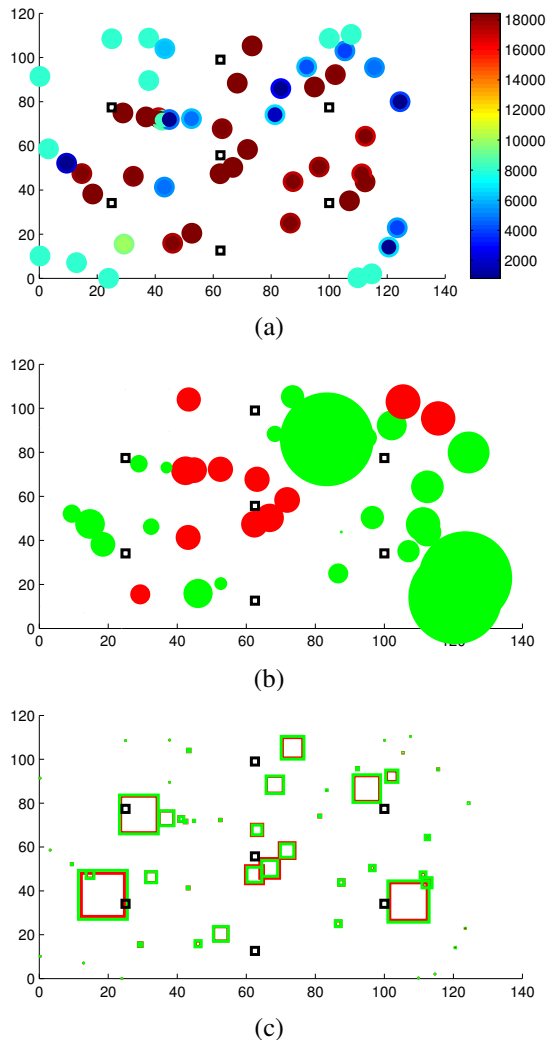


Fig. 5. A spatial plot of (a) converged packet length, (b) percent improvement over non-adaptive packet length, and (c) absolute throughput for a scenario with  $-89\text{dBm}$  noise.

The size of the red square is proportional to the throughput of the node when all nodes use maximum packet length, and the size of the green square is proportional to the throughput of the node when all nodes adapt. As seen, the 5 nodes with the greatest throughputs all have moderate throughput improvements. The 3 nodes with the largest percentage gains seen in Figure 5(b) are all in the bottom 40% in terms of overall throughput.

## VI. CONCLUSIONS AND FUTURE WORK

In this paper we have presented a detailed packet loss model for 802.11 networks, including channel fading and staggered collisions. We derive analytical expressions for throughput and its derivative as a function of packet length, and show that it can be estimated using the loss estimation techniques in [10]. We have proposed a packet length adaptation algorithm, and shown that it can achieve up to 25% throughput gains.

In our method, packet length is not adapted rapidly enough to track fast timescale channel changes; rather it is set over a

long period of time, i.e. 5 seconds, to achieve the best overall performance for the distribution of SNRs that are likely to occur over that time period. This will likely have interactions with adapted modulation rate at a fast timescale, but these are not taken into account in this paper. Although packet length adaptation cannot track these rapid channel changes, intelligent packet length selection based on an estimation of the channel distribution can yield gains with respect to using maximum packet length.

Future work includes development of a multi-parameter optimization to improve throughput using the  $P_C$  estimates, by adapting other parameters such as contention window, forward error correction, and transmit power, and combining with adaptation of other MAC parameters. Hardware implementation and experimentation using an open-source wireless card driver or a software radio platform are also important to verify the validity of our approach [13].

## REFERENCES

- [1] Jun Yin, Xiaodong Wang, and Dharma P. Agrawal, "Optimal Packet Size in Error-prone Channel for IEEE 802.11 Distributed Coordination Function", in *Proc. of IEEE WCNC 2004*, Atlanta, Georgia, USA, March 2004.
- [2] Xin He, Frank Y. Li, and Jiaru Lin, "Link Adaptation with Combined Optimal Frame Size and Rate Selection in Error-Prone 802.11n Networks", in *Proc. of IEEE ISWCS 2008*, Reykjavik, Iceland, October, 2008.
- [3] Amit K. Vyas, Fouad A. Tobagi, and Rajesh Narayanan, "Characterization of an IEEE 802.11a Receiver using Measurements in an Indoor Environment", in *Proc. of IEEE GLOBECOM 2006*, San Francisco, California, November 2006.
- [4] Feng Zheng and John Nelson, "Adaptive Design for the Packet Length of IEEE 802.11n Networks", in *Proc. of IEEE ICC 2008*, Beijing, China, May 2008.
- [5] Gerogi Atanasov Naydenov and Petko Stoyanov Stoyanov, "Bit Error Period Determination and Optimal Frame Length Prediction for a Noisy Communication Channel", in *AU J.T. 11(1): 7-13*, July 2007.
- [6] Dae Gil Yoon, Soo Young Shin, Wook Hyun Kwon, and Hong Seong Park, "Packet Error Rate Analysis of IEEE 802.11b under IEEE 802.15.4 Interference", in *Proc. of IEEE VTC 2006*, Melbourne, Australia, May 2006.
- [7] Wei Song, Michael N. Krishnan, and Avidesh Zakhor, "Adaptive Packetization for Error-Prone Transmission over 802.11 WLANs with Hidden Terminals", in *Proc. of IEEE MMSP 2009*, Rio De Janeiro, Brazil, October 2009.
- [8] Wei Dong, Xue Liu, Chun Chen, Yuan He, Gong Chen, Yunhao Liu, and Jiajun Bu, "DLPC: Dynamic Packet Length Control in Wireless Sensor Networks", in *Proc. of IEEE INFOCOM 2010*, San Diego, CA, March 2010.
- [9] Mehmet C. Vuran and Ian F. Akyildiz, "Cross-layer Packet Size Optimization for Wireless Terrestrial, Underwater, and Underground Sensor Networks", in *Proc. of IEEE INFOCOM 2008*, Phoenix AZ, April 2008.
- [10] Michael N. Krishnan, Sofie Pollin, and Avidesh Zakhor, "Local Estimation of Probabilities of Direct and Staggered Collisions in 802.11 WLANs", in *Proc. of IEEE GLOBECOM 2009*, Honolulu, Hawaii, December 2009.
- [11] Douglas O. Reudink, "Properties of Mobile Radio Propagation Above 400 MHz", in *IEEE Trans. of Veh. Technol.*, 23(4):143-60, November 1974.
- [12] Kevin Fall and Kannan Varadhan, "The ns Manual", available at <http://www.isi.edu/nsnam/ns/ns-documentation.html>.
- [13] Miklos Christine, Michael Krishnan, Sherman Ng, Ehsan Haghani, and Avidesh Zakhor, "Local Estimation of Collision Probabilities in 802.11 WLANs: An experimental Study", in *Proc. of IEEE WCNC 2011*, Cancun, Mexico, March 2011.