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, using shorter packets decreases efficiency with respect to overhead, introducing a tradeoff. 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 outline 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 the nodes compare with their local observations in order to estimate current network conditions. They then use these estimates to estimate the derivative of throughput with respect to packet length in the current network conditions and adapt their packet lengths accordingly. We demonstrate throughput gains of up to 25% 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 limitations of the design 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, longer packets become more susceptible to loss than shorter packets, introducing a potential reason to prefer shorter packet lengths.

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 packet at B, while being unable to sense the transmission of A. This can lead to *staggered collisions*, defined in [1], 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 impact of protocol 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, but very little practical application of the results. This is likely due to the fact that packet loss models used in much of the literature do not accurately reflect real-world scenarios. 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[2][3][4]. This assumes that most packet losses occur due to random bit errors in the packet payload. However, in [5], Vyas et. al. show experimentally that for lower modulation rates in 802.11a, most packet losses occur due to failure to synchronize to the packet preamble. This type of loss cannot be accounted for using a constant BER model, as it requires a model of channel fading. In [6], Zheng and Nelson use a fading model, but only assume channel coherence over symbols rather than packets and omit staggered collisions.

Various other channel models have also been introduced. In [7], Korhonen and Wang use a Poisson burst error model, and in [8] , Kim et. al. use a two-state Markov model for the channel, in which the channel alternates between two possible BERs in a Markov process. These models capture the existence of channel dynamics, but in a simplified manner. It is assumed that the channel has only two states, and that the adapting wireless node knows the proportion of time spent in each state.

In [9] Naydenov and Stoyanov use a periodic error model, which can model the traffic of a single hidden node, but in an oversimplified manner and without a model of channel fading. In [10], Yoon et. al. consider 802.11b networks with interference from 802.15.4, which behaves similarly to staggered collisions, and in [11], Song et. al. explicitly include staggered collisions into the model. However, both still use a constant BER channel model.

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. For example, if packets are primarily lost due to *direct collisions*, in which two nodes within sensing range of one another randomly decide to start transmission at the same time, there is no benefit to reducing packet length, as losses are occurring at the time when channel access is initiated. Additionally, using shorter packets would require more frequent channel access and more overhead per and therefore decreases throughput.

In [1], we propose a method for nodes in an 802.11 network with hidden terminals to estimate their collision probabilities based on shared information about channel occupancy by the access points. The basic idea behind the approach in [1] is that all nodes continually measure the occupancy of the channel around them; in addition, the access point (AP) periodically broadcasts its local channel occupancy information to all the nodes that it serves. The nodes then compare their own local measurements with those of the AP to obtain a spatial picture of the network traffic. Based on this information, they estimate the probability of occurrence of several different scenarios which lead to collisions, thus obtaining an accurate estimate of the probability of collisions of each type. In 802.11b, the overhead incurred by the AP broadcasts is less than 2%.

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 then propose a local packet length adaptation algorithm whereby each node computes estimates of its current probability of each type of packet loss, based on [1], then uses this information to compute the derivative of its throughput with respect to packet length, and adapts 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 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

In the majority of the packet length adaptation literature, the channel is assumed to be constant BER, which makes computing the optimal packet length simple, but unrealistic for wireless channels. The reason is that a wireless channel is highly variable, and the received signal to noise ratio (SNR) of different packets can vary wildly. In other parts of wireless literature, it is more common to model the channel as having a different gain for each packet, with the gains following a distribution such as Rayleigh, Ricean, or log-normal[12]. Successful reception probability is then the probability that the SNR for a given packet is sufficiently large. To accurately account for the effects of packet length with this type of model, it is necessary to compute packet error rate as a function of SNR for a given packet and assume a distribution on this SNR. Further complicating the issue is the fact that packets may also be lost due to collisions with other nodes. Collision probability is also affected by packet length in a way that requires modeling. In this section we discuss our complete packet loss model to be used in our packet length adaptation approach.

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 unfavorable 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

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

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. The following subsections discuss in detail how each type of loss occurs in carrier-sense multiple access protocols such as 802.11.

A. Collisions

There are three primary causes of collisions. The first is a direct artifact of the distributed coordination function. Since nodes access the channel randomly, there is a chance that two nodes begin their transmission at the same time, making neither of their packets decodable at the destination. These collisions are referred to as *direct collisions(DCs)* because the packets start at the same time and directly overlap.

The other types of collisions caused the hidden node problem, occur when multiple far-away nodes that cannot sense each other transmit at the same time. In these collisions, transmissions do not necessarily start at exactly the same time. As such, these collisions are referred to as *staggered collisions (SCs)*. SCs can be further subdivided in to two types, namely type 1 and type 2. 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 these two types of staggered collisions each have a different cause, and as a result they must be estimated and adapted to in different ways.

In our P_C estimation technique described in [1], nodes obtain spatial information about network traffic via periodically broadcast information from the AP, and use it to estimate the probability of each of these three types of collisions. In particular, each node as well as the AP in the network generates a binary-valued *busy-idle signal*, which is a function of time, taking value 1 when there is enough energy on the local channel that the node would not be able to transmit or successfully receive a packet, and taking value 0 otherwise. The AP periodically compresses and broadcasts its busy-idle signal to all its associated nodes. By comparing its local busy-idle signal to that of the AP, each node continually estimates the various components of collision probability for its packets locally.

B. Channel Errors

Consider a sequence of k bits being sent at a constant modulation rate R over a channel with SNR σ. The bit-error rate, as a function of R and σ, 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 σ can be computed as

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

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 $f_\sigma(s)$.

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

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 variability in the environment [12].

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. The figure on the left assumes a constant BER, and the figure on the right assumes SNR to have a log-normal distribution. The shapes of the curves are noticeably different. This is because in the case where SNR has a distribution, 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 4000 Bytes, a more accurate model including SNR distribution shows that maximum packet length would be superior.

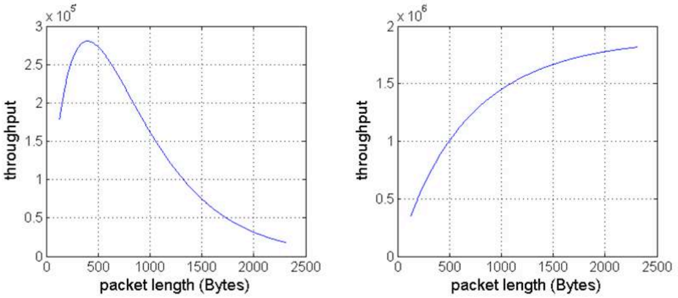


Fig. 1. *Throughput vs packet length assuming no collisions for SNR fixed at 9dB (left) and SNR with mean of 9dB with standard deviation of 3dB (right).*

III. Analytical Throughput Computation

The throughput of a wireless node can be modeled as

$$TP = C \cdot L \cdot sendFreq \cdot (1 - P_L) \tag{4}$$

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

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

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. $(1 - P_L)$ can be expanded into

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

where P_{SC2}, P_{DC}, P_{SC1}, and P_e are the probabilities of staggered collision of type 2, direct collision, staggered collision of type 1, and channel error, respectively.

If 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. 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 the expression for throughput from Equation (4) to four non-constant terms:

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

where $sendFreq$ 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. This makes it impossible to determine a single value of L to maximize this function analytically. However, it turns out that it is possible to estimate the derivative with respect to L for the current conditions given the same quantities computed in [1] 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:

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

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:

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

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

$$\begin{aligned} \tfrac{\partial}{\partial L} sendFreq &= -\tfrac{\frac{\partial}{\partial L}\overline{W}(P_L) + \frac{1}{R}}{(\overline{W}(P_L) + \overline{T}_B + T_{ov} + L/R)^2} \\ &= sendFreq^2 \cdot (\tfrac{\partial}{\partial L}\overline{W}(P_L) + \tfrac{1}{R}). \end{aligned} \tag{10}$$

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

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

$\overline{W}$ is a deterministic function of the number of attempts, which is in turn a function of P_L. It is straightforward to obtain 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

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

where the second equality comes from (8). The two terms in the sum are the same as the last two terms in Equation (9), so we can subtitute Equations (10), (11), and (12) into (9) and re-arrage to obtain

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

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

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

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

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

To obtain the final term in Equation (13), we must take the derivative of Equation (3). This gives us

$$\begin{aligned} &\tfrac{\partial}{\partial L}(1 - P_e) \\ &\quad = \int (1 - P^p_{e,h}(s)) \tfrac{\partial}{\partial L}(1 - BER_{R_p}(s))^L f_\sigma(s) ds \\ &\quad = \int (1 - P^p_e(s)) \ln(1 - BER_{R_p}(s)) f_\sigma(s) ds \\ &\quad \approx -\int (1 - P^p_e(s)) BER_{R_p}(s) f_\sigma(s) ds \end{aligned} \tag{16}$$

where the first equality comes from switching the derivative into the integral and noting that $P^p_{e,h}(s)$ does not depend on L; the second 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. As s decreases, the Taylor approximation becomes less accurate, but the terms in the integral for smaller s have lesser weight because as s decreases, $(1 - P^p_e(s)) \to 0$. The final expression 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 based on the estimate of P_e, obtained as in [1], and via a lookup table based on Equation (3). If it is a two-parameter distribution, it will require 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 re-arranging Equation (1) as

$$P_e = 1 - (1 - P_L)/(1 - P_C) \tag{17}$$

where P_L is estimated via empirical counting and P_C is estimated via the technique in [1]. 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:

$$\begin{aligned} \tfrac{\partial}{\partial L} TP &= TP \cdot \Big[\tfrac{1}{L} + \tfrac{sendFreq}{R} \\ &\quad + (sendFreq \cdot p(P_L) \cdot (1 - P_L) + 1) \\ &\quad \cdot \left(-\tfrac{m_2}{R \cdot (1-P_{SC1}} + \tfrac{\frac{\partial}{\partial L}(1-P_e)}{1-P_e} \right) \Big] \end{aligned} \tag{18}$$

where L is known, $sendFreq$ and P_L are estimated via empirical counting, m_2 is estimated via the technique in [1], P_e is computed from Equation (17) where P_C is estimated via the technique in [1], 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 [1]. As in [1], APs periodically broadcast medium occupancy statistics to all associated nodes, which the nodes use to estimate their probabilities of each type of loss. Nodes can then plug this information, along with local information into Equation (18) to compute the derivative of throughput with respect to packet length. The nodes then adjust their packet length, L, by

$$\Delta L = \alpha \cdot \frac{\partial}{\partial L} TP \tag{19}$$

where α is a proportionality constant which is updated according to the following schedule:

- if $sign(\frac{\partial}{\partial L} TP)$ has changed, decrease α by a factor of λ
- if $sign(\frac{\partial}{\partial L} TP)$ has not changed, increase α by a factor of γ

By using this schedule, nodes can have 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 choice of λ and γ can increase convergence rate and avoid oscillation. There is an additional caveat that if L is equal to L_{min} or L_{max}, α 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 α to grow exponentially. With such a large value of α, the node will 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 α decreased to a reasonable value. We additionally set a maximum step size max_step, and require that L remain between L_{min} and L_{max}.

Algorithm 1 The packet length adaptation algorithm

```
loop
    Transmit at current rate for 5 seconds
    Observe $sendFreq$ and $P_L$ over last 5 seconds
    Compute estimates of $P_C$ and $m_2$ via [1]
    Estimate $P_e$ from $P_L$ and $P_C$ via Equation (17)
    Compute estimate of $\frac{\partial}{\partial L} TP$ via Equation (18)
    if $sign(\frac{\partial}{\partial L} TP) \neq last\_dir$ then
        $h \leftarrow \alpha \cdot \gamma$
        $last\_dir \leftarrow sign(\frac{\partial}{\partial L} TP)$
    else if $L \in (L_{min}, L_{max})$ then
        $h \leftarrow \alpha / \lambda$
    end if
    $\Delta L \leftarrow sign(\frac{\partial}{\partial L} TP) \cdot min(|\frac{\partial}{\partial L} TP| \cdot h, max\_step)$
    $L \leftarrow median(L_{min}, L + \Delta L, L_{max})$
end loop
```

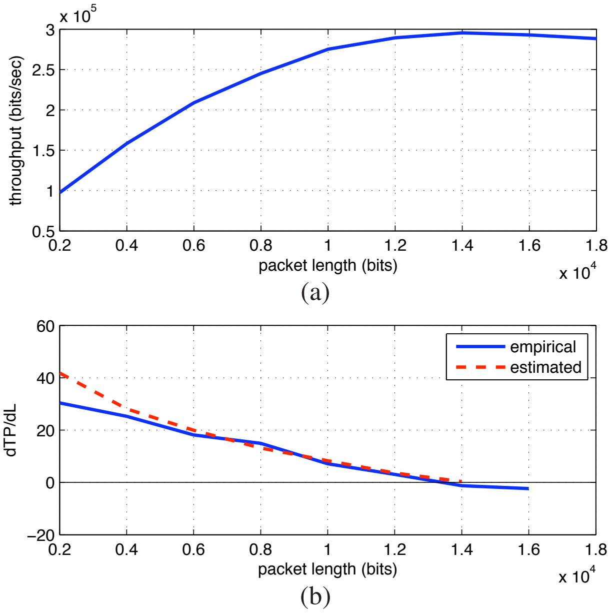
V. Simulation results

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 [1] 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 [13]

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. Plots for two representative nodes are shown in Figures 2(a) and 3(a). From our simulation data, we also compute the $\frac{\partial}{\partial L} TP$ estimates for each scenario and plot them in Figures 2(b) and 3(b). The estimates are shown as the red dotted line, while the solid blue line is computed as the average slope over three consecutive data points in the curve in Figures 2(a) and 3(a). These two examples are situations in which 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.

In the case of Figure 2, the estimate of $\frac{\partial}{\partial L} TP$ is quite accurate, while in the case of Figure 3, it is much less so. However, the most important feature of the derivative curve is the point at which it crosses zero. This is because the algorithm will converge to this packet length. It can be seen that even in the case of Figure 3, where the algorithm converges to an incorrect packet length of 7000 bits rather than 10,000 bits, the resulting throughput is greater than if the node had used maximum packet length, namely 270 kbps rather than 240 kpbs.

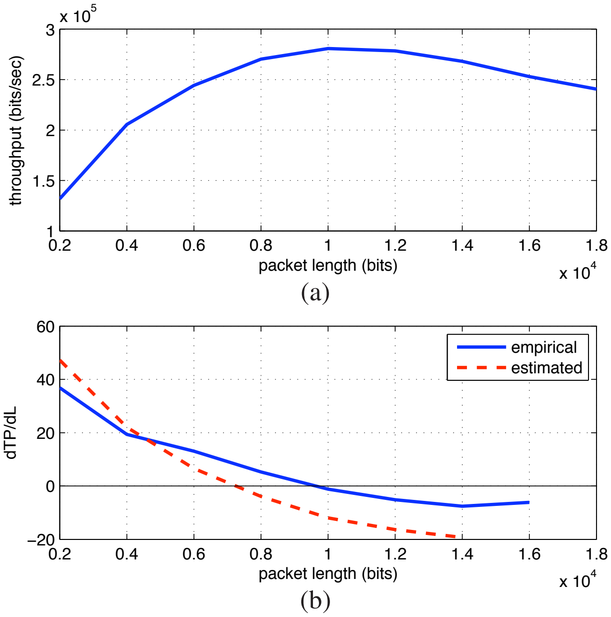
Having verified that the algorithm works for single-node adaptation, we now allow all nodes to adapt simultaneously. Here the dynamics of our algorithm come into play. We have empirically found that initializing α to 2000, and setting $\lambda = 2$, $\gamma = 1.25$, and $max\_step = 4000$ allows nodes 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.



(a)

(b)

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



(a)

(b)

Fig. 3. *(a) Throughput as a function of packet length and (b) its derivative for another single node with all other nodes sending at constant packet length.*

Depending on their specific locations, some nodes quickly converge to maximum packet length while other nodes converge to other packet lengths. Figure 4 shows the converged packet lengths of various nodes for one scenario with a noise power of -95dBm. 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; if this color is significantly different from the color of the center of the circle, this indicates slow convergence. It can be seen that in this case, 68% of the nodes quickly converge to maximum packet length. The nodes which converge to shorter packet lengths tend to be the nodes 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, giving them higher packet loss rates, making it more beneficial to select shorter packet lengths.

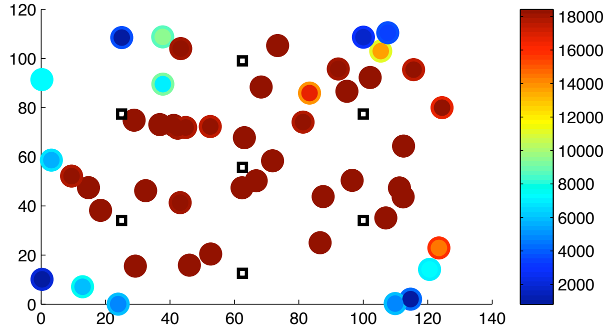
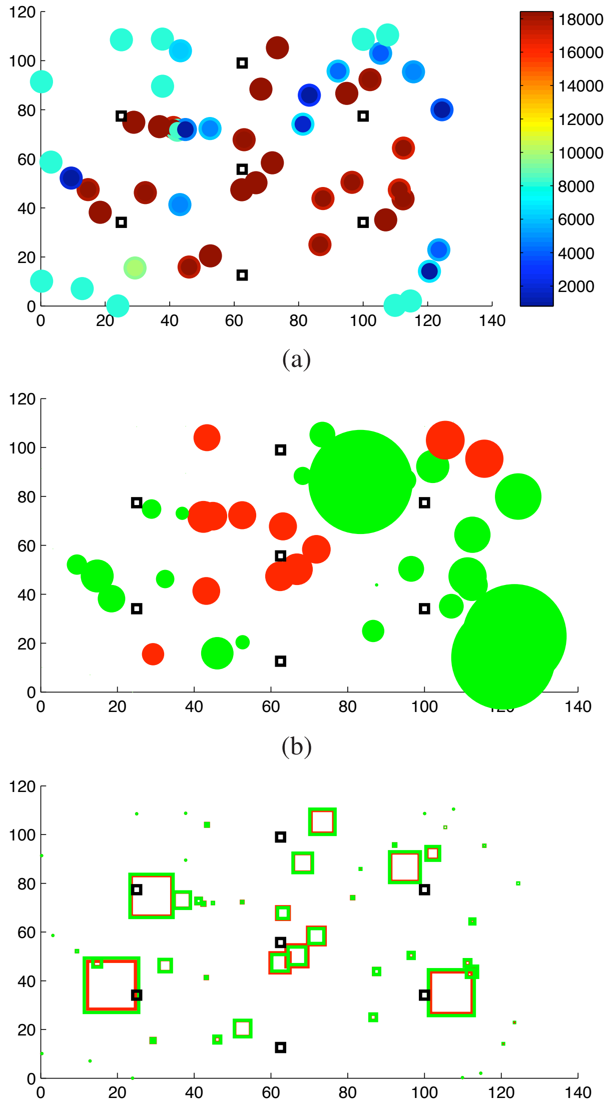


Fig. 4. *Spatial plot of adapted packet length for each node in a scenario with -95dBm noise.*

Figure 5(a) shows the converged packet lengths for the same topology for a higher noise power of -89dBm. 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. 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



(a)

(b)

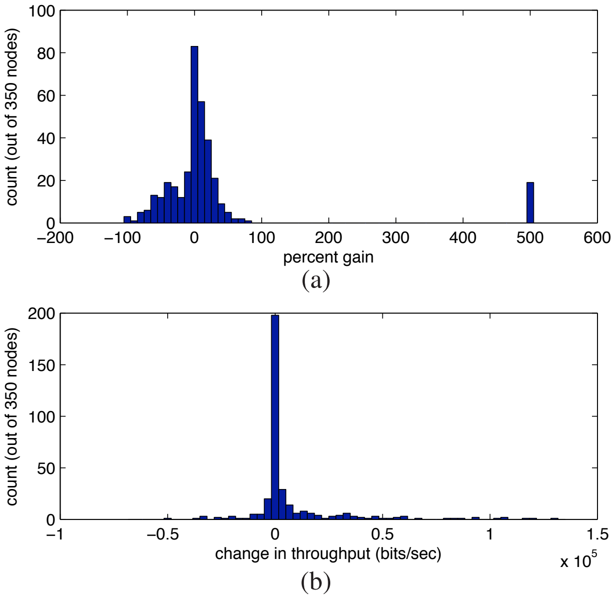
(c)

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 -89dBm noise.*

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.

Figure 6(a) shows histogram of the percent throughput improvement for all nodes over all 7 topologies at -89dBm noise, with values capped at 500%. The mode centers around zero, but about 20 nodes experience gains of greater than 500%. The nodes with these extreme percentage changes are nodes with negligible throughput. Figure 6(b) shows absolute gains in throughput for the same set of nodes. As seen, most nodes do not change significantly, but the positive tail is significantly larger than the negative tail. Only one node loses 50 kbps of throughput, while 18 gain 50 kbps or more.

Figure 7 shows some traces of packet length over time for several nodes. All nodes use an initial packet length of 8000 bits. Many nodes with favorable conditions behave like the red curve marked with circles, converging to maximum length after three iterations. Others take as many as 20-30 iterations, but once they converge, they stay relatively stable.



(a)

(b)

Fig. 6. *Histogram of (a) percentage (b) absolute throughput changes for each of 350 nodes at -89dBm.*

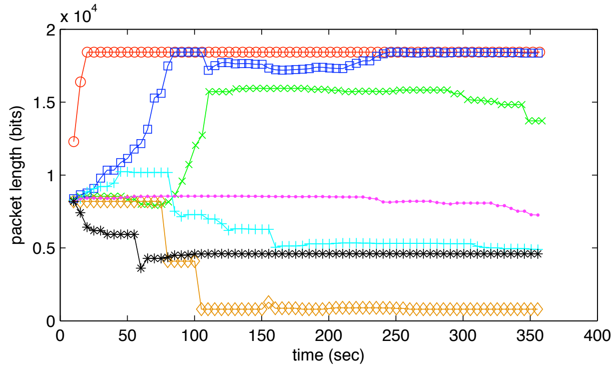


Fig. 7. *Packet length versus time for some adapting nodes.*

Figure 8 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 been 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. It can also been 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.

TABLE I
AVERAGE VALUES OF TOTAL NETWORK THROUGHPUT GAIN AND NUMBER OF NODES CHOOSING NON-MAX LENGTH FOR EACH NOISE LEVEL

noise power	throughput gain	# non-max length nodes
-98 dBm	1.2%	5.3
-95 dBm	5.4%	10.6
-92 dBm	8.7%	18.8
-89 dBm	11.6%	24.0
-86 dBm	16.3%	32.6

At lower noise powers, our algorithm results in most nodes choosing maximum packet length, which is optimal. Thus the total throughput it very close to when packet length is fixed at maximum for all nodes.

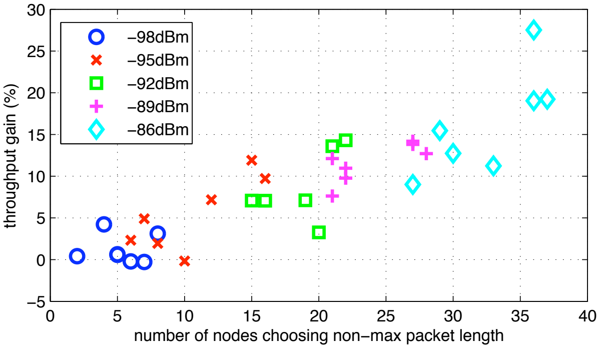


Fig. 8. *Throughput gain and number of nodes choosing non-max packet length for 7 different topologies, each at 5 different noise powers.*

Table I shows the average values of total network throughput gain and number of nodes choosing non-max length for each noise level. It can be seen that as noise increases, an increasing number of nodes choose shorter packet lengths, and the total network throughput gain is increased.

VI. CONCLUSIONS AND FUTURE WORK

In this paper we presented a detailed packet loss model for 802.11 networks, including channel fading and staggered collisions. We computed throughput as a function of packet length, and the derivative of this function, and showed that it can be estimated using estimates form [1]. We proposed a packet length adaptation aglorithm, and showed that it can achieve up to 25% throughput gains.

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 modulation rate [14] and carrier sense threshold [15]. 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.

REFERENCES

[1] Michael N. Krishnan, Sofie Pollin, and Avideh Zakhor, “Local Estimation of Probabilities of Direct and Staggered Collisions in 802.11 WLANs”, in *Proc. of IEEE GLOBECOM 2009*, Honolulu, Hawaii, December 2009.

[2] Xiao Liu and Wael Badawy, “A Novel Adaptive Error Control Scheme for Real Time Wireless Video Streaming”, in *Proc. of ITRE 2003*, August 2003.

[3] 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.

[4] 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.

[5] Amit K. Vyas, Fouad A. Tobagi, and Rajesh Narayanan, “Characterization of an IEEE 802.11a Receiver using Measurements in an Indoor Environmant”, in *Proc. of IEEE GLOBECOM 2006*, San Francisco, California, November 2006.

[6] 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.

[7] Jari Korhonen and Ye Wang, “Effect of Packet Size on Loss Rate and Delay in Wireless Links”, in *Proc. of IEEE WCNC 2005*, New Orleans, Louisiana, USA, May 2005.

[8] Eung-in Kim, Jung-Ryun Lee, and Dong-Ho Cho, “Throughput Analysis of Data Link Protocol with Adaptive Frame Length in Wireless Networks”, in *AEÜ Int. J. Electron. Commun., 51(1):1-8*, 2003.

[9] 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.

[10] Dae Gil Yoon, Soo Young Shin, Wook Hyun Kwon, and Hong Seong Park, “Packet Error Rate Analysis of IEEE 802.1lb under IEEE 802.15.4 Interference”, in *Proc. of IEEE VTC 2006*, Melbourne, Australia, May 2006.

[11] Wei Song, Michael N. Krishnan, and Avideh 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.

[12] Douglas O. Reudink, “Properties of Mobile Radio Propagation Above 400 MHz”, in *IEEE Trans. of Veh. Technol., 23(4):143-60*, November 1974.

[13] Kevin Fall and Kannan Varadhan, “The ns Manual”, available at http://www.isi.edu/nsnam/ns/ns-documentation.html.

[14] Michael N. Krishnan and Avideh Zakhor, “Throughput Improvement in 802.11 WLANs using Collision Probability Estimates in Link Adaptation,” in *Proc. of IEEE WCNC 2010*, Honolulu, Hawaii, April 2010.

[15] Adaptive Carrier-Sensing for Throughput Improvement in IEEE 802.11 Networks,” to appear in *IEEE GLOBECOM 2010*, Miami, FL, December 2010.