Dynamic Optimal Fragmentation for Goodput Enhancement in WLANs

Yusun Chang, Chris Lee, B. Kwon, and John A. Copeland
Communications System Center
Georgia Institute of Technology
Atlanta, GA 30332
{yusun, chris, bkkwon, copeland}@ece.gatech.edu

***Abstract*— To meet the demand for broadband wireless communication, wireless systems should work well in typical wireless environments, characterized by the path loss of the signals, multipath fading, interference to adjacent channels, and random errors. IEEE 802.11 WLANs use the unlicensed 2.4GHz industrial, scientific and medical (ISM) band, which is vulnerable to noise generated by TVs, microwaves, and cordless phones. This paper proposes an algorithm to enhance system goodput through the dynamic optimal fragmentation. The number of contending stations, packet collisions, packet error probabilities, and fragmentation overheads are modeled in the analysis. Using an adaptive SNR estimator, the sender estimates the SNR of the receiver, and chooses a fragmentation threshold to shape arbitrary sized packets into optimal length packets. Through the rigorous analysis and extensive experiments with implemented test-bed, we show that the dynamic optimal fragmentation enhances the goodput approximately 18% in a typical WLAN environment. The experiment results reinforce that the algorithm is a comprehensive analytical model applicable to any CSMA/CA based MAC protocol for next generation wireless networks, and a realistic approach that can be deployed without changing the IEEE 802.11 MAC protocol.**

Keywords-component; Fragmentation, Goodput, 802.11

I. INTRODUCTION

These days the desire for ubiquitous wireless connectivity is growing rapidly. The third generation (3G) network industries are accelerating the development and deployment of the wireless data network. Built on the cellular technologies, it promises wide coverage, seamless support of mobility, paging, and quality of service. However, it is a complex and costly connection-oriented networking that provides approximately 2 Mbps for indoor traffic.

To meet demands of high data rate users, many companies have provided wireless local area networks (WLAN) that give up to 50 Mbps. Since it uses the unlicensed spectrum with a simple design, the cost is much lower than 3G networks. On the other hand, WLAN covers only a few hundred meters and mobile users are unable to have a seamless connection when moving to other networks. Users in the future are expected to use both types of networks, one for wide coverage and reliable seamless connection, and the other for high data rate.

The low costs of wireless cards and access points of WLAN made it possible to use at home, in university access networks, and at hotspots such as train stations, hotels, coffee shops, and airports. Theoretically, the IEEE 802.11b [1] network can achieve 11 Mbps. However, the actual data rate may vary and depends on network configurations, channel states, and user behavior. Intersymbol interference (ISI) caused by multipath fading has been known to be the major obstacle to high-speed data transmission. To counter the ISI, the 802.11b receiver adopts the decision feedback equalizer (DFE) to compensate deep spectral nulls in frequency selective time dispersive channels. However, many vendors require a 65-ns delay spread for full–speed 11 Mbps performance at a reasonable frame error rate, and when the delay spread is large, many cards will reduce the transmission rate, resulting in degradation of the throughput. The IEEE 802.11b networks share the 2.4 GHz ISM band with other electronic devices, such as TVs, microwaves, and cordless phones. Considering the fact that the spectrum is inherently wide open to the interference from these devices, and no forward error correction (FEC) is used, it is not surprising that actual network throughput is far less than 11 Mbps.

Packet length adaptation is an approach that has been studied to increase the throughput of a WLAN by changing the frame size to the time varying wireless channels. If transmitters break messages into smaller fragments for sequential transmission, the each shorter duration fragment has a better chance of escaping burst interference and increases throughput. This simple technique also reduces the need for retransmission in many cases and can be used to reshape arbitrary sized packets into optimal length packets to improve wireless network performance. The 802.11 standard mandates that all receivers support fragmentation, but it leaves such functionality optional on transmitters. Full-time fragmentation in a transmitter makes it possible to design a less expensive receiver resulting in lower receiver sensitivity. However, it incurs overhead on every fragment rather than every frame, thereby reducing the aggregate throughout of the WLAN and the realizable peak throughput rate achieved between stations.

The architecture for adapting frame length to the time varying channel is proposed in [2]. It exploits the effect of BER and frame length on throughput in wireless network. Simple backoff based frame length adaptation [3] is proposed to adapt fragment sizes using the fragmentation threshold in time varying channels. In this algorithm, the next fragmentation threshold is set to half of the previous one, if an ACK is lost or timeout occurs. When the transmission is successful, the threshold is doubled in the next stage. Similar approaches, [4] and [5], have been proposed to tune fragment size to fit in a dwell time in the frequency hopping system. However, these approaches [2-5] are for a general MAC protocol, and they do not include the 802.11 distributed coordination function (DCF) protocol to calculate optimal fragmentation. In [6] and [7], a link adaptation strategy is studied to select the optimal combinations of the 802.11a [8] PHY mode and the fragment size to achieve the best goodput performance for different SNR conditions. This approach has a detailed analysis of DCF with fragmentation, and shows how the fragmentation affects goodput with different physical modes and SNR. Although the scheme achieves some degree of optimization, it excludes the effect of collisions. When a collision occurs in the network, it has the same impact on the exponential backoff procedure of the DCF as random packet drops due to a bad channel. Thus, contention based packet-sending probability and collision probability must be incorporated in the calculation of the goodput. J. Yin et al models the effect of the contentions among users, the collisions, and the random errors at the receiver in [9]. The analysis computes the optimum packet size to maximize the throughput in an error prone channel, and gives insight of how the network responds to the system parameters stated above. However, the MAC layer is not capable of reshaping arbitrary sized packets to the optimum sized packets, and if we apply fragmentation, the optimal fragment size will be different because of the fragmentation overhead and time spent for transmission.

In this paper, we propose a dynamic optimal fragmentation algorithm. The algorithm adapts packet length to maximize available goodput in given network conditions, such as packet length, number of users, contentions, collisions, and random errors. By using a detailed fragmentation analysis of the IEEE 802.11 DCF, an adaptive estimator at the sender estimates the SNR of the receiver, and slices higher layer packets into optimally sized packets in time varying channels. Since the UDP messages are used to estimate the SNR with received signal strength, no protocol change is required to deploy the scheme. These control packet overheads are reasonably low in a typical wireless LAN environment. If the channel is fast fading, which is an unusual case in a typical WLAN, it still can be incorporated with modified RTS/CTS schemes [10, 11, 12] without losing generality. To verify the performance of the algorithm, extensive experiments have been performed using an implemented test-bed, which consists of CISCO (Aironet 1231-G-A-K9 [13]) access point with three mobile stations and servers in the network.

The remainder of this paper is structured as follows. Section II presents an overview of the IEEE 802.11 MAC protocol with the basic access method and DCF operation. A detailed fragmentation and delay analysis that considers the trade-offs between packet size, overhead, and packet errors rates follows. The system design elements are discussed in Section III for the adaptive channel estimation, impact of imperfect channel estimation on the system goodput, and implementation issues. In Section IV, we discuss our wireless test-bed, the experiments performed, and the results. Finally, we state our conclusions in Section V.

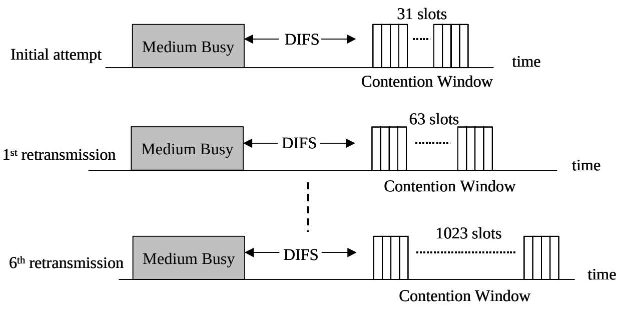

Fig. 1. Exponential backoff of the contention window

II. SYSTEM ANALYSIS

The fundamental access method in IEEE 802.11 medium is the DCF. The carrier sensing multiple access/collision avoidance (CSMA/CA) mechanism is used to avoid collisions in the medium while users contend to access the channel in the same basic service set (BSS). If contention free access is required, a point coordination function (PCF) built on the top of the DCF can be provided. We analyze the DCF in this section. Detailed fragmentation operation and the trade-off between overheads and packet error rate are investigated.

The performance of the optimal fragmentation on goodput and delay will be discussed later in Section II.

A. IEEE 802.11 MAC Overview

In DCF, stations about to transmit data sense whether the medium is idle or occupied by other stations. If the medium is idle for at least a DCF inter-frame space (DIFS) interval, the station decreases the backoff timer, which was randomly selected at the first attempt of the transmission. When this backoff timer expires, the station tries to access to the medium and transmit data. If the transmission is successful, the station resets the backoff timer and chooses a new time slot in the contention window as it does in the initial attempt. Otherwise, the station that has failed at the first round of transmission should exponentially backoff the contention window and chooses a new time slot in the window. When the medium is idle for at least a DIFS interval, the station counts down the backoff timer, and accesses to the medium when it expires. This exponential backoff of the contention window will repeat until it reaches the maximum of 1023 slots in the 802.11b direct sequence spread spectrum (DSSS) physical layer, as in Fig 1. Then it remains there unless the timer is reset by a successful transmission, or discarded by the retry counter. Because the DCF operates without a central coordinator, the medium access control is done independently. This typical exponential backoff ensures the stability of the network and guarantees long-term fairness even in the maximum saturated traffic with many contending stations in the same BSS.

Fig. 2 depicts how the DCF works with different inter-frame spaces defined in the standard [1]. If the medium idles longer than the DIFS interval, stations sensing the medium attempt to get access when their backoff timers expire. If only one station tries to access the channel when no other stations transmit data or interfere the sender, the station captures the channel for transmission.

B. IEEE 802.11 MAC Analysis

To analyze the performance, we consider the DCF in IEEE 802.11b under saturated traffic condition. For each time a station transmits a packet, we assume that the unsuccessful transmission probability p is constant at steady state in a generic slot. Let W_i be the contention window after i times of collision. Then, W_i is

$$W_i = (W_m * 2^i) - 1\,, \tag{1}$$

where W_m is 32 at the initial round of transmissions as in Fig 1. The backoff timer chooses a time slot uniformly between 0 to W_i after i collisions until the packet is successfully transmitted or discarded by the retry counter. We denote the probability of unsuccessful transmission p for each station in the channel. Then, the probability of success for each station after i times failure is

$$P_{success}(i) = (1-p)\,p^i\ . \tag{2}$$

To calculate the average waiting time $\overline{W}$ for a station to transmit a packet, we use (2) with the average contention window size $\frac{W_i}{2}$, and we get the following.

$$\overline{W} = \sum_{i=0}^{RC} \left(\frac{W_i}{2}\right) P_{success}(i)\ , \tag{3}$$

where RC is the retry counters defined in [1]. If we assume the average packet transmission probability of each node P_τ, which covers the whole backoff stage, can be calculated as a constant value at steady state, the average probability of each station to send a packet, P_τ [14] can be represented as

$$P_\tau = \frac{1}{(\overline{W}+1)}\ . \tag{4}$$

Let's consider the collision probability for a station to transmit a packet while competing with other $n-1$ stations. The collision probability, p_c is

$$p_c = 1-(1-P_\tau)^{n-1}\,. \tag{5}$$

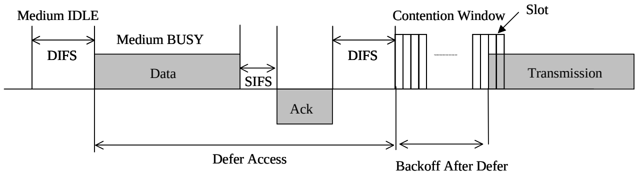

Fig. 2. Basic access method and inter frame space

Note that the integrity of the 802.11 frames is checked at the receivers by the frame check sequence (FCS). The receivers calculate the FCS that includes MAC header and frame body, and compare it to the received FCS. If the frames pass the integrity check, there is a high possibility that the frames were not damaged in transit. If it fails, the packets are corrupted by collisions or random errors due to interference and poor SNR. Therefore, the probability of an unsuccessful transmission should include both the collision probability p_c and packet loss due to random bit error. We denote this random bit error, p_b as BER in wireless channels. Then, the upper bound of the probability of the packet loss by random bit errors in L bit packet can be expressed as follows [15].

$$p_{pkt\_err} = 1-(1-p_b)^L\,. \tag{6}$$

If we assume p_c and p_{pkt_err} are independent, the probability of unsuccessful transmission p is given as

$$p = 1-(1-p_{pkt\_err})(1-p_c)\,. \tag{7}$$

Using (1) to (7) with a given number of users n, BER, and L-bit long packets, the two unknown values, p and p_c can be solved using numerical techniques. Let's consider the probability of a successful transmission associated with collisions when there is at least one station to transmit. The probability that there is a transmission among n stations in the channel is $1-(1-P_\tau)^n$. Since only one station transmits while other $n-1$ stations keep quiet, the probability of successful transmission without collision is given by [9].

$$P_s = \frac{n P_\tau (1-P_\tau)^{n-1}}{1-(1-P_\tau)^n}\,. \tag{8}$$

C. Fragmentation Analysis

To see the fragmentation effect on goodput and probability of errors, we consider overhead and IFS in Fig. 3. Assume L bit long MAC Service Data Unit (MSDU) is fragmented to j times MAC Protocol Data Units (MPDUs), L_{opt} . Then, it incurs $j-1$ times the additional overhead of $H+2SIFS+ACK$, where H is the header of the MSDU. Note that this overhead is additive, while the probability of

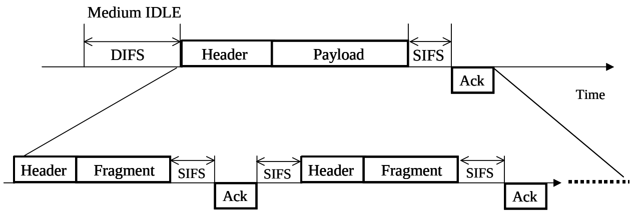

Fig. 3. Fragmentation operation

packet loss, p_{pkt_err} is reduced exponentially from $1-(1-p_b)^L$ to $1-(1-p_b)^{L_{opt}}$. This p_{pkt_err} has a big impact on p especially when the channel BER becomes worse, and random packet drops are the major component of the unsuccessful transmissions. As we can see in Fig. 4, p_{pkt_err} is dominant at the BER of 10^{-4}, whereas p_c mostly affects p at the moderate BER of 10^{-5}. For the same fragment with a given n, we have a little bit smaller p_c at 10^{-4} than that of 10^{-5}. As p is greater at 10^{-4} because of the worse p_{pkt_err}, we have a larger average waiting time $\overline{W}$. Consequently, the number of collisions is slightly reduced by the exponential backoff. However, the effect of the fragment size on p_c is decreased for a higher SNR, and p_c is primarily dependent on the number of users rather than BER. In Fig. 5, we can see the fragment (MPDU body) size has little impact on p_c while user population has a direct impact on it for the same BER of 10^{-5}. Note that the smallest p does not guarantee a maximum goodput if the fragment size is not the largest, and this is why we need to optimize p, fragment size, and BER for the number of users n.

Here, we define goodput as the fraction of time that the medium is occupied to transmit user data successfully. To obtain the goodput with fragmentation, we calculate various time components. The average idling time between two consecutive transmissions, t_{idle} can be represented by the mean of geometric distributions [9],

$$t_{idle} = \frac{1}{1-(1-P_\tau)^n} - 1. \tag{9}$$

In addition, consider sending a L bit packet with data rate R and packet header H'. We represent t_s as the time duration normalized to a slot time to transmit user data. Then, we have

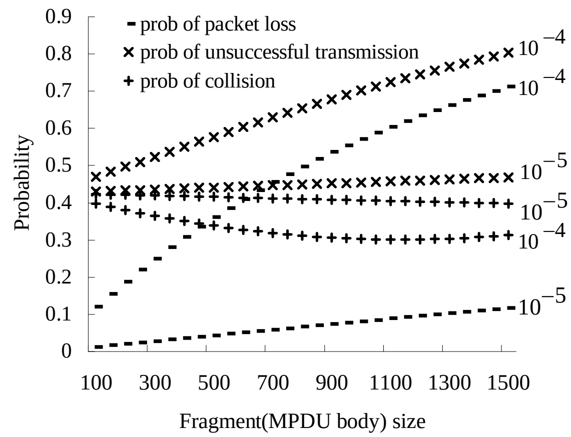

Fig. 4. Probability, fragment size, and BER for 20 users

$$t_s = \frac{L-H'}{R * t_{slot}}, \tag{10}$$

where H' is physical, MAC, and TCP/IP layer headers. Now, we compute the time interval, t_f, to send a L bit MSDU successfully with fragmentation. We denote the fragment (MPDU) that maximizes goodput as L_{opt}, and physical and MAC layer headers, H. If we fragment L into j times L_{opt}, we have an overhead of $(j-1)(H+2SIFS+ACK)$. Therefore, t_f is

$$t_f = DIFS + L + SIFS + ACK + (j-1)(H + 2SIFS + ACK). \tag{11}$$

In addition, the time duration to detect collisions at the receiver

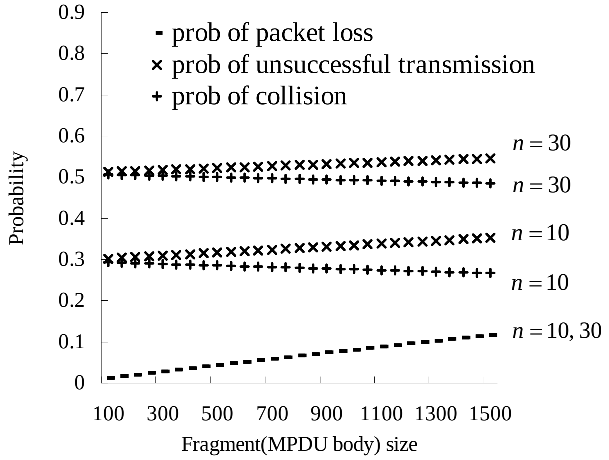

Fig. 5. Probability, fragment size, and number of users

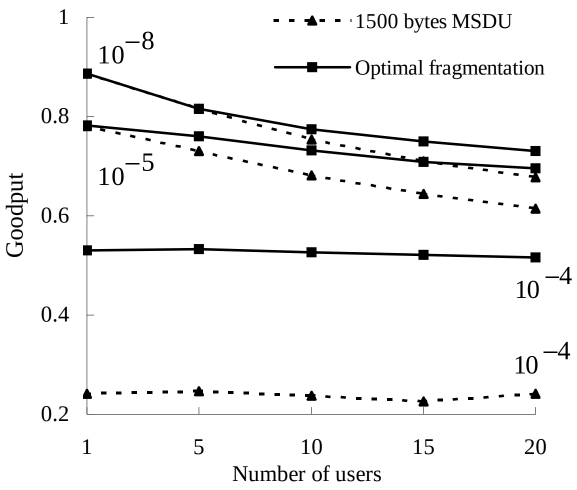

Fig. 6. Goodput versus number of users

t_c is given as

$$t_c = DIFS + L_{opt} + SIFS + ACK\ . \qquad (12)$$

Consequently, we have goodput G,

$$G = \frac{P_s(1-P_{pkt\_err})t_s}{t_{idle} + P_s(1-P_{pkt\_err})t_f + (1-P_s)t_c + P_sP_{pkt\_err}t_f} \qquad (13)$$

Given a packet size L, the number of users n, and a BER, the solution, optimal fragment L_{opt}, of the nonlinear system (13), can be uniquely determined using numerical approaches to find where G reaches the maximum value.

The impact of contention on goodput is depicted in Fig. 6. For the numerical analysis, we compare the goodput for 1500 bytes MSDU and the optimal packet L_{opt} in various BER. In a perfect channel, for example BER of 10^{-8} in Fig.6, we do not need fragmentation to avoid the overheads, if the number of users is less than 5. Since the probability of a random packet loss is negligibly small in this channel, an unsuccessful transmission probability, p, is almost equal to the collision probability, p_c, in (7). As we stated earlier, p_c is not a function of packet length if the channel is perfect, but a function of number of users in the network. Therefore, if there are no hidden terminals and no interference at the receivers, and all stations obey the basic access rules, we have a constant p_c regardless of the packet length, and performance will be degraded gracefully. However, p_c increases as the number of users increases. Thus, a longer packet needs more time to detect loss and recover from it. As the contentions become severe, the optimal fragmentation provides more benefit by adjusting the fragment size to mach the channel.

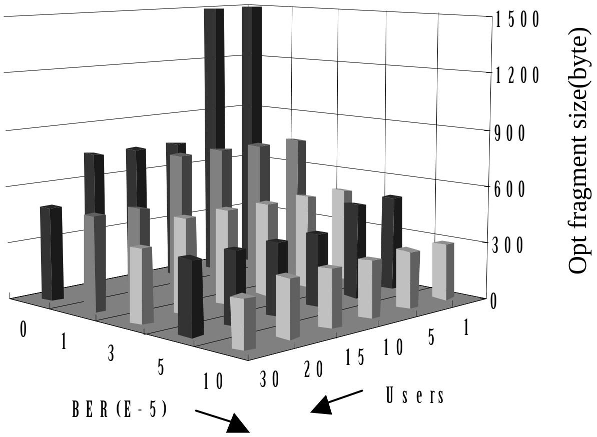

Fig. 7. Optimal MPDUs sizes for 1500 bytes MSDU with BER and number of users

As the channel worsens, the fragment size should decrease abruptly to compensate for the random errors. The contention among users also affects the optimal fragment size as mentioned earlier. However, the impact of the contention is smaller, since random errors play a bigger role on the exponential backoff procedure than collisions caused by contentions. Consequently, optimal fragmentation improves goodput more effectively as the channel becomes worse, or the number of users increases in the network. We present the performance of using the optimal fragmentation in Fig. 7 and Fig. 8 for the 1 Mbps 802.11b physical mode.

D. Delay Analysis

Packet delay is also an indicator of performance. For each user to send a packet with n contending nodes, if we assume each node shares equally the average stationary goodput, the

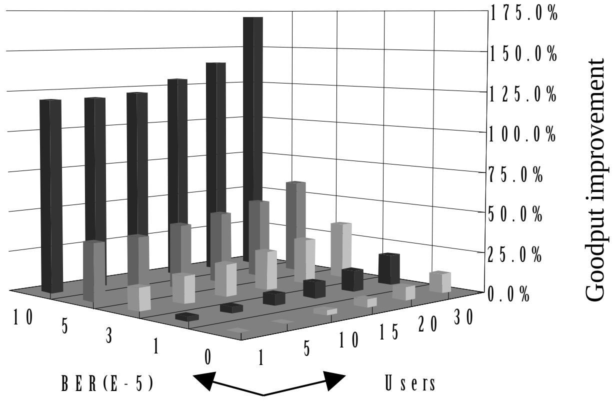

Fig. 8. Goodput improvement of optimal fragmentation

delay for a node to transmit a bit is n/G. Since the time to send user data in a packet is given at (10), we have the packet delay D_{packet},

$$D_{packet} = \frac{n}{G} t_s . \tag{14}$$

Fig. 9 shows the packet delay for the 1500 bytes MSDU in 802.11b [1], 1Mbps DBPSK physical mode. As the number of users grows, the delay performance deteriorates proportionally. For a typical WLAN environment having 20 users with BER of 10^{-5}, they experience 380 ms of packet delay for normal operation, and 330 ms with the optimal fragmentation. This yields a 13.15% performance gain, and it increases as the channel errors become worse. For example, optimal fragmentation improves goodput 33.16% for a BER of 5×10^{-5}.

III. System design

Since optimal fragmentation improves the goodput in a typical WLAN environment, we propose a dynamic optimal fragmentation algorithm that uses network parameters to select the optimal fragmentation using the SNR estimator. The network parameters considered in our model includes the incoming packet length, BER, number of users, and the transmission rates. In the following section, we detail our design of dynamic optimal fragmentation algorithm.

A. Adaptive channel estimation

In [10][11] and [12], a modified RTS/CTS exchange is used to feed back the channel conditions of the receiver, which requires modifications of the protocol. The link adaptation strategy [6][7] uses the received signal strength (RSS) of the frame from the AP to select the best transmission rate for the sender. This approach assumes that the RSS has a linear relationship with the SNR of the receiver. However, this assumption is not valid when the AP supports multiple rates for downlink channels. Since users may have different network cards, the transmission power of each user may be different. Therefore, SNR estimation with RSS for each station should be different, and the AP is not able to select proper rates individually for the stations. Furthermore, in the presence of interference at the receivers, strong RSS at the AP does not guarantee better SNR, and each user may experience different profile of interference.

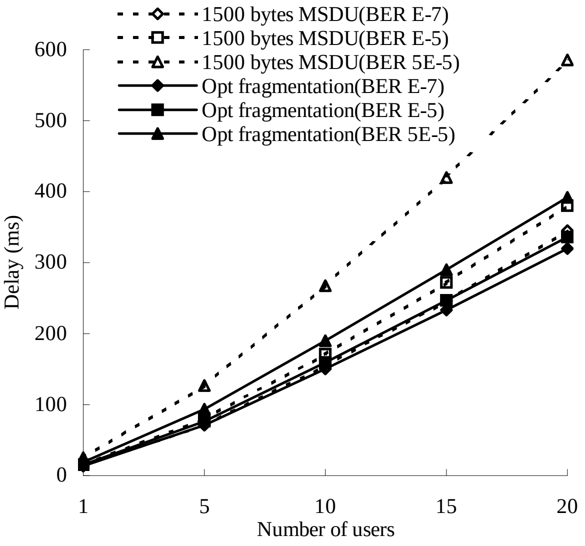

Fig. 9. Packet delay for 1Mbps, DBPSK physical mode

Since 802.11 MAC has no closed loop feedback channel for the channel quality, the sender needs to estimate the channel condition of the receiver. We propose an adaptive estimator to estimate the SNR of the receiver. We represent the RSS from the receiver as $y(k)$, and the average of RSS before $y(k)$ as $\bar{y}_{RSS}(k-1)$. Then, the estimation of the SNR for selecting optimal fragmentation threshold, $\hat{y}_{SNR}(k+1)$, can be expressed as

$$\begin{aligned} \hat{y}_{SNR}(k+1) &= \alpha\, \bar{y}_{RSS}(k-1) + (1-\alpha)\, y(k) \\ &\quad + \gamma \left\{ \bar{y}_{SNR}(k) - y(k) \right\} , \end{aligned} \tag{15}$$

where $\bar{y}_{SNR}(k)$ is the average SNR of the receiver, and $0 \le \alpha \le 1$. The term, $\gamma \ge 0$, is to compensate for estimation error and suppress temporary SNR fluctuations in the decision region for stable selection of the fragmentation threshold

Since the uplink and downlink channels are not always geographically symmetric, estimation by only using the observed RSS at the AP is not valid for selecting optimum rates and fragmentation threshold, even though there is no interference at the receiver. Thus, the adaptation algorithm needs to know current average SNR of the receiver, while reflecting the variation of the RSS with it. If the average SNR exceeds the previous one by a certain amount, the receiver notifies the sender of the current average SNR to reduce the estimation error. In Fig. 10, we can see the difference between

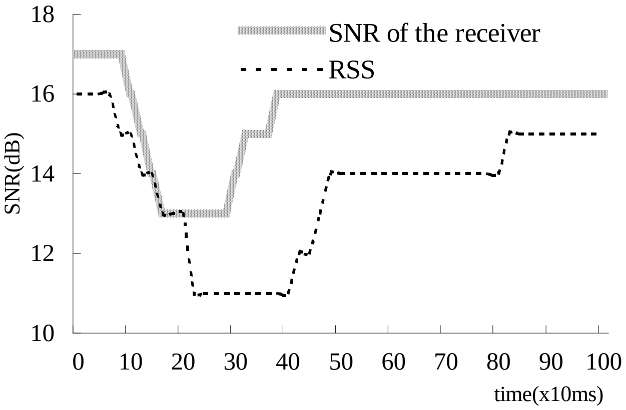

Fig. 10. Measurement of SNR & RSS with same Tx power -10dBm

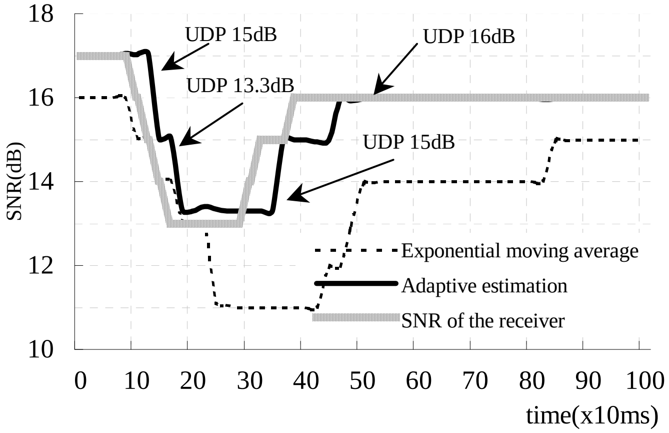

Fig. 11. Adaptive estimator with 4 UDP messages.

SNR of the receiver and the RSS at the sender for the same transmission power of -10dBm with LOS in a typical office environment. Total 50 samples of 1500 bytes MSDU are used to measure the SNR of the receiver and RSS at the sender for the same -95dBm noise power.

For the estimator in Fig. 11, we set $\alpha = 0.05$ and $\gamma = 1.0$ using numerical trial and error to minimize the estimation error. We can decide how quickly the algorithm tracks the channel conditions by choosing the values α and γ. However, finding optimal values of α and γ is out of this research scope and should be included in future work.

The average of five samples and three samples are used to calculate $\bar{y}_{RSS}(k-1)$ and $\bar{y}_{SNR}(k)$ respectively. The receiver informs $\bar{y}_{SNR}(k)$ to the sender in UDP message when the difference is greater than 1.5dB. This value may vary with respect to the BER performance variations between modulation schemes in the system and channel characteristics. In this experiment, the average estimation error for the proposed estimator is 0.33dB while exponential moving average using the RSS in [6] [7] incurs 1.94dB for each packet.

Typically, just a two-byte UDP message is enough to represent SNR with 0.1dB scale and less than 10 UDP packets are required per second, even for the worst channels in our measurements. However, these UDP messages can influence the goodput, and sacrifice bandwidth in the network. In Fig. 12, we show the impact of the UDP messages. We established three TCP flows of 1500 bytes MSDU in DBPSK with 4% of packet error rate, and three mobile users receiving the packets in NS-2 simulator. Each user transmits UDP control messages to estimate the SNR of the receiver during the TCP transmission. As we can see, even in the worst situation of 10 UDP messages in a second, the performance loss is less than 5% compared to the basic operation, and we get 5.23% improvement for the RTS/CTS based channel estimation as in [10, 11] and [12]. However, note that the total performance gain we achieve is greater than basic operation as we described earlier in Fig. 8.

B. Impact of imperfect channel estimation

The BER and SNR are the dominant components in the calculation of the optimal fragmentation. However, the algorithm does not use an explicit SNR estimator as in [10, 11, 12] to decide optimum values in the channel. Note that the BER is usually fixed by the design parameter in the rate adaptation algorithm, and even if the SNR fluctuates severely such that the BER also changes dramatically, the adaptive rate adaptation automatically switches the rate to maintain the BER above the target BER of the system. Furthermore, selecting the wrong optimal fragmentation is unlikely as the estimation algorithm has a very low approximation error (e.g., 0.33 dB in our previous measurement). For example, we have 750 bytes of optimal MPDU for 15 users at the BER of 1×10^{-5} in Fig.7. To select 500 bytes, which is the optimal value for 3×10^{-5}, 1.1dB estimation error is required. However, it still achieves 99.98% of the goodput for the optimal value. In case it selects 1500 bytes for 1×10^{-7}, it will use larger packet in a bad channel, and we lose 9% of goodput. However, that is unlikely to occur in the proposed adaptive estimator, since it would have to have an error of at least 4.61 dB.

C. Implementation issues

In the non-linear system given by (13), nodes must determine the optimal fragment, L_{opt}. However, solving this equation in real-time is a computationally expensive task. We can simplify the complexity by incorporating the knowledge that L_{opt} will be a divisor of the original packet length. For example, if the packet is 1500 bytes, the valid candidates of

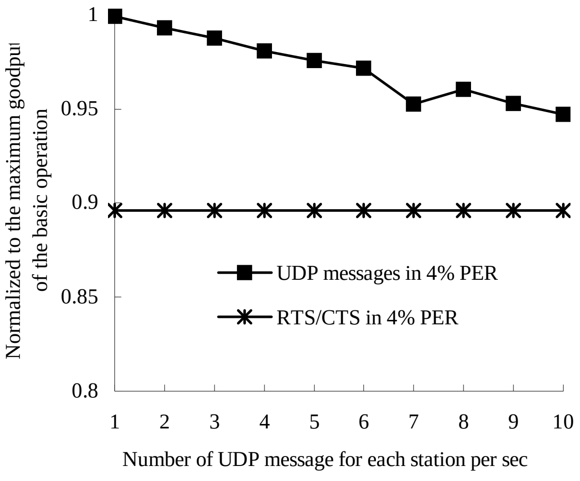

Fig. 12. UDP message overhead on goodput.

L_{opt} are 1500, 750, 500, and 300 bytes as we can see in Fig. 7. This allows us to use a small, efficient lookup table to map network conditions to a fragmentation size. We may further reduce the complexity of the system without performance degradation by setting a minimal packet length considered for fragmentation.

To do this mapping, the sender needs to estimate the SNR of the receiver, and the receiver uses UDP messages to update the sender on the receiver's SNR. Note that it is the sender, which sets the fragmentation threshold. Therefore, it does not incur any security problems, and the sender could be any of the wireless stations in WLANs (client stations, access points), or mobile nodes in peer-to-peer networks.

IV. EXPERIMENT

To verify the performance of the optimal fragmentation in a typical office environment, we executed extensive experiments with a CISCO Aironet 1231-G-A-K9 access point [13], servers, laptops, and a wired traffic generator as illustrated in Fig. 13. Each server generates 1500 bytes of MSDU, and establishes a TCP flow to one of the stations. With poisson-distributed inter-departure, each server sends 30 packets per second on average through the access point. As a result, we had three TCP flows through the access point to the stations, which is enough to saturate the maximum throughput of 1 Mbps for differential binary phase shift keying (DBPSK) in IEEE 802.11b PHY modes.

Then, we sliced the 1500 bytes MSDU into MPDUs to find the optimum fragment, L_{opt} . For each fragmentation threshold, we ran 20 trials of 100 seconds per trial for MPDUs of 300 through 1500 in 10 bytes increments for each given BER. To estimate the BER with our measured SNR, we used the BER equations found in [16] for a white Gaussian channel. The AWGN channel model is not realistic in WLANs, but this model is useful for reference purposes, since we do not have exact channel model of the experiment environment. For DBPSK,

$$P_{e,DPSK} = \frac{1}{2} e^{-E_b/N_0}, \qquad (16)$$

where E_b / N_0 is the SNR per bit.

For the first scenario, the three stations were placed such that the average SNR exceeded 15.42 dB to achieve 10^{-7} in (16). The experiment results in Fig.14 match up exactly to the proposed analysis. It clearly shows the overhead of the fragmentation, and importance of selecting L_{opt} carefully. In this channel, the optimal fragmentation threshold should be greater than 1500 bytes MSDU plus 28 bytes of MAC header, and 8 bytes of sub-network access protocol (SNAP) header.

It is interesting to see the first peak of the goodput at 750 bytes. If we slice the 1500 bytes payload of MSDU to 760 bytes, we have a slightly larger fragment more vulnerable to random errors, while sacrificing the same overhead as in 750 bytes. Likewise, for 740 bytes, although a shorter fragment is robust against random errors, we have one more overhead of $H + 2SIFS + ACK$, and this overhead has more influence on the goodput than the random errors.

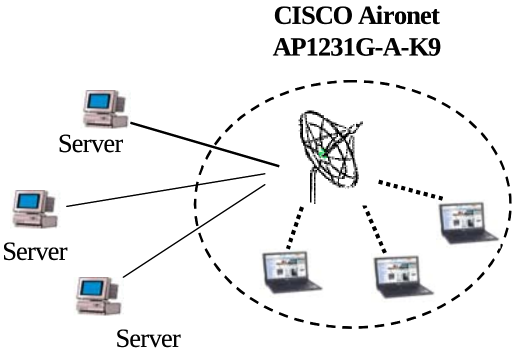

Fig. 13. WLAN topology for the experiment.

For the second scenario, we placed the stations in a typical WLAN environment, where an average SNR is 10.491 dB with standard deviation of 1.348 dB. This corresponds to a BER of 1.4×10^{-5} in (16). For comparison with the numerical results, the reference curve of 3×10^{-5} is drawn in Fig.14. This experimental results track the reference curve with an offset of 0.8 dB for an average SNR of 10.491 dB. This is reasonable if we consider the vulnerability of the low SNR channel against interferences in the office environment verses the ideal AWGN channel model in (16). For the BER of 3×10^{-5} , we observed the maximum goodput at an MPDU of 500 bytes, an optimal fragmentation threshold of 536 bytes when the MAC headers are considered. The total packet length to transmit on the air should be 560 bytes, which includes 24 bytes of physical-layer convergence protocol header and preamble. This result shows that the packet loss is dominant compared to the overhead of fragmentation with a worse BER. Given the number of users and BER in (13), the fragmentation thresholds that maximize the goodput are determined uniquely every case. By choosing these optimal values, we can achieve the optimum goodput for a given network environment.

To see the performance of the dynamic optimal fragmentation, we executed an experiment in various bit error rates. From a moderate BER of 1×10^{-5} to 5×10^{-5} in Fig. 15, we have found approximately 18.4% increase of the average goodput. In a situation where the channel becomes worse, for example, at BER of 10×10^{-5} , a 73% improvement in goodput can be obtained.

Since the 802.11 standard mandates all receivers support fragmentation, transmitters can apply this dynamic optimal fragmentation without any protocol change.

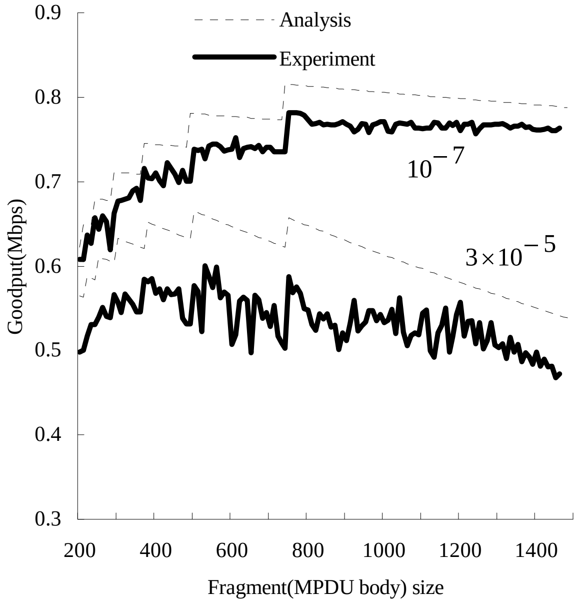

Fig. 14. Goodput with various MPDUs in WLAN.

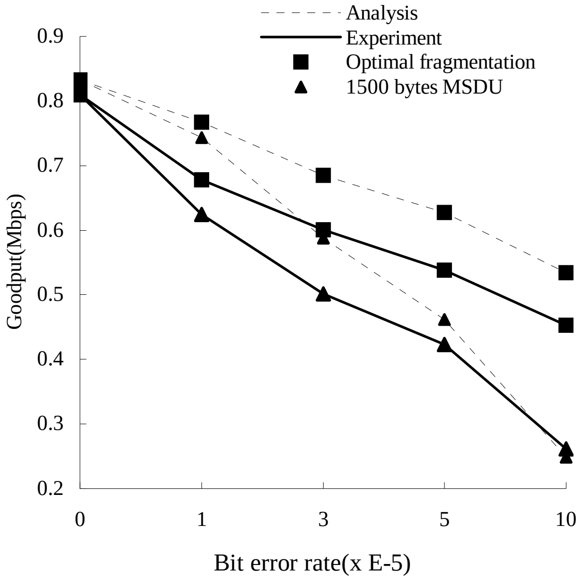

Fig. 15. Goodput with various bit error rates.

V. CONCLUSIONS

In this paper, we have presented an effective way to increase the system goodput of the IEEE 802.11 DCF. The proposed model reveals the impact of the number of contending stations, the packet collisions, the packet error probability, and the fragmentation overhead on the system goodput. Using an adaptive estimator, the algorithm dynamically selects optimally sized packets in time varying channels with minimal network overhead. Through rigorous analysis and extensive experiments, we have found the dynamic optimal fragmentation increases system goodput approximately 18.4% in a moderate BER and 73.1% at a BER of 10×10^{-5}. The proposed scheme is a comprehensive analytical model applicable to any CSMA/CA MAC protocol for next generation wireless networks, and a realistic approach that can be deployed without changing the IEEE 802.11 MAC protocol.

REFERENCES

[1] IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, P802.11, 1999.

[2] P. Lettieri and M. B. Srivastava, "Adaptive Frame Length Control for Improving Wireless Network Link Throughput, Range and Energy Efficiency," Proceedings of the IEEE INFOCOM 1998, March 1998.

[3] S. Ci and h. Sharif, "Adaptive approaches to enhance throughput of IEEE 802.11 wireless LAN with bursty channel," in Proc. IEEE LCN'00, Nov. 2000, pp. 44-45.

[4] J. Tourrihes, "Dwell adaptive fragmentation: How to cope with short dwells required by multidedia wireless LANs," in Proc. IEEE GLOBECOM'00, VOL. 1, 2000, PP. 57-61.

[5] J. Tourrihes, "Fragmentation adaptive reduction: Coping with various interferers in radio unlicensed bands," in Proc. IEEE ICC'01, VOL. 1, 2001, PP. 239-244.

[6] D. Qiao and S. Choi, "Goodput Enhancement of IEEE 802.11a Wireless LAN via Link Adaptation," Proc. IEEE International Conference on Communications, June, 2001.

[7] D. Qiao and S. Choi, and K. G. Shin, "Goodput ananysis and link adaptation for IEEE 802.11a wireless LANs," IEEE/ACM Transactions on Mobile Computing, vol. 1, pp. 278-291, Dec. 2002

[8] IEEE Std 802.11a-1999, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification:High-speed Physical Layer Extension in the 5GHz Band, P802.11, 1999.

[9] J. Yin, X. Wang and D.P. Agrawal, "Optimal Packet Size in Error-prone Channel for IEEE 802.11 Distributed Coordination Function," in Proc. The 2004 IEEE Wireless Communications and Networking Conference (WCNC'04), Atlanta, 21-25 March 2004.

[10] G. Holland, N. Vaidya, and P. Bahl, "A rate-adaptive MAC protocol for wireless networks," in *Proc. ACM MOBICOM'01*, Jul. 2001, pp. 236-251.

[11] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly, "Opportunistic meadia access for multirate ad hoc networks," in *proc. ACM MOBICOM'02*, Sep. 2002, pp.24-35.

[12] B. Kim, Y. Fang, T. Wong, and Y. Kwon, "Throughput Enhancement Through Dynamic Fragmentation in Wireless LANs," IEEE Transctions on Vehicular Technology, Vol. 54, No. 4, July 2005.

[13] Cisco Aironet 1231G-A-K9, http://www.cisco.com

[14] F. Cali, M. Conti and E. Grergori, Dynamic Tuning of the IEEE 802.11 Protocol to Achieve a Theoretical Throughput Limit, *IEEE/ACM Transactions on Mobile Compting*, Vol. 1, No. 1, Jan-Mar. 2002.

[15] J. L. Joshua, "Very Low Power Wireless Protocol Performance," M. S.thesis, Tufts University, May 2000

[16] T.S. Rappaport, "Mobile radio propagation: Large-scale path loss," in *Wireless Communications:Principles and Practices*. Upper Saddle River, NJ: Prentice-Hall, 1996, pp. 69-185