EXPLOITING OPPORTUNISTIC PACKET DELIVERY FOR RATE CONTROL IN 802.11 WIRELESS NETWORKS

Hoon Chang, Vishal Misra

Computer Science Department
Columbia University
New York, NY 10027
Email: {hchang,misra}@cs.columbia.edu

Dan Rubenstein

Electrical Engineering Department
Columbia University
New York, NY 10027
Email: danr@ee.columbia.edu

ABSTRACT

The throughput of the 802.11 protocol, which offers multiple transmission rates, is greatly influenced by the rate control algorithms. However, the high overheads associated with rate adjustment cause traditional rate selection mechanisms to select rates that are non-optimal. In this paper, we explore how packet combining can address this problem. Combining together and broadcasting packets originally destined different receivers exploits multiple opportunities of packet delivery and reduces the risk of poor choices of rates. As a result, our algorithm more aggressively seeks higher transmission rates and can adapt to fast changing channel conditions. Throughput enhancement in simulation indicates that throughput can be boosted by more than 50% on average in Rayleigh fading channels.

1. INTRODUCTION

The 802.11 wireless protocol [1] supports a wide range of transmission rates. Typical rate control algorithms [2, 3, 4] adjust the transmission rate based on retry numbers: the rate is incrementally increased (decreased) after the number of successful (failed) transmissions exceeds a certain threshold. The thresholds to increase and decrease the transmission rate are called the upscale and downscale thresholds.

Even though this threshold-based approach, when compared to throughput and FER-based rate control [4, 5], responds quickly to changing link conditions, the transmission can incur significant frame losses before an appropriate rate is found in an environment where there is channel fading. This is because channel fades cause sharp drops in the strength of a signal such that the rate needs to be decremented several times before transmissions are successful. To compensate, the upscale threshold is set to a large value so that the sender will increase its transmission rate in a conservative manner.

Our proposal is to send multiple application packets together within a single physical layer frame to save wasted bandwidth during this rate adaptation process. When the sender sends a packet to the receiver, it chooses a secondary packet to a different receiver and transmits them within the same physical layer frame. The secondary receiver should get the frame due to the broadcast nature of wireless channels. In order to select a packet as secondary, the sender consults the recent transmission history and the receiver who is expected to have the highest likelihood of successful transmission is chosen. This enhances transmission efficiency even when the frame is lost at the primary receiver. The increased efficiency makes it possible to implement more aggressive rate control by setting smaller upscale thresholds, since speculation of higher transmission rates has a lower cost than it would otherwise.

To implement our idea, Section 2 presents a packet combining algorithm that chooses the primary and secondary packets, as well as an ARQ protocol enhance the reliability of secondary packets. In Section 3, we demonstrate the improvement in throughput of our approach via extensive simulations in `Qualnet`. Section 4 discusses related work, including previous rate control schemes and packet multiplexing methods. Section 5 concludes our study.

This material was supported in part by the National Science Foundation under CAREER Award No. CNS-560153, ANI-0238299 and by support from Samsung, the Intel Information Technology Research Council, and Microsoft Research. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

2. SYSTEM OVERVIEW

2.1. Rate Control in 802.11

In this paper, we consider Lucent's Auto Rate Fallback (ARF) rate control mechanism [2]. The ARF protocol utilizes the threshold-based mechanism described in the introduction, with the upscale and downscale thresholds respectively set to 10 and 2. These settings enable the sender to quickly lower the rate when the signal strength drops, but slowly increase it when the strength increases.

Many research papers, including [3, 4], point out that the ARF protocol with large upscale thresholds is too conservative such that the rate used at any point in time is most often much lower than the optimal rate. However, the main reason that this conservative behavior is applied is to eliminate transmission losses that occur when transmitting at too high of a rate. As can be seen in Figure 1, such losses reduce the throughput more than can be made up in this protocol by attempting to keep the rate higher.

In Figure 1, the total channel throughputs of the ARF protocol with a small upscale threshold is plotted. For a given number of receivers, each simulation is repeated 10 times with different node placements and each measured throughput is plotted separately. Receivers are placed uniformly at random within 100×100 m^2 and moving to random way points. One sender, fixed at the geographic center of the square, transmits UDP packets to receivers. The maximum Doppler frequency is either 40 or 100 Hz for low speed movements. Throughputs are normalized to the default ARF upscale threshold of 10 with the same configurations.

Using a small upscale threshold, the frame error rate increases

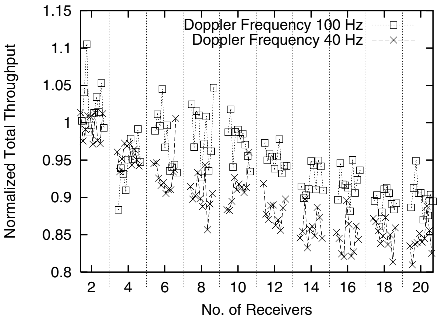


Fig. 1. Total throughputs with the upscale threshold 2.

due to frequent rate adaptation and the channel throughput drops. The negative effect of a small threshold increases with a growing number of receivers. Transmitting to many receivers, long gaps in time often transpire between successive transmissions for one receiver. Hence, the rate used for a previous transmission is more likely to fail for the next attempt, causing loss. This inaccuracy of the previous rate decision is another reason why employing a conservative approach to rate control is recommended, since slowly increasing the rate increases the probability of success of the following transmission.

2.2. Rate Control with Multiple Packets

If the time spent locating the optimal rate can be reduced, then higher throughputs can be achieved. To reduce the bandwidth waste, we propose to utilize *packet combining*. The idea is that when sending a packet to a *primary receiver*, the sender selects an additional packet for a *secondary receiver* whose channel conditions are good at that moment and transmits the two packets in a single physical layer frame. Even if the primary receiver fails to receive its packet, the secondary packet will likely be received, saving on bandwidth when searching for a good rate.

The primary packet is selected by the default scheduling policy such as FIFO, among packets in the queue. The secondary receiver is identified via a transmission history kept by the sender over the last T_h seconds. The sender identifies a successful transmission whose transmission rate is at least equal to the current rate (to be used by the primary transmission). If it has more than two candidates, the transmission with the most recent history is selected. Once a secondary receiver is chosen, the sender selects the first packet in the queue destined to this secondary receiver. If no secondary receiver can be found, the primary receiver is also used as the secondary receiver. Note that sending two packets for one receiver still reduces transmission overhead.

When the primary receiver gets the frame, it responds with an acknowledgment as in usual 802.11 transmissions. The sender adjusts its transmission rate according to the responses from the primary receiver. Without the secondary packet, the operation looks just like the original 802.11 DCF with a rate control algorithm.

If the secondary receiver gets the frame, it delivers the packet to the upper layer (if the packet is received in order) or keeps the packet in the reordering buffer. However, the secondary receiver does not acknowledge its reception. Receivers overhear all transmissions from the sender and simply take their packets sent as secondary. Because of the high delivery rate, instantaneous acknowledgments are not necessary and only a simple protocol for infrequent packet losses is needed. We propose a new ARQ protocol for primary and secondary packet losses and reordering.

2.3. ARQ Protocol with Minimal Overhead

As in the 802.11 protocol, the sender immediately retransmits the primary packet if not acknowledged. However, since the secondary packet is selected such that it will be delivered with high probability, the sender selects a new secondary packet when retransmitting the lost primary packet. For the transmitted secondary packet, instead, the sender starts a timer with timer value T_t. When the timer expires, the sender sends the secondary packet be resent as primary. Since the loss of the packet is caused by an outdated rate decision, the sender retransmits it as primary to invoke a new rate adjustment process.

Note that timer value T_t should be larger than time T_h. With smaller T_h, any receivers that have been selected only as secondary will become primary before the retransmission timer expires. As a primary receiver, it returns an acknowledgment for all received packets and that can confirm the reception of secondary packets to the sender.

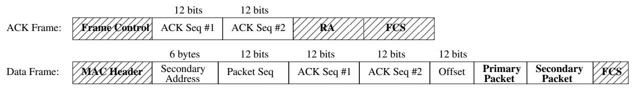


Fig. 2. ACK and Data frame format in our ARQ protocol.

Our new acknowledgment and data frame formats are depicted in Figure 2. For data frames, a new multiplexing header is added before the primary packet, which includes the address of the secondary receiver, the sequence number of the secondary packet, the offset to the secondary packet and two piggy-backed acknowledgment sequence numbers (for cases where the receiver is sending packets like TCP acknowledgments to the sender). The total length of the multiplexing header is only 12 bytes. Note that current technologies like Atheros Fast Frames [6] allow the transmission of $3,000$ data bytes (i.e., two Ethernet frames) within a single frame.

Our acknowledgment frames carry two sequence numbers because the sender transmits two packets together. These numbers identify the first two expected packets by the receiver. The sender retransmits at least the first one as primary. Once the first of lost packets is transmitted as primary, the sender can determine which packets should be retransmitted next from the following acknowledgment. Resending the packets specified in the acknowledgment as primary continuously causes other acknowledgments carrying the sequence numbers of the remaining lost packets. The sender can thus complete all needed retransmissions.

2.4. Performance Analysis

In this subsection, we explore how packet combining increases the average throughput of rate-adaptive sessions. Let us first compute the success probability of one transmission in Rayleigh fading channels. The success probability of a transmission equals $\pi_\Gamma \times P_l(\Gamma, T_p)$, where π_Γ is the probability that the received power at the beginning of the transmission is above a threshold Γ and $P_l(\Gamma, T_p)$ is the probability that a sufficient received signal power lasts for the full duration of the transmission, which is T_p. With additive Gaussian noise, π_Γ is equal to $\exp\left(-\frac{\Gamma}{\gamma_0}\right)$, where γ_0 is the average SNR. Using the finite-state model in [7], $P_l(\Gamma, T_p)$ equals $\exp\left(-\sqrt{2\pi} f_d T_p / \sqrt{\Gamma/\gamma_0}\right)$, where f_d is the maximum Doppler frequency.

Now, consider that the sender is adjusting the transmission rate. If the last transmission succeeds without error, the sender increases the transmission rate. Otherwise, the transmission rate decreases. Assume the sender adjusts the transmission rate from rate r_1 to r_2. Let the power thresholds for error-free reception at rates r_1 and r_2 be Γ_1 and Γ_2 respectively. After adjusting the rates, the probability that the received power at the receiver would be above the new power threshold Γ_2 is:

$$P_t(r_1 \rightarrow r_2) = \begin{cases} \frac{\pi_{\Gamma_2}}{\pi_{\Gamma_1}} & (\text{if } r_2 > r_1) \\ \frac{\pi_{\Gamma_2} - \pi_{\Gamma_1}}{1 - \pi_{\Gamma_1}} & (\text{if } r_2 < r_1). \end{cases} \quad (1)$$

Note that P_t is a conditional probability since with small T_p, we already know the received power is at least Γ_1 (if the last transmission has succeeded) or at most Γ_1 (if the last transmission has failed) at the point in time when the rates are changed.

Now, consider the probability of successful reception at the primary and secondary receivers. Let $T_p(n)$ be the transmission time of n combined packets. The probability of successful transmission for the primary receiver is the product of two probabilities where the received power is above a threshold and where the strong power lasts. Thus,

$$P_{primary} = P_t(r_1 \rightarrow r_2) \times P_l(\Gamma_2, T_p(n)). \quad (2)$$

For the secondary receiver, we assume that the last acknowledged transmission for the receiver was done T_w time ago. To account for more than 2 packets combining, we introduce multiple secondary receivers and assume that the last transmissions for the secondary receivers were continuously done. In other words, with $(n-1)$ secondary receivers, the last acknowledged transmission for secondary receiver i is assumed to have done $T_p(n) \times (i - 1) + T_w$ time ago. This permits optimistic results for combining multiple packets more than two, since the last transmissions are assumed to be done the most recently.

The success probability for a secondary receiver is at least the probability that a strong signal power lasts from the last transmission to the complete time of the current transmission, and can be approximated to this lower bound when T_w is short enough. Thus, assuming L is the packet length, we obtain the success probability and finally the throughput of n-packet combining as

$$P_{second}(i) \approx P_l(\Gamma_2, T_p(n) \times (i-1) + T_w) \quad \text{and} \quad (3)$$

$$\text{Thrp}(n) = \frac{L(P_{primary} + \sum_{i \leq n-1} P_{second}(i))}{T_p(n)}. \quad (4)$$

Figure 3 shows analysis results varying the average SNR and the number of combined packets. In analysis, we set L to 500 bytes and f_d to 100 Hz. Γ_{24} and Γ_{18} are 11.31 and 8.64 dB, which correspond to SNR thresholds for 24 and 18 Mbps in `Qualnet` simulator. We only present results when the transmission rate is changed from 24 to 18 Mbps and from 18 to 24 Mbps; results with different rates in 802.11a look similar. T_w is assumed to be 5 ms and the transmission times of 2 and 5 data packets at 18 Mbps are respectively 690.56 and 1357.22 μs. We assume the same average SNR value for all receivers in Figure 3(a) and 3(b).

Table 1. 802.11a Operation Parameters

Physical Layer		MAC Layer	
Frequency	5 GHz	MAC Protocol	DCF
Path Loss Model	Two Ray	Slot Time	9 μs
Shadowing Model	Constant	SIFS Time	16 μs
Shadowing Mean	4 dB	DIFS Time	34 μs
Fading Model	Rayleigh	CW_{min}	15
Temperature	290 K	CW_{max}	1023
Noise Factor	7	MAC Header	28 bytes
Tx Power	16 dBm	Mux. Header	12 Bytes
Rx Sensitivity	-87 dB	ACK Frame	15 bytes
Antenna	Omni.	Retry Number	7 times
Antn. Efficiency	0.8	Access Mode	Basic
Antenna Loss	0.5 dB	Connection	AP Mode
Antenna Height	1.5 m	Routing	AODV

As shown in Figure 3(a), throughput is increased most dramatically when the average SNR is smaller than the power thresholds. Even when the average SNR is larger, combining two packets still increases throughput. That is because the difference between the power thresholds of two adjacent rates in 802.11 are small. When the signal strength experiences a sharp drop, adjusting the transmission rate to the next lower rate is not sufficient; it must be lowered several levels for transmissions to succeed. Packet combining compensates for the losses during the adjustment.

Increasing the transmission rate (in Figure 3(b)), throughput worsens when more than two packets are combined. Note that actual throughput with multiple secondary packets should be worse because of the optimistic assumption. Adding too many secondary packets increases the total transmission time, thereby increasing the likelihood of transmission errors.

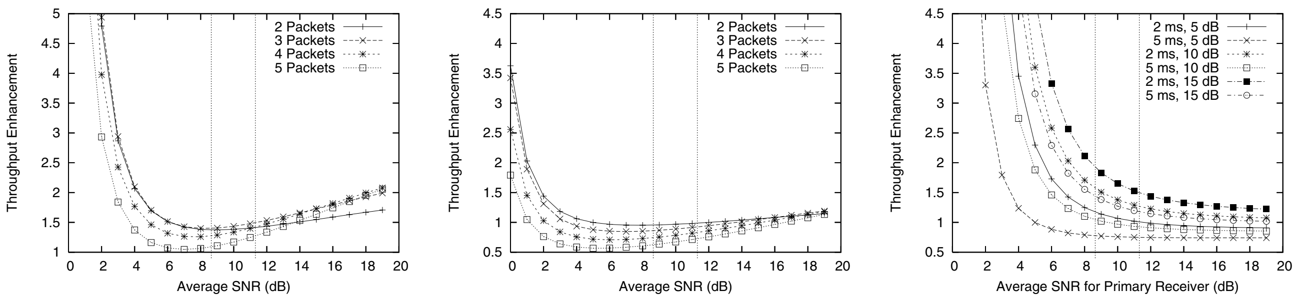
Figure 3(c) shows how the throughput improves as T_w is varied from 2 to 5 ms and the average SNR at the secondary receiver from 5 to 15 dB. The average SNR at the primary receiver is shown along the x-axis. Two-packet combining is assumed. The results demonstrate the importance of selecting a secondary receiver whose last transmission was the most recently successful. Selecting such a receiver can result in a better throughput even when its average SNR is worse. Shortly after a successful transmission, the receiver is likely to have good signal reception and sending a secondary packet to that receiver has a high likelihood of success.

3. SIMULATION

3.1. Simulation Parameters

We use `Qualnet` [8] for our simulations. In our simulations, multi-hop wireless connections are not considered; all packets go to receivers through a single access point. We run CBR applications, which generate 512-byte application packets to receivers at the same rate. The total generation rate is chosen so that the 802.11a channel is saturated. A 200-packet queue is used for incoming packets. T_h and T_h are set to 5 and 50 ms.

Receivers are placed uniformly at random in an area of size 100×100 m^2, with the access point placed in the geographic center. Receivers move toward random way points within the square, with two different maximum Doppler frequencies of 40 and 100 Hz. Other 802.11a simulation parameters are shown in Table 1.



(a) When decreasing Tx rate ($24 \rightarrow 18$)

(b) When increasing Tx rate ($18 \rightarrow 24$)

(c) When varying T_w and average SNR

Fig. 3. Throughput enhancement with packet combining. The ratio of throughputs with and without packet combining (i.e., $\mathrm{Thrp}(n)/\mathrm{Thrp}(1)$) is plotted. Required power thresholds Γ_{24} (11.31 dB) and Γ_{18} (8.64 dB) are also shown.

3.2. Throughput Enhancement

In simulation, Lucent's ARF protocol is used for rate control with the upscale threshold of 2. We normalize all throughput results in our figures with respect to those using the original 802.11 DCF and ARF protocol with the default threshold 10. Note that we already observed in Section 2 that using a small threshold reduces throughput due to the overhead of the rate adaptation process. When our packet combining and ARQ protocol is applied, however, the ability to adapt quickly enhances throughput, as in Figure 4(a).

Using our packet combining algorithm and ARQ protocol, total throughput is increased by 52% on average. Throughput is improved by 80% in some cases. Note that this improvement does not come at the expense of unfairness, as the average value of Jain's fairness index [9] is more than 0.99 in all of our simulations . Furthermore, the average end-to-end delay is around 80 ms even though our queue size is 200 packets.

To measure how well our algorithm utilizes the transmission history, packet reception rates are presented in Figure 4(b). Results in fast changing channel conditions (Doppler frequency 100 Hz) are shown. Primary packets in our protocol are dropped frequently due to the increased frequency with which rates are changed. However, secondary packets are successfully delivered 94.9% of the time on average, compensating for the primary losses. Figure 4(c) shows the average probability that at least one secondary receiver is available when combining. As the number of receivers goes up, the probability increases, approaching 0.8.

Figure 5 shows aggressive rate adaptations of our algorithm. We pick a 6-receiver example to show the distribution of transmission rates. The x-axis identifies the six receivers with their ID numbers. Each stacked bar shows the transmission number to the corresponding receiver. As shown in the figure, our algorithm enables high-rate transmissions; more than a half of frames are transmitted at 36 Mbps or higher while the ARF protocol with the default upscale threshold utilizes a lower rate most of the time.

Last, we note that the energy consumption of our algorithm is almost the same as that of the original ARF in all simulation runs. Figure 6 shows consumption normalized to that utilized by the original ARF protocol. The similar utilization is because the total transmission time of a larger frame is not much longer. Transmitting a two-packet frame at 54 Mbps requires only an additional

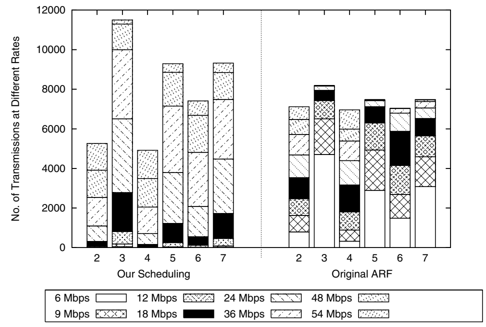


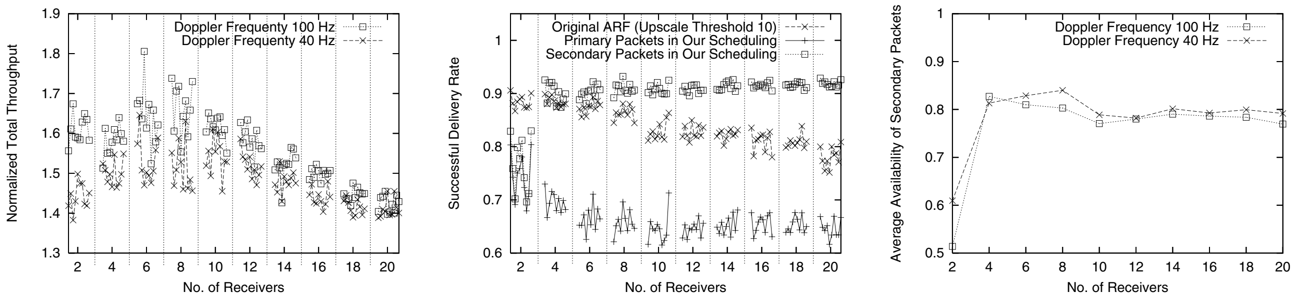
Fig. 5. Cumulative transmission numbers (f_d = 100 Hz).

80 μs. Moreover, the lower success probability for the primary receiver in our algorithm the frequency of transmitting due to the 802.11 backoff. The energy consumption of receivers is also similar, since they only transmit short acknowledgments.

4. RELATED WORK

Packet combining to reduce transmission overhead has been studied in the VoIP research community [10, 11]. Wang, *et al.*, presents multiplexing for VoIP applications in wireless LANs. Lin, *et al.*, introduces redundancy across packets in [12]. Both of the proposals transmit without acknowledgments from receivers and obtain significant throughput improvements. However, they do not consider the problem of rate control, and all multiplexed frames are transmitted at low rates to reach all their receivers because the sender does not get any feedback, such as 802.11 ACK frames.

Most rate control algorithms are based on the statistics of retry numbers. Lacage, *et al.* present an Adaptive ARF algorithm to adjust the upscale threshold [3]. In [13], Holland, *et al.* propose to use direct measurements of the link conditions. Haratcherev, *et al.* present a hybrid approach to take advantage of retry-based and SNR-based control methods [4]. Their algorithms apply to a single flow and focus on how to compute the optimal rate. One packet is transmitted at a time and multiple flows are not considered.



(a) Throughput enhancements

(b) Packet reception rates (f_d = 100 Hz)

(c) Secondary Packet Availability

Fig. 4. Simulation results. For each set of receivers, simulation is repeated 10 times with different node placements. Each data point shows a normalized throughput result to the one from the original ARF protocol.

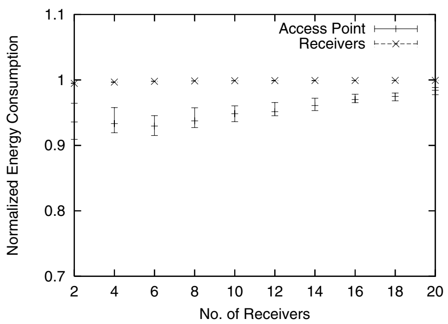


Fig. 6. Energy consumption (f_d = 100 Hz).

5. CONCLUSION

We have explored how to adjust the transmitting rate more aggressively without inducing significant losses, thereby increasing throughput. Typical rate control algorithms use large values for the upscale threshold and hence often transmit at a rate that is below the the ideal. We show that by implementing a novel packet combining algorithm and an ARQ protocol to support reliability, such large thresholds are not needed. As a result, the improved delivery rate of secondary packets can mitigate the inefficiency of the fast rate adjustment. To the best of our knowledge, this is the first work that considers rate adaptation involving multiple flows.

Simulations based on `Qualnet` show that our protocol can significantly enhance the total throughput by 52% on average for multiple UDP flows. Our scheduler chooses packets such that, on average, secondary packets are successfully delivered 95.1 % of the time. Even though packet losses rarely occur, we present a reliable ARQ protocol that would impose minimal overhead.

6. REFERENCES

[1] "Wireless LAN MAC and physical layer (PHY) specifications," IEEE Standard, 1999.

[2] A. Kamerman and L. Monteban, "WaveLAN-II: A high-performance wireless LAN for the unlicensed band," *Bell Labs Technical Journal*, pp. 118–133, 1997.

[3] M. Lacage, M. Manshaei, and T. Turletti, "IEEE 802.11 rate adaptation: a practical approach," in *MSWiM '04*, New York, NY, USA, 2004, pp. 126–134, ACM Press.

[4] I. Haratcherev, K. Langendoen, R. Lagendijk, and H. Sips, "Hybrid rate control for IEEE 802.11," in *MobiWac '04*, New York, NY, USA, 2004, pp. 10–18, ACM Press.

[5] B.E. Braswell and J.C. McEachen, "Modeling data rate agility in the IEEE 802.11a WLAN protocol," in *OPNETWORK 2001*, March 2001.

[6] Atheros Comm., "Super G: Maximizing wireless performance," Atheros White Paper, March 2004.

[7] Q. Zhang and S. Kassam, "Finite-state markov model for rayleigh fading channels," *IEEE Trans. on Communications*, vol. 47, no. 11, pp. 1688–1692, November 1999.

[8] Scalable Network Technologies, "Introduction to QualNet," Qualnet User's Manual, 2004.

[9] R. Jain, D. Chiu, and W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," DEC Research Report TR-301, Digital Equipment Corporation, September 1984.

[10] H. Sze, S. Liew, J. Lee, and D. Yip, "A multiplexing scheme for h.323 voice-over-ip applications," *IEEE Journal on Selected Areas in Comm.*, vol. 20, no. 7, pp. 1360 – 1368, 2002.

[11] W. Wang, S. Liew, and V. Li, "Solutions to performance problems in VoIP over a 802.11 wireless LAN," *IEEE Trans. on Vehicular Technology*, vol. 54, no. 1, pp. 366 – 384, 2005.

[12] C. Lin, H. Dong, U. Madhow, and A. Gersho, "Supporting real-time speech on wireless ad hoc networks: inter-packet redundancy, path diversity, and multiple description coding," in *WMASH '04*, New York, NY, USA, 2004, pp. 11–20, ACM Press.

[13] G. Holland, N. Vaidya, and P. Bahl, "A rate-adaptive MAC protocol for multi-hop wireless networks," in *MobiCom '01*, New York, NY, USA, 2001, pp. 236–251, ACM Press.