

# 802.11 Link Interference: A Simple Model and A Performance Enhancement

Hoon Chang and Vishal Misra

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

**Abstract.** Recent findings have revealed that the carrier-sensing range set in current major implementations does not detect all interference signals in 802.11 networks. In this paper, we investigate the effect of interference and develop a mathematical model for it. The accuracy of our model is verified via simulations. Based on an insight from our model, we present a simple protocol that operates on the top of 802.11 MAC layer and achieves more throughput than rate-adjustment schemes.

## 1 Introduction

It has been reported that IEEE 802.11 stations out of the carrier-sensing range of senders may interfere with receiver stations [5]. Consider two 802.11 senders, stations A and B that can not sense signals from each other but can interfere. During backoff times, two senders do not sense any active signals. They finally start their transmissions, which may overlap and collide. After collisions, the binary exponential backoff in 802.11 DCF grows their backoff windows up until one sender can send a packet without interference.

However, a binary exponential backoff does not provide fairness for a short term [3]. Assume sender B has successfully finished its transmission. A larger backoff window of sender A may allow sender B to capture the channel and Sender A's window size finally reaches the maximum. Note that packet corruption probability is not exactly 1, and sender A can eventually succeed and returns in early backoff stages. Their contention sends one of the senders to the maximum backoff window size again. Our simulation results support this observation and are the basis for our modeling approximation.

In the next section, we present the operational model and analysis on the behavior of 802.11 DCF under interference with two pairs of greedy senders and receivers. Section 3 presents our protocol to provide throughput enhancements. We finally conclude in section 4.

## 2 Modeling and Analysis

### 2.1 Operation Model and Upper Bound of Throughput

From our observation, we assume that after every successful transmission, a sender that has finished transmitting goes back to the first backoff stage while

the other sender moves to the last regardless of its current stage. In other words, one sender has the minimum backoff window ( $CW_{min}$ ) and the other has the maximum ( $CW_{max}$ ). If both senders make successful transmissions, then one sender is arbitrarily selected to go to the first stage.

The operational model has  $m$  states based on our analysis assumption. Each state corresponds to a pair of backoff stages of two senders in the network. In state  $(i, m)$ , one of senders stay in stage  $i$  and the other in the last stage  $m$ . In 802.11, their window sizes are  $(CW_{min} + 1) \cdot 2^{(i-1)} - 1$  and  $CW_{max}$ . If a collision occurs, two senders move to the next state  $(\min(i + 1, m), m)$  and increase their window size. If one of them makes a successful transmission, the senders go back to  $(1, m)$ . Analysis on this assumption can provide an upper bound of the channel throughput because setting two senders in the first and the last stages minimizes the conflict probability [1].

## 2.2 Average Transmission Time for a Single Packet

We present in this paper analysis for only basic accesses due to space limitations. Complete analysis for RTS/CTS accesses is found in [1].

Define very small virtual slots and assume that a sender in stage  $i$  starts a transmission at the beginning of a virtual slot with probability  $\lambda_i = 1/((W_i - 1)/2 \times T_{SLOT} + T_{DIFS})$ , where  $T_{SLOT}$  and  $T_{DIFS}$  are the length of 802.11 slots and DIFS in unit of virtual slots. At each slot, transmission probability  $\tau_i$  of two senders in state  $(i, m)$  is  $\lambda_i + \lambda_m - \lambda_i \lambda_m$ . Let total transmission time of a  $B$ -bit packet be  $T_{TX}$ , which includes time to send a data packet ( $T_{DA}$ ) and an ACK. For the average collision time, we assume that the second transmission starts at the beginning of any virtual slot within  $T_{DA}$  and it follows the uniform distribution. Let  $T_{OVER}$  be the average.

Given  $\lambda_i$  and  $\lambda_m$ , the probability  $p$  where a transmission is completed without overlapping is  $(\lambda_i \cdot (1 - \lambda_m)^{T_{DA}} + \lambda_m \cdot (1 - \lambda_i)^{T_{DA}})/\tau_i$ . Suppose that at least one sender begins a transmission in state  $(i, m)$  and let  $p_i$  be the conditional probability of successful transmissions occurring. We obtain  $p_i = p + (1 - p) \times (1 - P_{PER}^2)$ , where  $P_{PER}$  is error rate of an overlapped packet. Note that  $P_{PER} \leq 1$  and overlapped transmissions result in safe deliveries with probability  $(1 - P_{PER}^2)$ . Let  $q$  be the average number of packets that are overlapped but delivered.

Now we obtain  $C_j$ , which is time to reach stage  $(j, m)$ .  $C_1$  is equal to 0.

$$C_m = C_{m-1} + 1/\tau_{m-1} + T_{OVER} + (1/\tau_m + T_{OVER}) \times (1 - p_m)/p_m \quad (1)$$

$$C_i = C_{i-1} + 1/\tau_{i-1} + T_{OVER} \quad (2 \leq i \leq m - 1) \quad (2)$$

Note that senders must stay in the same state after conflicts in  $(m, m)$ , which happens  $(1/p_m - 1)$  times on average. Now consider the average delivery time  $S_i$  of one packet in state  $(i, m)$ . To reach the state  $(i, m)$  from  $(1, m)$  takes  $C_i$  and successful transmissions in  $(i, m)$  happen with probability  $p_i$ . Assuming successful transmissions, a successful transmission without overlapping occurs at  $p/p_i$  and takes  $1/\tau_i + T_{TX}$ . With overlapping,  $q$  packets are delivered in time

$1/\tau_i + T_{OVER}$ . Thus,  $S_i$  is:

$$S_i = p/p_i(C_i + 1/\tau_i + T_{TX}) + (1 - p)/p_i(C_i + 1/\tau_i + T_{OVER})/q \quad (3)$$

Now, let  $T$  be the average delivery time of one packet. We obtain  $T$  from  $S_i$  as follows and the throughput in this system is  $B/T_0$ .

$$\begin{aligned} T &= S_1 \times p_1 + (1 - p_1) \times [S_2 \times p_2 + (1 - p_2) \times (\dots)] \\ &= S_1 p_1 + \sum_{i=2}^{m-1} S_i p_i \prod_{j=1}^{i-1} (1 - p_j) + S_m \prod_{j=1}^{m-1} (1 - p_j). \end{aligned} \quad (4)$$

### 2.3 Simulation in Qualnet

In **Qualnet** [4], we used two-ray path loss model. From two-ray path loss model and BER mapping function in **Qualnet**, we can compute the acceptable range of SINR values and the interference range. 802.11 stations were placed at locations where packet loss probabilities for simulations can be achieved. All stations were stationary with omni-directional antennas and located on a flat plain without any obstacles. Fading models were not used.

Figure 1(a) shows the simulation results in terms of the cumulative channel throughput with varying packet size. Simulation result for RTS/CTS accesses is presented in [1]. Two senders and receivers transmitted data packets and ACKs at 11 Mbps. Each simulation, which runs with fixed packet size in the range of 100 to 1500 bytes, was performed for 100 seconds. We also varied the long and short retry numbers in 802.11; the graph tagged ‘BASIC (50)’ shows results setting both of long and short retry limits to 50. The operation with default retry numbers are also plotted in the graph ‘BASIC (7)’.

As expected in section 2, analysis results put the upper bounds on the channel throughput, that look very tight. By increasing retry numbers up to 50, the channel throughput with sending 1500-byte packets hit 98.28 percent of the expectation. This indicates that setting large retry numbers keeps one of senders in the last backoff stage longer than in the regular cases and it boosts up the chance to make a successful transmission.

## 3 Throughput Enhancement

The key idea of this enhancement protocol is emulating the analysis model to reduce conflicts and guarantee bounded waiting time as much as possible. When the protocol detects a packet loss, it delays supplying packets to the MAC layer for a random backoff time in the range of 400 to 2000 slots. After that, it tries to (re)transmit a packet until the packet delivered or it moves back to the delay state after a timeout. If it has delivered at least one packet and detects a loss again, it transits to the delay state. Full state diagram is found in [1].

Figure 1(b) shows performance of our protocol when RTS/CTS probing is used. Simulation result for basic accesses is present in [1]. As expected, our

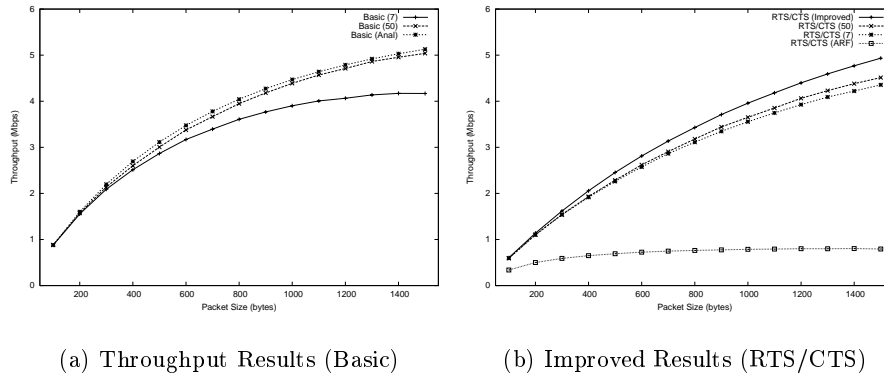


Fig. 1. Simulation Results at 11 Mbps

protocol enhanced the system throughput more than simple increase of retry numbers. We also run Lucent AutoRate Fallback (ARF) protocol [2] and denote the results *ARF* in the figure. Due to long consecutive losses, senders in ARF reduce the transmission rate until it reaches 1 Mbps, where transmissions are more vulnerable. Senders, thus, do not increase their transmission rates while our protocol effectively reduces conflicts and enhances the throughput.

## 4 Conclusion

In this paper, we investigated the behavior of 802.11 DCF on the interference channel. We presented our analysis model and performed simulations that show our analysis provides tight upper bounds. According to our analysis, delaying packets from the upper layer will mitigate the impact of packet conflicts. We presented a simple protocol that increases the system throughput by 30.9% while Lucent's ARF protocol, one of well-known rate adaptation schemes achieves much smaller than regular 802.11 DCF throughput without rate adjustment.

## References

1. Hoon Chang and Vishal Misra. 802.11b throughput with link interference. Technical report, Columbia University, 2004.
2. A. Kaman and L. Monteban. WaveLAN-II: A high-performance wireless LAN for the unlicensed band. *Bell Labs Technical Journal*, pages 118–133, 1997.
3. K. K. Ramakrishnan and Henry Yang. The ethernet capture effect: Analysis and solution. In *the IEEE 19th Local Computer Networks*, 1994.
4. Scalable Network Technologies. QualNet Developer, 2004. <http://www.qualnet.com>.
5. Kaixin Xu, Mario Gerla, and Sang Bae. How effective is the IEEE 802.11 RTS/CTS handshake in ad hoc networks? In *IEEE Globecom*, 2002.