802.11b Throughput with Link Interference

Hoon Chang, Vishal Misra

***Abstract*— IEEE 802.11 MAC is a CSMA/CA protocol and uses RTS/CTS exchanges to avoid the hidden terminal problem. Recent findings have revealed that the carrier-sensing range set in current major implementations does not detect and prevent all interference signals even using RTS/CTS access method together. In this paper, we investigate the effect of interference and develop a mathematical model for it. We demonstrate that the 802.11 DCF does not properly act on the interference channel due to the small size and the exponential increment of backoff windows. 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.**

I. INTRODUCTION

It has been reported that the carrier-sensing range and the interference range in IEEE 802.11b [4] are not matched. The interference range is usually larger and stations out of the carrier-sensing range may interfere with other stations receiving packets. This scenario occurs in many real world scenarios. They include urban environments that are dense in private 802.11 networks, which makes perfect channel partitioning infeasible; and also multi-hop 802.11 networks that have been used in extending network access to regions that have limited accessibility or infrastructure. This paper formally investigates interference problems caused by stations out of the carrier-sensing range. The main contributions of our paper are:

1) A mathematical model for the interference problem in 802.11 networks
2) Simple protocol modifications based on our modeling insights to boost performance

Our protocol enhancements are also compared with commercial solutions (Lucent's ARF) and we achieve significantly better performance.

Several analytical models for 802.11 have been proposed. Bianchi [1] provides an analytical model to compute the 802.11 DCF throughput in the assumption of finite stations and ideal channel conditions. The backoff behavior of a tagged station is described as a discrete Markov model. However, they assumes perfect channel model and requires perfect CCA (clear channel assessment) function, making the model inapplicable to the interference problem.

Hoon Chang and Vishal Misra are with the Department of Computer Science, Columbia University. {hchang,misra}@cs.columbia.edu

Kim and Hou's work [7] also models 802.11 DCF operations. It describes packet transmissions as the MAC fluid and expresses the average transmission attempt rate λ. From the analysis, they suggest an optimizing algorithm that gives artificial delays supplying data packets to MAC layer. This suggestion assumes ideal channel conditions. Moreover, they introduces artificial delay before sending every packets to increase the system throughput; our analysis indicates seperate waiting and transmitting states will achieve higher performance

Chhaya and Gupta [2] analyze the throughput and fairness issues of the DCF function concerning the effect of hidden terminals and capturing. They assume that a successful transmission will be completed only if no station within the capture area of a receiver and a sender do not transmit and exclude the possibility of parallel transmissions.

Xu *et al* [5] point out that 802.11 procotols can not prevent all interference as expected in theory. To reduce interference, they propose to limit the transmission range by having receivers closer to a sender, which does not always work in all possible cases.

Li *et al* [8] study the influence of large interference range to the ad hoc network capacity. They focus on the ideal and theoretical maximum capacity that depends on network topologies but do not try to analyze in detail.

Cali *et al* [3] investigate criteria to improve the protocol capacity of a IEEE 802.11 network by tuning its backoff algorithm. They adopt the p-persistent backoff algorithm to show that it is possible to tune at run time the backoff window size to obtain a capacity very close to the theoretical limit. However, they also assume that all stations can sense and detect the medium state.

To prohibit interference sources from emitting signals, Qiao *et al* [9] propose to transmit CTS packets at stronger power level in order to reduce interference range. Their simulation results, however, show that the proposed scheme would gain less than simple physical rate adaptation.

The rest of our paper is structured as follows: Section II describes 802.11 DCF. In section III, modeling and analysis of the interference problem is presented. Section IV presents simulation results with the network simulator *Qualnet* to support the analysis in the previous section. In Section V we present our protocol to provide thoughput enhancements. We finally conclude in section VI.

II. 802.11 DCF WITHOUT CARRIER-SENSING MECHANISM

Carrier-sensing mechanism plays an important role for this random backoffing in CSMA/CA. A sender selecting the smallest backoff time usually wins the contention and is guaranteed to monopolize the medium for its one transmission. After the sender starts transmission, the others must sense the busy medium and defer their pending transmissions. They should suspend their back-off timers and resume the timers after the medium is released.

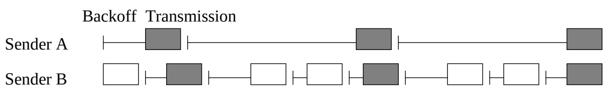

Fig. 1. Contiguous Conflict Problem

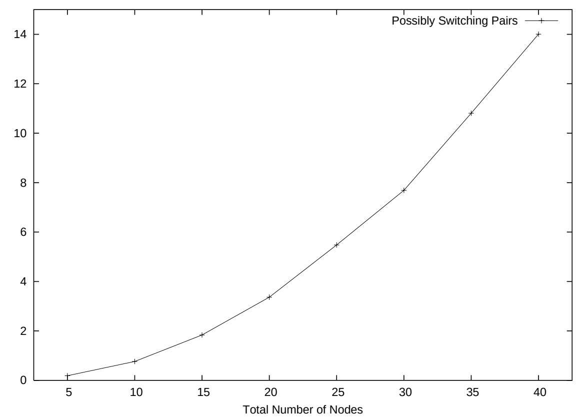

Fig. 2. Contiguous Conflict Problem

If Sender A and B as shown in Figure 1 cannot not sense the busy medium or signal from each other, they start their transmission after their backoff timer has expired. Note that in Figure 1, shaded boxes correspond to corrupted packets and white ones to packets safely delivered. Lines ending before boxes indicate backoff time. In this case, a successful transmission is not achieved until one of two senders chooses a backoff time value longer than the other one's packet transmission time. Otherwise, transmissions conflict and both of them move forward to their next backoff stage.

The first few backoff stages of 802.11 DCF certainly do not provide window size big enough to resolve these kind of conflicts. For smaller packets transmitted at higher speeds, large difference between backoff window sizes causes another problem. Let two senders be in the last stage. If one of them makes a successful transmission, it moves back to the first backoff stage and selects its backoff time from its backoff window, which is much smaller than the other's current window. As shown in Figure 1, a likely scenario is that sender B is able to make several successful transmissions because of its shorter backoff window. Moreover, sender B is able to recover from the occasional collisions by being in an early backoff stage and retransmitting while sender A in the last backoff stage waits.

However, this does not indicate that sender A starves forever. Since packet corruption probability is not exactly 1, sender A can succeed with a small probability and goes back to the first backoff stage. Two senders are now in early backoff stages and their contention leaves one of senders in the last stage again. This leads to short term unfairness, however, by repeating this process for a long time, the channel is very fairly shared while senders suffer large variation of delivery time. Our simulation results support this observation and are the basis for our modeling approximation.

Figure 2 plots the number of pairs of two senders that can not sense but may interfere with each other. Nodes were placed in an 2000-by-2000 square meters area. Carrier-sensing and transmission ranges is set to them in `Qualnet`. The results show that a significant number of nodes could experience this switching process.

III. MODELING AND ANALYSIS

A. Operation Model and Upper Bound of Throughput

In this section, we develop a model and perform analysis with four 802.11 stations to compute the system throughput. We assume two of them are greedy senders and the other two are receivers. Senders can not sense each other and they are in the interference ranges of receivers. If senders use RTS/CTS accessing, receivers do not either sense a signal or receive packets from senders that they are not associated with. Let P_{PER} be the average error probability from given packet size, where each of two packets will be corrupted when transmissions collide. In the analysis, P_{PER} is assumed to be close to 1.

To simplify the whole analysis, we make three assumptions. The first one is 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 stage regardless of its current backoff stage. If both senders make successful transmissions, then one of senders is arbitrarily selected and goes to the first stage; and the other sender goes to the last stage. This is based on our key observation that in scenarios that we are studying, senders are typically found either in the first backoff stage or the last one. Second, we assume that each conflict involves only two transmissions. Two senders wait until the end of conflicted transmissions and start the backoffing process. Last, we assume that time is divided into small discrete slots, and the two senders start transmission in each slot with some probability instead of selecting a random backoff values. That is, we assume that the distribution of backoff values in 802.11 approximates a geometric distribution, and the expected length of the geometric distribution is exactly equal to the expected length in the real backoff process in 802.11.

Clearly, this sort of stage transitions does not match with 802.11 DCF behavior. Analysis on this assumption, however, can provide a reasonable upper bound of the average throughput because this process will minimize the average length of conflict phases and maximize that of transmission phases. Let a conflict phase be time from the first conflict of sequential collisions to the first successful transmission. Let a transmission phase start just after a successful transmission and end before the first conflict after that. Figure 3 depicts an example of two phases. Let L be the packet transmission time in the unit of 802.11 time slots, which is longer than the first-stage backoff window size and shorter than a half of the last-stage window size. Most of packets transmitted at 11 and 5.5 Mbps comply with this.

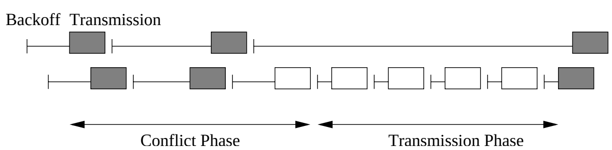

Fig. 3. Conflict and Transmission Phase

After making a successful transmission, a transmission phase starts. A sender in the first stage will choose a new backoff value and start its next transmission after its timer expires. Because L is larger than the sender's window size, a transmission from the other sender in the last stage will collide with one from the first sender. Keeping the second sender in the last stage and making it restart backoff process after every successful transmission will reduce transmit probability of the sender to the minimum and reduce conflict probability. Thus, by the first assumption, our model achieves the maximal average length of transmission phases.

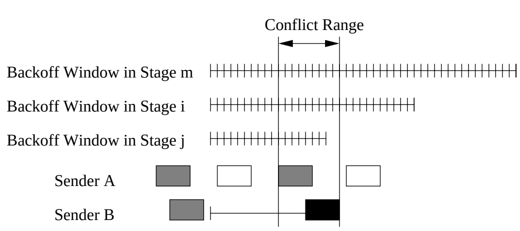

Fig. 4. Conflict Probability after the First Collision

When a collision occurs, a conflict phase begins. Let sender A in Figure 4 be in stage i and sender B in stage j, where $0 \leq i, j \leq m$. Since this is the first conflict after at least one successful transmission, one of two senders must be in the first stage at the beginning of the conflict. Let sender B be in the first stage at the first conflict. Since we assume that a conflict involves only two transmissions, each sender makes a single transition to the next backoff stage at every conflict. Thus, i is always greater than or equal to j in the whole conflict phase.

At the end of one conflict, sender B chooses one backoff value. In Figure 4, a black box depicts the next packet transmission scheduled by sender B. *Conflict Range* represents the range of backoff values where a transmission from sender A would conflict. Assume stage i has backoff values beyond the conflict range. The number of values in the conflict range is $2L - 1$. The conflict probability in stage i, P_i is $(2L - 1)/W_i$, where W_i is the window size of stage i. The probability in stage m is P_m, which is smaller than P_i.

Assume that stage i does not have backoff values beyond the conflict range. Note that now $j \leq i < m$. Let sender A have $(L + x)$ backoff values in the conflict range, where $0 \leq x < L$. Then, the conflict probability P_i is $(L + x)/W_i$. Since $W_m \geq 2(W_i + 1)$, $P_m \leq (2L-1)/(2(W_i+1)) < L/W_i < P_i$. Thus, P_m is always smaller than P_i and the model, switching between the first and the last stage will minimize the average length of conflict phases.

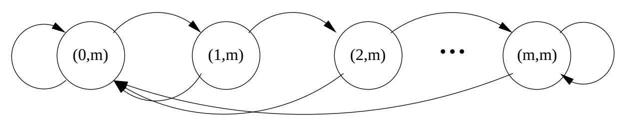

Fig. 5. Backoff Stages in Analysis Model

Figure 5 shows $(m + 1)$ backoff stages based on our analysis assumption. Each stage corresponds to a pair of backoff stages of two senders in the network. In stage (i, m), one of senders stay in the last stage m and the other in stage i, where $0 \leq i \leq m$. If a collision occurs, two senders move to the next stage $(\min(i + 1, m), m)$. If one of them makes a successful transmission, the senders go back to the first stage, $(0, m)$.

In the next section, we compute the average time to deliver a single packet. Let the average time of a successful delivery, T_i be the average time from two senders enter stage (i, m) until they come back to the first stage $(0, m)$, and let B be the data size in a packet. Then, the average throughput is given by B/T_0.

B. Average Transmission Time for a Single Packet

1) Basic Accesses: 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. Since one of two senders is in stage i and the other in stage m, transmission probability τ_i of two senders in stage (i, m) is $\lambda_i + \lambda_m - \lambda_i \lambda_m$ and the average idle time before starting a transmission from any of two senders is $1/\tau_i$.

Let total transmission time of a B-bit packet be T_{TX} in unit of virtual slots, which includes time to exchange a data packet and an ACK and time of SIFS in the middle of the exchange. Let also T_{DA} be the transmission time of one data packet. During a packet being transmitted, if the other sender keeps waiting and idle, this transmission is successfully done with an ACK from a receiver. Otherwise, a transmission from the other overlaps the first transmission. With error rate of interfered packets, P_{PER}, each transmission fail at P_{PER} or succeed at $1 - P_{PER}$. Given that at least one sender begins a transmission, let p_i be the conditional probability of at least one successful transmission occurring in stage (i, m). We obtain p_i as follows:

$$\begin{aligned} p &= (\lambda_i \cdot (1-\lambda_m)^{T_{DA}} + \lambda_m \cdot (1-\lambda_i)^{T_{DA}})/\tau_i \\ p_i &= p + (1-p) \times (1 - P_{PER}^2) \end{aligned} \quad (1)$$

Note that if P_{PER} is not one, overlapped transmissions result in safe deliveries with probability $(1 - P_{PER}^2)$. In this case, $(1 + (1-P_{PER})^2/(1-P_{PER}^2))$ packets are delivered on the average. Let q be this average.

Let us now consider the average delivery time in stage (i, m). Without overlapping, a successful transmission including acknowledging takes $1/\tau_i + T_{TX}$. When the second transmission overlaps the first one, the second one starts at the beginning of any virtual slot within T_{DA} and it follows the uniform distribution. Let T_{OVER} be the average of the total transmission time of two overlapped packets including ACK timeouts. Computing the conditional expectation on the overlapping, T_{OVER} is:

$$\begin{aligned} T_{OVER} &= T_{TX} + \sum_{j=0}^{T_{DA}-1} (j/T_{DA}) \\ &= T_{TX} + (T_{DA} - 1)/2 \end{aligned} \quad (2)$$

To get the upper bound of the throughput, we take the minimal ACK timeout value equal to ACK transmission time plus time of SIFS. Let C_{i-1} be time to reach stage (i, m). Then,

$$C_{m-1} = C_{m-2} + (1/\tau_i + T_{OVER}) \times (1-p_m)/p_m \quad (3)$$
$$C_i = C_{i-1} + 1/\tau_i + T_{OVER} \quad (4)$$
$$C_1 = 1/\tau_1 + T_{OVER}, \quad (5)$$

where $1 < i < m - 1$. Note that senders must stay in stage (m, m) after conflicts, which happens $(1-p_m)/p_m$ times on average.

Now consider the average delivery time of one packet from stage $(1, m)$, assuming it is successfully delivered in stage (i, m). Let S_i be this average. To reach the stage (i, m) from $(1, m)$ takes C_i and successful transmissions in stage (i, m) happen with probability p_i. Assuming successful transmissions, a successful transmission in stage (i, m) 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:

$$\begin{aligned} S_i &= p/p_i \times (C_{i-1} + 1/\tau_i + T_{TX}) + (1-p)/p_i \times \\ &\quad (C_{i-1} + 1/\tau_i + T_{OVER})/q \end{aligned} \quad (6)$$
$$\begin{aligned} S_1 &= p/p_1 \times (1/\tau_1 + T_{TX}) + (1-p)/p_i \times \\ &\quad (1/\tau_1 + T_{OVER})/q, \end{aligned} \quad (7)$$

where $1 < i \leq m$.

Now, let T be the average delivery time of one packet. We obtain T from S_i for $1 \leq i \leq m$ as follows and the throughput in this system is B/T_0, where each packet carries B bytes of data.

$$\begin{aligned} T &= S_1 \times p_1 + (1-p_1) \times \\ &\quad [S_2 \times p_2 + (1-p_2) \times (\ldots)] \\ &= S_1 \cdot p_1 + \sum_{i=2}^{m-1} S_i \cdot p_i \cdot \prod_{j=1}^{i-1} (1-p_j) + \\ &\quad S_m \cdot \prod_{j=1}^{m-1} (1-p_j) \end{aligned} \quad (8)$$

2) RTS/CTS Accesses: Assume senders probe the channel with RTS/CTS messages. Let T_{TX}^{rts} be transmission time for a RTS and a CTS packet as well as a data packet, an ACK and three SIFS intervals between them, and T_{RCD} be equal to $T_{RCS} + T_{SIFS} + T_{DA}$, where T_{RCS} is time to send a RTS and a CTS packet including SIFS time. Without overlapping, a transmission is safely done and takes $1/\tau_i + T_{TX}^{rts}$ time units. Let p^{rts} be the probability where only single transmission would happen.

There are two categories of overlappings for RTS/CTS accesses as shown in Figure 6. One is that two senders begin transmitting RTSs before either one get a CTS from their receiver. The other is that one of them already have completed sending a RTS.

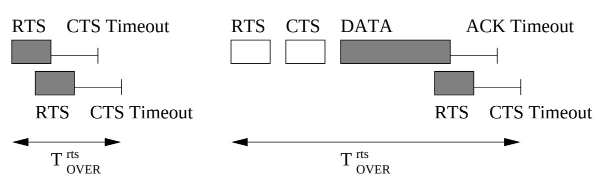

Fig. 6. Two kind of overlaps for RTS/CTS accesses

Let P_{PER}^{rts} be probability where a RTS is corrupted. Since RTS packets are much shorter than data packets,

P_{PER}^{rts} should be smaller than P_{PER}. The first overlapping in Figure 6 ends in a collision with probability $(P_{PER}^{rts})^2$. With probability $2(1-P_{PER}^{rts})P_{PER}^{rts}$, either one of them continues and transmits a data packet. Two RTSs will be delivered safely at $(1-P_{PER}^{rts})^2$. All possible scenarios where two RTSs overlap are depicted in Figure 7.

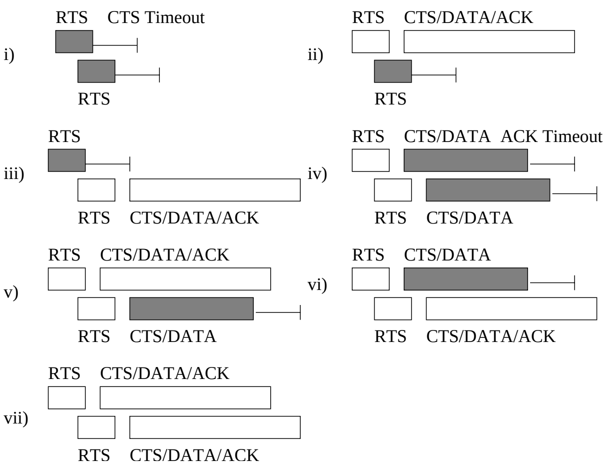

Fig. 7. RTS Packet Overlaps

In Figure 7, it is easy to observe that both of transmissions fail only when two RTSs are corrupted or two data packets are corrupted as in case i) and case iv). Thus, when RTSs from two senders overlap, v^{rts}, the probability of at least one successful transmission is:

$$v^{rts} = 1 - ((P_{PER}^{rts})^2 + (1-P_{PER}^{rts})^2 \cdot P_{PER}^2) \tag{9}$$

Let us consider the average overlapping time. In the first case in Figure 7, the average overlapping time is $T_{RCS} + (T_{RTS} - 1)/2$, since the starting time of the second RTS packet follows the uniform distribution within the first RTS transmission time. For case ii), the average overlapping time is exactly equal to T_{TX}^{rts}. The average overlapping time in all other cases in Figure 7 is $T_{TX}^{rts} + (T_{RTS} - 1)/2$ by the similar computation for case i). Let T_{OVER_1} be the first average overlapping time and T_{OVER_2} be the second one.

Now, let C_{i-1}^{rts} be the average time to reach stage (i, m) and $S_i^{rts}(R^2)$ be the average delivery time of a data packet for all success cases in Figure 7. Then, $S_i^{rts}(R^2)$ is:

$$\begin{aligned}
r &= (1-P_{PER}^{rts})P_{PER}^{rts}/v^{rts} \\
s &= 1 + (1-P_{PER}^{rts})^2 \times \\
&\quad (1-P_{PER})^2/(v^{rts}(1-r)) \\
S_i^{rts}(R^2) &= r \cdot (C_{i-1} + 1/\tau_i + T_{TX}^{rts}) + \\
&\quad (1-r) \cdot (C_{i-1} + 1/\tau_i + T_{TX}^{rts} + \\
&\quad (T_{RTS} - 1)/2)/s && (10) \\
S_0^{rts}(R^2) &= r \cdot (1/\tau_0 + T_{TX}^{rts}) + \\
&\quad (1-r) \cdot (1/\tau_0 + T_{TX}^{rts} + \\
&\quad (T_{RTS} - 1)/2)/s, && (11)
\end{aligned}$$

where $0 < i \le m$.

Note that r is the conditional probability where case ii) happens, given that at least one successful transmission is done. s is the average number of packets delivered, given that at least one successful transmission is done and case ii) does not happen.

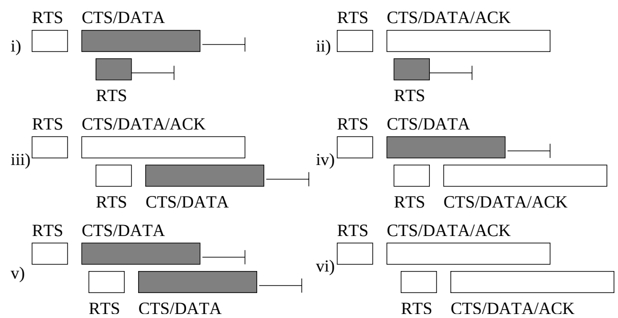

Fig. 8. RTS/Data Packet Overlaps

Figure 8 shows all possible cases for overlapping of a RTS and a data packet. Among all six cases, the first and the fifth cases are the only that have both transmissions finish with an error. Let u^{rts} be the success probability for overlapping of a RTS and a data packet. Then,

$$\begin{aligned}
u^{rts} &= 1 - ((P_{PER}^{rts})^2 + (1-P_{PER}^{rts}) \cdot P_{PER} \cdot \\
&\quad (P_{PER}^{rts} + (1-P_{PER}^{rts}) \cdot P_{PER})) && (12)
\end{aligned}$$

Note that in case v) of Figure 8, the data packet of the first transmission would be corrupted by the RTS or the data packet of the second. Thus, the corrupting probability of the first data packet is either P_{PER}^{rts} by the second RTS packet) or $(1-P_{PER}^{rts}) \cdot P_{PER}$ by the second data packet.

Let us compute the average length of overlappings. The first and second cases will have the same average. The other four cases will also have the same average length. Repeating the similar steps used above, we compute the average length of overlappings as follows:

$$\begin{aligned}
T_{OVER_3} &= T_{TX}^{rts} + T_{RTS} \times \\
&\quad (T_{RTS} - 1)/(2(T_{RCD} - T_{RTS})) && (13) \\
T_{OVER_4} &= T_{TX}^{rts} + T_{RTS} + \\
&\quad (T_{RCD} - T_{RTS} - 1)/2 && (14)
\end{aligned}$$

Let $S_i^{rts}(RD)$ be the average delivery time of a data packet for all success cases in Figure 8. Then, $S_i^{rts}(RD)$ is:

$$\begin{aligned} t &= P_{PER}^{rts}(1-P_{PER}^{rts}) \\ u &= (1-P_{PER}^{rts})(1-P_{PER}) \times \\ &\quad (1+(1-P_{PER}^{rts})P_{PER}) \\ v &= 1+(1-P_{PER}^{rts})^2(1-P_{PER})^2/u \\ S_i^{rts}(RD) &= (t\cdot(C_{i-1}+1/\tau_i+T_{OVER_3})+ \\ &\quad u\cdot(C_{i-1}+1/\tau_i+ \\ &\quad T_{OVER_4})/v)/u^{rts} \qquad (15) \\ S_0^{rts}(RD) &= (t\cdot(1/\tau_0+T_{OVER_3})+ \\ &\quad u\cdot(1/\tau_i+T_{OVER_4})/v)/u^{rts} \qquad (16) \end{aligned}$$

where $0 < i \le m$.

Note that $t+u=u^{rts}$ and v is the average number of delivered packets, given that at least one successful transmission is done and case ii) does not happen.

Now, we are ready to compute p_i^{rts} and T_i^{rts}. Let us first consider p_i^{rts}, the conditional success probability in stage (i,m), given that at least one of senders begins a transmission. We already know that probability of a single transmission is p^{rts}, the success probability in cases where two RTS overlap is v^{rts}, and the success probability where a RTS and a data packet overlap is u^{rts}. Thus, p_i^{rts} is:

$$\begin{aligned} p^{rts} &= (\lambda_i\cdot(1-\lambda_m)^{T_{DA}}+ \\ &\quad \lambda_m\cdot(1-\lambda_i)^{T_{DA}})/\tau_i \qquad (17) \\ q^{rts} &= (\lambda_i\cdot((1-\lambda_m)^{T_{RTS}}-(1-\lambda_m)^{T_{DA}})+ \\ &\quad \lambda_m\cdot((1-\lambda_i)^{T_{RTS}}-(1-\lambda_i)^{T_{DA}}))/\tau_i \qquad (18) \\ r^{rts} &= (\lambda_i\cdot(1-(1-\lambda_m)^{T_{RTS}})+ \\ &\quad \lambda_m\cdot(1-(1-\lambda_i)^{T_{RTS}})-\lambda_i\lambda_m)/\tau_i \qquad (19) \\ p_i^{rts} &= p^{rts}+r^{rts}\cdot v^{rts}+q^{rts}\cdot u^{rts}, \qquad (20) \end{aligned}$$

where $0 \le i \le m$.

Note that r^{rts} is the conditional probability where two RTS overlap. Similarly, q^{rts} is the conditional probability where a RTS overlaps with a data packet. Also, $p^{rts}+q^{rts}+r^{rts}=1$.

Next, the average delivery time of a single packet in stage (i,m), S_i^{rts} is computed as follows:

$$\begin{aligned} S_i^{rts} &= (p^{rts}(C_{i-1}+1/\tau_i+T_{TX}^{rts})+ \\ &\quad (r^{rts}-p^{rts})\cdot v^{rts}\cdot S_i^{rts}(R^2)+ \\ &\quad (q^{rts}-p^{rts})\cdot u^{rts}\cdot S_i^{rts}(RD))/p_i^{rts} \qquad (21) \end{aligned}$$

where $0 \le i \le m$.

Finally, C_i^{rts} is computed by the following:

$$\begin{aligned} w &= (P_{PER}^{rts})^2 \\ x &= (1-P_{PER}^{rts})^2P_{PER}^2 \\ y &= (P_{PER}^{rts})^2 \\ z &= (1-P_{PER}^{rts})P_{PER} \times \\ &\quad (P_{PER}^{rts}+(1-P_{PER}^{rts})P_{PER}) \\ C_i^{rts} &= C_{i-1}^{rts}+(r^{rts}\cdot w\cdot(1/\tau_i+T_{OVER_1})+ \\ &\quad r^{rts}\cdot x\cdot(1/\tau_i+T_{OVER_2})+ \\ &\quad q^{rts}\cdot y\cdot(1/\tau_i+T_{OVER_3})+ \\ &\quad q^{rts}\cdot z\cdot(1/\tau_i+T_{OVER_4}))/(1-p_i^{rts}) \qquad (22) \\ C_0^{rts} &= (r^{rts}\cdot w\cdot(1/\tau_0+T_{OVER_1})+ \\ &\quad r^{rts}\cdot x\cdot(1/\tau_0+T_{OVER_2})+ \\ &\quad q^{rts}\cdot y\cdot(1/\tau_0+T_{OVER_3})+ \\ &\quad q^{rts}\cdot z\cdot(1/\tau_0+T_{OVER_4}))/(1-p_0^{rts}) \qquad (23) \end{aligned}$$

where $0 < i < m$. It is easy to see that $w+x=1-v^{rts}$ and $y+z=1-u^{rts}$.

With S_i^{rts} and C_i^{rts}, we can find T_i for all i and thus compute the average channel throughput.

$$\begin{aligned} c_m^{rts} &= (r^{rts}\cdot w\cdot(1/\tau_m+T_{OVER_1})+ \\ &\quad r^{rts}\cdot x\cdot(1/\tau_m+T_{OVER_2})+ \\ &\quad q^{rts}\cdot y\cdot(1/\tau_m+T_{OVER_3})+ \\ &\quad q^{rts}\cdot z\cdot(1/\tau_m+T_{OVER_4}))/(1-p_m^{rts}) \qquad (24) \\ T_m^{rts} &= S_m^{rts}+c_m^{rts}\times(1-p_m^{rts})/p_m^{rts} \qquad (25) \\ T_i^{rts} &= S_i^{rts}\cdot p_i^{rts}+T_{i+1}^{rts}\cdot(1-p_i^{rts}) \qquad (26) \end{aligned}$$

IV. SIMULATION

A. Simulation in `Qualnet`

1) Radio Ranges: Table I shows these ranges in `Qualnet` physical layer model.

Range	Distance
Carrier-Sensing Range	519.395 m
Receive Sensitivity Range	504.239 m (-93 dBm)
TX Range with PER $= 0.05$	475.733 m (1 Mbps) 364.010 m (2 Mbps) 345.952 m (5.5 Mbps) 294.792 m (11 Mbps)

TABLE I
DEFAULT RANGES IN QUALNET

In `Qualnet` [12], we used two-ray path loss model, which represents the path loss in an wide open area. Assume d is the distance between a sender and a receiver. When the receiver is close to the sender, receiving signal power is inversely proportional to d^2. When the receiver goes away farther than some distance threshold (e.g. outside of Freznel zone [10]), the receiving signal power is then inverse proportional to d^4. The carrier-sensing range in Table I indicates an area where the strength sum of a emitted signal and normal noise is larger than the carrier-sensing threshold (-93 dBm) after path loss reduction.

Receive sensitivity in `Qualnet` is how much signal a receiver needs to receive in order to work at any speed level. The physical layer discards received packets with signal strength less than the required receive sensitivity. Table I shows the receive sensitivity range where a signal being received is not ignored.

`Qualnet` also provide an accurate function that maps SNR of a received signal to a bit-error rate for each modulation scheme in 802.11b [11]. Within the transmission ranges in Table I, the error rate of 1024-byte packets is computed less than 0.05 assuming no fading channels and other interference sources. If larger packets are transmitted, then the packet error rate (PER) at that distance is higher than 5 %.

From two-ray path loss model and BER mapping function, we can compute the acceptable range of SINR (Signal to Interference and Noise Ratio) and the interference range where receivers may have interference sources. For example, a receiver located at a distance 284.620 m from its sender might be interfered with another sender, 513.285 m away from the receiver. That is, the receiver would loss 95 % of 1024-byte received packets at 11 Mbps on the average because interfering signals from 513.285 m away would severely reduce SINR. Without the interferer, the receiver at a distance 284.620 m would get 100 % of transmitted packets. Note that interfering sources 513.285 m away are too far to catch receiver's CTS at any transmission rate.

2) Protocol Stack and Simulation Parameters: Table II shows the protocol stack in this simulation. The application layer generates fixed-size packets and always makes MAC layer have data to send. STAR routing protocol. (Source Tree Adaptive Routing) operates in LORA mode, where STAR attempts to provide viable, if not necessarily optimal (according to performance, delay metrics) paths to each destination. This protocol exchanges routing messages as broadcasting messages. We set and control message flows to prevent the protocol from obstructing data transmissions.

Layer	Protocol in `Qualnet`
Application	Application/Traffic/CBR
Transport	Agent/UDP
Routing	STAR/LORA
MAC	802.11 MAC
Physical	802.11b Physical Layer

TABLE II
PROTOCOL STACK IN SIMULATION

In `Qualnet`, MAC protocol is performed as specified by the IEEE 802.11b standard. Table III shows selected operation parameters for this simulation. Note that PLCP Data Rate indicates the transmission rate for 48-bit Physical Layer header. Data payloads following the Physical Layer header can be transmitted at from 1 Mbps to 11 Mbps, which is controlled by `Qualnet` simulation scripts. RTS/CTS probing is determined by comparing packet size with RTS/CTS threshold, which is also one of simulation parameters. Several simulations will be performed with basic access and RTS/CTS access methods in the later of this section.

Parameters	Value
Backoff Window Size	32 – 1024
Slot Time	20 us
RxTxTurnaround Time	5 us
SIFS Time	10 us
Preamble Length	144 bits
PLCP Header Length	48 bits
PLCP Data Rate	1 Mbps

TABLE III
PARAMETERS IN 802.11B OPERATION

We use 802.11 Physical Layer protocols provided by `Qualnet`. The implementation performs both physical and virtual carrier sense. Assuming omni-directional antennas and two-way ground propagation model, all stations are stationary and located on a flat plain without any obstacles. Fading models are not used.

B. Simulation Results

1) Basic Accesses: We run simulations with two senders and two receivers. Distances between stations are shown in the following table.

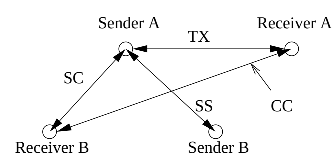

Fig. 9. Station Topology for Throughput Simulations

Figure 10 shows the simulation results in terms of the cumulative channel throughput with varying packet size. Two senders and receivers transmit data packets and ACKs at 11 Mbps except their physical headers. Each simulation, which run with fixed packet size in the range of 100 to 1500 bytes, was performed for 100 seconds. After simulations, we computed how many data bytes MAC layer delivered to the upper layer on the average. 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

Distance	Meter
Sender to Sender (SS)	519.933
Sender to Receiver Associated with the Other Sender (SC)	504.969
Receiver to Receiver (CC)	633.221
Sender to Receiver Associated with itself (TX)	284.000
Error Probability of Half-Overlapped 1500-byte Packets	0.95
Error Probability of Fully Overlapped RTSs	0.61

TABLE IV
SIMULATION SETTING 1

Distance	Meter
Sender to Sender (SS)	519.933
Sender to Receiver Associated with the Other Sender (SC)	519.933
Receiver to Receiver (CC)	706.270
Sender to Receiver Associated with itself (TX)	338.000
Error Probability of Half-Overlapped 1500-byte Packets	0.95
Error Probability of Fully Overlapped RTSs	0.39

TABLE V
SIMULATION SETTING 2

the graph ‘BASIC (7)’. Analysis results in section III are also displayed in Figure 10.

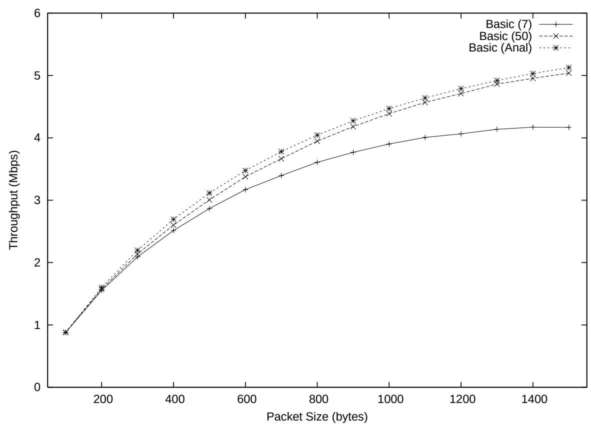

Fig. 10. Throughput at 11 Mbps (Basic Access, Setting 1)

As expected in section III, analysis results put the upper bounds on the channel throughput, that looks very tight. Sending 1500-byte packets, senders achieved a 81.35 percent of analysis expectation in simulation. With 100-byte packets, simulation resulted in a 99.76 percent of expected throughput. However, by increasing retry numbers up to 50, the channel throughput with sending 1500-byte packets hit a 98.28 percent of the expectation.

This indicates that large retry numbers keeps senders in the last backoff stage longer than in the regular cases and it boosts up the chance to make a successful transmission due to the large backoff window size in the stage. In other words, going back to the first stage after giving up a transmission shrinks the backoff window size and increase the possibility of another collision. Since we can not change the window sizes in 802.11, having large retry numbers will be a good and simple solution to get higher throughput on interfered channels.

Another simulation at 5.5 Mbps was set up like in Table V and performed. In Figure 11, analysis and simulation results are shown. At 5.5 Mbps, senders achieve the highest throughput by sending 800-byte packets with regular retry numbers and 1200-byte packets with 50 retry limits. It is interesting that smaller packets are not preferred to avoid conflicts. Because 802.11 imposes large overhead on packetizing, this severely drops the efficiency of the channel compared to sending larger packets. Sending larger packets on interfered channels, however, suffers more conflicts; as the packet size increases, this conflict-and-backoff overhead finally overwhelms the packetizing gain and results in lower throughput.

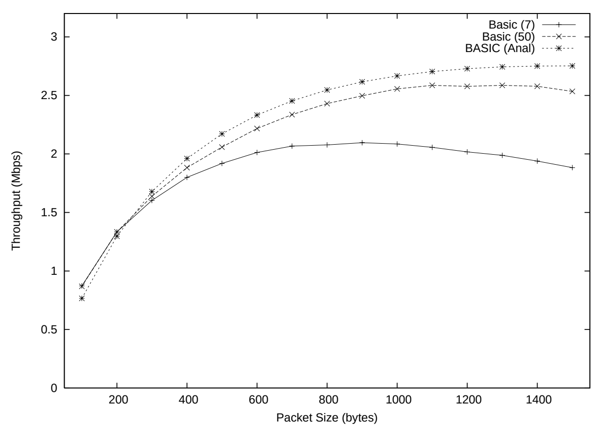

Fig. 11. Throughput at 5.5 Mbps (Basic Access, Setting 2)

Simulation results in Figure 10 and 11 show that our analysis model provide approximate upper bounds even in a case of sending large packets. The error probability in setting 1 and 2 is 0.95 on the average for 1500-byte packets. For basic accesses at 11 Mbps, packet reception at a distance 198 m can get interfered and corrupted at 0.95 by a station out of the carrier-sensing range centered in an associated sender. The channel throughput must be the same if the error probability is not changed, and thus a receiver in the range of 198 to 284 m, which is more than 51 % of the transmission range, may have such interference sources. Our analysis provides upper bounds for them.

2) RTS/CTS Accesses: Simulation results performed with RTS/CTS exchanges are plotted in Figure 12 and Figure 13. Compared to basic accessing, using RTS/CTS accesses performs better with default retry numbers and gains less as the number of retransmissions increases.

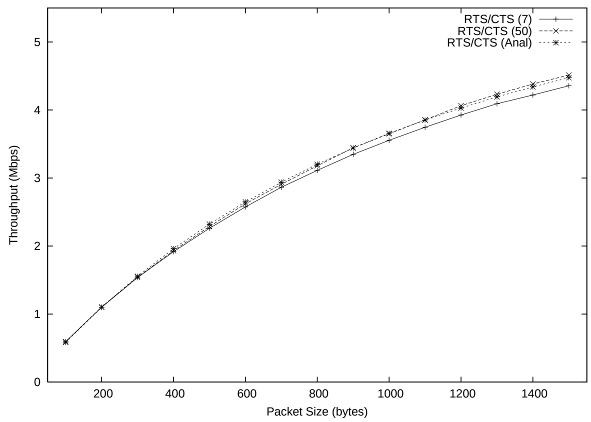

Fig. 12. Throughput at 11 Mbps (RTS/CTS Access, Setting 1)

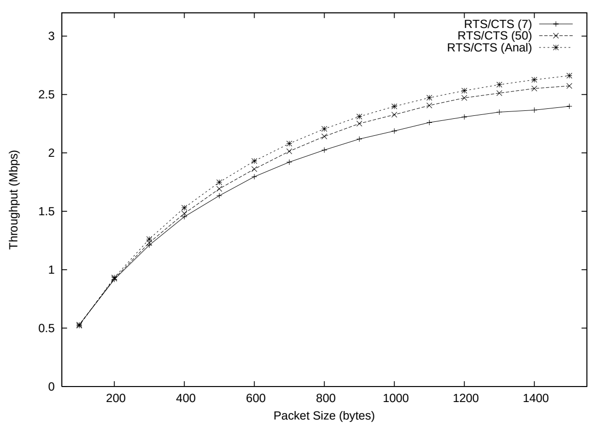

Fig. 13. Throughput at 5.5 Mbps (RTS/CTS Access, Setting 2)

The channel throughput on the interfered channel is determined by the length of conflict and transmit phases as shown in section III. Especially, with small number of retries, senders frequently go back to the first stage after packet drops and thus the length of conflict phases become more critical. The length of conflict phases decreases if two senders grow up the backoff window sizes faster. Multiple RTS/CTS probes involved a single conflict have senders make backoff stage transitions much faster compared to using basic accesses. Note that senders in setting 2 loss almost 40 % of transmitted RTSs. Figure 11 and 13 show that senders using RTS/CTS accesses perform better than using basic accesses when retry numbers are set to default values.

When retry numbers become large, two senders are more often in transmit phases. In the phases, one sender continuously transmits packets in the first backoff stage without interference and the other waits. Given waiting time, probing with RTS/CTS packets increases packet transmission time and works as overhead. Thus, with large retries, advantage of fast stage transition is limited by RTS/CTS probing overhead. Results in Figure 13 show that RTS/CTS accessing achieves performance just comparable to using basic access.

In simulation setting 1, increasing retry numbers does not make much difference, either. The error probability of RTS packets is more than 0.61 and this commonly makes multiple transitions. Moreover, receivers in setting 1 can sense both of senders. Note that the distance SC is less than the carrier-sensing range, 519.395 m. Receivers will discard all received RTSs while the channel is sensed as busy. It makes senders try multiple probes through a conflict phase and move to next backoff stage much faster. Thus, the closer RTS loss probability during a conflict gets to 1, the better throughput RTS/CTS probing with default retry numbers achieves. Another case where receivers are close to senders is considered below.

Distance	Meter
Sender to Sender (SS)	519.932
Sender to Receiver Associated with the Other Sender (SC)	470.934
Receiver to Receiver (CC)	558.182
Sender to Receiver Associated with itself (TX)	263.000
Error Probability of Half-Overlapped 1500-byte Packets	0.50
Error Probability of Fully Overlapped RTSs	0.196

TABLE VI
SIMULATION SETTING 3

In simulation setting 3, receivers are closer to two senders. Now, they can sense signals from both of senders. Signals from their associated senders are strong and the receivers only lose 50 % of packets received with interference. Figure 14 shows simulation results and analysis expectation.

As mentioned above, increasing retry numbers does not help much. Even though RTSs under interference signals can be received with probability more than 0.8, receivers will ignore them during the channel is busy and it makes fast stage transitions of senders.

Note that our analysis still provides a reasonable approximation in Figure 12 and 14. Although we assumed a single backoff stage transition for a sender in each conflict, the analysis model works well for cases where multiple transitions are commonly performed. We believe that multiple transitions can be done in the first few backoff stages even when sending large packets. Increasing retry numbers and keeping a sender in the last stage also reduce the probability of multiple transitions. The limited effect of multiple transitions substantiates

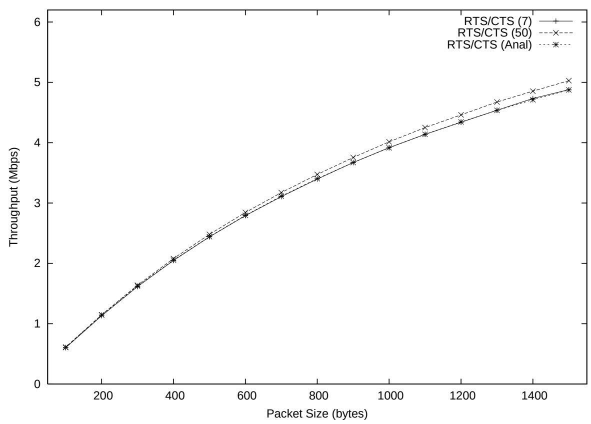

Fig. 14. Throughput at 11 Mbps (RTS/CTS Access, Setting 3)

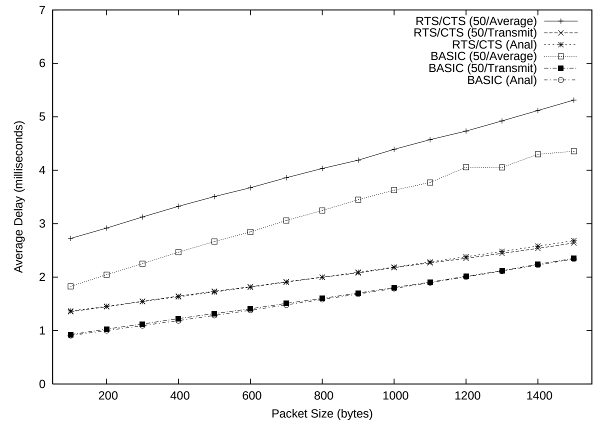

Fig. 15. Delay at 11 Mbps (Setting 1)

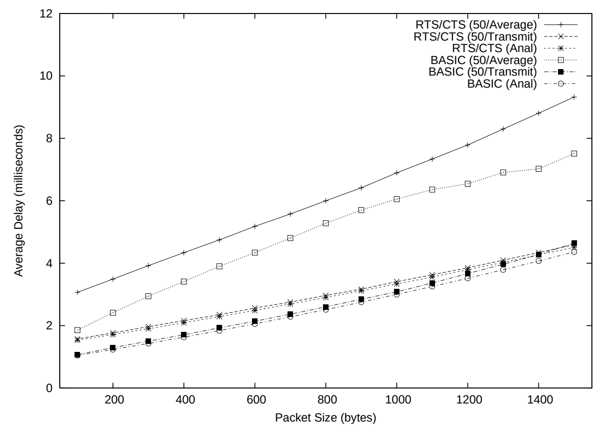

Fig. 16. Delay at 5.5 Mbps (Setting 2)

our analysis model for 802.11 DCF behavior.

3) Transmission Delay: Figure 15 and 16 plot the average delivery time from simulation results. We compute from our analysis the expected delivery time of packets transmitted in transmission phases, denoted *Anal* in the figures. We also measured the average delivery delay of 'younger' packets. Each of two senders has one packet on the table to deliver. Everytime a packet is delivered, a sender fetches another packet from its buffer and records the current time. If the packet has been delivered and its fetch time is earlier than that of a pending packet that the other sender has, it means that the first sender probably now enters the first backoff stage and the second one stays in the last stage. On the other hand, if the fetch time of delivered packet is later than that of the second sender, the first one may have been staying in the first stage. Thus, the average delivery delay of younger packets will be an approximation of the delivery time in transmission phases. Note that delivery time is defined as time difference between fetch and delivery. The average delivery time of younger packets is denoted *Transmit* in Figure 15 and 16.

In Figure 15 and 16, our analysis expectation exactly matches up the delivery time of younger packets. Surprisingly, the real average time is much higher than the delivery time. That means sending 'older' packets takes very long time and senders with older ones must stay in one of the last few stages for that long time. The large difference between delivery time of younger and older packets implies that two senders must be in the one of two states: continuously transmitting and long waiting states. Switching behavior in 802.11 DCF between two states strongly reinforces our analysis model.

V. Throughput Enhancement

Based on the close match between simulation results and our analysis model in the previous section, we can reasonably assert that our modeling approximation, of switching between the first and last backoff stage after a successful transmission is justified. Our analysis model also provides key idea to enhance the system throughput. We already know that switching between the first and the last backoff stage under our assumption achieves the maximum possible throughput. The more number of backoff stages, the more enhancement we can have.

Figure 17 shows an enhancement protocol from our analysis model. This protocol operates on the top of 802.11 MAC layer. The protocol has three states: Waiting, Switching and Transmitting. When a station has data to send or it detects the existence of unknown interference sources, it enters into Waiting state. In the state, it chooses a backoff value and waits. Note that in [7] every packet is delayed whereas we only delay in Waiting state.

After backoff timer expires, it moves to Switching state. It sets 802.11 long and short retry limits to 1 and restarts Switching timer. Note that setting the numbers

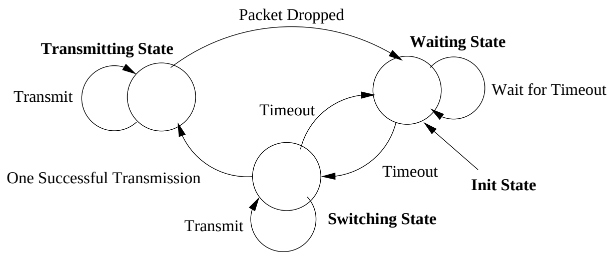

Fig. 17. State Diagram of Enhancement Protocol

to 1, 802.11 MAC will transmit packets only once and notify the upper layer of the delivery status. Then, it supplies one data packet to MAC layer. If it is given notification of failure, the protocol supplies the same data packet again until the packet is delivered or Switching timer expires. Packet delivery makes the protocol move to Transmitting state; otherwise, it goes back to Waiting state.

In Transmitting state, it supplies data packets until delivery failure is informed. Then, the protocol puts the packet back to the buffer and moves to Waiting state.

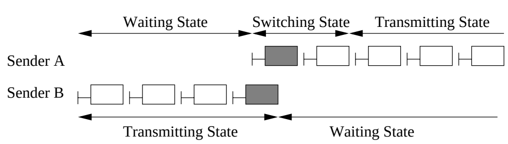

Fig. 18. Protocol Operation

Figure 19 shows results using simulation setting 4. The protocol chooses backoff values in the range of 2000 to 10000 slots, which is about ten times of the window size in the last backoff stage. Compared to increasing retry numbers, our protocol enhances the channel throughput more than 21.4 % in sending 1500-byte packets.

Distance	Meter
Sender to Sender (SS)	520.000
Sender to Receiver Associated with the Other Sender (SC)	322.000
Receiver to Receiver (CC)	124.000
Sender to Receiver Associated with itself (TX)	198.000
Error Probability of Half-Overlapped 1500-byte Packets	0.95

TABLE VII
SIMULATION SETTING 4

Note that a distance between a sender and an associated receiver is only 198 m. The distance is really short and transmissions at 5.5 Mbps will not be interfered by any senders out of the carrier-sensing range. In Figure19, however, sending at 5.5 Mbps shows performance worse than 11-Mbps transmissions with default retry numbers. The reason is that ACKs from an adjacent receiver and data packets from an associated sender conflicts. 802.11 DCF must respond to received data packets with an ACK regardless of the channel status. Assume that receiver A has finished receiving and receiver B is in the middle of data packet reception. The distance between receiver A and B is only 124 m and An ACK from receiver A collides with the packet being received. The conflicts make senders retransmit and drop the channel efficiency.

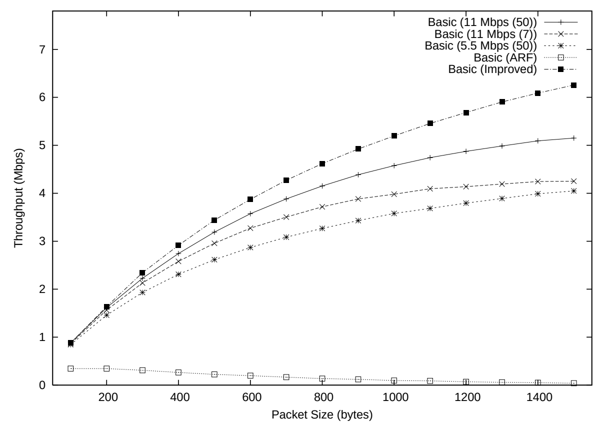

Fig. 19. Throughput at 11 Mbps (Setting 4)

We also run Lucent AutoRate Fallback (ARF) protocol [6] in the same simulation setting. The results is denoted *ARF* in Figure 19. In ARF protocol, loss of two consecutive ACKs makes a sender reduce transmission rate. At 5.5 Mbps, senders suffer packet conflicts with ACKs and reduce the transmission rate again. Since transmissions at 1 and 2 Mbps in 802.11 are more vulnerable than at 11 and 5.5 Mbps, the transmission rate finally drops to the lowest rate but conflicts are still there. Thus, low transmission speed and high packet error rate severely downs the efficiency.

Figure 20 shows performance of our protocol. Again, our protocol enhances the system throughput by more than 30 % while Lucent ARF protocol wastes the channel bandwidth retransmitting packets. Note that, in setting 2, the distance between receivers is too far and neither ACKs nor CTSs do collide with RTSs and data packets. Rather than that, multiple RTSs collides with a single data transmission and it makes ARF protocol reduce the transmission rate. From the results, we can also expect that, if received RTSs are ignored due to the busy channel, then ARF protocol will severely drop the throughput. Thus, the higher the loss probability of RTSs, the more our protocol will improve the throughput, compared to ARF.

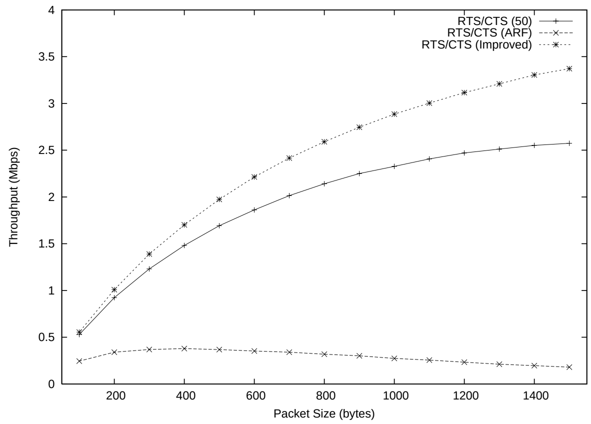

Fig. 20. Throughput at 5.5 Mbps (Setting 2)

VI. CONCLUSION

In this paper, we investigated the behavior of 802.11 DCF on the interference channel. With interference sources out of the carrier-sensing range, 802.11 DCF suffers a lot of overhead and short-term unfairness. We presented our analysis model and performed simulations that shows our analysis provides tight upper bounds for basic and RTS/CTS accesses.

We also analyzed the effect of increasing the retry limits. With large retry numbers, stations using basic access will retain the largest window size in the last backoff stage and have more chances to make successful transmissions. However, for RTS/CTS accesses, the enhancement is limited by the probing overhead and the multiple stage transitions.The higher the loss probability of RTSs, the less increasing retry limits has advantages.

According to our analysis, delaying packets from the upper layer will mitigate the impact of packet conflicts. We present a simple protocol that shows an outstanding performance. Our protocol 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 without rate adjustment.

REFERENCES

[1] G. Bianchi. Performance analysis of the ieee 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications*, 18(3), 2000.

[2] Harshal S. Chhaya and Sanjay Gupta. Performance modeling of asynchronous data transfer methods of ieee 802.11 mac protocol. *Wirel. Netw.*, 3(3):217–234, 1997.

[3] Marco Conti Frederico Cali and Enrico Gregori. Dynamic tuning of the ieee 802.11 protocol to achieve a theoretical throughput limit. *IEEE/ACM Trans. Netw.*, 8(6):785–799, 2000.

[4] Approved draft 802.11-1999 (revisions to IEEE std 802.11-1997). IEEE Standard, 1999.

[5] Mario Gerla Kaixin Xu and Sang Bae. How effective is the ieee 802.11 rts/cts handshake in ad hoc networks? In *IEEE Globecom*, 2002.

[6] A. Kamerman and L. Monteban. Wavelan-ii: A high-performance wireless lan for the unlicensed band. *Bell Labs Technical Journal*, pages 118–133, 1997.

[7] Hwangnam Kim and Jennifer C. Hou. Improving protocol capacity with model-based frame scheduling in ieee 802.11-operated wlans. In *Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 190–204. ACM Press, 2003.

[8] Jinyang Li, Charles Blake, Douglas S.J. De Couto, Hu Imm Lee, and Robert Morris. Capacity of ad hoc wireless networks. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 61–69. ACM Press, 2001.

[9] Daji Qiao, Sunghyun Choi, Amit Jain, and Kang G. Shin. Miser: an optimal low-energy transmission strategy for ieee 802.11a/h. In *Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 161–175. ACM Press, 2003.

[10] T. Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall, 1996.

[11] Mineo Takai and Jay Martin. Effects of wireless physical layer modeling in mobile ad hoc networks.

[12] Scalable Network Technologies. Introduction to qualnet. Qualnet User's Manual, 2004.