Acknowledgement-Aware MPR MAC Protocol for Distributed WLANs: Design and Analysis

Arpan Mukhopadhyay, Neelesh B. Mehta, *Senior Member, IEEE*, Vikram Srinivasan, *Member, IEEE*

***Abstract*—Multi-packet reception (MPR), in which a receiver can decode multiple simultaneous transmissions, significantly improves the uplink throughput of wireless local area networks (WLANs). However, the medium access control (MAC) layer must be redesigned to encourage, and not avoid, simultaneous transmissions. Asynchronous MPR MAC protocols, in which nodes independently access the channel so long as the number of ongoing transmissions is less than a threshold, are promising solutions for enabling MPR in IEEE 802.11-based WLANs. In this paper, we highlight the problem of acknowledgment (ACK) delays that arises in asynchronous MPR when multiple nodes transmit in succession without the channel becoming idle. We propose a novel asynchronous MAC protocol that reduces the ACK delays, increases throughput, and retains the distributed nature of the 802.11 distributed coordination function (DCF). An accurate renewal theoretic fixed-point analysis that leads to general analytical expressions for the saturation throughput is also developed.**

I. Introduction

IEEE 802.11 wireless local area networks (WLANs) are very popular, but are facing increasing demands for higher data rates and greater spectral efficiency. These WLANs use the distributed coordination function (DCF) and its variants such as enhanced DCF (EDCF) for medium access control (MAC) [1]–[3]. The design of the DCF is based on the premise that when multiple stations transmit simultaneously in a time slot, a *collision* occurs and the receiver cannot decode any transmission. Therefore, the DCF uses carrier sense multiple access with collision avoidance (CSMA/CA) to discourage transmissions by more than one user at any time. This is accomplished by freezing the backoff timer anytime a node senses a busy channel.

However, wireless receivers today are capable of employing sophisticated signal processing and multi-user detection technologies such as successive interference cancellation (SIC), space-time coding, and multiple antennas [4], and can receive multiple packets simultaneously. This capability has been referred to as multi-packet reception (MPR) in the literature [5]–[9]. MPR promises significant throughput gains in the uplink, i.e., the link from the stations to the access point (AP) [10]. However, it also engenders a fundamental redesign of the MAC protocol, which must now facilitate – and not discourage – simultaneous transmissions by multiple users. At the same time, it is desirable that the MAC retains its distributed nature, which is a key reason for the success of IEEE 802.11 DCF.

While MPR has been considered in the literature, e.g., [7], [10], [11], a synchronous access scenario is often assumed. In it, multiple nodes can start transmitting simultaneously, but no node can start a transmission if it senses the channel to be busy. This leads to the MPR capability not being fully harnessed. This problem was first addressed by Babich and Comisso in [12], who analyzed an asynchronous protocol using a Markov chain model. In it, a node decrements its backoff timer as long as the number of ongoing transmissions sensed by it is below a threshold. The number of receive antennas in the node determines the number of transmissions that it can sense. However, acknowledgements (ACKs) were not modeled. It was implicitly assumed that a node knows whether its transmission was successful or not as soon as its packet transmission ends.

In fact, in the asynchronous MPR setup, ACKs can get significantly delayed. This is because multiple transmissions to the AP can now occur in succession without any idle period in between. Thus, the AP, which is a half-duplex node, must continue to receive these time overlapping transmissions even after a particular node completes its transmission. It can acknowledge all the packets it successfully received only after the channel becomes idle. However, a node chooses its backoff timer parameters and schedules its next transmission only after it receives its ACK or when it times out waiting for the ACK. Therefore, this delay can degrade the protocol's performance. Furthermore, in [12], a memoryless distribution of packet lengths was assumed in order to facilitate a Markovian analysis. This assumption does not hold under heavy traffic situations when there are many retransmissions.

In this paper, we first highlight a problem related to ACK delays that is unique to an asynchronous MPR MAC protocol. To show that the ACK delays can degrade the performance of such protocols we compare the performances of two protocols. The first one is similar to the one analyzed in [12] except that ACKs are now incorporated in the protocol. The second protocol, which we propose in this paper, reduces ACK delays by modifying the rules that govern when the timer should be decremented and frozen. We show that the modifications reduce the ACK delays and increase the saturation throughput, while retaining the distributed nature of conventional DCF.

Another important contribution of this paper is a fixed-point analysis of the proposed asynchronous MPR MAC protocol based on renewal theory [13]. Unlike [12], it captures the effect of ACK delays on the protocol's performance and leads

A. Mukhopadhyay is with the Dept. of Electrical and Computer Eng. at the Univ. of Waterloo, Canada. N. B. Mehta is with the Dept. of Electrical Communication Eng. at the Indian Institute of Science (IISc), Bangalore, India. V. Srinivasan is with Bell Labs Research, Alcatel-Lucent, Bangalore, India. A. Mukhopadhyay was at IISc during the course of this work.
Emails: arpanmu@gmail.com, nbmehta@ece.iisc.ernet.in

978-1-4673-0921-9/12/$31.00 ©2012 IEEE

to general analytical expressions for the throughput and packet collision probability. It also generalizes the analysis in [14], which is applicable only to conventional DCF. The packet lengths are no longer required to be geometrically distributed in our model. Further, the effect of packet dropping after a finite number of retransmissions is now explicitly included. Thus, our approach provides a more general analysis of the asynchronous MPR MAC.

The paper is organized as follows. Section II develops the system model. The proposed protocol is analyzed in Section III. Simulation results are presented in Section IV, and are followed by our conclusions in Section V. Mathematical derivations are relegated to the Appendix.

II. System Model and Protocol Description

Consider an IEEE 802.11 network that consists of n contending nodes. We assume that the AP can decode up to two simultaneous transmissions, and each node can correctly estimate whether there is zero, one, or more than one ongoing transmission in the channel, as was also assumed in [12].[1] This is easily achieved when each node is equipped with at least two antennas by means of direction of arrival (DOA) techniques based on eigen-decomposition of the received signal's array correlation matrix [16] [12, Sec. VI].

All nodes follow the binary exponential backoff scheme, as specified in 802.11 DCF. Before each packet transmission, a node selects the backoff period in multiples of a slot duration δ. The multiple is chosen uniformly from $\{0, 1, \ldots, w-1\}$, where w is called the *contention window*. It depends on the number of failed transmissions of a packet. In the first attempt, w is set equal to the minimum contention window $CW_{\min}$. After each unsuccessful transmission, w is doubled, up to a maximum value of $CW_{\max} = 2^m CW_{\min}$. A packet is dropped by a node after $K+1$ failed transmission attempts.

A. Protocol 1: Asynchronous MPR Protocol in [12] with ACKs

In the following we describe a protocol similar to that in [12] but with ACKs incorporated in it. As mentioned earlier, the AP is assumed to be able to decode up to two simultaneous transmissions. In this protocol, a node having a packet to transmit samples a backoff timer value and starts decrementing it once it senses the channel to be idle for a distributed inter-frame space (DIFS) of duration T_{DIFS}. The backoff timer is decremented as long as the node senses 0 or 1 transmissions in the channel. When the sensed number of transmissions becomes greater than or equal to 2, the node freezes its backoff timer. It is resumed from the last stored value as soon as the number of sensed transmissions again falls to 1. The node transmits its packet when the backoff timer value becomes zero.

[1]We focus on the case with two simultaneous transmissions since it is both analytically rich and insightful and practically relevant. As we show, it leads to close to 100% gains in saturation throughput and delay over conventional DCF. Further, it requires nodes to have just two antennas, which is feasible today [4], [15]. The analysis can be generalized to handle more than two simultaneous transmissions, though the expressions become more involved.

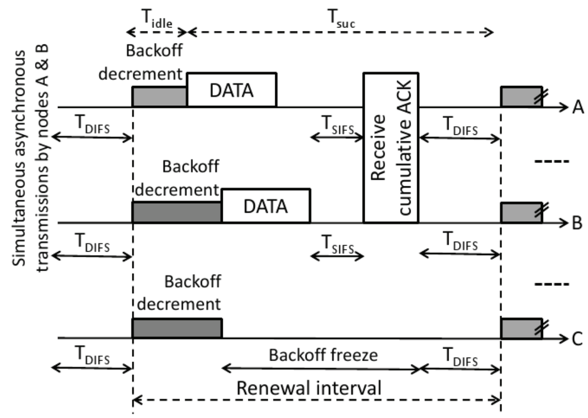

Fig. 1. Timing diagram for the proposed scheme showing a system-wide renewal interval in which two simultaneous asynchronous transmissions to the AP occur.

When the channel becomes idle after the end of the overlapping transmissions, the AP sends a cumulative ACK of duration T_{ACK}, which acknowledges all the successful transmissions together [7]. As in conventional DCF, it waits for the channel to remain idle for a short inter-frame space (SIFS) of duration T_{SIFS} before commencing its transmission of the ACK. By setting $T_{\text{DIFS}} > T_{\text{SIFS}}$, the ACK gets priority over other transmissions when the channel is idle. However, unlike conventional DCF, a node expects its ACK to arrive within a duration $T_{\text{OUT}} = T_{\text{DIFS}}$ after the channel becomes idle, and not after it has completed its packet transmission. In case the ACK does not arrive, the node times out, updates its contention window, chooses a new backoff timer value, and starts decrementing it.

B. Protocol 2: Proposed Protocol

We now propose an asynchronous MPR MAC protocol that reduces the ACK delays. As in the previous protocol, a node with a packet samples a backoff timer and starts decrementing it when it finds the channel to be idle for T_{DIFS} duration. It transmits its packet when the timer reaches zero. However, the proposed protocol differs with respect to the conditions under which the backoff timer is frozen and remains frozen. The rule is as follows: *During the backoff phase, a node freezes its timer once the number of ongoing transmissions exceeds two or once it decreases. Thereafter, the node resumes decrementing its timer only when the channel has remained idle for a duration T_{DIFS}.* The operation of the AP remains the same as before.

The above protocol is illustrated in Figure 1 for three nodes A, B, and C. In it, A transmits first. While A is transmitting, B and C continue decrementing their timers since there is only one ongoing transmission in the channel. Once B starts transmitting, C freezes its timer. The timer of C remains frozen even after A's transmission ends. Only after an idle period of duration T_{DIFS}, do all the nodes resume decrementing their timers. In Protocol 1 instead, C will start decrementing its timer as soon as A's transmission ends. As a result, its timer may become zero while B is transmitting.

III. Analysis

We now analyze the proposed MPR scheme in saturated traffic conditions in which transmission queue of each node is assumed to be always non-empty. This gives a limit on the system throughput in heavy traffic loads, and has been extensively analyzed for conventional DCF; see, for example, [1], [14]. To simplify the analysis, the nodes are assumed to be statistically identical and the transmission rate is fixed at Ω [1], [14]. A transmitted packet is received successfully unless it is involved in a collision. To get compact analytical results, we assume that a data packet transmission lasts for a duration of λ slots. This also illustrates that a non-memoryless packet length distribution can be analyzed, unlike the Markov chain approach of [1], [12]. We also assume that ACKs are successfully received by all the nodes.

As all the nodes use the same backoff parameters, we make the following two classical decoupling approximations, which enable a fixed-point analysis [1], [14]:

1) Each transmitted packet suffers a collision with a probability γ, regardless of the number of its retransmissions and independent of all other nodes. We shall call γ as the *conditional packet collision probability*.
2) Each node attempts a transmission in a slot where it is allowed to transmit with a probability β that is independent of all other nodes. The parameter β will be referred to as the *attempt rate*.

Note that over a sufficiently long time, γ can be interpreted as the ratio of the total number of collisions occurring in the network to the total number of transmission attempts by all nodes under saturated traffic conditions. Similarly, β, in the long run, is the average (taken over all nodes) of the ratio of the total number of transmission attempts by a node to the total number of slots in which the node is in its backoff phase. Based on these decoupling approximations, the following two renewal processes can be constructed.

Node-specific renewal process: Consider a given node, which we henceforth call the *tagged* node. Let A_j and B_j respectively denote the number of attempts and total backoff duration (in slots) needed by the tagged node to transmit its j^{th} packet. From the first approximation it follows that the backoff process of a tagged node is a renewal process with renewal lifetimes B_j, $j \geq 1$, and with the time instants at which the node starts the final transmission of its j^{th} packet as renewal epochs. If we treat A_j, $j \geq 1$, as the reward gained in each renewal cycle, then from the renewal reward theorem [13], we have $\beta = \frac{\mathbb{E}[A_j]}{\mathbb{E}[B_j]}$, where $\mathbb{E}\left[\cdot\right]$ denotes expectation. It can be shown that the attempt rate β as a function of γ is given by

$$\beta \triangleq G(\gamma) = \frac{1 + \gamma + \cdots + \gamma^K}{b_0 + \gamma b_1 + \cdots + \gamma^K b_K}, \tag{1}$$

where $b_k = \frac{1}{2}\left(2^k \mathrm{CW}_{\min} - 1\right)$, for $0 \leq k \leq K$, is the mean backoff duration (in slots) before the $(k+1)^{\text{th}}$ transmission attempt of a packet. The proof is similar to that in [14], and is not shown here to conserve space.

System-wide renewal process: Consider the aggregate attempt process by all the n nodes on the channel. Due to the decoupling approximations, the aggregate process is another renewal process. As shown in Figure 1, a renewal interval starts when all the nodes start decrementing their backoff timers and ends with either an ACK timeout (in case of a collision) or the transmission of a cumulative ACK by the AP (in case of a success) followed by an idle of duration T_{DIFS}.

Unlike conventional DCF, more than one packet can get transmitted in a renewal interval. We, therefore, define the first packet and second packet of a renewal interval as follows. A packet is called the *first packet* in a renewal interval if the channel is idle when its transmission commences. A packet is called the *second packet* if there is already one ongoing transmission in the channel when its transmission commences. Given that a tagged node has transmitted a packet, let α denote the probability that the packet is the first packet in the renewal interval.

Lemma *1:* Given that a tagged node transmits in a renewal interval, the probability α that it is the first packet in the interval is given by

$$\alpha = \frac{K_1(\beta)}{K_1(\beta) + K_2(\beta)}, \tag{2}$$

where $K_i(\beta)$ is the probability that the i^{th} packet in a renewal interval is the tagged node's packet. Further, $K_1(\beta) = \frac{\beta}{1-(1-\beta)^n}$ and $K_2(\beta) = \frac{(n-1)\beta^2(1-\beta)^{n-1}(1-(1-\beta)^{(\lambda-1)(n-1)})}{(1-(1-\beta)^n)(1-(1-\beta)^{n-1})}$.

Proof: The proof is relegated to Appendix A. ■

Theorem *1:* The conditional packet collision probability, γ, as a function of β is given by

$$\gamma \triangleq \Gamma(\beta) = \alpha P_1(\beta) + (1-\alpha) P_2(\beta), \tag{3}$$

where $P_i(\beta)$, for $i = 1, 2$, denotes the probability of collision of the i^{th} transmitted packet in a renewal interval. Further,

$$P_1(\beta) = \frac{\left(1 - (1-\beta)^{n-1} - (n-1)\beta(1-\beta)^{n-2}\right)}{1-(1-\beta)^{n-1}} \times \left(1 - (1-\beta)^{\lambda(n-1)}\right), \tag{4}$$

$$P_2(\beta) = 1 - (1-\beta)^{n-2}. \tag{5}$$

Proof: The proof is relegated to Appendix B. ■

Hence, by combining (1), (2), and (3), we get the following fixed-point equation: $\gamma = \Gamma(G(\gamma))$. Since $\Gamma(G(\gamma))$ is a continuous mapping from the closed set $[0,1]$ to itself, Brouwer's fixed-point theorem implies that there exists a fixed point in $[0,1]$ [14]. Solving this equation numerically yields γ. Then, (1) directly yields β.[2]

A. Saturation Throughput

Let ζ denote the amount of successfully transmitted data in a renewal interval of duration T. From the renewal reward theorem [13], the saturation throughput, S, is given by

$$S = \frac{\mathbb{E}\left[\zeta\right]}{\mathbb{E}\left[T\right]}, \tag{6}$$

[2]We have observed that the fixed point is unique for the parameters of interest. However, proving uniqueness remains a challenging problem.

We now develop expressions for $\mathbb{E}[\zeta]$ and $\mathbb{E}[T]$. As shown in Figure 1, a renewal interval of length T starts with an idle period of duration T_{idle}. It is followed by a busy period of duration T_{busy}, which includes packet transmission(s), a cumulative ACK (if success occurs), and an idle duration of length T_{DIFS}. Depending on whether a success or collision has occurred, we refer to the busy period that occurs after the idle period as a success period (of duration T_{suc}) or a collision period (of duration T_{col}). Further, let $T_{\text{col}}^{\min}$ and $T_{\text{suc}}^{\min}$ denote the minimum values of T_{col} and T_{suc}, respectively. It can be seen that $T_{\text{col}}^{\min} = \lambda\delta + T_{\text{DIFS}}$ and $T_{\text{suc}}^{\min} = \lambda\delta + T_{\text{SIFS}} + T_{\text{ACK}} + T_{\text{DIFS}}$.

Lemma *2:* The expected length of the renewal interval is

$$\mathbb{E}[T] = \mathbb{E}[T_{\text{idle}}] + D_{\text{col}} + D_{\text{suc}}, \tag{7}$$

where D_{col} and D_{suc} are the contributions to the average busy period duration from the collision and success events, respectively. Further, $\mathbb{E}[T_{\text{idle}}] = \frac{1}{1-(1-\beta)^n}$,

$$\begin{aligned} D_{\text{col}} = &\left[\frac{1-(1-\beta)^n - n\beta(1-\beta)^{n-1}}{1-(1-\beta)^n} - \frac{n(n-1)\beta^2(1-\beta)^{n-2}}{2\left(1-(1-\beta)^n\right)}\right] T_{\text{col}}^{\min} \\ &+ \frac{n\beta\left(1-\beta\right)^{n-1}\left(1-(1-\beta)^{n-1} - (n-1)\beta(1-\beta)^{n-2}\right)}{\left(1-(1-\beta)^n\right)\left(1-(1-\beta)^{n-1}\right)} \\ &\times \left[\left(1-(1-\beta)^{(\lambda-1)(n-1)}\right) T_{\text{col}}^{\min} \right. \\ &\left. + \frac{1-(1-\beta)^{(\lambda-1)(n-1)}\left(\lambda-(\lambda-1)(1-\beta)^{n-1}\right)}{\left(1-(1-\beta)^{n-1}\right)}\delta\right], \end{aligned} \tag{8}$$

and

$$\begin{aligned} D_{\text{suc}} = &\frac{n(n-1)\beta^2(1-\beta)^{n-2} + 2n\beta(1-\beta)^{\lambda(n-1)}}{2\left(1-(1-\beta)^n\right)} T_{\text{suc}}^{\min} \\ &+ \frac{n(n-1)\beta^2(1-\beta)^{n-1}(1-\beta)^{n-2}}{\left(1-(1-\beta)^n\right)\left(1-(1-\beta)^{n-1}\right)} \\ &\times \left[\left(1-(1-\beta)^{(\lambda-1)(n-1)}\right) T_{\text{suc}}^{\min} \right. \\ &\left. + \frac{1-(1-\beta)^{(\lambda-1)(n-1)}\left(\lambda-(\lambda-1)(1-\beta)^{n-1}\right)}{\left(1-(1-\beta)^{n-1}\right)}\delta\right]. \end{aligned} \tag{9}$$

Proof: The proof is relegated to Appendix C. ■

Lemma *3:* The expected amount of data transmitted in a renewal interval, with a transmission rate of Ω, is given by

$$\begin{aligned} \mathbb{E}[\zeta] = 2\lambda\delta\Omega &\left(\frac{n(n-1)\beta^2(1-\beta)^{n-2}}{2\left(1-(1-\beta)^n\right)} \right. \\ &\left. + \frac{n(n-1)\beta^2(1-\beta)^{n-2}\left[(1-\beta)^{n-1} - (1-\beta)^{\lambda(n-1)}\right]}{(1-(1-\beta)^{n-1})(1-(1-\beta)^n)}\right) \\ &+ \lambda\delta\Omega\frac{n\beta(1-\beta)^{\lambda(n-1)}}{1-(1-\beta)^n}. \end{aligned} \tag{10}$$

Proof: The proof is relegated to Appendix D. ■

The expression for the normalized saturation throughput then follows directly from Lemmas 2 and 3.

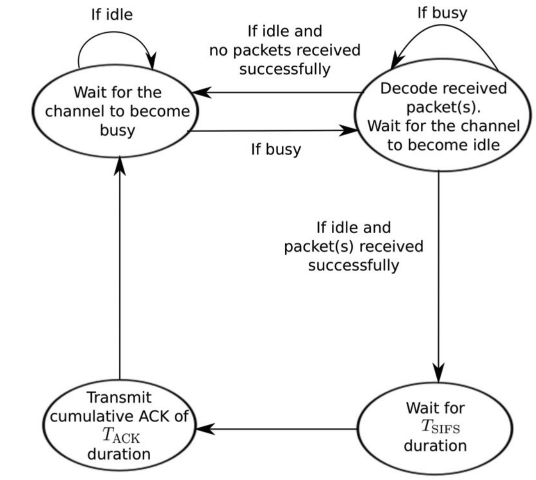

Fig. 2. Proposed protocol: Finite state machine for the access point

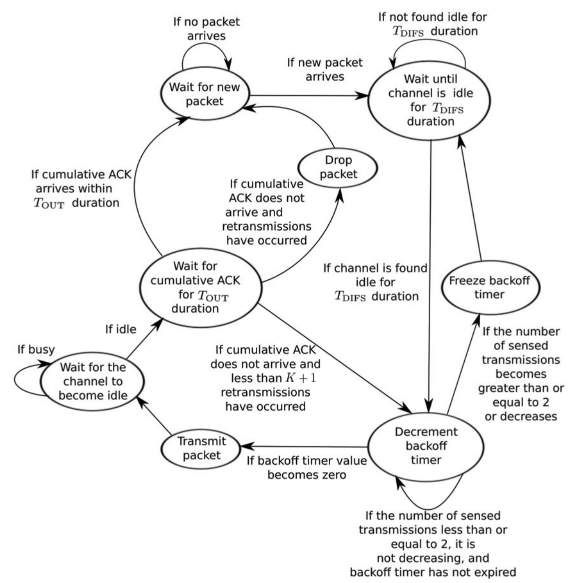

Fig. 3. Proposed protocol: Finite state machine for a node

IV. Numerical Results

We now present the results obtained from Monte Carlo simulations that use 50,000 samples for different MAC protocols, and compare them with analytical results. An event-driven simulation platform written in C programming language was used to implement the MPR protocols described in Section II. This provides independent verification of the analytical results. In it, each node is modeled as a finite state machine (FSM) having several states as per the protocols. The FSMs that model the AP and the nodes for the proposed protocol are shown in Figures 2 and 3, respectively. The parameter values used in the simulations are: $\delta = 20\ \mu$s, $T_{\text{DIFS}} = T_{\text{OUT}} = 50\ \mu$s, $T_{\text{SIFS}} = 10\ \mu$s, $T_{\text{ACK}} = 304\ \mu$s, $CW_{\min} = 32$, $CW_{\max} = 1024$, $K = 7$, and $\lambda = 400$ slots.

Figure 4 plots the normalized saturation throughput, S/Ω,

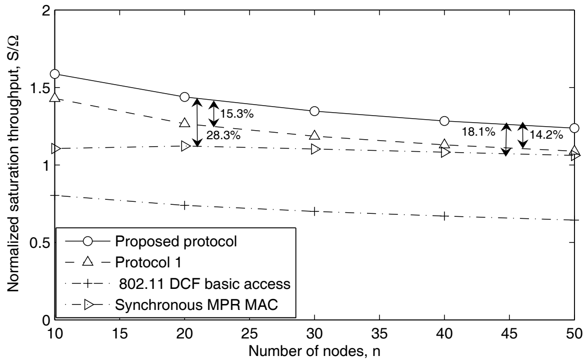

Fig. 4. Saturation throughput as a function of the number of nodes

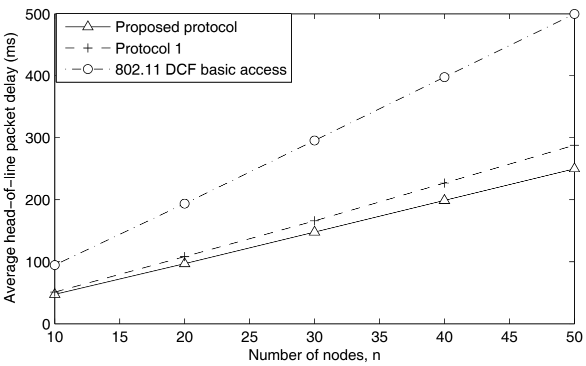

Fig. 5. Saturation delay as a function of the number of nodes

as a function of the number of contending nodes, n, for conventional DCF, Protocol 1, and the proposed Protocol 2. Also plotted is the throughput of a synchronous MPR protocol in which the nodes freeze their backoff timers if they sense the channel to be busy regardless of the number of ongoing transmissions in the channel. However, if one or two nodes transmit simultaneously, their packets will be received successfully by the AP. The ACKs are incorporated in all the protocols in order to ensure a fair comparison. We see that the throughput of the proposed protocol is twice that of conventional DCF, 10-15% more than Protocol 1, and 16-38% more than the synchronous MPR version.[3]

Figure 5 plots the average head-of line packet delay, i.e., the average time spent by a packet at the head of a node's queue until the node receives an ACK confirming successful transmission or the node drops the packet. Again, the proposed protocol outperforms the two benchmark protocols. The head-of-line delay is an important performance metric that affects the performance of higher layers of the protocol stack [17].

The conditional packet collision probability as a function of

[3]For the general case where the AP can decode up to $L \geq 2$ overlapping transmissions, Protocol 2 can be generalized as follows. A node freezes its timer when the number of transmissions in the channel exceeds $L-1$ or when the number of transmissions starts decreasing. For $L = 3$, 4, and 5, the throughput of Protocol 2 is 2.96, 3.95, and 4.98 times, respectively, more than the throughput of conventional DCF. It is at least 10% more than the that of Protocol 1.

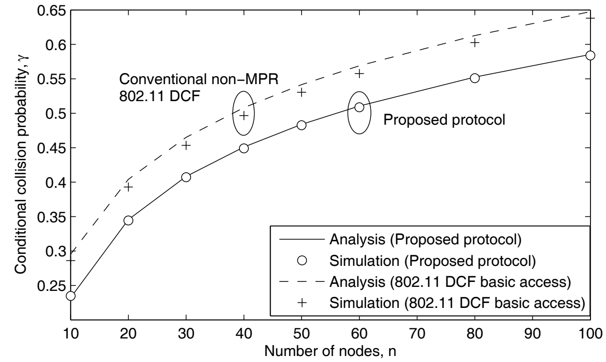

Fig. 6. Conditional packet collision probability as a function of the number of nodes

n is shown in Figure 6. Notice that the analysis and simulation results match each other well. Further, as n increases, the percentage error decreases. This is in consonance with the results in mean field interaction theory, which provides a mathematical justification for this behavior [18].

V. Conclusions

We saw that making the nodes freeze their backoff timers once the number of ongoing transmissions in the channel has crossed a threshold value or decreased from its previous value reduces the ACK delays in the asynchronous MPR setup. It restricts the number of ovelapping transmissions in a renewal interval, and leads to saturation throughput gains over the asynchronous MPR MAC protocol considered in the literature and over conventional DCF. We saw that our renewal-theoretic fixed-point analysis is accurate and general. It enables the modeling of packet dropping after K retransmissions. Also, unlike the Markovian analysis, it does not need to assume a memoryless packet length distribution.

Future work includes evaluating the effect of adaptive modulation and coding and imperfect estimation of the number of ongoing transmissions in the channel, and determining the non-saturated throughput. Corresponding throughput improvements in the downlink, which can be achieved using transmissions techniques such as superposition coding or spatial multiplexing at the AP, are also worth investigating.

Appendix

A. Proof of Lemma 1

Expression for $K_1(\beta)$: Let the tagged node transmit in slot t ($t \geq 1$) of the renewal interval. This occurs with probability $(1-\beta)^{n(t-1)}\beta$, since none of the n nodes should have transmitted in the slots $1, \ldots, t-1$ and the tagged node must transmit in slot t. Hence,

$$K_1(\beta) = \sum_{t=1}^{\infty} \beta(1-\beta)^{n(t-1)}\beta = \frac{\beta}{1-(1-\beta)^n}.$$

Expression for $K_2(\beta)$: Let the first transmission from exactly one node other than the tagged node begin in slot t_1 of the renewal interval, where $t_1 \geq 1$, and let the tagged node transmit in slot t_1+t_2+1. Clearly, $0 \le t_2 \le \lambda-2$, since the proposed protocol does not permit the tagged node to transmit once the channel becomes idle. The probability of this event is

$$(1-\beta)^{n(t_1-1)}(n-1)\beta(1-\beta)^{n-2}(1-\beta)(1-\beta)^{(n-1)t_2}\beta.$$

Summing the probabilities over t_1 and t_2 yields the desired expression for $K_2(\beta)$.

The expression in (2) for α in terms of $K_1(\beta)$ and $K_2(\beta)$ follows directly from Baye's rule.

B. Proof of Theorem 1

A packet transmitted by the tagged node suffers a collision under the following two scenarios.

1) It is the first packet in the interval: In this case, the packet suffers a collision only if, in any of its λ slots, at least two among the other $n-1$ nodes transmit. The probability that the first i slots ($0 \le i \le \lambda-1$) of the transmitted packet are free from collisions is $(1-\beta)^{i(n-1)}$. And, the probability that two or more nodes transmit in the $(i+1)^{\text{th}}$ slot is $1-(1-\beta)^{n-1}-(n-1)\beta(1-\beta)^{n-2}$. Thus, we have $P_1 = \sum_{i=0}^{\lambda-1}(1-\beta)^{i(n-1)}\left(1-(1-\beta)^{n-1}-(n-1)\beta(1-\beta)^{n-2}\right)$, which simplifies to (5).

2) It is the second packet in the interval: In this case, a collision can occur only in the first slot of the packet. This happens when at least one among the remaining $n-2$ nodes transmits in the slot, which occurs with probability $1-(1-\beta)^{n-2}$.

C. Proof of Lemma 2

Let $P[\cdot]$ denote probability.

Expression for $\mathbb{E}[T_{idle}]$: Clearly, $P[T_{\text{idle}} > x] = (1-\beta)^{nx}$, for $x = 0, 1, \ldots$, because no node must transmit for at least x slots. Hence, we get $\mathbb{E}[T_{\text{idle}}] = \sum_{x=0}^{\infty} P[T_{\text{idle}} > x] = \sum_{x=0}^{\infty}(1-\beta)^{nx} = \frac{1}{1-(1-\beta)^n}$.

Expression for D_{col}: When two or more nodes start transmitting simultaneously just after the idle period ends, we have $T_{\text{col}} = T_{\text{col}}^{\min}$. This occurs with probability $\frac{1-(1-\beta)^n-n\beta(1-\beta)^{n-1}-\frac{n(n-1)}{2}\beta^2(1-\beta)^{n-2}}{1-(1-\beta)^n}$. The denominator term of $1-(1-\beta)^n$ arises due to conditioning on the event that the idle period has ended. Now, $T_{\text{col}} = T_{\text{col}}^{\min} + x\delta$, for $1 \le x \le \lambda-1$, when exactly one among the n nodes starts transmitting after the idle period is over, none of the remaining $n-1$ nodes transmit in the next $x-1$ slots, and at least two of the remaining $n-1$ nodes transmit in the following slot. This happens with probability $\frac{n\beta(1-\beta)^{n-1}(1-\beta)^{(x-1)(n-1)}(1-(1-\beta)^{n-1}-(n-1)\beta(1-\beta)^{n-2})}{1-(1-\beta)^n}$. Summing over x results in (8).

Expression for D_{suc}: If just after the idle period, only one node transmits in the whole renewal interval or exactly two nodes start transmitting simultaneously then a success occurs and $T_{\text{suc}} = T_{\text{suc}}^{\min}$. Its probability of occurrence is $\frac{\frac{n(n-1)}{2}\beta^2(1-\beta)^{n-2}+n\beta(1-\beta)^{n-1}(1-\beta)^{(\lambda-1)(n-1)}}{1-(1-\beta)^n}$. Similarly, it can be shown that $P[T_{\text{suc}} = T_{\text{suc}}^{\min} + x\delta] = \frac{n\beta(1-\beta)^{n-1}(1-\beta)^{(x-1)(n-1)}(n-1)\beta(1-\beta)^{n-2}}{1-(1-\beta)^n}$, for $1 \le x \le \lambda-1$. Summing over x yields the desired expression in (9).

D. Brief Proof of Lemma 3

If a collision occurs in a renewal interval, then the AP cannot decode any of the transmitted packets. Hence, in this case $\zeta = 0$. Otherwise, ζ equals $i\lambda\delta\Omega$ if $i \in \{1, 2\}$ nodes transmit in the renewal interval. The probabilities of these events have been derived in Appendix C. Therefore,

$$\mathbb{E}[\zeta] = \frac{n\beta(1-\beta)^{\lambda(n-1)}}{1-(1-\beta)^n}\lambda\delta\Omega + \left(\frac{\binom{n}{2}\beta^2(1-\beta)^{n-2}}{1-(1-\beta)^n} + \sum_{x=1}^{\lambda-1}\frac{n(n-1)\beta^2(1-\beta)^{n-2}(1-\beta)^{x(n-1)}}{1-(1-\beta)^n}\right)2\lambda\delta\Omega. \quad (11)$$

The expression above simplifies to (10).

References

[1] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Sel. Areas Commun.*, vol. 18, pp. 535–547, Mar. 2000.

[2] K. Medepalli and F. A. Tobagi, "Towards performance modeling of IEEE 802.11 based wireless networks: a unified framework and its applications," in *Proc. INFOCOM*, pp. 1–12, Apr. 2006.

[3] F. Cal, M. Conti, and E. Gregori, "IEEE 802.11 protocol: design and performance evaluation of an adaptive backoff mechanism," *IEEE J. Sel. Areas Commun.*, vol. 18, pp. 1774–1786, Sep. 2000.

[4] M. Z. Siam and M. Krunz, "An overview of MIMO-oriented channel access in wireless networks," *IEEE Wireless Commun.*, vol. 15, pp. 63–69, Feb. 2008.

[5] L. Tong, Q. Zhao, and G. Mergen, "Multipacket reception in random access wireless networks: from signal processing to optimal medium access control," *IEEE Commun. Mag.*, vol. 39, pp. 108–112, Nov. 2001.

[6] G. D. Celik, G. Zussman, W. F. Khan, and E. Modiano, "MAC for networks with multipacket reception capability and spatially distributed nodes," in *Proc. INFOCOM*, pp. 1436–1444, Apr. 2008.

[7] W. L. Huang, K. B. Letaief, and Y. J. Zhang, "Joint channel state based random access and adaptive modulation in wireless LANs with multipacket reception," *IEEE Trans. Wireless Commun.*, vol. 7, pp. 4185–4197, Nov. 2008.

[8] F. Shad, T. D. Todd, V. Kezys, and J. Litva, "Dynamic slot allocation (DSA) in indoor SDMA/TDMA using a smart antenna basestation," *IEEE/ACM Trans. Networking*, vol. 9, pp. 69–81, Feb. 2001.

[9] D. Truhachev, S. Nagaraj, and C. Schlegel, "Throughput/reliability tradeoffs in spread spectrum multi-hop ad hoc wireless networks with multi-packet detection," in *Proc. ICC*, Jun. 2009.

[10] S. Ghez, S. Verdu, and S. C. Schwartz, "Stability properties of slotted ALOHA with multipacket reception capability," *IEEE Trans. Automat. Contr.*, vol. 33, pp. 640–649, Jul. 1988.

[11] P. X. Zheng, Y. J. Zhang, and S. C. Liew, "Multipacket reception in wireless local area networks," in *Proc. ICC*, pp. 3670–3675, Jun. 2006.

[12] F. Babich and M. Comisso, "Theoretical analysis of asynchronous multi-packet reception in 802.11 networks," *IEEE Trans. Commun.*, vol. 58, pp. 1782–1794, Jun. 2010.

[13] R. W. Wolff, *Stochastic Modeling and the Theory of Queues*. Prentice Hall, 1989.

[14] A. Kumar, E. Altman, D. Miorandi, and M. Goyal, "New insights from a fixed point analysis of single cell IEEE 802.11 WLANs," *IEEE/ACM Trans. Networking*, vol. 15, pp. 588–601, Jun. 2007.

[15] T. Kaiser, "When will smart antennas be ready for the market? part I," *IEEE Sig. Proc. Mag.*, vol. 22, pp. 87–92, Mar. 2005.

[16] L. C. Godara, "Application of antenna arrays to mobile communications–part II: beam forming and direction-of-arrival considerations," *Proc. IEEE*, vol. 85, pp. 1193–1245, Aug. 1997.

[17] M. M. Carvalho and J. J. Garcia-Luna-Aceves, "Delay analysis of IEEE 802.11 in single-hop networks," in *Proc. IEEE Intl. Conf. Network Protocols*, pp. 146–155, Nov. 2003.

[18] M. Benaim and J.-Y. L. Boudec, "A class of mean field interaction models for computer and communication systems," *Perform. Eval.*, vol. 20, pp. 243–302, Nov. 2008.