Wireless Networks 7, 159–171, 2001

A Capacity Analysis for the IEEE 802.11 MAC Protocol

Y.C. TAY and K.C. CHUA
National University of Singapore

Abstract. The IEEE 802.11 MAC protocol provides shared access to a wireless channel. This paper uses an analytic model to study the channel capacity – i.e., maximum throughput – when using the basic access (two-way handshaking) method in this protocol. It provides closed-form approximations for the probability of collision p, the maximum throughput S and the limit on the number of stations in a wireless cell.

The analysis also shows that: p does not depend on the packet length, the latency in crossing the MAC and physical layers, the acknowledgment timeout, the interframe spaces and the slot size; p and S (and other performance measures) depend on the minimum window size W and the number of stations n only through a gap $g = W/(n-1)$ – consequently, halving W is like doubling n; the maximum contention window size has minimal effect on p and S; the choice of W that maximizes S is proportional to the square root of the packet length; S is maximum when transmission rate (including collisions) equals the reciprocal of transmission time, and this happens when channel wastage due to collisions balances idle bandwidth caused by backoffs.

The results suggest guidelines on when and how W can be adjusted to suit measured traffic, thus making the protocol adaptive.

Keywords: IEEE 802.11 MAC protocol, capacity analysis, saturation throughput, closed-form approximation, analytic validation, window size adaptation

1. Introduction

With the proliferation of mobile computers, their limited computing resources, and the popularity of Internet access, there is a growing need for these computers to be networked. In response, the IEEE 802.11 study group proposed a standard for wireless local area networks [8]. This standard specifies the characteristics of the physical layer, as well as the medium access control (MAC) protocols in the link layer.

There are essentially two MAC protocols in the proposal – a *basic access* method that uses two-way handshaking (DATA-ACK) and a RTS/CTS variant that uses request-to-send and clear-to-send messages in a four-way handshake (RTS-CTS-DATA-ACK). This paper analyzes the former but not the latter, for two reasons: (1) the basic access method is mandatory, whereas RTS/CTS is an optional variant; (2) the performance for RTS/CTS is significantly different from that for basic access [2], and therefore requires a separate analytic model. We also do not discuss the no-ACK option meant for broadcasts and multicasts, nor the *point coordination function*, which is an optional polling scheme defined on top of basic access.

Basic access uses carrier-sensing multiple access with collision avoidance (CSMA/CA). There are numerous CSMA protocols, and their performance under low load conditions are usually similar [22,24]. The many variations arise because of efforts to improve on performance and push back the limits. Our analysis therefore focuses on the most important such limit – namely, the maximum (or saturation) throughput, which measures the capacity when the protocol is used to access the channel, and which is lower than the raw bandwidth for the physical medium itself.

Our model considers the case where multiple stations use the protocol to share a wireless channel without a coordinating base station. It assumes that the stations are homogeneous in traffic generation, channel noise is negligible, and there are no hidden terminals. A scenario that may fit these assumptions would be a classroom or meeting in which students or executives exchange information on their laptops. From the modeling perspective, it is not difficult (but somewhat tedious) to take noise into consideration; also, hidden terminals require a separate model and, in any case, should be analyzed together with RTS/CTS because the two are closely related [2,3,7].

In contrast to previous simulation studies of the 802.11 MAC protocols [2,13,23], the performance analysis we present here is based on a mathematical model. This model not only differs from previous analytic models of the 802.11 protocols [3,6,7,10], it is also different from the other techniques in the CSMA literature [1,5,12,14,15,17–20]. Whereas these studies use stochastic analysis (e.g., Markov chains), our model uses the average value for a variable wherever possible – this is a methodology that is commonly used in the performance analysis of computer systems. This technique is simple, yet effective: It provides closed-form expressions for the probability of a collision and the saturation throughput, thus facilitating the analysis of various issues, such as the choice of window size, the limit on the number of stations, and the tradeoff between collisions and backoffs. It also yields two rules of thumb: halving the initial window size W (for the exponential backoff) is similar in effect to doubling the number of stations, and the optimum choice of W is proportional to the square root of packet size.

A performance model is usually validated by comparing its numerical predictions with simulation results. For our model, we also check its analytical conclusions against simulated performance – we call this *analytic validation*. We use Bianchi et al. simulator [2] for the validation. This simulator is comprehensive in capturing the many details in the 802.11 protocol, and although our model omits many of these details and relies on many approximations, the comparison shows that it is accurate both numerically and analytically.

We begin in section 2 by describing the protocol, and introduce the performance model in section 3. We then check the numerical accuracy of the model in section 4, before using it to analyze the protocol in section 5; there, we constantly use the simulator to check the results from the analysis. Section 6 summarizes our conclusions.

2. Protocol description

The basic access method for the 802.11 MAC protocol works as follows: To send a packet, a station $\mathcal{X}$ first listens to the channel for time T_{DIFS} (DIFS is *distributed interframe space*). If there is silence for T_{DIFS}, $\mathcal{X}$ proceeds with the transmission (e.g., station A in figure 1); otherwise, $\mathcal{X}$ waits for the first T_{DIFS} of silence after the current busy period, then backs off for a random interval (e.g., station C in figure 1).

For each packet, $\mathcal{X}$ initializes a *contention window size* $\widetilde{W}$ to be W, the *minimum window size*. $\mathcal{X}$ sets a timer to a random integer uniformly distributed over $0, 1, \ldots, \widetilde{W} - 1$, and decrements it after every T_{slot} period of silence, but suspends it if another station $\mathcal{Y}$ begins transmission – this suspension spans the acknowledgment as well (see below); when the timer reaches 0, $\mathcal{X}$ begins transmission of its packet (e.g., stations B, D and E in figure 1). Time is thus discretized by T_{slot} to support backoff timers, and a transmission typically occupies multiple slots.

The packet is transmitted in its entirety, even if there is a collision, since $\mathcal{X}$ does not do collision detection. The receiver uses the CRC (cyclic redundancy check) bits in each packet to check for collisions and, if no error is detected, sends an ACK (acknowledgment) after time T_{SIFS} (SIFS is *short interframe space*; $T_{\text{SIFS}} < T_{\text{DIFS}}$). If the sender does not detect an ACK within an ACK-timeout, it enters a *retransmit backoff*: if $\widetilde{W}$ is smaller than the *maximum window size* $2^m W$, then $\widetilde{W}$ is doubled (thus, m is the number of retransmission attempts before $\widetilde{W}$ reaches its maximum size – thereafter, the window size remains unchanged until it is reinitialized to W for a new packet); $\mathcal{X}$ sets a timer to a value uniformly chosen from less than the new $\widetilde{W}$, and (re)transmission is postponed to when this timer expires, as before.

Finally, a station must separate two consecutive transmissions by a random backoff, even if the channel is idle for DIFS after the first transmission (e.g., station B in figure 1.)

3. The performance model

The signal propagation delay in a local area network is very small – about 1 μsec in our simulations – so a carrier sensing protocol may be expected to have a negligible probability of collision. However, channel sensing and hardware switching take time, so time is slotted (with the help of synchronizing beacons) to accommodate these delays and transmissions start only at the beginning of slots. Hence, a window size of 32 slots means that there are only 32 possible choices of transmission times within a range of $32T_{\text{slot}}$. If, say, 10 stations are choosing transmission times from the same 32 slots, then the probability of a collision p is very high.

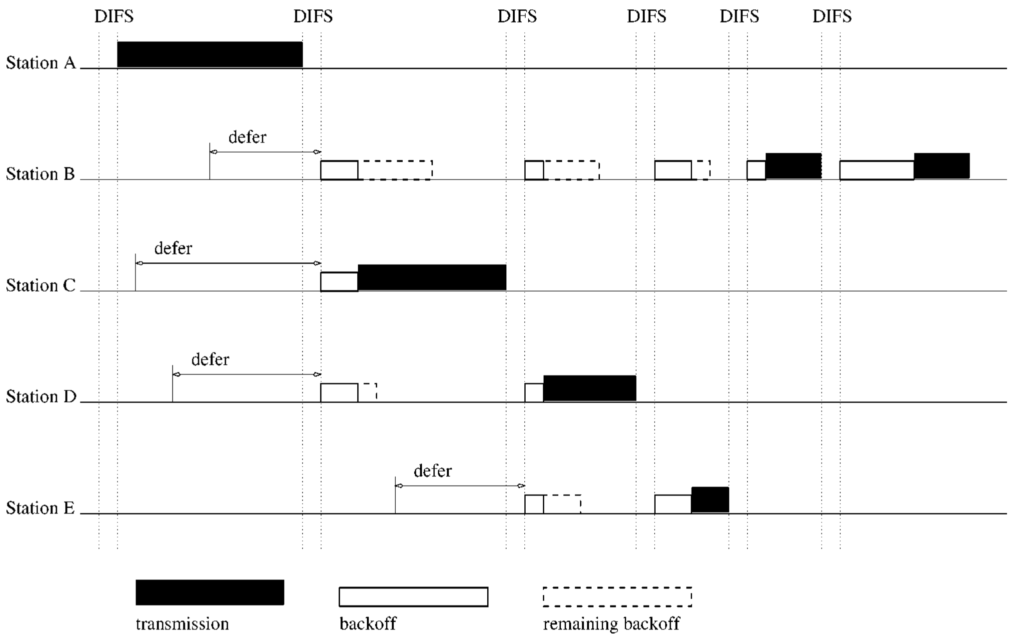


Figure 1. Basic access in the MAC protocol (with SIFS and ACK omitted).

Basic access in the 802.11 MAC protocol is designed to avoid collisions (e.g., using a random backoff if the channel is found to be busy), but collisions can still occur. While it is possible to stress the protocol with a workload that causes a high collision rate, the protocol is clearly not designed for such workloads. Specifically, workloads that cause p to exceed, say, 0.5 should probably be considered incompatible with the protocol, and require more access coordination, such as that provided by RTS/CTS.

Our performance analysis for basic access is therefore restricted to workloads for which p is less than 0.5. Note that this is not the same as saying workload is low, since our analysis is focused on the saturation throughput; in other words, saturation happens for small p as well.

Indeed, simulation results show that our model is accurate even when the saturation throughput is decreasing as more stations are added. It is a common misconception that throughput degrades (only) because of too many collisions when, in fact, this can also happen because senders spend too much time waiting for each other in backoffs, and it is something to guard against in a CSMA/CA protocol. (A similar phenomenon occurs in transaction processing [21].)

We now present our performance model. The performance analysis of any real system requires approximations if it is to be tractable. Only idealized queues (e.g., M/M/1) and networks (e.g., product-form) can be analyzed exactly [11]. For example, stochastic analysis of a protocol must adopt assumptions (e.g., exponential distributions, independent variables) and techniques (e.g., state space aggregation) that are – in effect – approximations of reality.

Our model makes numerous approximations, some implicitly, but these approximations are carefully chosen by using the simulator to check their effect on the model's accuracy. The tradeoff is between simplicity (to facilitate analysis) and accuracy, since the conclusions drawn from the approximate model are acceptable only if it is numerically accurate.

We begin by estimating the probability of a collision, and table 1 lists the variables used in our analysis. At saturation, a transmitting station will always have a queue of packets to send, so every transmission is preceded by a backoff (see the final remark in section 2 protocol description). Since the backoff is uniformly distributed over $0, 1, \ldots, W-1$ for the first attempt, the backoff timer is $W/2$ (slots), on average; for simplicity, we use $W/2$ instead of $(W-1)/2$.

Each transmission has probability p of collision, and a station transmits a packet multiple times until it receives an acknowledgment (thus indicating a successful transmission), so we can model the number of transmissions per packet as geometrically distributed with probability of success $1-p$. Furthermore, each collision causes a dilation of the contention window until the maximum is reached, so the backoff window size is

$$W_{\text{backoff}} = (1-p)\frac{W}{2} + p(1-p)\frac{2W}{2} + \cdots + p^m(1-p)\frac{2^m W}{2} + p^{m+1}\frac{2^m W}{2} = \frac{1-p-p(2p)^m}{1-2p}\,\frac{W}{2}. \qquad (1)$$

Note that the summation can be evaluated at $p = 0.5$, although the latter expression seems ill-defined for $p = 0.5$.

Consider now the probability that when station $\mathcal{X}$ begins transmission, it collides with another station $\mathcal{Y}$. Since $\mathcal{X}$'s backoff timer is suspended whenever $\mathcal{Y}$ is transmitting, it appears to $\mathcal{X}$ that $\mathcal{Y}$'s transmission occupies only one slot (the first slot of $\mathcal{Y}$'s transmission). On $\mathcal{X}$'s time-line, $\mathcal{Y}$ thus appears to be silent except for every W_{backoff}th slot, as illustrated in figure 2. (As mentioned in the introduction, our technique uses average values wherever possible.)

Assuming there are sufficiently many other stations so that $\mathcal{X}$ and $\mathcal{Y}$'s transmissions are not synchronized, then $\mathcal{X}$ could begin transmission anywhere along this time-line, so its probability of colliding with $\mathcal{Y}$ is $1/W_{\text{backoff}}$. The probability that $\mathcal{X}$ collides with any of the other stations can therefore be approximated as $1-(1-1/W_{\text{backoff}})^{n-1}$, i.e.,

$$p = 1 - \left(1 - \frac{2(1-2p)}{1-p-p(2p)^m}\,\frac{1}{W}\right)^{n-1}. \qquad (2)$$

The next task is to derive the saturation throughput and other performance measures. If r_{success} is the rate of successful transmissions and r_{xmit} the rate of transmissions (including collisions), then the average number of transmissions per packet is $r_{\text{xmit}}/r_{\text{success}}$, so the geometric distribution gives

$$\frac{1}{1-p} = \frac{r_{\text{xmit}}}{r_{\text{success}}}. \qquad (3)$$

Table 1
Glossary

n	the number of stations in a wireless cell
W	minimum window size
m	maximum window size is $2^m W$
W_{backoff}	(average) backoff window size
T_{slot}	slot time
T_{SIFS}	time duration of short interframe space
T_{DIFS}	time duration of distributed interframe space
T_{payload}	time to transmit payload bits
T_{physical}	time to transmit packet (including headers)
T_{ACK}	transmission time for an acknowledgement
T_{cycle}	time between the start of two payload transmissions
r_{success}	rate of successful (i.e., uncollided) transmissions
$r_{\text{collision}}$	rate of collisions
r_{xmit}	rate of transmissions
p	probability of a collision
S	channel utilization by successful transmission of payload bits
u_{total}	total channel utilization (including collisions)

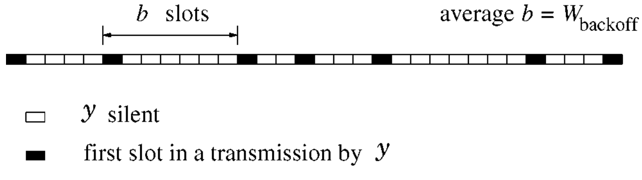


Figure 2. $\mathcal{Y}$'s activity as seen on $\mathcal{X}$'s time-line.

We count multiple transmissions that collide as one collision; to a first approximation, each collision is between just two transmissions, so the rate of collisions $r_{\text{collision}}$ is given by

$$r_{\text{xmit}} - r_{\text{success}} = 2r_{\text{collision}}. \quad (4)$$

The transmission episodes along the time-line occur at rate $1/T_{\text{cycle}}$ – where T_{cycle} is the time between two payload transmissions – and consist of successful and collided transmissions, so

$$\frac{1}{T_{\text{cycle}}} = r_{\text{success}} + r_{\text{collision}}. \quad (5)$$

Equations (3)–(5) give

$$r_{\text{success}} = \frac{2(1-p)}{2-p}\frac{1}{T_{\text{cycle}}}, \quad (6)$$

$$r_{\text{xmit}} = \frac{2}{2-p}\frac{1}{T_{\text{cycle}}}, \quad (7)$$

$$r_{\text{collision}} = \frac{p}{2-p}\frac{1}{T_{\text{cycle}}}. \quad (8)$$

The time T_{cycle} consists of the physical transmission time, a SIFS, an acknowledgment, a DIFS (the last three are also used to approximate the ACK-timeout for collided transmissions) and a silent interval. At saturation, most transmissions are preceded by a minimum backoff of W; when n stations uniformly choose a time in W, the separation between choices has mean $W/(n+1)$. In particular, the station that picks the earliest slot breaks the channel silence after time $WT_{\text{slot}}/(n+1)$, so we use the approximation

$$T_{\text{cycle}} = T_{\text{physical}} + T_{\text{SIFS}} + T_{\text{ACK}} + T_{\text{DIFS}} + \frac{W}{n+1}T_{\text{slot}}. \quad (9)$$

We can now derive expressions for performance measures such as the average number of retransmissions, the average access delay to the channel, etc. Of these, the two that are most important are total channel utilization

$$u_{\text{total}} = r_{\text{success}}(T_{\text{physical}} + T_{\text{ACK}}) + r_{\text{collision}}T_{\text{physical}} \quad (10)$$

and saturation throughput

$$S = r_{\text{success}}T_{\text{payload}} = \frac{2(1-p)}{2-p} \times \frac{T_{\text{payload}}}{T_{\text{physical}} + T_{\text{SIFS}} + T_{\text{ACK}} + T_{\text{DIFS}} + \frac{W}{n+1}T_{\text{slot}}}, \quad (11)$$

i.e., the fraction of channel bandwidth that is used to successfully transmit payload bits if every station's buffer is always occupied.

4. Numerical validation

For validation, we use Bianchi et al. discrete-event simulator (written in C++). The number of stations n is fixed, all packets are of the same size and not fragmented, the time between packet generation is exponentially distributed, and there is no noise and no hidden terminal.

Table 2
Packet format and parameter values used in the simulation.

packet payload	8184 bits
MAC header	272 bits
PHY header	128 bits
ACK length	240 bits
Channel Bit Rate	1 Mbits/sec
Propagation Delay	1 μsec
RxTx_Turnaround_Time	20 μsec
Busy_Detect_time	29 μsec
SIFS	28 μsec
DIFS	130 μsec
ACK_Timeout	300 μsec
Slot Time	51 μsec

The packet format and parameter values used in the simulation are shown in table 2. The latter values are those for the frequency hopping physical layer. In table 2, RxTx_Turnaround_Time is the time between when the MAC transmission request is sent to the physical layer and when the first bit is transmitted; the Busy_Detect_Time is the time between when the channel changes state and when the MAC layer receives notification.

In the figures that follow, each sample point represents the average from 10 simulation runs; each run is for 10 sec of simulated time, with the first 5 secs of transient behavior ignored. The 95% confidence interval for each sample point is usually too small to be shown in the figures. The saturation throughput S is obtained by choosing packet generation rates high enough that the send buffers are always occupied.

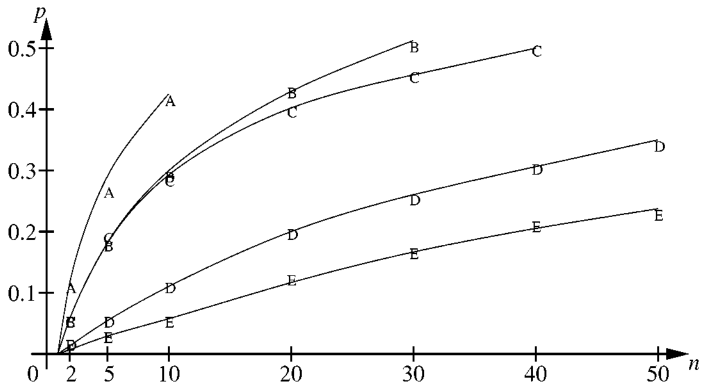
To verify the numerical accuracy of our model, we compare its prediction of the two most important performance measures p (see figure 3) and S (see figures 4 and 5) to results from the simulator. The curves in each graph are plotted by solving (2) and evaluating (11); for $W = 16$ and $W = 32$, they are truncated at $p = 0.5$, since higher values of p are of no practical interest. The numerical values for these graphs are tabulated in tables 8 and 9.

The figures show that the model is reasonably accurate, despite its many approximations. Moreover, the accuracy holds good even when S drops as we increase n (figure 4) or W (figure 5). In particular, for $W = 128$ and $W = 256$ (D and E curves) S is maximum when p is small (around 0.1 – compare figures 3 and 4), thus indicating the oppressive effect of backoffs. Indeed, we will show later (in the proof for claim 5) that S reaches a maximum when

$$p \approx \frac{2}{2 + \sqrt{\frac{T_{\text{physical}} + T_{\text{SIFS}} + T_{\text{ACK}} + T_{\text{DIFS}}}{T_{\text{slot}}}}},$$

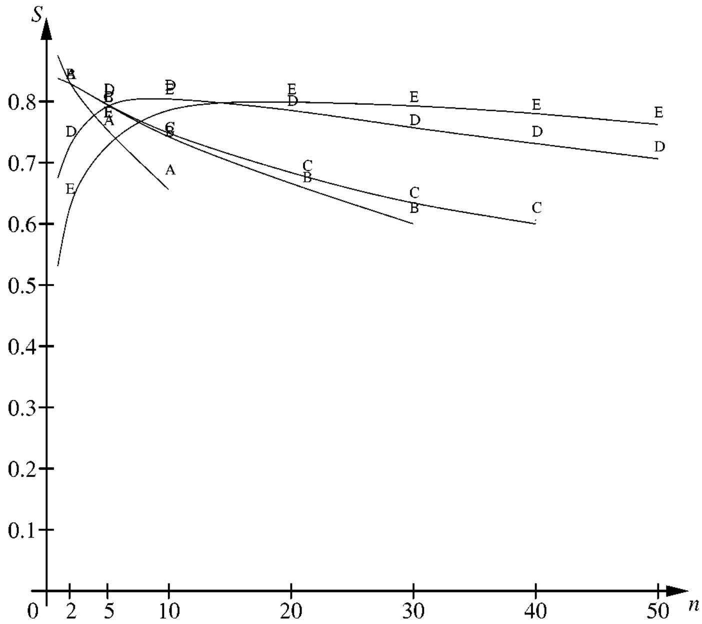
which is small if $T_{\text{physical}} \gg T_{\text{slot}}$.

Figures 4 and 5 have another significance: In essence, the model makes several first-order approximations, so it is possible that we go overboard and end up with a monotonic approximation that is good only for light workloads. These



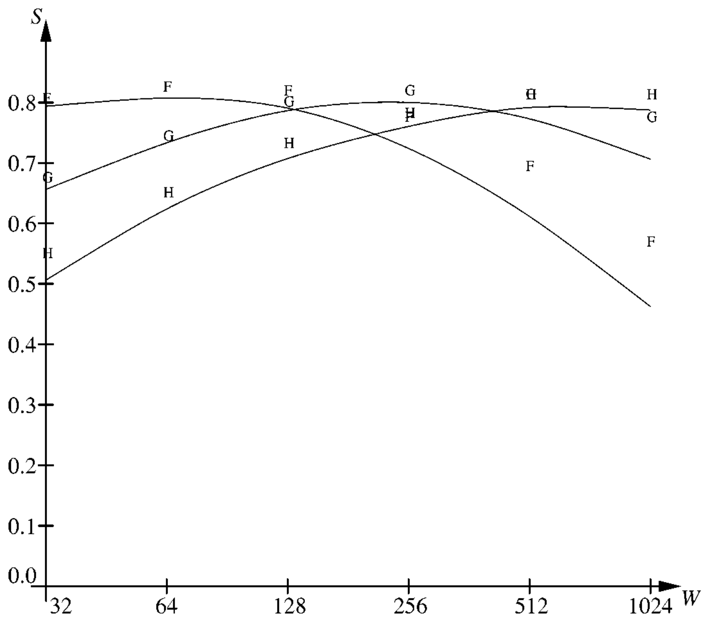
	W	m
A	16	3
B	32	3
C	32	5
D	128	3
E	256	3

Figure 3. Comparison of model's prediction (solid line) against simulation results for p.



	W	m
A	16	3
B	32	3
C	32	5
D	128	3
E	256	3

Figure 4. Comparison of model's prediction (solid line) against simulation results for S.



	n	m
F	5	3
G	20	3
H	50	3

Figure 5. Comparison of model's prediction (solid line) against simulation results for S.

graphs show that, in fact, the approximations are not overdone, that they can still capture the non-monotonic behavior in S.

We will present more data from the simulator, and study the interesting shapes in figures 4 and 5, in the next section.

5. Protocol analysis

For many analytic models in the literature, once the model is numerically validated, it is used to analyze the system by solving for various input values, and examining the resulting graphs. This is not very different from analyzing graphs produced by simulators, except possibly faster. We believe an analytic model must do better: it should be *analytically tractable*, i.e., serve as a tool for dissecting a system so that one can mathematically analyze the various forces inside that determine system behavior through their interaction.

We now use our model to do such an analysis of the protocol. As with any approximate performance model of a system, one must be careful that the properties deduced from the model are in fact properties of the system, and not just properties of the approximation itself. In the following, we always make an attempt at analytic validation, i.e., verify that the results from our analysis apply to the simulated behavior of the protocol.

Claim 1. The probability of a collision does not depend on packet length, the latency in crossing the MAC/PHY layers (i.e., RxTx_Turnaround_Time and Busy_Detect_Time), the acknowledgment timeout, the interframe spaces and the slot size; it only depends on W, m and n.

Proof. This follows from (2). □

To validate this claim, we ran the simulator for various choices of RxTx_Turnaround_Time, Busy_Detect_Time, T_{payload}, T_{SIFS}, T_{DIFS} and T_{slot}. Table 3 shows that only W, m and n affect p. (In fact, the effect of m is minimal – see claim 3.)

Even the dependence on W and n can be simplified:

Claim 2. When n is large (say, $n \geqslant 5$) but smaller than $2^{m-1}W$, the protocol's performance (p, r_{success}, r_{xmit}, $r_{\text{collision}}$, u_{total}, S) depends on W and n only through the gap $g = W/(n-1)$.

Proof. Taking a first-order approximation of (2), we get

$$p = \frac{2(1-2p)}{1-p-p(2p)^m}\frac{n-1}{W}. \qquad (12)$$

Now let

$$f(x) = x\frac{1-x-x(2x)^m}{2(1-2x)} - \frac{n-1}{W}.$$

Then

$$f(0) = -\frac{n-1}{W} < 0 \quad \text{and}$$
$$f(1) = \frac{2^{m-1}W-(n-1)}{W} > 0.$$

Moreover, from (1),

$$f(x) = \frac{1}{2}\left(x + x^2\sum_{k=0}^{m-1}(2x)^k\right) - \frac{n-1}{W}$$

is an increasing and continuous function of x, so $f(x) = 0$ has exactly one root in $(0, 1)$, as illustrated in figure 6. Thus, (12) gives a valid and unique value for p. Furthermore, for large n, we can approximate (11) by

$$S = \frac{2(1-p)}{2-p} \times \frac{T_{\text{payload}}}{T_{\text{physical}} + T_{\text{SIFS}} + T_{\text{ACK}} + T_{\text{DIFS}} + \frac{W}{n-1}T_{\text{slot}}}. \qquad (13)$$

The claim follows from (12) and (13). □

The requirement $n < 2^{m-1}W$ is not severe: even for modest values of m and W ($m = 3$ and $W = 16$, say, so $2^{m-1}W = 64$), the condition accommodates a number of stations that is large for a wireless local area network.

The approximation in (12) is gross, and the one in (13) is arbitrary, so claim 2 may, in fact, not hold for the simulation

Table 3
Validating claim 1: p only depends on W, m and n.

RxTx	BusyDT	T_{slot}	T_{SIFS}	T_{DIFS}	ACK_T	T_{payload}	n	W	m	p
10	15	27	18	120	270	512	10	256	3	0.062
20	29	51	28	130	280	512	10	256	3	0.065
20	29	51	28	130	280	8184	10	256	3	0.061
40	50	92	48	150	300	512	10	256	3	0.065
40	50	92	48	150	300	8184	10	256	3	0.067
10	15	27	18	120	270	8184	5	32	3	0.175
20	29	51	28	130	280	8184	5	32	3	0.179
20	29	51	28	130	280	512	5	32	3	0.181
40	50	92	48	150	300	8184	5	32	3	0.178
40	50	92	48	150	300	512	5	32	9	0.180
20	29	51	28	130	280	8184	5	32	9	0.186

RxTx is RxTx_Turnaround_Time, BusyDT is Busy_Detect_Time, ACK_T is ACK-timeout.

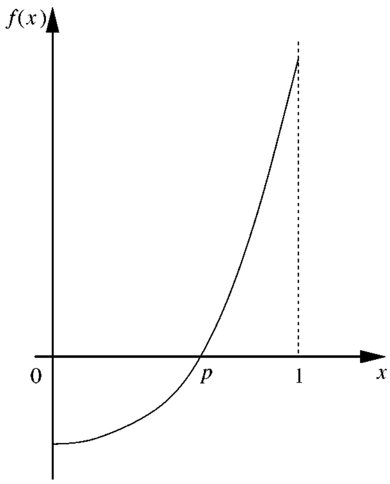


Figure 6. For $n < 2^{m-1}W$, approximation (12) gives exactly one p between 0 and 1.

data. To validate this claim, the simulator's values for p and S in figures 3 and 4 are plotted against g in figures 7 and 8 for $n \geqslant 5$. We see that the data points do in fact form a single curve, despite large variations in W (16 to 256) and n (5 to 50).

Claim 2 now makes the complicated, unintuitive behavior in figure 4 easy to understand: all the curves are manifestations of the same curve in figure 8. For $W = 16$ and $W = 32$, the A, B and C curves are taken from the pre-maximum segment in figure 8 (because W is small), so there are no maximums; the D and E curves for $W = 128$ and $W = 256$ are two segments of the same curve that are scaled, reflected, translated and superimposed in figure 4. The choice of m splits the curve for $W = 32$ (B and C) in figure 4 when n is comparable to W because the relevant segment (near $g = 0$) in figure 8 is steep in this range and sensitive to the window size.

Claim 2 thus reduces the three input parameters W, n and m that determine p (claim 1) to just two: the gap g and m. Intuitively, if n points are equally spaced over an interval of length W, with one point at each end, then $g = W/(n-1)$ is the separation between two points – hence the term *gap*. In other words, the protocol's performance is determined by how the minimum window size is divided among the stations; the gap thus encapsulates the effect of shared access. (A. Weiss pointed out to us that a parameter similar to g was observed in the Aloha protocol as well [4,16].)

It is intuitively obvious that n has a strong effect on S, and one would expect that the effect of W is secondary. After all, the protocol can enlarge the window beyond its minimum size through backoffs. However, Bianchi et al. have observed that throughput performance is strongly dependent on W. The following corollary of claim 2 shows that, in fact, W and n have *equivalent* effect on S:

Corollary 1. Halving the initial window size has similar impact on saturation throughput as doubling the number of stations.

Table 4
Validating corollary 1: halving W has the same effect on S as doubling n.

n	W	m	$S_{(2n,W)}$	$S_{(n,w/2)}$
5	16	3	0.70	0.71
5	16	5	0.71	0.72
5	32	5	0.76	0.77
10	32	3	0.68	0.70
15	32	3	0.63	0.64
10	128	3	0.80	0.80
15	128	3	0.77	0.77
10	256	3	0.82	0.83
50	256	3	0.73	0.78

Proof. This follows from (12) and (13) and the fact that

$$\frac{W/2}{n-1} \approx \frac{W}{2n}. \qquad \square$$

Let S be the saturation throughput for some choice of W, m and n, $S_{(n,W/2)}$ the new S when W is halved, and $S_{(2n,W)}$ the new S when n is doubled. Table 4 shows simulation measurements of $S_{(n,W/2)}$ and $S_{(2n,W)}$ for a wide range of values in W and n; indeed, each pair is approximately equal, thus supporting the corollary.

This is another example of what we mean by analytical validation of the model: Analysis of our model leads to the claim that $S_{(n,W/2)} = S_{(2n,W)}$, and this claim is then checked with simulated values of $S_{(n,W/2)}$ and $S_{(2n,W)}$. This is unlike the numerical validation in the previous section, where the comparison is between the model's prediction and the simulation's result.

Corollary 1 points out the importance of the minimum window size. We now show that, in contrast, the maximum window size has only a small effect on p and S.

Claim 3. Suppose $p < 0.5$. The choice of maximum window size has minimal effect (namely, $\mathrm{O}(p(2p)^m)$) on the collision rate and saturation throughput.

Proof. Suppose

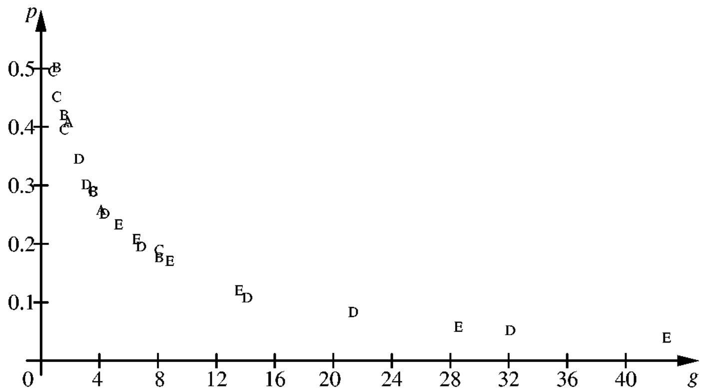
$$p_m \frac{1 - p_m - p_m(2p_m)^m}{1 - 2p_m} = 2\frac{n-1}{W} = p_\infty \frac{1 - p_\infty}{1 - 2p_\infty},$$

i.e., p_m is the root of (12) for maximum window size $2^m W$ and p_∞ is the root for unbounded window size (using $2p < 1$, so $\lim_{m\to\infty}(2p)^m = 0$). Let $\Delta_p = (p_\infty - p_m)/p_m$. Ignoring the term Δ_p^2, this gives

$$\Delta_p = -p_m(2p_m)^m \frac{1 - 2p_m}{1 - 2p_m + 2p_m^2(1 - (2p_m)^m)},$$

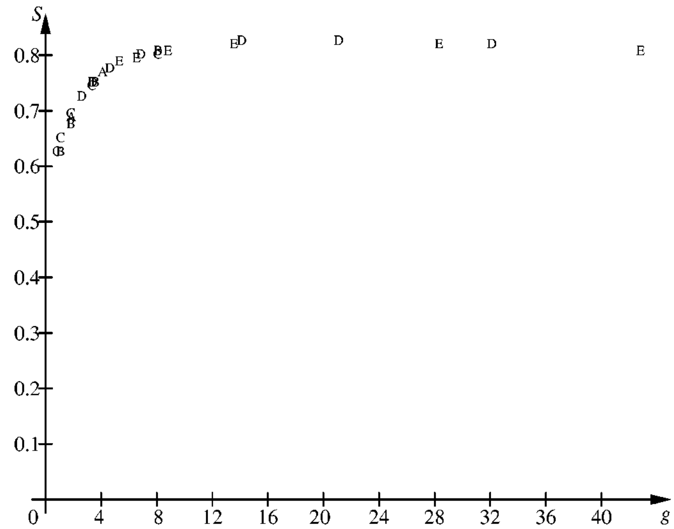
so

$$|\Delta_p| < p_m(2p_m)^m. \qquad (14)$$



	W	m
A	16	3
B	32	3
C	32	5
D	128	3
E	256	3

Figure 7. Validating claim 2: p is a function of W and n only through $g(= W/(n-1))$. Values for W, m and n are as in figure 3; $n \geqslant 5$. One data point is added for D and for E at $n = 7$ ($g = 21.3$ and $g = 42.7$).



	W	m
A	16	3
B	32	3
C	32	5
D	128	3
E	256	3

Figure 8. Validating claim 2: S is a function of W and n only through $g(= W/(n-1))$. Values for W, m and n are as in figure 4; $n \geqslant 5$. One data point is added for D and for E at $n = 7$ ($g = 21.3$ and $g = 42.7$).

Similarly, if S_m and S_∞ are the corresponding saturation throughputs and $\Delta_S = (S_\infty - S_m)/S_m$, then we get from (13)

$$\Delta_S = \frac{2(1-p_m(1+\Delta_p))}{2-p(1+\Delta_p)} \frac{2-p_m}{2(1-p_m)} - 1 = -\frac{p_m \Delta_p}{(2-p_m(1+\Delta_p))(1-p_m)}.$$

Since $\Delta_p < 0$, we have

$$|\Delta_S| < \frac{p_m|\Delta_p|}{(2-p_m)(1-p_m)}. \tag{15}$$

For $p < 0.5$, this bound is smaller than

$$\frac{2}{3}|\Delta_p| = \mathrm{O}\big(p_m(2p_m)^m\big). \qquad \square$$

For example, if $m = 3$ and $p_m = 0.3$, then the right-hand side of (14) is less than 0.065 and the right-hand side of (15) is less than 0.017, showing that the choice of m (between 3 and ∞) has minimal impact on p and S.

To validate this claim with the simulator, table 5 examines the effect of choosing between $2^m W$ and $2^i W$ as the maximum window size for some values of W and n. The simulations show that the effect is even smaller than the tight bounds derived with our model.

Using claim 3, we can now express p and S explicitly in terms of the parameters in table 1:

Claim 4. For large n, the saturation throughput can be approximated by

$$S = \frac{2(1-p)}{2-p} \times \frac{T_{\text{payload}}}{T_{\text{physical}} + T_{\text{SIFS}} + T_{\text{ACK}} + T_{\text{DIFS}} + gT_{\text{slot}}}, \tag{16}$$

where

$$p = \frac{1}{2}\left(1 + \frac{4}{g} - \sqrt{1 + \left(\frac{4}{g}\right)^2}\right), \tag{17}$$

$$g = \frac{W}{n-1}.$$

Table 5
Validating claim 3: The minimal effect of maximum window size on p and S.

W	n	g	m	i	p_m	p_i	$\|(p_i - p_m)/p_m\|$	S_m	S_i	$\|(S_i - S_m)/S_m\|$
16	5	2.7	3	10	0.268	0.268	0.0%	0.769	0.769	0.0%
32	5	5.3	3	9	0.179	0.186	3.9%	0.809	0.805	0.0%
32	5	5.3	5	9	0.183	0.186	1.6%	0.806	0.805	0.0%
32	2	10.7	3	9	0.053	0.053	0.0%	0.848	0.848	0.0%
128	10	11.6	3	7	0.108	0.110	1.9%	0.829	0.828	0.0%
128	30	4.1	3	7	0.257	0.254	1.2%	0.773	0.774	0.0%
256	30	8.3	3	6	0.170	0.165	2.9%	0.808	0.810	0.0%

Proof. Since the choice of m has minimal impact on p, we can approximate (12) by

$$\frac{p(1-p)}{1-2p} = \frac{2}{g}. \tag{18}$$

This has solution

$$p = \frac{1}{2}\left(1 + \frac{4}{g} - \sqrt{1 + \left(\frac{4}{g}\right)^2}\right).$$

(The positive square root gives $p > 1$, which is impossible.) The claim follows from (13). □

Equations (16) and (17) are plotted in figures 9 and 10, which show that the explicit expressions are reasonably accurate in estimating p and S. The numerical values for the figures are tabulated in tables 8 and 9.

Figure 8 shows that there is a value of g for which S is maximum. The next claim locates this maximum.

Claim 5. Suppose $T_{\text{physical}} + T_{\text{SIFS}} + T_{\text{ACK}} + T_{\text{DIFS}} \gg 4T_{\text{slot}}$. The saturation throughput is maximum when

$$W = \sqrt{\frac{T_{\text{physical}} + T_{\text{SIFS}} + T_{\text{ACK}} + T_{\text{DIFS}}}{T_{\text{slot}}}(n-1)}.$$

Proof. Let $b = (T_{\text{physical}} + T_{\text{SIFS}} + T_{\text{ACK}} + T_{\text{DIFS}})/T_{\text{slot}}$. From (16),

$$\frac{dS}{dg} = \frac{-2}{2-p}\frac{T_{\text{payload}}}{(b+g)T_{\text{slot}}}\left(\frac{1}{2-p}\frac{dp}{dg} + \frac{1-p}{b+g}\right),$$

so the maximum occurs when

$$\frac{dp}{dg} = -\frac{(2-p)(1-p)}{b+g}.$$

By (18), we have

$$\frac{dp}{dg} = -\frac{p^2(1-p)^2}{2(1-2p+2p^2)}.$$

These two equations give

$$b = \frac{4(1-3p+4p^2-p^3)}{p^2(1-p)} \approx \left(\frac{2(1-p)}{p}\right)^2, \quad \text{i.e.,}$$
$$p \approx \frac{2}{\sqrt{b}+2}.$$

Since $\sqrt{b} \gg 2$, we get $p \approx 2/\sqrt{b}$. This is small, so (18) gives $g \approx 2/p \approx \sqrt{b}$, and the claim follows. □

For the parameter values in the simulation, claim 5 puts the maximum at $g = W/(n-1) \approx 13$. With $g = 13$ and using $W = 32$ (as recommended in the 802.11 specification), the optimum n is just 3 stations in a cell; a class of 15 students, say, should instead set $W = 182$ (assuming the parameter values and traffic assumptions apply). Since the curve for S is very flat around the maximum, the value 13 here is just a "ballpark" figure – it wouldn't matter much to S if the ratio deviates somewhat from 13.

Because the maximum is flat, randomness in the sample average S makes locating the maximum – and hence validating claim 5 – difficult. However, the claim is consistent with an analysis by Bianchi et al., who show that, for constant backoff (i.e., no exponential enlargement of the backoff window size), S is maximum at

$$W \approx n\sqrt{2\frac{T_{\text{physical}} + T_{\text{SIFS}} + T_{\text{ACK}} + T_{\text{DIFS}}}{T_{\text{slot}}}}.$$

It follows that, for the same n, saturation throughput under constant backoff is maximum at a larger (initial) window size than under exponential backoff; this formally confirms one's expectation that, with exponential backoff to help adjust the approximate window size, the protocol can afford to be aggressive and start with a smaller W.

Claim 5 leads to the following instructive rules:

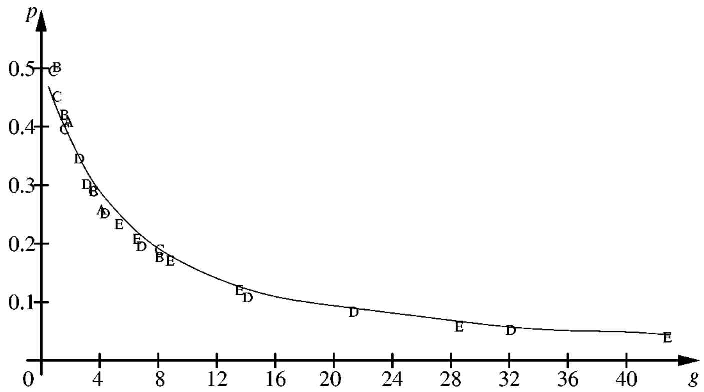
Corollary 2. For maximum throughput,

(i) W should increase linearly with n and

(ii) W should be proportional to the square root of the packet size (in slots), assuming $T_{\text{physical}} \gg T_{\text{SIFS}} + T_{\text{ACK}} + T_{\text{DIFS}}$.

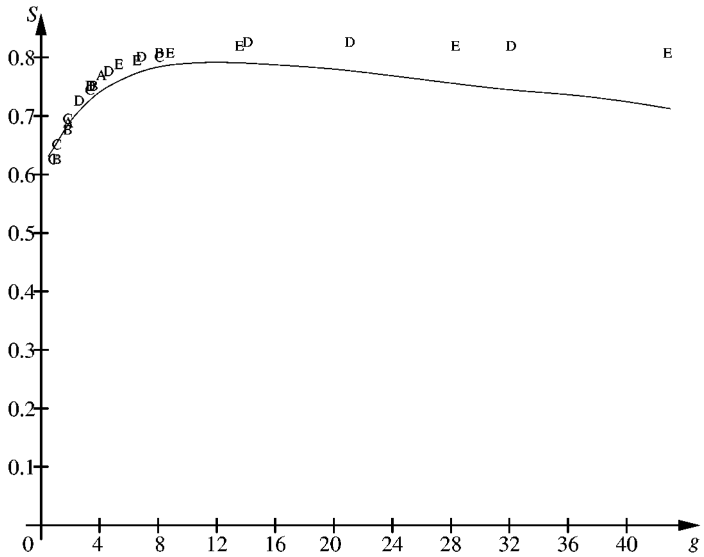
Although packet sizes do not affect p (claim 1), they do affect S, thus leading to corollary 2(ii).

We have established in Corollary 1 that a small W is equivalent to a large n, thus causing excessive collisions that increase the variance in channel access time. On the other hand, a large W leads to idleness and large access times. The protocol should therefore try to adjust W to suit the conditions, and corollary 2 suggests how this can be done: by monitoring the traffic, the nodes in a cell can estimate n [2]



	W	m
A	16	3
B	32	3
C	32	5
D	128	3
E	256	3

Figure 9. Validation of closed-form expression (17) for p. (Data points are as in figure 7.)



	W	m
A	16	3
B	32	3
C	32	5
D	128	3
E	256	3

Figure 10. Validation of closed-form expression (16) for S. (Data points are as in figure 8.)

and the packet size, and scale W up or down when they increase or decrease. Later, we will see (from claim 7) when this adjustment should be made.

The protocol's behavior is thus driven by two underlying forces, and the maxima in S is a tradeoff between bandwidth wastage by collisions and by backoffs. The next claim confirms this intuition:

Claim 6. Suppose $T_{\text{physical}} \gg T_{\text{ACK}}$. Bandwidth wasted by collisions exceeds idle bandwidth caused by backoffs if and only if

$$r_{\text{xmit}} > \frac{1}{T_{\text{physical}}}.$$

Proof. Let $u_{\text{collision}}$ be the fraction of bandwidth wasted by collisions. We can estimate the channel silence due to backoffs by $1 - u_{\text{total}}$. The difference between the two is

$$\begin{aligned}\delta &= u_{\text{collision}} - (1 - u_{\text{total}}) \\ &= r_{\text{collision}} T_{\text{physical}} - 1 + \big(r_{\text{success}}(T_{\text{physical}} + T_{\text{ACK}}) \\ &\quad + r_{\text{collision}} T_{\text{physical}}\big) \quad \text{by (10)} \\ &\approx (2r_{\text{collision}} + r_{\text{success}}) T_{\text{physical}} - 1 \\ &\quad \text{since } T_{\text{ACK}} \ll T_{\text{physical}} \\ &= r_{\text{xmit}} T_{\text{physical}} - 1, \quad \text{by (4).}\end{aligned}$$

Hence, $\delta > 0$ if and only if $r_{\text{xmit}} T_{\text{physical}} > 1$, and the claim follows. □

The claim is intuitively appealing because $1/T_{\text{physical}}$ is the maximum transmission rate if packets do not overlap, so the result says that to exceed this rate is to waste bandwidth on collisions, while a lower rate causes excess idle bandwidth. (Note that u_{total} need not be 1 when $r_{\text{xmit}} = 1/T_{\text{physical}}$, because some transmissions may collide and overlap; in other words, there may be significant idle bandwidth even if $r_{\text{xmit}} = 1/T_{\text{physical}}$.)

As validation, table 6 shows the simulator's value of $r_{\text{xmit}} T_{\text{physical}}$ decreasing from one side of 1 to the other as W increases, with the crossover happening near where W is optimum (i.e., S is maximum).

Having examined the choice of W for a fixed n, we now consider: for a given W and traffic rate of λ packets/sec from each station, what is the *congestion point*, i.e., the value

Table 6
Validating claim 6: $r_{\text{xmit}}T_{\text{physical}} = 1$ when W is optimum.

$n = 5,\ m = 3$

W	32	64	128	256	512	1024
S	0.81	<u>0.83</u>	0.83	0.78	0.70	0.57
$r_{\text{xmit}}T_{\text{physical}}$	<u>1.04</u>	<u>0.98</u>	0.91	0.84	0.74	0.60

$n = 20,\ m = 3$

W	32	64	128	256	512	1024
S	0.68	0.73	0.80	<u>0.82</u>	0.82	0.77
$r_{\text{xmit}}T_{\text{physical}}$	1.24	1.15	<u>1.04</u>	<u>0.99</u>	0.91	0.84

$n = 50,\ m = 3$

W	32	64	128	256	512	1024
S	0.56	0.65	0.73	0.79	0.81	<u>0.83</u>
$r_{\text{xmit}}T_{\text{physical}}$	1.48	1.30	1.16	1.06	<u>1.01</u>	<u>0.94</u>

Table 7
Validating claim 7: Comparing the throughput at the congestion point n^* and at saturation.

λ per station	W	m	n_{sim} from simulation	n^* eqn. (19)
3.0	32	3	28	27.5
3.0	32	6	29	27.5
3.0	64	6	30	29.3
6.0	32	3	15	14.6
6.0	64	3	16	15.7
6.0	64	6	16	15.7
6.0	1024	3	16	18.2
9.0	32	3	10	10.2
9.0	32	9	10	10.2
9.0	128	3	11	11.4
12.0	8	3	7	6.9
12.0	8	9	7	6.9
12.0	16	3	7	7.3
12.0	32	3	8	7.8

Table 8
Numerical values for graphs.

	W	m	n	$W/(n-1)$	p sim.	p eqn. (2) fig. 3	p eqn. (17) fig. 9	S sim.	S eqn. (11) fig. 4	S eqn. (16) fig. 10
	16	3	1			0.000			0.872	
A	16	3	2	16.0	0.110	0.110	0.110	0.839	0.833	0.787
	16	3	5	4.0	0.268	0.286	0.293	0.769	0.748	0.738
	16	3	10	1.8	0.411	0.425	0.394	0.688	0.660	0.681
	32	3	1			0.000			0.837	
B	32	3	2	32.0	0.053	0.059	0.059	0.848	0.834	0.748
	32	3	5	8.0	0.179	0.182	0.191	0.809	0.796	0.780
	32	3	10	3.6	0.291	0.302	0.310	0.758	0.737	0.729
	32	3	20	1.7	0.425	0.432	0.399	0.681	0.655	0.678
	32	3	30	1.1	0.503	0.511	0.432	0.631	0.595	0.656
	32	5	1			0.000			0.837	
C	32	5	2	32.0	0.053	0.059	0.059	0.848	0.834	0.748
	32	5	5	8.0	0.186	0.181	0.191	0.805	0.797	0.780
	32	5	10	3.6	0.288	0.293	0.310	0.758	0.743	0.729
	32	5	20	1.7	0.395	0.401	0.399	0.699	0.677	0.678
	32	5	30	1.1	0.456	0.461	0.432	0.662	0.635	0.656
	32	5	40	0.8	0.495	0.502	0.449	0.636	0.603	0.644
	128	3	1			0.000			0.672	
D	128	3	2	128.0	0.017	0.015	0.015	0.764	0.731	0.524
	128	3	5	32.0	0.055	0.057	0.059	0.824	0.791	0.748
	128	3	7	21.3	0.085	0.082	0.085	0.826	0.801	0.777
	128	3	10	14.2	0.108	0.116	0.121	0.829	0.803	0.789
	128	3	20	6.7	0.196	0.203	0.215	0.801	0.782	0.772
	128	3	30	4.4	0.257	0.265	0.278	0.773	0.755	0.745
	128	3	40	3.3	0.305	0.313	0.321	0.750	0.729	0.723
	128	3	50	2.6	0.340	0.352	0.351	0.731	0.707	0.707
	256	3	1			0.000			0.532	
E	256	3	2	256.0	0.008	0.008	0.008	0.665	0.615	0.370
	256	3	5	64.0	0.027	0.030	0.030	0.783	0.725	0.658
	256	3	7	42.7	0.039	0.044	0.045	0.807	0.756	0.717
	256	3	10	28.4	0.057	0.063	0.065	0.824	0.780	0.758
	256	3	20	13.5	0.120	0.121	0.127	0.822	0.798	0.789
	256	3	30	8.8	0.170	0.167	0.178	0.808	0.792	0.783
	256	3	40	6.6	0.209	0.206	0.219	0.793	0.779	0.770
	256	3	50	5.2	0.233	0.239	0.253	0.784	0.766	0.757

Table 9
Numerical values for graphs.

	W	m	n	p sim.	p eqn. (2)	S sim.	S eqn. (11) fig. 5
F	32	3	5	0.179	0.182	0.809	0.796
	64	3	5	0.110	0.106	0.828	0.812
	128	3	5	0.055	0.057	0.824	0.791
	256	3	5	0.027	0.030	0.783	0.725
	512	3	5	0.015	0.015	0.699	0.613
	1024	3	5	0.008	0.008	0.573	0.465
G	32	3	20	0.425	0.432	0.681	0.655
	64	3	20	0.302	0.309	0.752	0.732
	128	3	20	0.196	0.203	0.801	0.782
	256	3	20	0.120	0.121	0.822	0.798
	512	3	20	0.061	0.067	0.821	0.775
	1024	3	20	0.032	0.035	0.781	0.704
H	32	3	50	0.600	0.611	0.560	0.509
	64	3	50	0.472	0.478	0.651	0.621
	128	3	50	0.340	0.352	0.731	0.707
	256	3	50	0.233	0.239	0.784	0.766
	512	3	50	0.145	0.147	0.816	0.795
	1024	3	50	0.083	0.083	0.822	0.784

of n at which saturation sets in? Since S varies with n (figure 4), we cannot determine the congestion point by dividing S's bit rate by λ's bit rate. Instead, we can use the following result:

Claim 7. Suppose $T_{\text{physical}}+T_{\text{SIFS}}+T_{\text{ACK}}+T_{\text{DIFS}} \gg WT_{\text{slot}}$ and each station sends packets at rate λ. Congestion (meaning throughput is at saturation) occurs when $n \approx n^*$ where

$$n^* = \frac{1}{\lambda T'}\left(1 - \frac{1}{3 + W\lambda T'}\right) \quad \text{and}$$
$$T' = T_{\text{physical}} + T_{\text{SIFS}} + T_{\text{ACK}} + T_{\text{DIFS}}. \tag{19}$$

Proof. For all traffic to get through, we require $n\lambda T_{\text{payload}} < S$. By (16), and since $T_{\text{physical}} + T_{\text{SIFS}} + T_{\text{ACK}} + T_{\text{DIFS}} + W/(n-1)T_{\text{slot}} \approx T'$, congestion is reached when

$$n\lambda = \frac{2(1-p)}{2-p}\frac{1}{T'}, \quad \text{or} \tag{20}$$

$$p = \frac{2 - 2n\lambda T'}{2 - n\lambda T'}. \tag{21}$$

On the other hand, from (12),

$$\frac{p(1-p)}{2(1-2p)} = \frac{n-1}{W} \approx \frac{n}{W}$$
$$= \frac{2(1-p)}{2-p}\frac{1}{T'}\frac{1}{\lambda W} \quad \text{from (20)},$$

so $2p\lambda WT' \approx 4(1-2p)$. The claim follows when p is eliminated from this equation and (21). □

Let n_{sim} be the maximum number of stations (each sending λ packets/sec) for which the simulator reaches steady state; specifically, with $n_{\text{sim}}+1$ stations, the delivered traffic is less than 95% of the requested traffic, so the stations have a backlog in their buffers and can no longer send at rate λ. Hence, n_{sim} is the congestion point for the simulator.

Table 7 compares n_{sim} to n^* calculated with (19). They are in close agreement, except when $W = 1024$, which violates the hypothesis underlying claim 7.

Like for corollary 2, (19) suggests a way of making W adaptive: By monitoring n and the average packet rate, the nodes in a cell can compare n to n^* and adjust W accordingly. If $n \ll n^*$, W can be reduced, thus reducing backoffs and improving access time; if n approaches n^*, the congestion can be relieved by increasing W. However, there is a limit to how far the congestion point n^* can be pushed by increasing W, since $\lim_{W\to\infty} n^* = 1/\lambda T'$.

6. Conclusion

We draw two sets of conclusions from our analysis, one on the analytic technique, and one on the protocol's behavior.

The technique of using average values is simple (section 3), yet reasonably accurate (figures 3–5, 9 and 10); the accuracy remains good even when the saturation throughput is dropping. It provides closed-form expressions (claims 4, 5 and 7) and reveals several properties of the protocol.

These properties, first proved with the technique, are confirmed by the simulator. They include: how hardware features (e.g., RxTx_Turnaround_Time and Busy_Detect_Time), protocol parameters (e.g., ACK-timeout and interference spaces) and traffic characteristics (e.g., payload) affect – or do not affect – performance (tables 3 and 5); how window size W and the number of stations n affect performance through the single parameter $g = W/(n-1)$ when n is large (figures 7 and 8); and how the optimum W is determined by the tradeoff between collisions and backoffs (table 6). The model also provides instructive guidelines on the effect of W (table 4), when to adjust it (claim 7 and table 7) and how to do so (corollary 2).

A performance model should help us *analyze* a system, so that we can *understand* and *control* its behavior. The model we present here substantially fulfills these three requirements.

Acknowledgement

Originally, our approximation for p was given by

$$p\frac{1-p-p(2p)^m}{1-2p} = \frac{2}{W}\left(1+\frac{2}{3}n\right)\frac{n-1}{n},$$

and our protocol analysis was done with this approximation and $g = W/(n+1)$. (The original versions of claims 1–7 and the corollaries were therefore slightly different from the ones presented here.) One reviewer suggested the alternative approximation in (2), and the editor (Pierre Humblet) supported it, as well as suggesting figure 2. We thank the reviewer for his insightful comments, and the editor for his helpful suggestions.

We also thank Giuseppe Bianchi for giving us a copy of their simulator, Matteo Oliveri for explaining the simulator and Mooi Choo Chuah for explaining the MAC protocol.

This work was funded in part by the National University of Singapore under ARF grants RP950626 and RP960683.

References

[1] E. Arthurs and B.W. Stuck, A theoretical performance analysis of polling and carrier sense collision detection communication systems, in: *Local Computer Networks*, eds. P.C. Ravasio, G. Hopkins and N. Naffah (North-Holland, Amsterdam, 1982) pp. 415–437.

[2] G. Bianchi, L. Fratta and M. Oliveri, Performance evaluation and enhancement of the CSMA/CA MAC protocol for 802.11 wireless LANs, in: *Proc. PIMRC*, Taipei, Taiwan (October 1996) pp. 392–396.

[3] H.S. Chhaya and S. Gupta, Performance modeling of asynchronous data transfer methods of IEEE 802.11 MAC protocol, Wireless Networks 3(3) (1997) 217–234.

[4] M. Cottrell, J.C. Fort and G. Malgouyres, Large deviations and rare events in the study of stochastic algorithms, IEEE Transactions on Automatic Control 28(9) (September 1983) 907–920.

[5] J. Hastad, T. Leighton and B. Rogoff, Analysis of backoff protocols for multiple access channels, in: *Proc. Symposium on Theory of Computing*, New York (May 1987) pp. 241–253.

[6] T.S. Ho and K.C. Chen, Performance analysis of IEEE 802.11 CSMA/CA medium access control protocol, in: *Proc. PIMRC*, Taipei, Taiwan (October 1996) pp. 407–411.

[7] K.C. Huang and K.C. Chen, Interference analysis of nonpersistent CSMA with hidden terminals in multicell wireless data networks, in: *Proc. PIMRC*, Toronto, Canada (September 1995) pp. 907–911.

[8] P802.11, IEEE Draft Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, D6.1 (May 1997).

[9] R. Jain, *The Art of Computer Systems Performance Analysis* (Wiley, New York, NY, 1991).

[10] J.H. Kim and J.K. Lee, Performance of carrier sense multiple access with collision avoidance protocols in wireless LANs, Wireless Personal Communications 11(2) (November 1999) 161–183.

[11] L. Kleinrock, *Queuing Systems, Vol. I: Theory* (Wiley-Interscience, New York, NY, 1975).

[12] L. Kleinrock and F.A. Tobagi, Packet switching in radio channels: Part I – Carrier sense multiple-access modes and their throughput-delay characteristics, IEEE Transactions on Communications 23(12) (December 1975) 1400–1416.

[13] S.M. Nor and I. Ismail, Performance of CSMA-CA MAC protocol for distributed radio local area networks, in: *Proc. PIMRC*, Toronto, Canada (September 1995) pp. 912–916.

[14] I.E. Pountourakis, E.D. Sykas and E.N. Protonotarios, Stability and throughput optimization of multichannel slotted nonpersistent CSMA protocol, Computer Communications 18(12) (December 1995) 949–958.

[15] R. Rom and M. Sidi, *Multiple Access Protocols: Performance and Analysis* (Springer-Verlag, 1990).

[16] A. Shwartz and A. Weiss, *Large Deviations for Performance Analysis* (Chapman & Hall, London, UK, 1995).

[17] W. Szpankowski, A multiqueue problem: bounds and approximations, in: *Performance of Computer-Communication Systems*, eds. H. Rudin and W. Bux (North-Holland, Amsterdam, 1984) pp. 349–364.

[18] H. Takagi and L. Kleinrock, Throughput analysis for persistent CSMA systems, IEEE Transactions on Communications COM-33(7) (July 1985) 627–638.

[19] H.H. Tan and K. Tsai, Packet output process of CSMA and CSMA/CD protocols, IEEE Transactions on Communications 44(4) (April 1996) 464–474.

[20] S. Tasaka, *Performance Analysis of Multiple Access Protocols* (MIT Press, Cambridge, MA, 1986).

[21] Y.C. Tay, N. Goodman and R. Suri, Locking performance in centralized databases, ACM Transactions on Database Systems 10(4) (December 1985) 415–462.

[22] F.A. Tobagi, Multiaccess protocols in packet communication systems, IEEE Transactions on Communications 28(4) (April 1980) 468–488.

[23] M. Visser and M. El Zarki, Voice and data transmission over an 802.11 wireless network, in: *Proc. PIMRC*, Toronto, Canada (September 1995) pp. 648–652.

[24] T. Vo-Dai, Throughput-delay analysis of the nonslotted and nonpersistent CSMA-CD protocol, in: *Local Computer Networks*, eds. P.C. Ravasio, G. Hopkins and N. Naffah (North-Holland, Amsterdam, 1982) pp. 459–476.

Y.C. Tay received his B.Sc. degree from the University of Singapore and Ph.D. degree from Harvard University. He has a joint appointment with the Departments of Mathematics and Computer Science at the National University of Singapore. His main research interest is performance modeling (transaction processing, multimedia load-sharing, parallel rendering and main memory). Other recent interests are: correctness in distributed and parallel computing, routing protocols for mobility support and ad hoc wireless networks, and application of data mining to online optimization.
E-mail: tay@acm.org
WWW: //www.math.nus.edu.sg/∼mattyc

Kee Chaing Chua received the BE degree with First Class Honours in electrical and electronic engineering from the University of Auckland, New Zealand, the MEng degree in electrical engineering from the National University of Singapore (NUS), and the Ph.D. degree in engineering from the University of Auckland. He is an Associate Professor in the Department of Electrical Engineering at NUS, and is concurrently the Deputy Director of the Centre for Wireless Communications, a national R&D centre. His research interests are primarily in computer communication networks, with current focus in wireless/mobile multimedia networking. He has published more than 70 technical papers in both internationally refereed journals and conference proceedings. Dr. Chua has been an active member of the IEEE Communications Society. He served as the Chairman of the Singapore Chapter of the Society in 1993–1994. He also served in the Society's Member Affairs Council and acted as the Region 10 Chapters Chair Coordinator. He has also served as the Finance Chair of IEEE Globecom'95 and in the TPC of some IEEE INFOCOM and PIMRC conferences. He is currently chairing the organising committee for the 2000 IEEE Singapore International Conference on Communication Systems (ICCS2000).
E-mail: eleckc@nus.edu.sg