IEEE 802.11 Distributed Coordination Function Discrete-Event Simulator against State-of-the-Art Analytical Model

Luca Mosetti
University of Trento, DISI
luca.mosetti-1@studenti.unitn.it

Shandy Darma
University of Trento, DISI
shandy.darma@studenti.unitn.it

***Abstract*—In this report, we present our Discrete-Event Simulator (DES) of Distributed Coordination Function (DCF), detail its utilization, and compare its results to the established analytical model, specifically Tinnirello's analytical model [3].**

The outcomes of our simulator and the analytical model exhibit similar trends, but our studies reveal a discrepancy that needs to be further investigated.

***Index Terms*—IEEE 802.11 Distributed Coordination Function (DCF), Discrete-Event Simulator (DES), throughput, probability of collision**

I. Introduction

The de facto standard for wireless Internet access is currently the IEEE 802.11 Wireless Local Area Networking (WLAN) technology also known to the general public as Wi-Fi. Wi-Fi connectivity is integrated by default in every modern portable computer, laptop, and smartphone. One of the key factors underlying the broad acceptance of Wi-Fi is the simplicity and robustness of the Medium Access Control (MAC) protocol.

The 802.11 MAC, called Distributed Coordination Function (DCF), is based on carrier sense multiple access with collision avoidance (CSMA/CA). This means that each device manages its own connection to the channel. Additionally, when transmissions from several stations collide, binary exponential backoff rules will be enforced to minimize further collisions.

It is important to study how the protocol is going to behave with different parameters. Understanding the behavior enables optimizing the protocol to get better performances. The behavior can be understood by either building a simulator or implementing an analytical model.

In this report we validate the state-of-the-art DCF analytical model, specifically Tinnirello's analytical model [3], by comparing its predictions with our simulation results.

II. Related Work

Several works have been done on IEEE 802.11 DCF protocol. One of the well-known studies is by Bianchi in 2000 [1]. This work proposed an analytical model for the protocol to measure saturation throughput and collision probability. The proposed model is a function of number of nodes, minimum contention window, and maximum backoff stage. Additionally, the model also works on two different mechanisms for packet transmission, namely basic access (BAS) mechanism and "request-to-send / clear-to-send" (RTS/CTS) mechanism.

In 2005, Bianchi *et al.* refined the work [2]. This work introduced the notion of maximum number of retries. When a transmission has collided and exhausted the maximum allowable number of retries, the transmission will be dropped and replaced with a new one. Furthermore, this work revised the previous model to accurately model the backoff freezing operation stated by the protocol. This aspect of the work is further explained in later work.

In 2009, Tinnirello *et al.* [3] elaborated more on previous revisions. This work explained that on certain conditions, a slot does not behave normally. The slot after a successful transmission can only be utilized by the same station that successfully transmitted. The slot after a collision cannot be accessed by any stations. Given the identified anomalies, the study proceeded to introduce a new analytical model.

III. Distributed Coordination Function

The IEEE 802.11 DCF follows the "listen-before-talk" baseline principles of the Carrier Sense Multiple Access mechanisms with Collision Avoidance (CSMA/CA). To avoid collision, each station (STA) which has to transmit will check if the medium is idle or busy. Each station will wait until the medium is sensed idle for a period of time equal to a Distributed Interframe Space (DIFS) period. At this point, if the station does not have a backoff counter, it will draw a new one. The backoff counter will be decremented when the medium is detected idle for σ amount of time. If the medium is detected busy, the backoff counter will be frozen until the medium is idle for a DIFS amount of time, then the backoff counter is decremented. When the backoff counter reaches zero, the station will attempt to transmit.

This protocol uses backoff counter as a mean to avoid collision. After a successful transmission, the station will draw a backoff counter from a uniform distribution between 0 to CW. Initially CW equals to the minimal contention window W. When a station is involved in a collision, CW will be doubled. CW cannot go higher than a maximum contention window $2^m W$. Eventually, CW will be reset to W either when the transmission complete successfully or when the

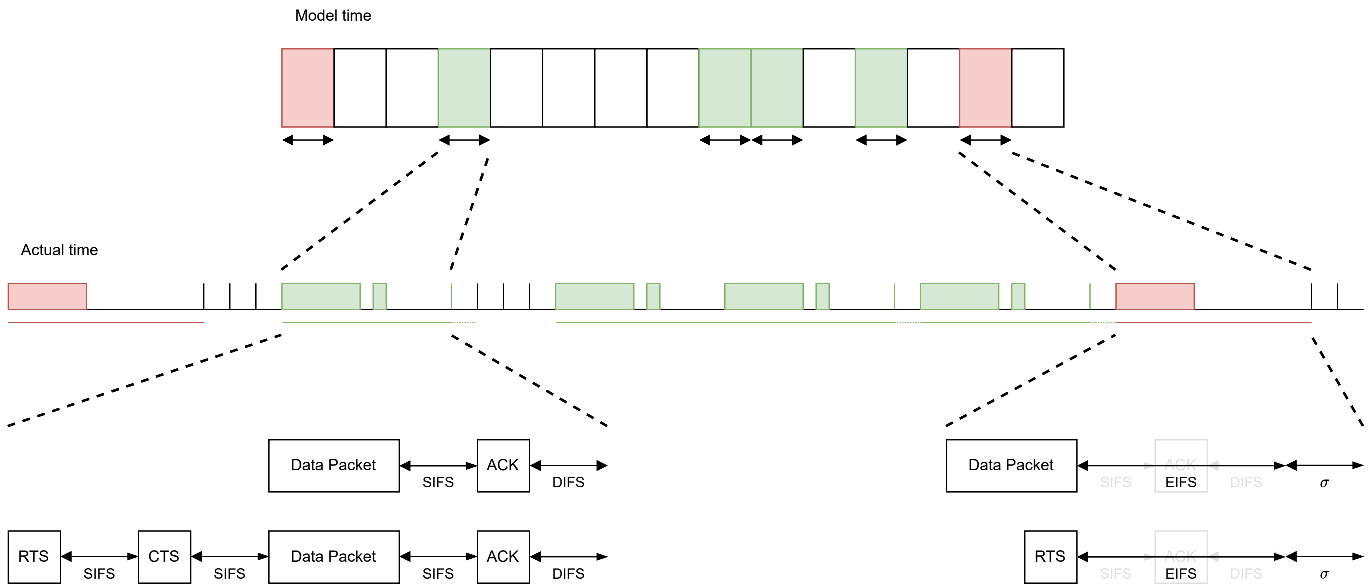

Fig. 1. DCF time scheme according to specifications from Tinnirello *et al*. work [3]. Both BAS and RTS/CTS modes are depicted.

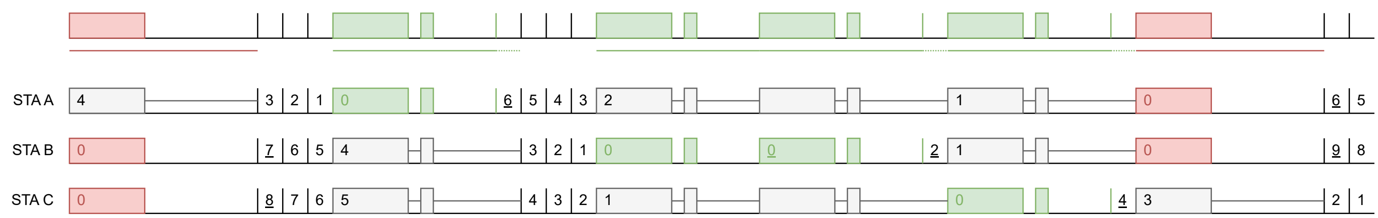

Fig. 2. DCF behaviour in BAS mode with three stations. The numbers represent the backoff value. When underlined, they represent a new extraction.

transmission has been retried more that the retry limit R. In the latter case, the transmission is discarded.

Furthermore, this protocol has two different access modes: basic access (BAS) and “request-to-send / clear-to-send” (RTS/CTS). On BAS, if a station has backoff counter equal to 0, the station will start transmitting its packet. After that, it waits for an ACK reply in a Short Interframe Space (SIFS) period. The ACK reply is necessary because the station is incapable of detecting collision by listening to its transmission. After the ACK reply is confirmed, the station waits for a DIFS duration, then it draws a new backoff counter up to W. However, if it does not receive any ACK reply, it also draws a new backoff counter but based on its current CW.

On RTS/CTS, instead of immediately transmitting its packet, the station first sends a short RTS packet. If the RTS transmission is successful, the station will wait for SIFS duration and receive a CTS packet. Consequently, the station sends its packet, similar to BAS. This access mode allows stations to reserve the medium first, preventing the possibility of collision in the subsequent transmission.

IV. Problem Definition and System Setup

In this section, we elaborate on how the simulation works, how we measure the metrics, the technique used to ensure robust and reliable results, and the analytical model used to evaluate our results.

A. Simulator

For this work we have implemented a simulator based on Tinnirello *et al*. description of the DCF protocol [3]. The simulator is implemented in Python. It is designed as a Discrete-Event Simulator (DES), where each discrete-event represents either an idle slot, a successful transmission, or a collision. Figure 1 shows how to represent the DCF protocol as discrete-events. Since every event is represented as a discrete-event, a single simulation is independent of any time-related specifications, such as SIFS, DIFS, and EIFS duration.

As for the analytical model, also for the simulator certain assumptions apply. In particular, the number of the stations is fixed, the transmission queue of each station is always nonempty, the packet size is constant, and the channel is in an ideal situation, that is without either hidden terminals or capture effect.

When the simulation starts, all stations are set with maximum retries and backoff counter equals zero. This is done to initiate the simulation with a discarded transmission, after which all stations begin by drawing a new backoff counter.

To progress each discrete event in the simulation, the waiting counter of every station is decremented by the lowest

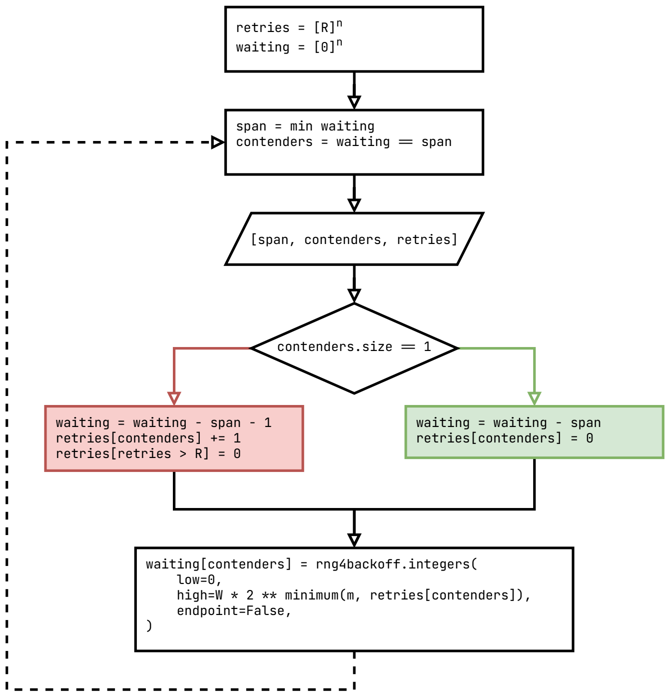

Fig. 3. DES of DCF according to Tinnirello *et al.* specifications [3]

count among them. Then, the simulation will check how many stations have backoff counter equal to zero. If there is only one station, then this station will successfully transmit. Otherwise, a collision occurs, resulting in an increment of the retry counter for each involved station. Afterwards, these stations will draw a new backoff counter based on the updated retry counter. Figure 3 illustrates the logic explained above.

In practice, the simulator produces CSV files where each row represents a contention round. These simulations are then "instantiated" with network specifications, such as payload size, channel bit rate, etc. Finally, metric analyses are computed and charts and graphs are depicted.

B. Metrics

1) Probability of collision: A collision occurs when several stations transmit their packet through a shared medium simultaneously. Collisions should be minimized because when a collision occurs, the medium is not transmitting any useful information, and thus lowers performance of the network. Collision probability measures the probability that a transmitted packet collides. Collision probability of a station is measured by calculating the ratio of its collided transmissions to its total number of transmissions. The collision probability of the network is measured by averaging the collision probability of each station. Example of this metric measurement is shown in fig. 4. In this figure, there are two plots. The lower plot shows the collision probability of each station (grey lines), and the resulting collision probability of the network (blue line). The upper plot describes the contention rounds. Each row represents the outcome of each station, where green means successful transmission and red means collision. It is evident that a collision will always regard at least two stations.

2) Throughput: Throughput measures the ratio of the useful bits successfully transmitted through the medium over a period of time. This metric takes into account the payload of any packet, regardless of the origin station. Example of this metric measurement is shown in fig. 5. In this figure, there are two plots. The lower plot demonstrates throughput as previously explained. The upper plot describes the channel usage, where green means successful transmission and red means collision. Ideally, there should be blanks where the stations are decreasing their backoff countdown and the channel is idle. However, the short duration of each idle slot makes it insignificant compared to the successes and collisions times.

C. Non-overlapping batch over means

Non-overlapping batch over means is an output analysis technique. The simulation generates a large number of results, which are subsequently split into multiple batches.

$$\underbrace{Y_1, \ldots, Y_s}_{B_1}, \underbrace{Y_{s+1}, \ldots, Y_{2s}}_{B_2}, \cdots, \underbrace{Y_{(b-1)s+1}, \ldots, Y_{bs}}_{B_b}$$

From each batch the batch mean is computed, say Z_i. These individual batch means are averaged to calculate a grand mean, say $\overline{Z}_b$, with a confidence interval (CI).

$$\overline{Z}_b = \frac{1}{b} \sum_{i}^{b} Z_i \qquad \left[\overline{Z}_b \pm t_{b-1, \frac{1+\gamma}{2}} \sqrt{\frac{\hat{V}}{b}} \right]_{\gamma}$$

Where $t_{d,c}$ is the c quantile of the student's t-distribution with d degrees of freedom, and $\hat{V}$ is defined as follows:

$$\hat{V} = \frac{1}{b-1} \sum_{i}^{b} (Z_i - \overline{Z}_b)^2$$

This technique is capable of handling initialization bias issues. Given that the simulation is run over a long period of time, it will reach a steady state, where the initialization bias disappears. When the result is split into batches, the batches with steady state result will have greater impact compared to the batches with initialization bias result. Example of this technique is shown in Figure 6. Mean of each batch is displayed as small grey lines, to describe the deviations of throughput against the grand throughput.

The results from this report are always complete with 95% CIs. Unless stated otherwise, probability of collision has been measured with $b = 100, s = 5{,}000$, while saturation throughput with $b = 500, s = 1{,}000$.

D. Analytical Model

As previously stated, our objective is to compare the result from our simulations with the analytical model from Tinnirello *et al.* [3]. In particular, we will evaluate collision probability p and saturation throughput S. Beforehand, we must first establish several key variables, as outlined below:

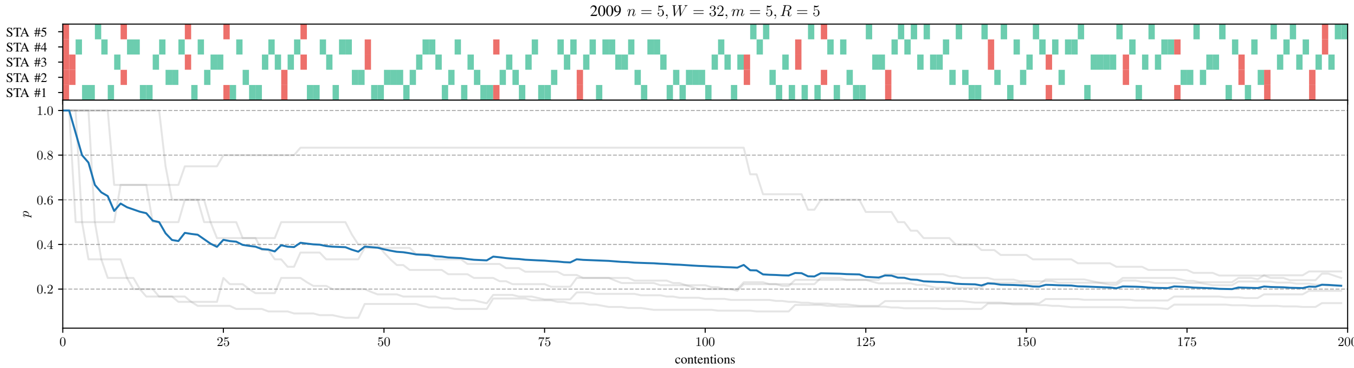

Fig. 4. Measurement of collision probability p. Upper figure shows whether a transmission or a collision occurred in each contention round. Lower figure shows the collision probability, with grey lines describe probability of each stations while blue line describes the final mean of all probabilities.

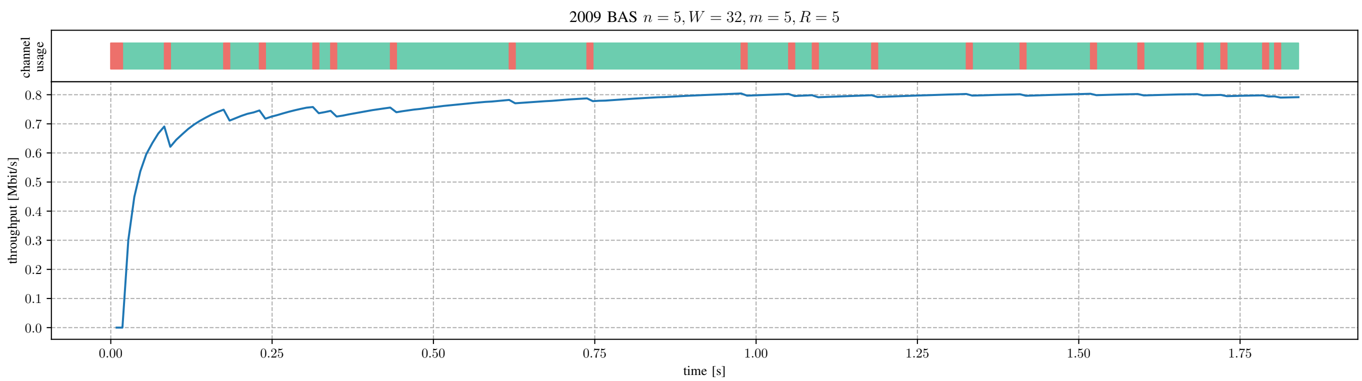

Fig. 5. Measurement of saturated throughput S. Upper figure shows the occurrences of transmissions and collisions. Lower figure shows the saturated throughput of the channel. Each transmission improves throughput while each collision worsens throughput.

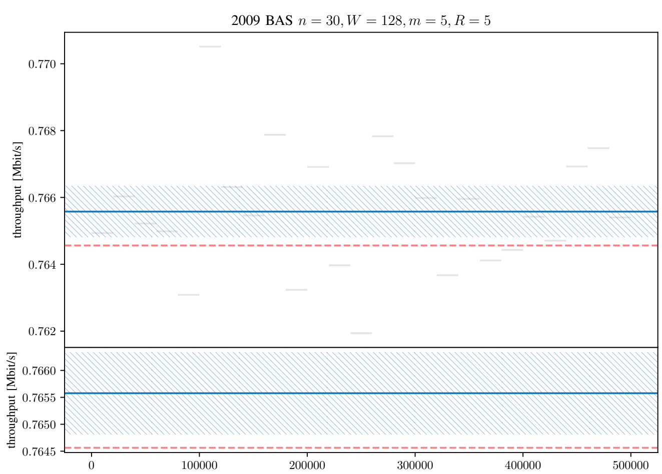

Fig. 6. Measurement of saturated throughput S with confidence interval CI. Upper figure shows saturated throughput with its CI, with the throughput of each batch displayed as small lines. Lower figure zooms in closer to observe the value in more detail.

σ	slot time
L	payload size
T_{MPDU}	the time to transmit the MPDU (including MAC header, PHY header, L, and/or tail)
T_{ACK}	the time to transmit an ACK
T_{RTS}	the time to transmit a RTS frame
T_{CTS}	the time to transmit a CTS frame
SIFS	the SIFS time
DIFS	the DIFS time
n	number of stations
W	minimum contention window
m	maximum backoff stage, where maximum contention window equals $2^m W$
R	maximum number of retries, after which the transmission is dropped
p	probability of collision
τ	probability that a station will transmit
S	saturation throughput

Probability of collision is defined as follows:

$$p = 1 - (1 - \tau)^{n-1}$$

τ, $\mathbf{S}$ are illustrated in fig. 14. Appendix A devotes some comments on how Tinnirello *et al*. justifies these formulae.

However, we noticed that τ does not involve m. Based on the calculation provided in the work, we derived an additional τ. The result from this revised analytical model will also be used to evaluate the simulations.

$$\tau = \frac{1}{1 + \frac{1-p}{2(1-p^{R+1})}\left[\sum_{j=0}^{R} p^j \cdot (\mathfrak{W}_j - 1) - (1 - p^{R+1})\right]}$$

$$\mathfrak{W}_j = W \cdot 2^{\min(m,j)} \quad j \in [0, R]$$

V. Results

Figure 7 plots the saturated throughput against the number of stations contending for the medium.

The simulation results are represented as points with CI, which are compared with two different analytical models: the original model proposed in Tinnirello *et al*. work [3] and the revised model.

The original model is represented in solid lines, while the revised model is represented in dashed lines. From the figure, it is apparent that when $m = R$, the results from both models are the same and the simulation result follows the same pattern. However, when $m \neq R$, the simulation results are closer to our revised model compared to the original model. Since our revised model includes m as a parameter while the original model does not, we can speculate that m actually impacts saturation throughput of DCF protocol.

Similarly, fig. 8 also measures saturated throughput. The distinction lies in the access mechanism: fig. 7 uses BAS while figure fig. 8 uses the RTS/CTS. Regardless, this figure displays similar patterns as the previous figure. It is evident that for both BAS and RTS/CTS, m still influences saturation throughput.

Furthermore, fig. 9 plots collision probability against the number of stations. Unlike saturation throughput, the access mechanism does not influence collision probability, thus this figure applies for both BAS and RTC/CTS mechanisms. Nevertheless, the figure shows that the simulation results are closer to our revised model compared to the original model. Thus, it is evident that m also influences the collision probability.

As demonstrated in fig. 13, the collision probability suffers from initialization bias. It is apparent that as the number of stations grows, the initialization bias effect persists for an extended duration. Fortunately, this effect is regulated by using the non-overlapping batch over means technique explained in section IV-C.

To further support our hypothesis, we conducted another measurement where the only changing variable is m. For these experiments $b = 500, s = 10{,}000$. Figure 10, fig. 11, and fig. 12 show saturation throughput for BAS, saturation throughput for RTS/CTS, and collision probability, respectively. While the original model remains constant regardless of m (solid line), both the revised model (dashed lines) and the simulation result (points with CI) show different results. As m decreases, saturation throughput decreases while collision

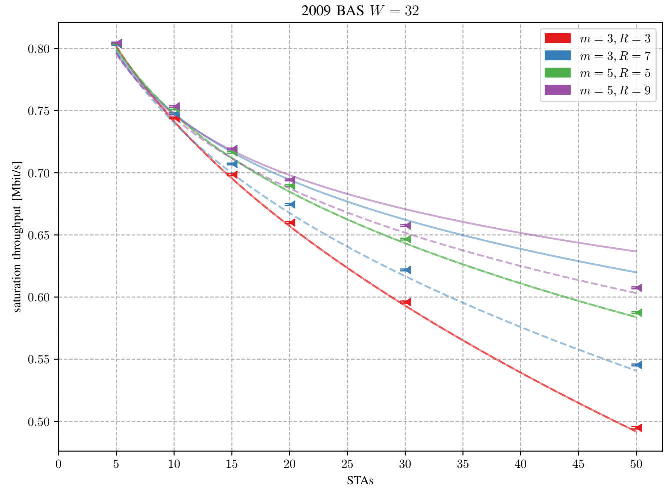

Fig. 7. Saturated throughput against number of contending stations.

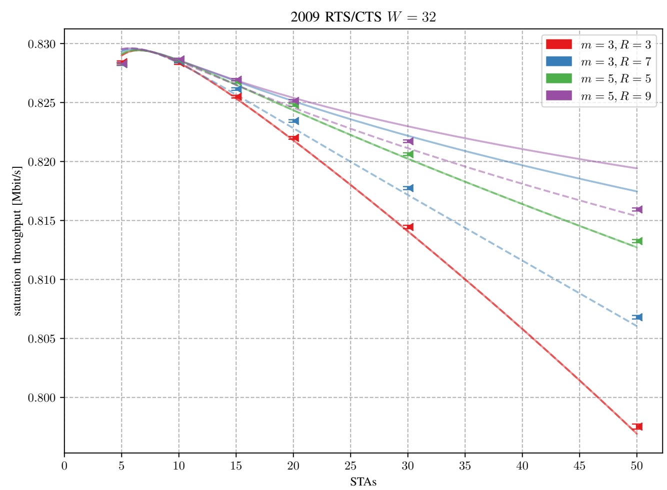

Fig. 8. Saturated throughput against number of contending stations.

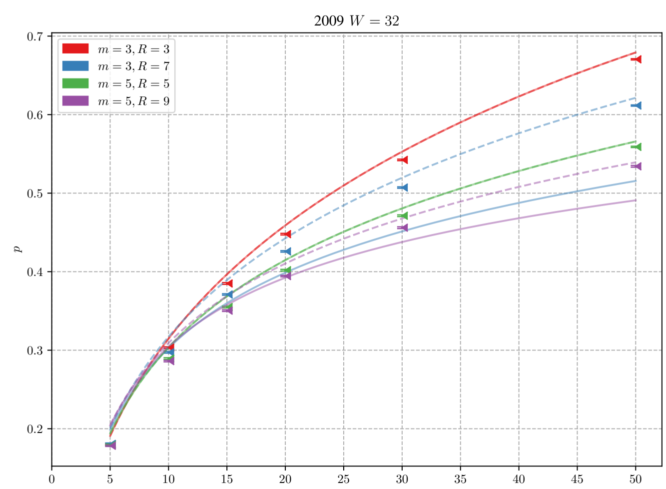

Fig. 9. Collision probability against number of contending stations.

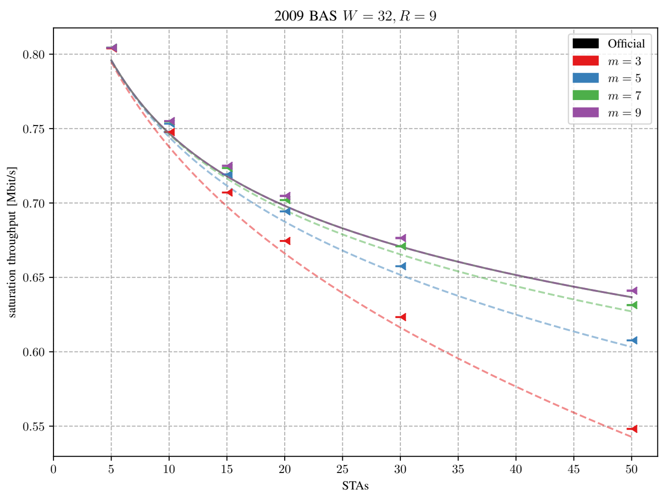

Fig. 10. Saturated throughput against number of contending stations.

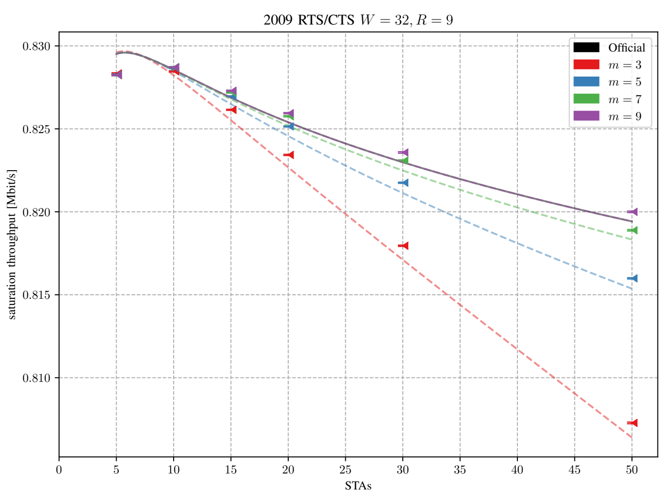

Fig. 11. Saturated throughput against number of contending stations.

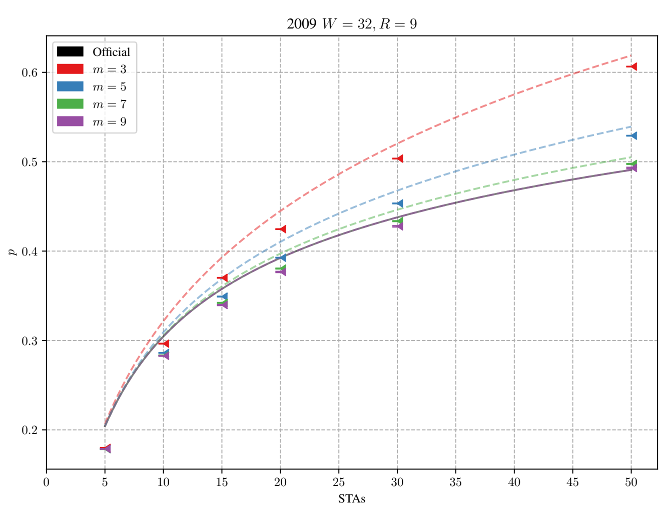

Fig. 12. Collision probability against number of contending stations.

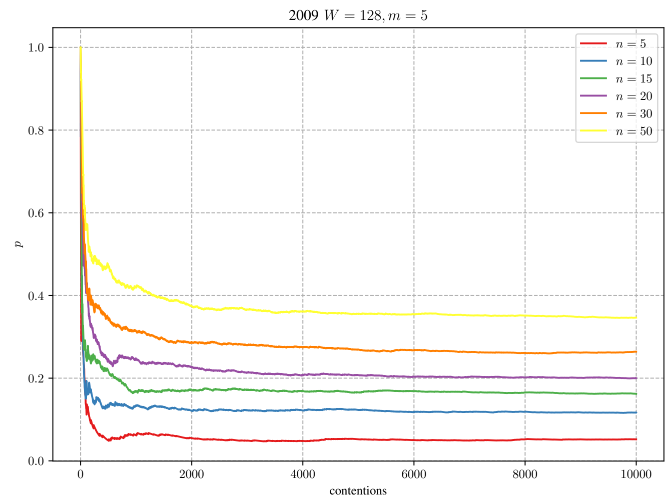

Fig. 13. Probability of collision: trends.

probability increases. This change is not reflected in the original model.

In summary, although the simulation results follows the trend of the analytical model, they do not match perfectly. The difference between the simulation results and the model is not negligible. To explain why this is the case, we suggest two alternative hypotheses:

1) The first possibility is that our simulator might contain inaccuracies. During the development process, we conducted a thorough examination but were unable to pinpoint any factors that could account for its inaccuracy. It is worth noting that this simulator builds upon another one, as detailed in appendix A. With the original simulator, the results aligned seamlessly with the analytical model presented by Bianchi in 2000 [1]. Consequently, if indeed the issue resides within our current simulator, the previous version can serve as a reliable baseline for initiating the inspection.
2) The analytical model is too approximate. As show in fig. 14, the analytical model does not include m as a parameter. However, were our simulator correct, the results would show that the maximum backoff stage m does influence the DCF protocol performances. Therefore, it would be advisable to inspect the analytical model to ensure that it is robust.

VI. Conclusion

We have implemented a DES simulator for the IEEE 802.11 DCF protocol according to the Tinnirello *et al.* specifications [3], along with the analytical model that the work proposed.

Upon comparing the results derived from the simulator to those of the analytical model, we observed a consistent trend between them, albeit without perfect alignment. The disparity between the two is non-negligible. This discrepancy can be attributed to one of two possibilities.

TABLE I
SIMULATOR PARAMETERS

Parameter	2000 **Simulator**	2009 **Simulator**
L	1023 Byte	1023 Byte
MAC header	34 Byte	34 Byte
PHY header	16 Byte	16 Byte
ACK	14 Byte + PHY header	14 Byte + PHY header
RTS	20 Byte + PHY header	20 Byte + PHY header
CTS	14 Byte + PHY header	14 Byte + PHY header
Channel Bit Rate	1 Mbit/s	1 Mbit/s
σ	50 μs	50 μs
SIFS	28 μs	28 μs
DIFS	SIFS $+ 2 \cdot \sigma$	SIFS $+ 2 \cdot \sigma$
δ	1 μs	–
$\mathsf{ACK}_{\mathsf{Timeout}}$	300 μs	–

In the first scenario, our simulations may be inaccurate. To address this, we recommend a comprehensive examination of the differences with the work presented for Bianchi's 2000 work [1] - better explained in appendix A - which perfectly aligns with Bianchi's model.

In the second situation, the simulator is correct while the analytical model is too approximated. Specifically, the analytical model makes imprecise estimations of the effects of certain protocol parameters. A possible parameter that reflects this description could be the maximum stage number m. Although the analytical formula does not consider m, our research shows that m significantly affects the protocol performance. When m is explicitly incorporated into the formula, the modified analytical model aligns better with our simulations. Hence, in this scenario, the initial focus should be on examining the impact of this parameter.

References

[1] Bianchi, Giuseppe. "Performance analysis of the IEEE 802.11 distributed coordination function." IEEE Journal on selected areas in communications 18.3 (2000): 535-547.

[2] Bianchi, Giuseppe, and Ilenia Tinnirello. "Remarks on IEEE 802.11 DCF performance analysis." IEEE communications letters 9.8 (2005): 765-767.

[3] Tinnirello, Ilenia, Giuseppe Bianchi, and Yang Xiao. "Refinements on IEEE 802.11 distributed coordination function modeling approaches." IEEE Transactions on Vehicular Technology 59.3 (2009): 1055-1067.

[4] Li, Zhifei and Das, Amitabha and Gupta, Anil and Nandi, Sukumar. "Performance Analysis of IEEE 802.11 DCF: Throughput, Delay, and Fairness."

Appendix
DCF according to Bianchi's 2000 work

Tinnirello *et al*. work [3] is a refinement of the Bianchi's work [1]. In this appendix we discuss the differences between the analytical models they proposed and the differences between the simulators we developed.

A. Differences between the analytical models

Figure 14 shows side by side the formulas from Bianchi's work [1], and Tinnirello *et al*. work [3]. The common parameters are written in *this form*, the Bianchi's results in this form, while the Tinnirello's results in **this form**.

In the original Bianchi's 2000 work

$$\mathsf{T_s^{BAS}}, \mathsf{T_c^{BAS}}, \mathsf{T_s^{RTS/CTS}}, \mathsf{T_c^{RTS/CTS}}$$

all include the propagation delay δ. However - as first pointed by Li in [4] - this parameter is already taken into account in SIFS. Indeed, in the most recent version

$$\mathbf{T_s^{BAS}}, \mathbf{T_c^{BAS}}, \mathbf{T_s^{RTS/CTS}}, \mathbf{T_c^{RTS/CTS}}$$

do not include such a parameter anymore. Other than that, while the success times are identical, the collision times have changed. This is because Bianchi's model consider the time during which the channel is sensed busy by the *noncolliding* stations:

> $\mathsf{T_c}$ is the period of time during which the channel is sensed busy by the *noncolliding* stations. We neglect the fact that the two or more colliding stations, before sensing the channel again, need to wait an $\mathsf{ACK}_{\mathsf{Timeout}}$, and thus the $\mathsf{T_c}$ for these *colliding* stations is greater than that considered here (the same approximation holds in the following RTS/CTS case, with a $\mathsf{CTS}_{\mathsf{Timeout}}$ instead of the $\mathsf{ACK}_{\mathsf{Timeout}}$).

On the other hand, under Tinnirello assumptions, both *colliding* and *noncolliding* stations have to wait the same amount:

> We assume that all stations that listen to a collision detect a PHY-RXEND indication, which returns an error and thus resumes the backoff process after an EIFS time from the previous channel activity. In this assumption, neither the transmitting stations nor other stations can use the first slot after an EIFS time from the previous transmission. In fact, by considering a basic rate of 1 Mbit/s, the EIFS duration is equal to $\mathbf{ACK_{Timeout}}$ + DIFS $- \sigma$. Thus, after an EIFS from a previous collision, an station involved in the collision and waiting for the ACK timeout has to wait for a further backoff slot before extracting a new backoff value. Moreover, stations sensing the collision resume the backoff counter to the frozen value, which is different from zero, because they were not transmitting. Hence, they cannot use the first slot after an EIFS time either.

Both analytical models are derived from a Markov model that describes the DCF model. In both cases, the Markov model states represent the current stage and backoff counter of a station, while the transitions represent their change. A transition in the Markov model occurs during each slot time. If the backoff counter is greater than 0, the only possible transition is the decrement of the backoff counter itself. In contrast, backoff counter equals 0 will either trigger a transmission or a collision, where the stage will be reset to 0 in the former case, or the stage will be incremented or unchanged in the latter case. In both cases, the station will draw a new backoff counter.

Despite that, there are differences in the details on how the Markov model is described. There are two factors that influence the difference between the models: when the backoff counter is decremented and the retry limit.

$$\tau = \frac{2}{1 + W + pW \sum_{i=0}^{m-1} (2p)^i} \qquad \tau = \frac{1}{1 + \frac{1-p}{2(1-p^{R+1})}\left[\sum_{j=0}^{R} p^j \cdot (2^j W - 1) - (1 - p^{R+1})\right]}$$

$$S = \frac{P_s P_{tr} L}{(1 - P_{tr})\sigma + P_{tr} P_s T_s + P_{tr}(1 - P_s) T_c} \qquad S = \frac{P_s \overline{E[L]}}{(1 - P_b)\sigma + P_s \overline{T_s} + (P_b - P_s)\overline{T_c}}$$

$$P_{tr} = 1 - (1 - \tau)^n \qquad P_s = \frac{n\tau(1 - \tau)^{n-1}}{P_{tr}} \qquad P_b = 1 - (1 - \tau)^n \qquad P_s = n\tau(1 - \tau)^{n-1}$$

$$\overline{E[L]} = L\frac{W}{W-1} \qquad \overline{T_s} = T_s\frac{W}{W-1} + \sigma \qquad \overline{T_c} = T_c + \sigma$$

$$\begin{aligned} T_s^{BAS} &= T_{\text{MPDU}} + \delta + \text{SIFS} + T_{\text{ACK}} + \delta + \text{DIFS} \\ T_c^{BAS} &= T_{\text{MPDU}} + \delta + \text{DIFS} \end{aligned} \qquad \begin{aligned} T_s^{BAS} &= T_{\text{MPDU}} + \text{SIFS} + T_{\text{ACK}} + \text{DIFS} \\ T_c^{BAS} &= T_s^{BAS} \end{aligned}$$

$$\begin{aligned} T_s^{RTS/CTS} &= T_{\text{RTS}} + \delta + \text{SIFS} + T_{\text{CTS}} + \delta + \text{SIFS} \\ &+ T_{\text{MPDU}} + \delta + \text{SIFS} + T_{\text{ACK}} + \delta + \text{DIFS} \\ T_c^{RTS/CTS} &= T_{\text{RTS}} + \delta + \text{DIFS} \end{aligned} \qquad \begin{aligned} T_s^{RTS/CTS} &= T_{\text{RTS}} + \text{SIFS} + T_{\text{CTS}} + \text{SIFS} \\ &+ T_{\text{MPDU}} + \text{SIFS} + T_{\text{ACK}} + \text{DIFS} \\ T_c^{RTS/CTS} &= T_{\text{RTS}} + \text{SIFS} + T_{\text{ACK}} + \text{DIFS} \end{aligned}$$

Fig. 14. Definitions from Bianchi 2000 (on the left) and Tinnirello *et al.* 2009 (on the right) works [1, 3]

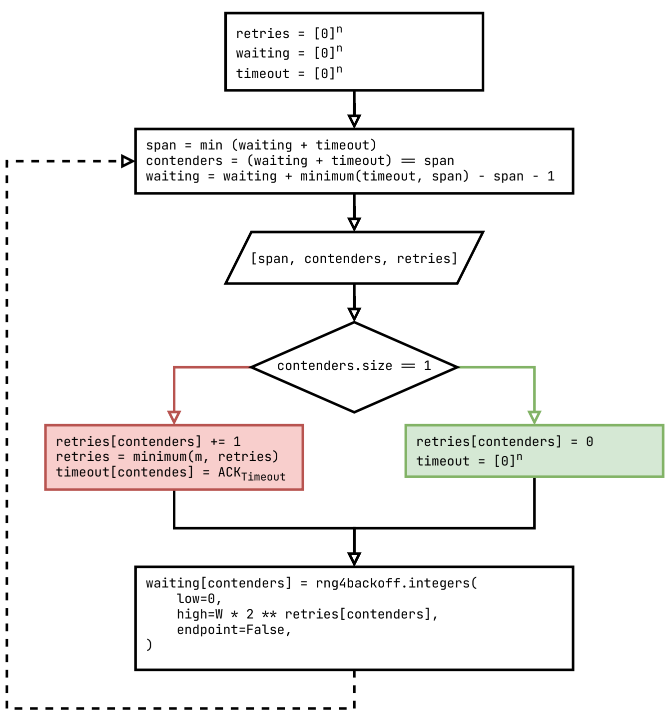

Fig. 15. DES of DCF according to Bianchi 2000 specifications [1]

In [1], the backoff counter is decremented at the beginning of each slot. However, [3] specified that according to the IEEE 802.11 DCF standard, the backoff counter is supposed to be decremented at the end of each slot. Moreover, it also pointed out that the backoff counter is suspended when the medium is observed to be busy. [3] explained further that this means that after a successful transmission an anomalous slot occurs. The slot can only be accessed by the station that successfully transmitted in the first place. Consequently, if this station draws 0 as its backoff counter, the new transmission will be successful. This is because during a successful transmission, the backoff counter of the non-sending stations must be at least 1, and their counter will only be decremented at the end of the anomalous slot.

This phenomenon is reflected in the change in Markov model transition. In [1], when a successful transmission occurs, the transition will change to the state where the stage is 0 and the backoff counter is drawn uniformly between $[0, W)$. In contrast, for [3], when a successful transmission occurs and the backoff counter draws 0, the Markov chain does not execute a transition due to the anomalous slot phenomenon explained earlier. Instead, the backoff counter will be drawn uniformly only from $[1, W)$ to consider a transition.

Moreover, this phenomenon is also reflected in how the throughput is measured. One of the main components for measuring throughput is the expected length of a successful slot time T_s. Assume that the payload size is constant. T_s for [1] is rather straightforward, by summing up all the necessary parameters. In contrast, [3] introduces $\overline{T_s}$ so to account for consecutive successful transmissions caused by the anomalous slots, in which there might be multiple of them, albeit its unlikeliness.

[3] introduces the notion of retry limit. When the packet has been retried and reached the retry limit, the packet will be dropped, the stage is reset to 0, and a new backoff counter will be drawn uniformly from $[0, W)$. Markov model in [1] does

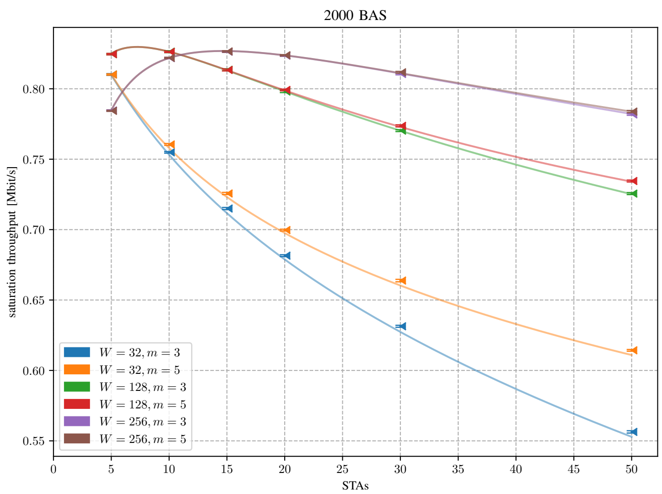

Fig. 16. Saturation throughput: analysis versus simulations.

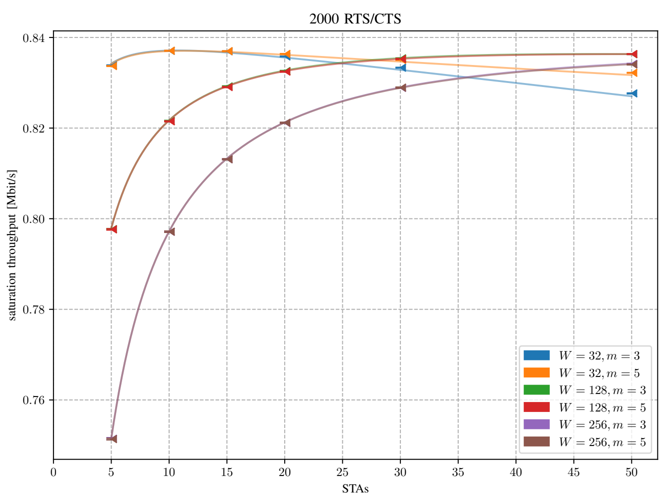

Fig. 17. Saturation throughput: analysis versus simulations.

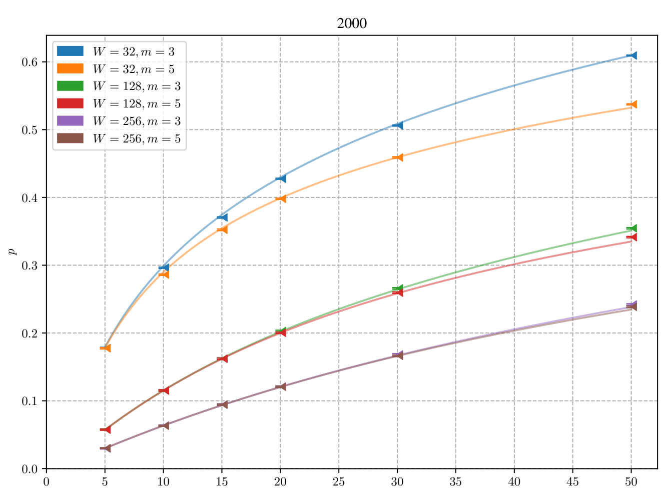

Fig. 18. Probability of collision: analysis versus simulations.

not take retry limit into account, while Markov model in [3] does. This is shown in how the Markov model is described in [3]. The model possesses two different sets of states for when the stage equals 0. The first set of states is represented as 0^+. This set of states is reached after a successful transmission. In this set of states, the backoff counter belongs to the range $[0, W-1)$. In contrast, the last set of states is represented as 0^-, which is achieved when the retry limit is hit. Unlike the previous, in this set of states, the backoff counter belongs to the range $[0, W)$.

B. Differences between the simulators

We have implemented two distinct DCF protocol simulators. They are based on [1] and [3], respectively. Since [3] is made as an improvement of [1], it has several refinements that were not available on the previous work, and consequently also reflected in the simulators that we made.

1) Retry limit: When a station fails to send a packet for R times, the packet will be dropped. After the packet is dropped, the retry counter is reset to zero, and thus it changes the range where the backoff counter will be drawn. This functionality is not present in the 2000 simulator, because it was not brought up by the former work. In contrast, the 2009 simulator has this functionality.

2) Backoff counter decrement: As DESs, the counter will decrease at every "tick". When the counter of one or more station(s) reaches zero, either a transmission or a collision will occur. In both simulators, to speed up the process, instead of decreasing the counter one-by-one, the simulators will decrease the counters of all stations by their lowest counter. However, [1] and [3] have different ideas for the counter decrement. The former executes decrement at the beginning of the slot. To reflect that, the 2000 simulator does an additional decrement. In contrast, the latter executes decrement at the end of the slot, so the 2009 simulator performs an additional decrement only in case of collisions.

3) ACK Timeout: The DCF protocol expects the station to wait for an ACK after a transmission. According to Bianchi's specification [1], when two or more transmissions collide, the stations involved must wait for an ACK (which will not arrive) for an $\text{ACK}_{\text{Timeout}}$ period. If another transmission takes place in the meantime, these stations will realise that the packet has been lost and consequently will reschedule the packet transmission according to the given backoff rules. This logic is implemented in the 2000 simulator, but not in the 2009 simulator. The reason is that, according to Tinnirello *et al.* specifications [3], when a collision occurs, all stations (*colliding* and *noncolliding*) have to wait for the same period before resuming the backoff process, that is $\text{EIFS} + \sigma$.

C. Results

Figure 16 and fig. 17 compare throughput from simulations with throughput from Bianchi's 2000 analytical model [1]. The first figure uses DCF BAS mode, while the second figure uses RTS/CTS mode. It is apparent that the result from simulations

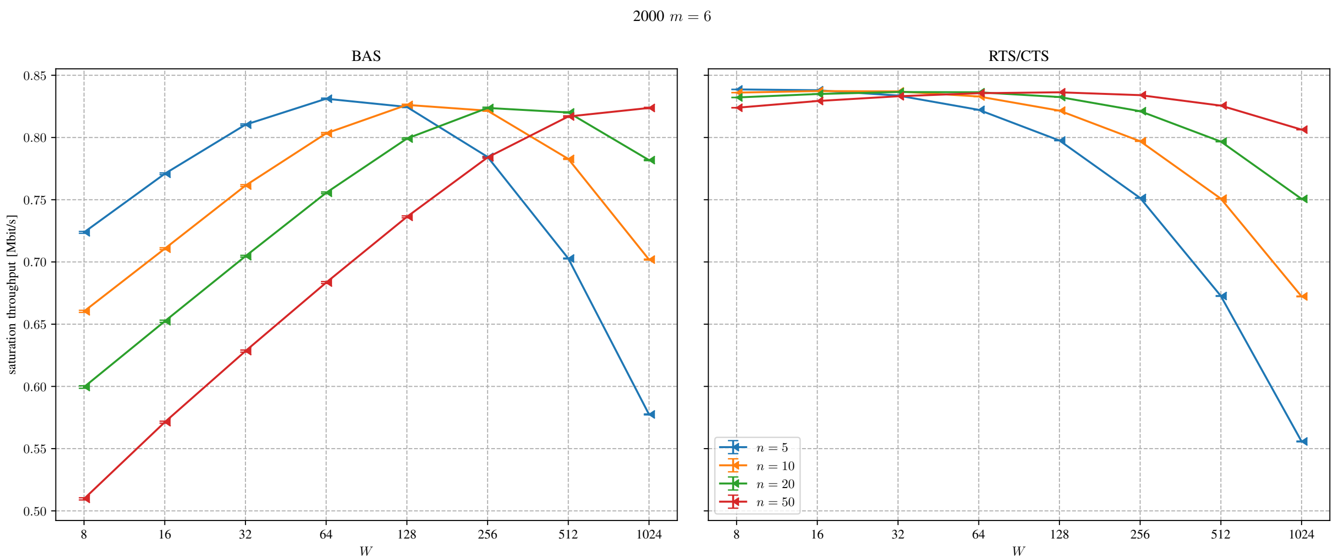

Fig. 19. Saturation throughput versus initial contention window size.

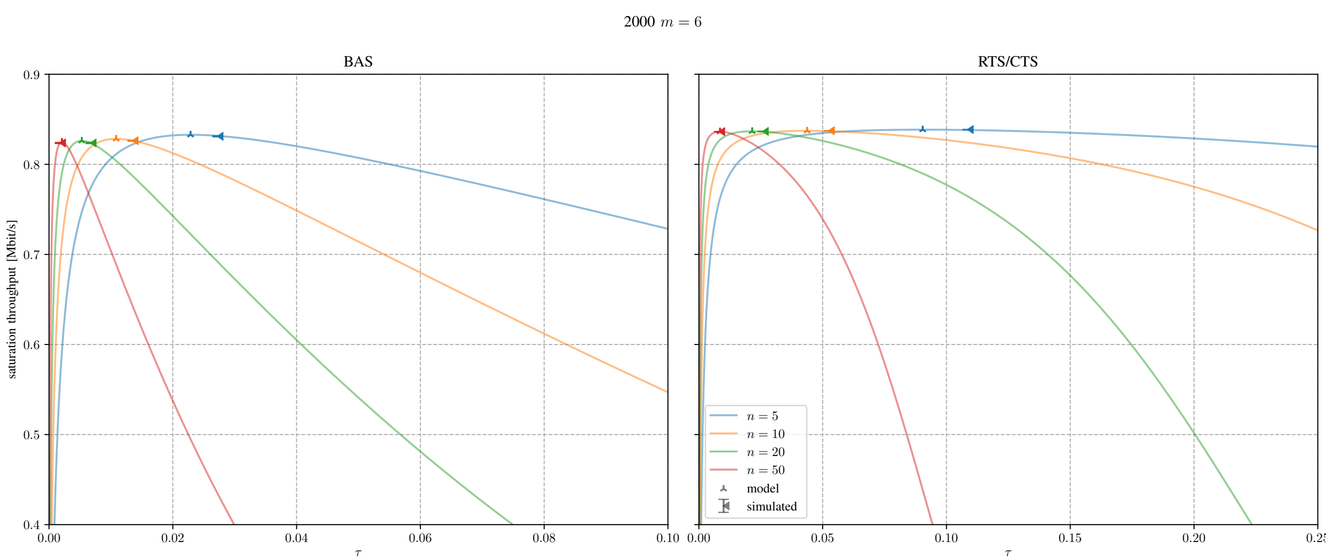

Fig. 20. Saturation throughput versus the transmission probability τ.

(points with CIs) and analytical model (lines) has negligible difference, if any.

Figure 18 compares collision probability from simulations (points with CIs) with collision probability from analytical model (lines). Similar to the comparison of throughput, the result from simulations and analytical model has negligible difference, if any.

Lastly, to ensure that our simulator accurately models the DCF protocol as described in [1], we repeated an experiment from the work mentioned. The experiment aims to verify if the analytical model correctly predicts the maximum saturation throughput achievable with some fixed parameters, namely the number of stations n and the maximum backoff stage m. The simulation results in fig. 19 show that network with different n achieve peak throughput with different initial backoff window W. These results (points with CIs) are then compared with analytical results in fig. 20. In these figures, lines depict the analytical value of S according to different transmission probability τ. Indeed, higher τ leads to drop in throughput, i.e. higher amount of transmissions leads to higher amount of collisions. The peak of these curves indicate the theoretically optimal configuration (points without CIs). As in [1], the analytical model predictions are extremely similar to simulation results, with little to no difference.

Hence, also this experiment confirms the validity of our simulator.