Wi-Fi: Understand the Bianchi model and compare simulation results against analysis.

Motivation

Wi-Fi, standardized by IEEE 802.11, is integral to modern connectivity; it is the primary means for internet access. Practically every laptop, tablet and mobile phone is Wi-Fi enabled. The simplicity and the robustness of the 802.11 MAC protocol enables numerous devices to coexist and communicate in the unlicensed band. Yet, it isn't without flaws: wireless medium use is inefficient, access is unequal, and performance lacks guarantees. With the continuous rise in Wi-Fi users and the shift towards high bandwidth applications, the need for a theoretical model is clear. Such a model is crucial for deepening our understanding of medium usage, achieving equitable access, and ensuring dependable performance in the face of ever-increasing network usage.

Objective

Understand the “Bianchi” formula obtained via a stochastic analysis of IEEE 802.11 protocol and evaluate its predictions against the results from NetSim simulations.

Introduction

In this experiment, we are concerned with the saturation throughput analysis of single cell IEEE 802.11 DCF wireless local area networks. We consider a single cell WLAN; single cell meaning that all nodes are within control channel range of each other, and every packet transmission can be heard by every other node. There can be only one successful transmission in the channel at any time, and the network does not support spatial reuse. The IEEE 802.11 standard defines a CSMA/CA based distributed medium access control (MAC) protocol, called the distributed coordination function (DCF). DCF permits nodes to have a single queue each and all have identical backoff parameters (which governs the channel access). We are interested in the expected throughput performance of the DCF WLAN when the nodes always have packet to transmit (i.e., saturation assumption).

The Distributed Coordination Function (DCF)

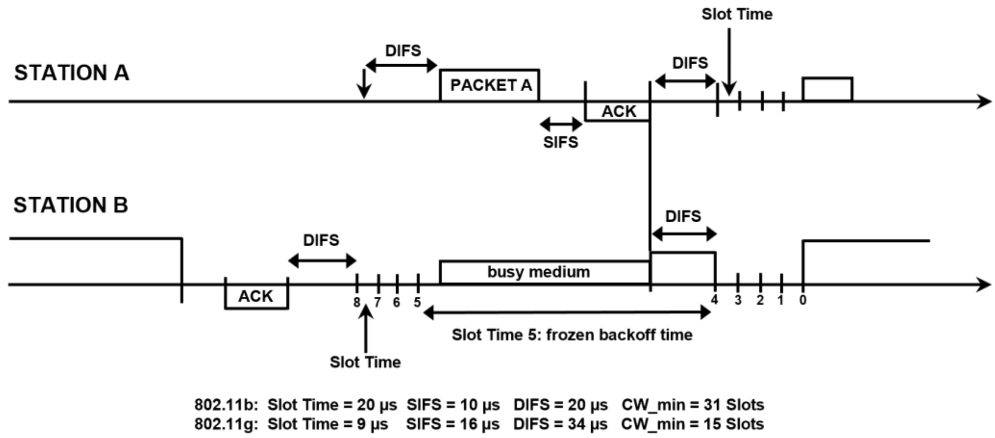

Figure 1: Illustration of the 802.11 DCF Basic Access Mechanism. This figure demonstrates the sequence of data packet transmission, backoff procedures, and acknowledgment (ACK) transmission, along with the corresponding timing specifics for the 802.11b and 802.11g standards.

Figure 1 provides a detailed illustration of the 802.11 Basic Access Mechanism, highlighting the sequence of events in data packet transmission, backoff procedures, and acknowledgment (ACK) transmission.

The IEEE 802.11 DCF follows the “listen-before-talk” baseline principles of the Carrier Sense Multiple Access mechanisms with Collision Avoidance (CSMA/CA). A station (STA) with a frame to transmit shall invoke the carrier-sense mechanism to determine whether the wireless medium is busy or idle.

In the case of a busy medium, the STA shall defer transmission until the medium is idle without interruption for a period of time equal to a distributed interframe space (DIFS). After this period, the STA shall generate a random backoff period for an additional deferral time before transmitting. The backoff period is slotted for efficiency reasons and is expressed in terms of an integer number of elementary backoff slots. Such a number, which is called the backoff counter, is decremented as long as the medium is sensed idle, "frozen" when a transmission is detected on the channel, and reactivated when the medium is sensed idle again for more than a DIFS. The STA transmits when the backoff time reaches zero.

At each transmission, the backoff time is uniformly chosen in the range $[0, CW]$, where CW is the current backoff window size. At the very first transmission attempt, CW is equal to the initial backoff window size CW_{min} (e.g., 31 in 802.11b). After each unsuccessful transmission, $CW := 2(CW + 1) - 1$, until a maximum backoff window size value CW_{max} is reached. Once it reaches CW_{max}, CW shall remain at the value CW_{max} until it is reset. CW shall be reset to CW_{min} after every successful attempt to transmit, or the retransmission counter reaches a predefined retry limit, which is referred to as R hereinafter. When the retry limit is reached, the frame is dropped.

Since the CSMA/CA does not rely on the capability of STAs to detect a collision by hearing their own transmission, a positive acknowledgment (ACK) is transmitted by the destination STA to signal the successful packet reception. The ACK is immediately transmitted at the end of the packet after a period of time called short interframe space (SIFS). As the SIFS is shorter than the DIFS, no other STA is able to detect the channel to be idle for a DIFS until the end of the ACK. If the transmitting STA does not receive the ACK within a specified ACK timeout or it detects the transmission of a different frame on the channel, it reschedules the frame transmission according to the previous backoff rules.

The previously described two-way handshaking technique for the packet transmission is called the basic access mechanism. [3]

The Bianchi Model

The Bianchi model abstracts these detailed sequences into a probabilistic framework. The key approximation in the Bianchi model is that, at each transmission attempt, and regardless of the number of retransmissions suffered, each packet collides with constant and independent probability p. This probability p, will be referred to as *conditional collision probability*, meaning that this is the probability of a collision seen by a packet being transmitted on the channel.

In the analysis, we assume a fixed number of stations, each always having a packet available for transmission. In other words, we operate in *saturation* conditions, i.e., the transmission queue of each station is assumed to be always nonempty.

Let us denote $W = \mathbb{E}[CW_{min}]$, and m "maximum backoff stage," be the value such that $CW_{max} = 2^m \cdot W$, and adopt the notation $W_i = 2^i \cdot W$ where i is called "backoff stage." From [1], the probability, τ that a station transmits in a randomly chosen slot time, is

$$\tau = \frac{2(1-2p)}{(1-2p)(W+1) + pW(1-(2p)^m)} \tag{1}$$

The derivation of the above formula involves using Markov chains to model the CSMA/CA protocol and is complex. We exclude it from this document and readers can consult [1]. From section 3 of [2], we can rewrite (1) for 802.11b and 802.11g as

$$\tau = \frac{\overbrace{1 + p + p^2 + p^3 + p^4 + p^5 + p^6 + p^7}^{7\ retransmissions\ limit}}{1 + \left(\frac{CW_{min}+1}{2}\right) + p \cdot \left(\frac{CW_{min}+1}{2}\right) \cdot \left(1 + 2p + (2p)^2 + (2p)^3 + (2p)^4 + \underbrace{(2p)^5 + (2p)^5 + (2p)^5}_{CW_{max}\ (1024)\ limit}\right)} \tag{2}$$

since 802.11 states that $CW_{max} = 1024$, and supports a maximum of 7 retransmissions.

In the above equation, τ depends on the conditional collision probability p, which is still unknown. To find the value of p, it is sufficient to note that the probability that a transmitted packet encounters a collision, is the probability that, in a time slot, at least one of the remaining $n-1$ stations transmit. At steady state, each remaining station transmits a packet with probability p. Therefore

$$p = 1 - (1-\tau)^{n-1} \tag{3}$$

Numerical method to find τ

Equations (2) and (3) define a nonlinear system with two unknown variables, for which algebraic methods fail due to the complexity of the equations. Numerical techniques are required, and the fixed-point iteration method is one viable option. To circumvent potential problems of non-convergence we apply the *relaxed fixed-point iteration* method and the algorithm for this technique as follows:

1. Initialize $i = 0$ and choose and initial guess for p_0 . Since p is a probability, $0 \le p_0 \le 1$. Let us choose $p_0 = 0.5$
2. Set the convergence threshold δ and the relaxation parameter α. The condition on alpha is given in Section IVA of [2]. We choose $\alpha = 0.95$.
3. Repeat the following steps till $|p_i - p_{i+1}| \le \delta$ (i.e., repeat till convergence)
    a. Substitute p_i into equation (2) to obtain τ_i.
    b. Substitute τ_i into equation (3) to obtain p_{i+1} .
    c. Update p_{i+1} using the relaxation: $p_{i+1} \leftarrow (1-\alpha)\cdot p_{(i+1)} + \alpha \cdot p_i$.
    d. Substitute $p_{(1+1)}$ into equation (2) to obtain $\tau_{(1+1)}$.
    e. Increment i by 1
4. End Repeat when convergence criterion is met

The collision probabilities obtained from relaxed fixed-point method are tabulated in the results section and compared against NetSim.

Network Setup in NetSim

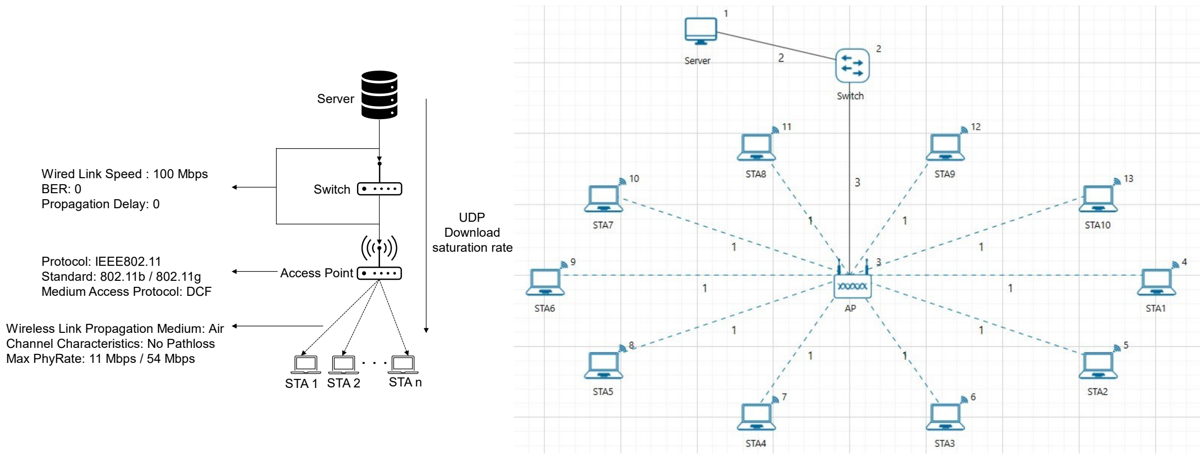

Figure 2: Network scenario with 1 AP connected to STAs. The number of STAs are varied and collision probability is analysed.

Case 1: 802.11b

Procedure

The following set of procedures were done to generate this sample.

Step 1: Click on Internetworks on the home screen. Once you enter set the grid length to 70m × 70m.

Step 2: A network scenario is created in NetSim GUI comprising of 1 Wired Node, 1 L2 Switch, 1 Access Point and 1 Wireless Node. The wired node is named Server, the wireless node is named STA, and the access point is named AP.

Step 3: In AP and STA, the Interface 1 (Wireless) > Physical Layer, Protocol Standard is set to IEEE802.11b. Then in Interface 1 (Wireless) > Datalink Layer check that the Medium Access Protocol is set to DCF.

Step 4: Right-click the link ID (of a wired/wireless link) and select Properties to access the link's properties. The parameters are set according to the values given in the below in Table 1 and Table 2.

Wireless Link Properties	
Channel Characteristics	No Pathloss

Table 1: Wireless Link Properties

Wired Link Properties	
Max Uplink Speed (Mbps)	100
Max Downlink Speed (Mbps)	100
Uplink BER	0
Downlink BER	0
Uplink Propagation Delay (µs)	0
Downlink Propagation Delay (µs)	0

Table 2: Wired Link properties

Step 5: Configure applications between any nodes by selecting an application from the Set Traffic tab. Right click on App1 CBR and select the following properties. Source: Server with Node ID 1, Destination: STA with Node ID 4, Packet Size: 1460 Bytes and Inter Arrival Time: 973.3 µs. These packet size and inter packet arrival time settings results in traffic generation rate that equals 12 Mbps. The Transport Protocol is set to UDP.

Step 6: Run simulation for 10 sec.

Step 7: Note down the following output metrics: (i) Application Throughput from Application Metrics the Results Window, and (ii) Packet Transmitted and Packet Collided from Link Metrics in the Result window.

Step 8: Similarly increase the number of STAs 1, 2, 3, 4, 5, 10, 15, 20 and note down the Throughput (Mbps) and packets transmitted, and packets collided from Result window as explained in Step 7.

Case 2: 802.11g

Procedure

Step 1: Consider the same scenario as case 1.

Step 2: In AP and STA, the Interface 1 (Wireless) > Physical Layer, Protocol Standard is set to IEEE802.11g.

Step 3: Configure applications between any nodes by selecting an application from the Set Traffic tab. Right click on App1 CBR and select the following properties. Source: Server with Node ID 1, Destination: STA with Node ID 4, Packet Size: 1460 Bytes and Inter Arrival Time: 212.36 µs. These packet size and inter packet arrival time settings results in traffic generation rate that equals 55 Mbps. The Transport Protocol is set to UDP.

Step 4: Run simulation for 10 sec.

Step 5: Note down the following output metrics: (i) Application Throughput from Application Metrics the Results Window, and (ii) Packet Transmitted and Packet Collided from Link Metrics in the Result window.

Step 6: Similarly increase the number of STAs 1, 2, 3, 4, 5, 10, 15, 20 and note down the Throughput (Mbps) and Packet Transmitted and Packet Collided from Result window as explained in Step 5.

Results

Case 1: Results for IEEE 802.11b

No of STAs	NetSim Simulation Results				Collision Probability (solving the FPE using MATLAB)
	Sum Throughput (Mbps)	Data Packets			
		Packets Transmitted	Packets Collided	Collision Probability (p)	
1 STA	5.9276	5075	0	0.0000	0.0000
2 STA	6.1600	5620	346	0.0616	0.0588
3 STA	6.1729	5952	666	0.1119	0.1132
4 STA	6.1589	6171	898	0.1455	0.1621
5 STA	6.0678	6400	1205	0.1883	0.2047
6 STA	6.0479	6529	1351	0.2069	0.2410
7 STA	5.9696	6689	1577	0.2358	0.2716
8 STA	5.8891	6841	1799	0.2630	0.2975
9 STA	5.8902	6897	1854	0.2688	0.3195
10 STA	5.7781	7076	2129	0.3009	0.3384

Table 3: Sum throughput and collision count with STAs transmitting 1460B packet size saturation traffic in the uplink

Case 2: Results for IEEE 802.11g

No of STAs	NetSim Simulation Results				Collision Probability (solving the FPE using MATLAB)
	Sum Throughput (Mbps)	Data Packets			
		Packet Transmitted	Packets Collided	Collision Probability (p)	
1 STA	29.2175	25015	0	0.0000	0.0000
2 STA	28.9454	28364	3582	0.1263	0.1109
3 STA	28.6627	30398	5858	0.1927	0.2022
4 STA	28.3065	31776	7540	0.2373	0.2697
5 STA	27.7587	33058	9292	0.2811	0.3181
6 STA	27.4305	33907	10422	0.3074	0.3541
7 STA	27.0170	34721	11589	0.3338	0.3821
8 STA	26.7308	35370	12483	0.3529	0.4050
9 STA	26.4879	36021	13343	0.3704	0.4242
10 STA	26.2029	36572	14137	0.3866	0.4408

Table 4: Sum throughput and collision count with STAs transmitting 1460B packet size saturation traffic in the uplink

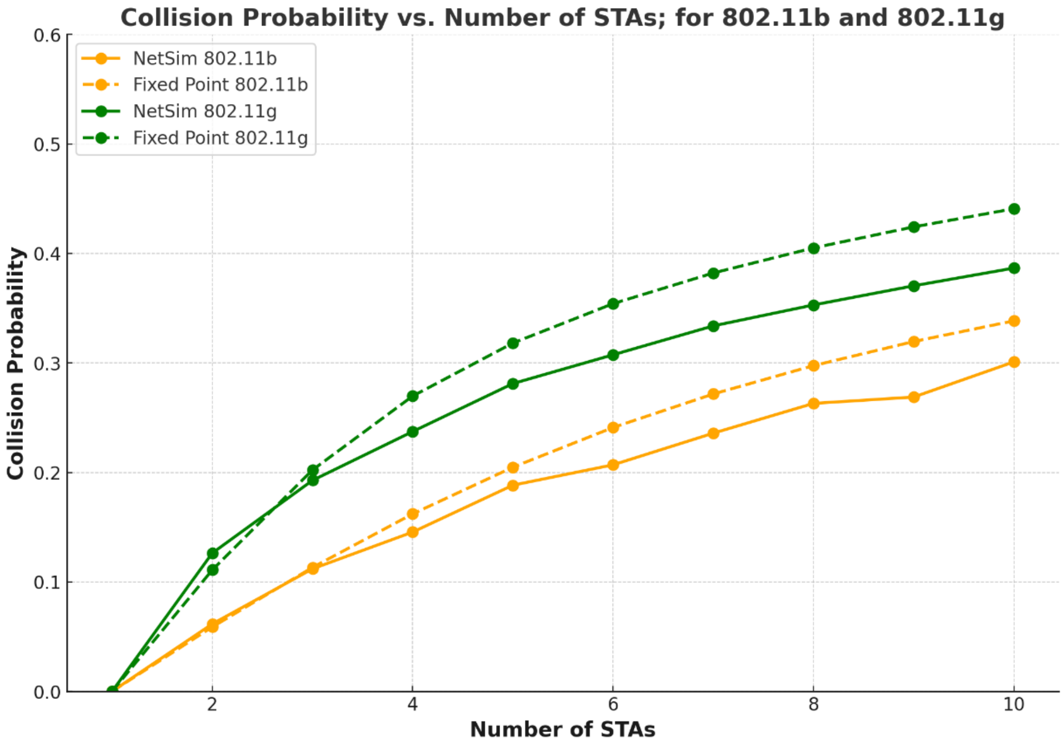

Figure 3: Plot for collision probability vs. number of STAs. We compare NetSim results against the Bianchi model predictions

Discussion

Analysis of the data presented in Figure 3 reveals a close correspondence between the simulation results and the theoretical predictions across various node counts, underscoring the effectiveness of the Bianchi model in approximating the behaviour of the IEEE 802.11 protocol.

Model refinement

In summary, while the simulation results show a close correspondence with theoretical predictions, a noticeable trend emerges where the simulation indicates marginally lower collision probabilities than theoretical values, especially at higher node counts. This deviation is in line with observations documented in literature and suggests that the assumptions inherent in the Bianchi model may not be fully representative of real-world conditions. Two aspects to consider include:

- The Bianchi model assumes a constant and independent collision rate for each node over time. However, studies highlight short-term unfairness in 802.11, where some nodes get to use the channel for extended durations while locking out the other nodes
- There is ambiguity in the model regarding the precise moment of backoff counter decrement, at the beginning or end of a slot, leading to potential inaccuracies [3].

The above points suggest possible avenues to enhancing the Bianchi model by incorporating these second-order effects, and thereby model real-world Wi-Fi networks with higher precision.

Additionally, it is important to recognize that the Bianchi model primarily addresses saturated traffic conditions. Yet, typical network scenarios are often non-saturated and heterogeneous. Therefore, extending the model to encompass non-saturated environments could broaden its applicability and relevance.

References

[1] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Sel. Areas Commun,* vol. 18, no. 3, pp. 535-547, 2000.

[2] 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. Netw,* vol. 15, no. 3, pp. 588-601, 2007.

[3] I. Tinnirello, G. Bianchi and Y. Xiao, "Refinements on IEEE 802.11 Distributed Coordination Function Modeling Approaches," *IEEE Transactions on Vehicular Technology,* vol. 50, no. 3, 2010.

Appendix: Download URL

The configuration files (scenario, settings, and other related files) of the examples discussed in this analysis are available for users to import and run in NetSim.

Users can download the files from NetSim's git-repository.
Link: https://github.com/NetSim-TETCOS/WiFi-experiment_Bianchi-and-FPE-analysis_v14/archive/refs/heads/main.zip

1. Click on the link given and download the folder.
2. Extract the zip folder. The extracted project folder consists of one NetSim Experiments file, namely *WiFi-experiment_Bianchi-and FPE-analysis_v14.netsimexp*
3. Import per steps given in section 4.9.2 in NetSim User Manual

MATLAB code: Fixed_Point_Iteration.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                 %
%         Collision Probability using Fixed Point Analysis         %
%                                                 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Initialization and Local Variables
CWMin = 31;         % CWmin for 802.11b is 31, 802.11g is 15
W = (CWMin +1) / 2;
tau = 0;          % Initializing tau
p = 0.5;          % Initial guess for collision probability
alpha = 0.95;       % The fixed point relaxation parameter

% Number of STA (Station) values to analyze
n_values = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];

% Loop through different STA values
for n = n_values

  % Iterate through 300 cycles
  for i = 1:300

    % Calculate Numerator and Denominator for the collision probability
    numerator = 1 + p + p^2 + p^3 + p^4 + p^5 + p^6 + p^7;
    denominator = 1 + W + p * W * (1 + 2 * p + (2 * p)^2 + (2 * p)^3 + (2 * p)^4 + (2 * p)^5 + (2 * p)^5 + (2 * p)^5);

    % Calculate tau
    tau = numerator / denominator;

    % An intermediate calculation
    p_int = 1 - (1 - tau)^(n - 1);

    % Update p using the relaxed fixed point iteration method
```

```
        p = (1 - alpha) * p_int + alpha * p;

        % Display the current collision probability
        disp(['Collision Probability (n = ', num2str(n), ', iteration = ', num2str(i), '): ', num2str(p)]);
    end
end
```