Performance Modeling of Radio Access in Unsaturated WLANs

Loïc Chassin de Kergommeaux, Malory Marin, Rémi Watrigant, Thomas Begin, Anthony Busson
ENS de Lyon, Université Claude Bernard Lyon 1, CNRS, Inria, LIP, UMR 5668, 69342, Lyon cedex 07, France
Email: firstname.lastname@ens-lyon.fr

***Abstract*—Wireless Local Area Networks (WLANs), based on the IEEE 802.11 standard, are increasingly deployed in large and dense environments where multiple Access Points (APs) often operate on overlapping radio channels. This creates contention for medium access that can severely impact network performance. While the performance modeling of saturated WLAN has been extensively studied, the unsaturated case remains largely unresolved due to the complexity introduced by heterogeneous traffic loads and contention patterns. In this paper, we propose a scalable and conceptually simple performance model for CSMA/CA-based WLANs that estimates the radio channel access rate of each AP based on their individual load and a conflict graph representing the contending APs. This model may provide a useful tool for network design tasks such as channel assignment and congestion detection. We compare our method with two state-of-the-art solutions and validate our approach through ns-3 simulations across various network sizes and traffic scenarios.**

***Index Terms*—WLAN, Performance Evaluation, Modeling, IEEE 802.11**

I. Introduction

WLANs (Wireless Local Area Networks) are ubiquitous in our daily life. WLANs rely on the IEEE 802.11 standardized protocol, commercially branded as Wi-Fi technology by manufacturers. Radio resources are shared among the nodes of the WLAN according to the Carrier-Sense Multiple Access with Collision Avoidance (CSMA/CA) mechanism. CSMA/CA is a distributed algorithm that works in a listen-before-talk fashion: nodes must sense the radio channel as being free to transmit; otherwise, they defer their transmissions. In the infrastructure mode of 802.11, WLANs are made of two types of nodes: Access Points (APs) to which end-user Stations (STAs) are associated.

WLANs tend to be increasingly large and dense. For instance, universities and companies may deploy an AP every 10 meters or so within facilities to ensure permanent connectivity for STAs.

Ideally, every AP should operate on a different radio channel than its nearby APs since adjacent APs cannot transmit (successfully) simultaneously.

However, due to the limited number of radio channels, adjacent APs frequently operate on the same channel. This leads to conflicts among these APs for the radio channel access, potentially causing significant performance degradation for their associated STAs. Although the introduction of the 5 GHz and 6 GHz frequency bands in the recent amendments of IEEE to the 802.11 standard has enabled new radio channels, their number may remain scarce in large and dense WLANs. This issue can be sharpened due to the common practice of aggregating (also known as channel bonding) two or more radio channels of 20 MHz width into a 40 MHz or wider channel to increase the overall radio channel output.

While the performance evaluation of saturated WLANs (i.e., all APs always have frames ready to transmit) has been extensively studied with well-established results, the case of large and dense WLANs in unsaturated conditions remains largely an open issue. Their analysis is hampered by several obstacles, including (i) The distributed nature of CSMA/CA; (ii) The intricate interactions between adjacent APs (operating on the same channel); (iii) The uneven distribution of the load on the APs with some APs requiring a lot of access to the radio channel to serve their STAs and others being mostly silent with limited needs.

In this paper, we propose a conceptually simple and scalable method for the performance modeling of CSMA/CA-based networks. Our method can handle WLANs comprising several tens of unsaturated (or saturated) APs. It relies on the analysis of the independent sets found in the graph deriving from the APs locations and it accounts for the level of loads (possibly saturated) of each AP. Our method returns the rate of access (i.e., proportion of access per time unit) to the radio channel that each AP will attain. We have implemented an online version of our method, which is publicly accessible at [15]. We run discrete-event simulations in the network simulator ns-3 using various types of traffic, including real-life traces, to assess the accuracy of our method, and we compare it with two state-of-the-art solutions. The numerical results show that our method performs similarly to the best state-of-the-art solution but, unlike them, it sheds lights on the relationship between the graph of APs and the WLAN performance. Our method can find natural applications in the issue of channel assignments (selecting the radio channel and its width for each AP of a WLAN), as well as in the detection & localization of potential congestion in a WLAN (i.e., some APs can fail to transmit their traffic because of the WLAN configuration).

The remainder of the paper is organized as follows. Section II reviews the related literature. In Section III, we present our performance model for unsaturated multi-AP WLANs along with its algorithmic solution. Section IV evaluates the accuracy of our solution by comparing its results with those produced by the ns-3 discrete-event simulator as well as those from two state-of-the-art solutions. Section V concludes this paper.

979-8-3195-0167-7/26/$31.00 ©2026 IEEE

II. State of the Art

Early studies on CSMA/CA modeling primarily focused on scenarios involving a unique AP serving multiple STAs. Seminal works such as [4], [6], [18] developed analytical frameworks to assess key performance metrics including the maximum capacity, defined as the highest achievable load (ranging from 0 to 1) that the WLAN can sustain. While the models in [6], [18] provided only the mean values of this metric, Bianchi's approach [4] introduced a discrete-time Markov chain model to capture the channel access mechanism, enabling more detailed performance analysis. Collectively, these studies established relationships between maximum capacity and various WLAN configuration parameters, such as the contention window size and the number of STAs.

Subsequent efforts extended this line of work to **unsaturated** WLANs—still consisting of a single AP and multiple STAs—offering analytical results for capacity and packet delay under arbitrary levels of load [10], [11], [17].

The rapid growth of WLANs has led to increasingly dense deployments, where multiple neighboring APs often operate on the same radio channel. The methodologies proposed in [4], [6], [10], [11], [17], [18] can be adapted to **multi-AP** scenarios, under the strong assumption that all APs lie within each other's detection range, effectively forming a clique. However, real-world WLAN topologies are typically more complex. To better capture these, conflict graphs are commonly employed to model interactions among APs operating on the same channel. These graphs vary significantly across deployments, depending on factors such as spatial layout and radio environment.

Several studies have investigated how to compute the maximum capacity of multi-AP networks. For example, [5], [8], [19] addressed **ad-hoc or multi-hop** wireless networks, where packets traverse multiple radio hops before exiting the network. This differs from traditional WLANs, where each AP is directly connected to the wired infrastructure. Under strong independence assumptions regarding channel usage, these works derived closed-form expressions for network capacity.

In the classical **single-hop multi-AP** WLAN case, to the best of our knowledge, [13] was the first to provide a closed-form analytical evaluation of WLAN capacity. However, its main limitation lies in the assumption that all APs are fully saturated. In this paper, we build upon the foundational results of [13], such as Equation (1), and extend them to address the more general and practically relevant case of unsaturated WLANs.

Only a few studies have specifically focused on **unsaturated WLANs involving multiple APs**. Building on [13], the authors of [12] proposed a performance model for such scenarios. Although they derived a closed-form solution, its utility is constrained by the fact that their notion of load cannot be directly mapped to the actual input load at each AP. The authors of [9] model an unsaturated WLAN as a saturated WLAN whose backoff-to-transmission ratios must be carefully selected to reflect the actual load at each AP. Then, they derive the WLAN capacity using the work of [13]. However, the algorithm complexity makes this approach unusable on large WLANs (say beyond 30 APs). In [16], the system is modeled using a Markov chain to capture the activity (transmitting or idle) of each AP under a given conflict graph and unsaturated traffic conditions. To manage the complexity induced by the conflict graph, the authors decompose the network into a set of saturated sub-WLANs, allowing them to derive a closed-form expression for AP throughput. Nevertheless, this solution remains computationally demanding and is practical only for networks of limited size (typically up to a dozen APs). In [7], the authors introduce a Markov chain to study the stability region of the WLAN. They obtain numerical results for the case of a WLAN comprising three APs. However, in the general case, the Markov chain does not possess a closed-form expression for its stationary distribution. As a result, the authors need to simulate the Markov chain to evaluate the stationary distribution and derive the WLAN capacity.

In summary, accurately modeling the performance of multi-AP WLANs operating on a shared radio channel, with complex conflict graph topologies and under unsaturated traffic conditions, remains an open research challenge. This paper makes three key contributions to this problem: (i) We demonstrate that, unlike in saturated scenarios, the radio channel access rate achieved by each AP in an unsaturated network depends not only on its average load but also on the statistical nature of the traffic; (ii) For a specific type of traffic, we propose a simple and accurate method to compute the rate of access attained by each AP; (iii) Through empirical evaluation, we show that the radio channel access rate derived under this specific type of traffic provides a good approximation for other traffic patterns, including Poisson arrivals, and real traffic traces, and that our method represents a compelling alternative to the limited number of state-of-the-art solutions currently available.

III. Performance Modeling

A. System under study and notation

We consider a WLAN composed of multiple APs operating on the same radio channel and using the CSMA/CA protocol for channel access. If the WLAN operates over several independent radio channels, it can simply be modeled as multiple disjoint WLANs, each corresponding to a dedicated channel.

Two APs are considered to be in **conflict** if they can detect each other's transmissions. We capture these relationships among APs using a **conflict graph** $G = (V, E)$, where each vertex in V corresponds to an AP, and each edge in E represents a conflict between two APs. Figure 1 shows the conflict graph for a simple WLAN with four APs, based on their respective detection range. In this example, APs a, b, and c can all detect each other's transmissions, forming a clique. In contrast, AP d can only detect transmissions from AP c.

As in many previous studies, we model only the APs and disregard the associated STAs. This modeling choice is supported by two main factors: (i) STAs are generally transient, frequently joining or leaving the WLAN and (ii) WLAN traffic

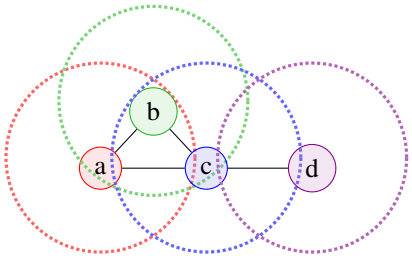


Fig. 1: Example of conflict graph for a four AP network. The dotted lines represent the detection range of APs.

is predominantly downstream (i.e., from APs to STAs). Note that our ns-3 simulations in Section IV include STAs and upstream traffic and their results empirically establish that this assumption does not introduce a significant bias on the performance.

We assume that all APs transmit, on average, at the same physical data rate.

We denote by C the **transmission capacity** (expressed in frames per unit of time) that represents the maximal rate at which AP can transmit its frames when there is no conflict with other APs. The value of C can be easily derived from the physical data rate, the average size of frames and the IEEE 802.11 amendments in use.

For every vertex (i.e., AP) v of the conflict graph, we introduce the following notation:

- ρ_v: The **normalized load**, defined as the mean generation rate of frames at vertex v, divided by C.
- u_v: The **access rate**, defined as the rate of frames successfully transmitted by vertex v, divided by C.
- p_v: The **saturation access rate**, which is similar to u_v but assumes that all vertices have their normalized load ρ_v equal to 1 (i.e. are saturated).

Note that the conflict graph G, the transmission capacity C and the set of loads ρ_v constitute the input parameters of our study, while the radio channel access rates u_v and saturation access rates p_v are the output parameters to be determined.

By definition, $\rho_v \in [0, 1]$ and $u_v \leq \rho_v$. We say that a vertex v is **congested** if and only if $u_v < \rho_v$. Otherwise, this vertex is described as **satisfied**. Once the values of u_v are known, the actual throughput attained by each AP can be obtained by scaling u_v according to the AP's physical data rate and average frame size.

B. Computation of the p_v in the saturated case

We first consider the case of a **fully saturated** WLAN, where each AP always has a frame ready for transmission. Following the methodology proposed in [13], we compute the values of p_v based only on the structure of the conflict graph G. Let $\mathcal{I}(G)$ denote the set of all **independent sets** of G, i.e., subsets of vertices $K \subseteq V$ such that no two vertices in K are adjacent—that is, for all $u, v \in K$, the edge uv does not exist. In the example of Figure 1, we have $\mathcal{I}(G) = \{(a, d), (b, d), (a), (b), (c), (d), \emptyset\}$. The saturation access rate for each vertex v is given by:

$$p_v = \frac{\sum_{K \in \mathcal{I}(G) | v \in K} \theta^{|K|}}{\sum_{K \in \mathcal{I}(G)} \theta^{|K|}}, \tag{1}$$

where θ denotes the transmission-to-backoff ratio, i.e., the ratio between the duration of a frame transmission (including its acknowledgement, often denoted by ACK) and the mean duration of the backoff delay. In practice, θ is generally large (say greater than 10) and tends to increase with modern IEEE 802.11 amendments that support the grouping of multiple frames into a large frame (i.e., frame aggregation) and the introduction of Orthogonal Frequency-Division Multiple Access (OFDMA).

Note that as $\theta \to +\infty$, the **maximum independent sets, (MISs)**, which represent the independent set of largest cardinality, significantly contribute to the computation of p_v. In this limit, we obtain for every vertex v:

$$\lim_{\theta \to +\infty} p_v = \frac{\text{Number of MIS containing } v}{\text{Number of MIS in } G}. \tag{2}$$

Equation (2) highlights the strong dependence of p_v's on the topology of the conflict graph.

Calculating the exact values of p_v using Equations (1) and (2) requires the enumeration of all independent sets. This process is computationally intensive, making exact computation infeasible for large graphs. To address this challenge, we present an approximate algorithm in Section III-E that enables the efficient computation of p_v.

C. Our solution for the unsaturated case

We now address the more complex case of an **unsaturated** WLAN, where not all APs have continuously frames ready for transmission.

The rationale behind our approach is to consider a representative time interval for the traffic generated at the APs.

We denote by t the relative time within this interval, where $t = 0$ corresponds to the start and $t = 1$ to the end. To ease the performance analysis, we assume that every vertex v generates an amount of traffic equals to the product of its normalized load ρ_v and the interval duration at $t = 0$, and then aims to transmit all this traffic before $t = 1$. We analyze the behavior of the WLAN within such a single interval, assuming a stationary regime: all intervals are statistically identical, and the WLAN behaves similarly across them. Then, our algorithm proceeds as follows (the formal pseudo-code is provided in Algorithm 1):

1) **Initialize**. Set the time variable t to 0 and mark all vertices as active, i.e., having frames ready for transmission.
2) **Assume saturation**. For all the active vertices, the WLAN can be considered as saturated, and each vertex v transmits at its saturation access rate p_v. We rely on the subroutine *computeSaturated()* which approximates Equation (1) from the current conflict graph, using a randomized sampling algorithm discussed in Section III-E.

3) **Drain traffic loads**. Every active vertex reduces its residual normalized load ρ_v at its rate p_v until either:
   - at least one vertex completed the transmission of its traffic load, or
   - the interval ends (i.e., $t = 1$).
4) **Update the conflict graph**. When a vertex v completes the transmission of its load (i.e., $\rho_v = 0$), it is removed from the conflict graph and no longer competes for access during the rest of the interval.
5) **Advance time**. Increase the current time t by the amount of time that was required for the vertex that just left the conflict graph to complete its traffic load, i.e., $\min_{v \in \text{active}} \left(\frac{\rho_v}{p_v}\right)$.
6) **Repeat**. Return to Step 2 and continue the process until either:
   - all vertices have completed the transmission of their load, or
   - the interval ends (i.e., $t = 1$).
7) **Compute the access rates**. At the end of the interval, the access rates u_v's are calculated as the total amount of load transmitted over the whole interval duration.

This iterative procedure captures how vertices progressively transmit their load and adjust their behavior to the evolving contention environment as other vertices complete the transmission of their load.

Algorithm 1: Computing the access rates u_v's of APs

Input: Graph G
Output: Performances $\{u_v\}_{v \in V(G)}$
$t = 1$;
foreach $v \in V(G)$ **do**
  $u_v = 0$;
while $t > 0$ **do**
  $A = \{v \in V(G) \mid \rho_v > 0\}$ #Set of active vertices ;
  $\{p_v\}_{v \in A}$ = computeSaturated($G[A]$) #See Section III-E;
  foreach $v \in A$ **do**
    if $p_v \neq 0$ **then**
      $\delta_v = \rho_v$ / p_v;
    else
      $\delta_v = \infty$;
  min_time = $\min(\min_{v \in A}(\delta_v), t)$;
  foreach $v \in A$ **do**
    $u_v = u_v + min_time \cdot p_v$;
    $\rho_v = \rho_v - min_time \cdot p_v$;
  $t = t - min_time$;
return $\{u_v\}_{v \in V(G)}$;

D. Simple example

We illustrate our solution using the simple example shown in Figure 2. For ease of understanding, in this simple example, we use Equation (2) in place of Equation (1) to compute the saturation access rates p_v. We set the normalized loads ρ_v for vertices a, b, c and d to:

$$\rho_a = 0.9, \rho_b = 0.3, \rho_c = 0.4 \text{ and } \rho_d = 0.8.$$

At time $t = 0$, all vertices are active and the WLAN can be considered as saturated. Hence, the conflict graph features two MIS (i.e., (a, d) and (b, d)). Applying Equation (2), we obtain the following initial saturation access rates:

$$p_a = 0.5, p_b = 0.5, p_c = 0, \text{ and } p_d = 1.$$

Each vertex begins draining its load at its corresponding rate p_v. The time required for each vertex to transmit its entire load is $\frac{\rho_v}{p_v}$ (for $p_v > 0$). Since vertex b has the smallest load-to-rate ratio $\frac{\rho_b}{p_b} = 0.6$, it is the first to complete the transmission of its load. This marks the end of the first iteration at time $t = 0.6$ and we remove vertex b from the conflict graph. The remaining vertices have transmitted a fraction of their load proportional to their respective saturation access rate. At this point, the transmitted load and residual load for each vertex are:

- a: transmitted $0.6 \times p_a$ = 0.3, residual load= 0.6,
- c: transmitted $0.6 \times p_c$ = 0, residual load= 0.4,
- d: transmitted $0.6 \times p_d$ = 0.6, residual load= 0.2.

In the updated conflict graph containing only vertices a, c, and d, we recompute the saturation rates and we obtain:

$$p_a = 1, p_c = 0, \text{ and } p_d = 1.$$

With these new rates, vertex d is the next vertex to complete its residual load of 0.2 in an additional $\frac{0.2}{1}$ time units. At the end of the second iteration, the total elapsed time is $t = 0.8$ and only vertices a and c remain, with residual loads of 0.5 and 0.4, respectively. In the corresponding conflict graph with only these two vertices, the saturation access rates become:

$$p_a = 0.5, \text{ and } p_c = 0.5.$$

With only 0.2 time unit left in the interval, each of these vertices can transmit an additional $0.2 \times 0.5 = 0.1$ of their load. Therefore, at the end of the interval (i.e., at $t = 1$), vertices a and c remain with residual loads of 0.4 and 0.3, respectively. This indicates that they are both congested. Finally, the access rates u_v, which reflect the overall channel access rate of each vertex, are computed as:

$$u_a = 0.5, u_b = 0.3, u_c = 0.1, \text{ and } u_d = 0.8.$$

E. Breaking the time complexity

The time complexity of Algorithm 1 strongly depends on the complexity of the subroutine $computeSaturated(.)$ used to compute the p_v values for the current conflict graph. The *while* loop calling $computeSaturated(.)$ is run at most $|V(G)|$ iterations since the number of active vertices decreases at each iteration. Denoting by $T(G)$ the time complexity of the function $computeSaturated(.)$, the overall complexity of

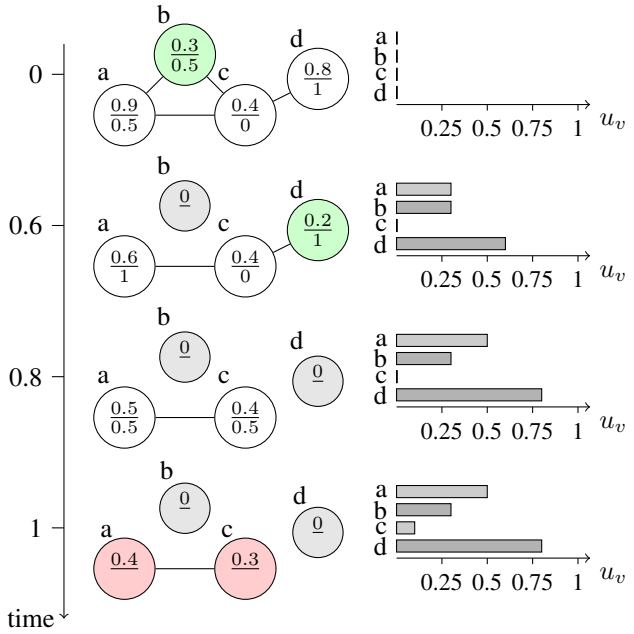


Fig. 2: Illustration of Algorithm 1 applied to a conflict graph with four vertices. For each iteration, the left panel shows the current conflict graphs, while the right panel displays the corresponding attained channel access rates u_v. When defined, the ratio ρ_v/p_v is displayed inside each vertex. **Green** vertices represent those that will the next to complete their transmissions. **Grey** vertices indicate those that are already completed and are thus excluded from the conflict graph; their p_v values are no longer computed. **Red** vertices are those that remain congested when the interval ends since they were unable to transmit all their load during the interval duration.

Algorithm 1 is thus $O(|V(G)|\cdot T(G))$. In the general case, the complexity $T(G)$ is exponential in the number of vertices[1].

We now focus on the complexity analysis for the particular case of geometric graphs, and more specifically, on **unit disk graphs**. In a unit disk graph, each vertex is associated with a unit disk in the plane, and two vertices are adjacent (i.e., there is an edge) if and only if their corresponding disks intersect. These graphs are commonly used to model wireless networks. For unit disk graphs, [14] presents a randomized sampling algorithm which, given a graph G, returns an independent set $K \in \mathcal{I}(G)$ with probability approximately $\theta^{|K|}/Z_G(\theta)$, where $Z_G(\theta) = \sum_{I\in\mathcal{I}(G)} \theta^{|I|}$. This algorithm runs in sub-exponential time $2^{O(\sqrt{|V(G)|})}$. Based on this algorithm, parameterized with $\theta = 10$, we estimate the p_v values by repeatedly sampling independent sets and computing the frequency with which each vertex appears in a sample. Compared to classical algorithms with exponential time complexity $2^{O(|V(G)|)}$, the sub-exponential approach provides a substantial gain. For example, in a conflict graph with 100 vertices, an exponential-time algorithm would require approximately 2^{100} iterations, whereas a sub-exponential algorithm would require only about 2^{10} iterations, up to constant factors. This complexity remains computationally tractable for graphs with several tens of vertices.

[1]unless the Exponential Time Hypothesis (ETH) fails.

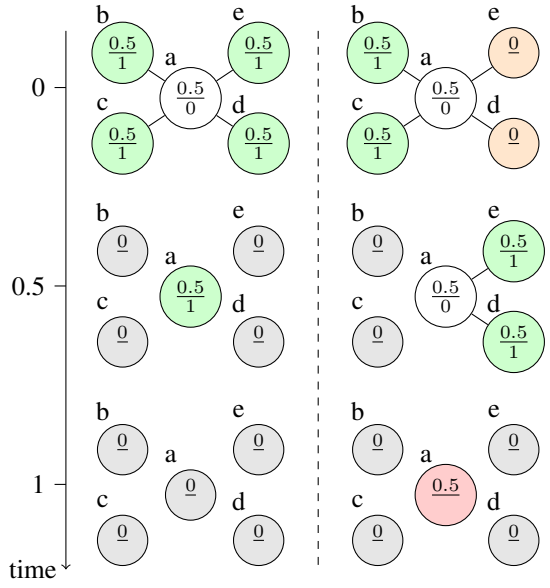


Fig. 3: Influence of traffic distribution on channel access rates u_v. All vertices generate their traffic at $t = 0$ except for **orange** vertices that generate their traffic at $t = 0.5$. Other colors (green, red, grey) follow the rule given in Figure 2. For a same conflict graph and the same normalized loads of 0.5 at every vertex, the resulting values of u_v may differ if the statistical nature of the loads differ. On the left panel, all vertices generate their traffic at $t = 0$ whereas on the right panel, the vertices d and e generate their traffic at $t = 0.5$.

F. Influence of the traffic distribution on the radio channel access rates u_v

For saturated WLANs, the computation of the channel saturation access rates, namely p_v, has been formally established (see Section II). Under these conditions, the values of p_v's are independent of the statistical properties of the traffic. In contrast, this independence does not hold in unsaturated WLANs and we demonstrate it through a counter-example.

Consider an unsaturated WLAN containing five vertices (see Figure 3). The corresponding graph consists of a central vertex a adjacent to four mutually-independent vertices b, c, d, and e. Each vertex v is assigned a normalized load ρ_v of 0.5.

First, we consider the case where, similarly to the assumption made in Algorithm 1, all traffic is generated at the start of the time interval, i.e., $t = 0$. Then, during the first iteration, which lasts until $t = 0.5$, we have $p_a = 0$ while $p_b = p_c = p_d = p_e = 1$. Consequently, by $t = 0.5$, all vertices except a have completed their transmissions. In the next iteration, a is the sole remaining active vertex and is able to transmit its entire load before the interval ends, i.e., $t = 1$.

As a result, all vertices achieve a radio channel access rate u_v of 0.5.

Now, we consider that the traffic of a, b, and c is generated at $t = 0$, while the traffic for d and e is generated later, at $t = 0.5$. In this setting, a is permanently unable to access the radio channel throughout the interval, as it always has at least two neighbors with active traffic. Therefore, we obtain: $u_a = 0$, while the other vertices achieve to transmit their load, resulting in $u_b = u_c = u_d = u_e = 0.5$.

Overall, this simple example illustrates that the actual values of u_v's are sensitive to the statistical nature of the traffic at each vertex. In our analytical solution, we assume a particular type of traffic (i.e., each AP generates all its traffic at the beginning of the interval) and although not exact, the examples shown in the next section suggest that this is a reasonable approximation for various types of traffic.

IV. Numerical Results

A. Measures for accuracy

In this section, we assess the ability of our method to analytically predict the performance of an unsaturated multi-AP WLAN. More precisely, through three distinct scenarios, we show the ability of our model to accurately evaluate the radio channel access rate u_v's, as well as its ability to determine whether a given AP v can handle its normalized load ρ_v.

We use the discrete-event simulator ns-3 [1] as the ground truth for the u_v values. In addition to our model, we implemented two solutions from the literature: (i) The algorithm proposed in [9], hereafter referred to as **EAI13**. Owing to its high computational complexity, we were only able to execute this algorithm on one of the three scenarios, where the number of APs is relatively small (namely, 20 APs). Indeed, the algorithm requires enumerating all independent sets of the WLAN's conflict graph, which becomes intractable for networks with more than 30 APs. (ii) The second solution is the Markov chain model described in [7], referred to as **PEVA14**. For each AP, the packet arrival times follow a Poisson process, and the backoff as well as the transmission times follow an exponential distribution. Unlike EAI13 and our proposed solution, and despite memoryless assumptions for the packet inter-arrival, transmission and backoff times, the PEVA14's model does not possess a closed-form expression for its stationary distribution, so that simulations must be used, akin to a discrete-event simulator.

The three algorithms use θ as an input parameter.

However, its definition and usage slightly vary among the algorithms. In our model, θ serves as a parameter for the function that computes the saturation access rates p_v. In PEVA14, θ is defined as the ratio of the average transmission time to the average backoff time. In EAI13, θ is initialized with a specific value and then iteratively updated to reflect the actual load at each AP. Finally, in a real WLAN, each AP may exhibit a different value of θ, dependent on its traffic and associated STAs. However, these values are typically unknown a priori. Hence, for the three algorithms, we set $\theta = 10$, as this value can frequently be observed in WLANs operating under 802.11ax standard. We implemented our solution as well as PEVA14 [7] in C++ and EAI13 [9] in Python and made the source code used for all our experiments (models and ns-3) publicly available [2].

TABLE I: ns-3 parameters.

Parameter	Value
ns-3 version	ns-3.43
Wi-Fi	IEEE 802.11ax
MCS	HT-MCS 0
Aggregation	Disabled
Frequency band	5 GHz
Channel width	40 MHz
Guard interval	800 ns
Frames size (in case of Poisson traffic)	1440 bytes
Physical data rate	10 Mbps
Attenuation between adjacent APs	10 dB
Attenuation between non-adjacent APs	200 dB

For each scenario, we compute several measures of accuracy to compare the outcomes of each solution, namely EAI13, PEVA14, and our model, against those delivered by the ns-3 simulations.

We compute the **deviation on** u_v**'s**, defined as the difference between the u_v's delivered via simulation and those predicted by each of the three considered solutions. This standard measure is effective to quantitatively validate the accuracy of our model.

B. Simulation setup

In the simulation setup, the propagation path loss between adjacent APs is set to 10 dB, and to 200 dB between non-adjacent APs to effectively eliminate interference. For each scenario, the WLAN is simulated over a period of 30 seconds. Every AP is associated with a single STA to which it transmits frames. Table I summarizes the parameters used for ns-3 experiments.

In the ns-3 simulator, the frame arrival process has to be specified at each AP. We consider two types of traffic: (i) Poisson, where inter-frame intervals follow an exponential distribution and the size of frames is of 1440 bytes. (ii) Real trace, based on real-life WLAN traffic that determines both the frames arrivals and sizes. For the real trace, we use data collected from an AP operating in library's WLAN of the Université Claude Bernard Lyon 1 (UCBL). This trace was captured via a mirroring port on the switch connected to the AP, enabling us to record only the downstream traffic arriving at the wired interface. The trace was anonymized and made publicly available to ensure reproducibility [2]. In simulations, each AP replays a randomly selected segment of the real-life trace. We simulate the real trace experiments five times using different excerpts of the trace before averaging them for the presented results. Note that the coefficient of variation of inter-frame intervals is by definition 1 for Poisson and approximately 1.7 for the real trace. For both types of traffic, we scale the inter-frame intervals using a multiplicative factor to match the normalized load ρ_v assigned to each AP.

The conflict graphs used in all scenarios are unit disk graphs generated as follows: Given an integer $n \in \mathbb{N}$ and a real number $r \in \mathbb{R}$, we uniformly sample n points within the unit square $[0, 1]^2$. Each point corresponds to a vertex, and an edge is added between two vertices if the Euclidean distance between their corresponding points is at most $2r$. The parameter n determines the number of APs (i.e., the graph's cardinality), while r controls the expected **edge density**, defined as the ratio of the number of actual edges to the total number of possible edges in the graph.

We consider three distinct scenarios presented in the three next subsections. To account for variability, we perform 10 independent replications for each scenario where the conflict graph and the load assigned to each AP are generated randomly.

C. Scenario 1: 20 APs with uniformly distributed load

In the first scenario, we consider a relatively dense WLAN consisting of 20 APs. The conflict graph has an edge density of 0.3, meaning that on average each AP is in conflict with 30% of the other APs, namely 6 in our case. The load at each AP v is drawn uniformly from the interval [0.1, 0.4] (i.e., $\rho_v \sim U[0.1, 0.4]$). This corresponds to a moderately loaded WLAN, with about 60% of the APs being congested. In the ns-3 simulations, the traffic is generated as a UDP flow from the AP to its STA.

Figure 4 depicts the distribution of deviation on u_v's between simulation results and the predictions made by each of the three solutions for both the case of Poisson traffic and real trace for the first scenario. Overall, the figure shows our model's ability to fairly predict the values of u_v. Nevertheless, we observe a slight bias in the model, which tends to slightly underestimate u_v for some APs. Figure 4 also indicates that the three solutions, namely our model, EAI13, and PEVA14, provide accurate predictions on the channel access rates u_v.

To better assess the accuracy of our model, Table II reports the mean and median values of the deviation between predicted and simulated values of u_v, in addition to its distribution. We observe that the mean deviations committed by our model are small: about 0.02 for both type of traffic. Equally important, looking at the last column of Table II, the deviation stays below 0.1 for almost all APs, regardless of the traffic type considered.

TABLE II: Scenario 1 - Mean, median, and distribution of the absolute deviation on the channel access rates u_v.

Traffic	Algorithm	Mean	Median	<0.02	<0.05	<0.10
Poisson	our model	0.022	**0.009**	67.5%	80.0%	99.5%
	PEVA14	**0.017**	0.010	**72.5%**	**91.0%**	**100.0%**
	EAI13	0.022	**0.009**	68.0%	81.5%	99.5%
Real trace	our model	0.022	**0.005**	62.5%	78.0%	99.5%
	PEVA14	**0.017**	0.007	**68.5%**	**88.5%**	**100.0%**
	EAI13	0.021	**0.005**	65.0%	81.5%	99.5%

D. Scenario 2: 50 APs with upstream traffic

The second scenario considers a larger WLAN composed of 50 APs with an edge density of 0.3. The normalized loads ρ_v at each AP are drawn from a mixture of two Gaussian distributions: 70% of APs are sampled from a distribution with a mean and standard deviation of 0.2, while the remaining 30% with a mean and standard deviation of 0.5. Under this configuration, approximately 65% of APs are congested. Unlike Scenario 1, in the ns-3 simulations, we include an upstream traffic, implemented as a UDP flow that comprises 10% of all frames. This scenario aims at validating the assumption of our model (also made by the state-of-the-art solutions) to consider only the APs when constructing the conflict graph. Note that in this scenario, we only compare our model to PEVA14, as it was computationally impossible to run the EAI13 algorithm on a WLAN of 50 APs.

TABLE III: Scenario 2 - Mean, median, and distribution of the absolute deviation on the channel access rates u_v.

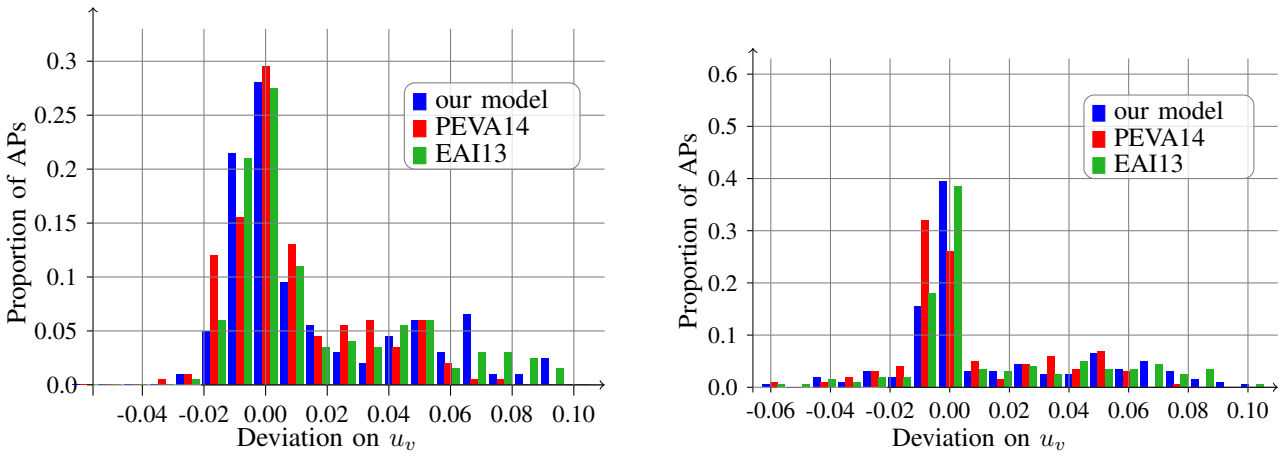
Traffic	Algorithm	Mean	Median	<0.02	<0.05	<0.10
Poisson	our model	0.041	0.038	27.0%	64.2%	96.2%
	PEVA14	**0.034**	**0.031**	**32.6%**	**76.2%**	**99.6%**
Real trace	our model	**0.017**	**0.010**	**70.4%**	**92.0%**	**99.8%**
	PEVA14	0.021	0.011	64.8%	86.6%	98.4%

We omit the detailed distribution of the deviation on u_v's between the predictions and the simulations, as the results are largely similar to those observed in Figure 4 for Scenario 1. Instead, we summarize the absolute deviation on u_v's in Table III. The average deviation on u_v varies between 0.017 and 0.041 for our model depending on the traffic type, and between 0.021 and 0.034 for PEVA14. Not surprisingly, PEVA14 brings its better results for Poisson traffic (since it assumes itself Poisson arrivals for frames) whereas our model is more accurate when dealing with the real trace.

E. Scenario 3: 50 APs with spatially correlated load and TCP traffic

Similarly to Scenario 2, our third scenario considers a WLAN comprising 50 APs but an edge density of 0.2. Also, the load distribution here is spatially correlated as this may be the case in real WLAN deployments in which certain geographical areas may experience higher traffic demands than others. To model this, we partition the graph into four distinct zones. Within each zone, the normalized load at each AP is drawn from a different Gaussian distribution (μ_1=0.2, σ_1=0.3, μ_2=0.1, σ_2=0.2, μ_3=0.3, σ_3=0.1, μ_4=0.4, σ_4=0.2). Overall, this setup produces a lightly loaded WLAN, with an average of about 50% of APs being congested, although this proportion varies across the zones. To be more realistic, in our ns-3 simulations, the traffic between APs and their STAs use TCP connections. As a result, frames transmitted by APs cause TCP ACKs returning from the STAs towards to the APs.

Table IV presents the absolute deviation committed on the estimation of the channel access rates u_v by our model and by PEVA14. We observe that both solutions provide a similar degree of accuracy. In the case of the real trace, the absolution deviation made by our model remains below 0.1 for more than 97% of APs. More precisely, the median absolute deviation is



(a) Poisson arrivals with constant frame sizes

(b) Real trace for frame arrivals and sizes

Fig. 4: Scenario 1 - Distribution of the deviations between the channel access rates u_v predicted by our model, PEVA14 and EAI13 and those obtained from ns-3 simulations.

TABLE IV: Scenario 3 - Mean, median, and distribution of the absolute deviation on the channel access rates u_v.

Traffic	Algorithm	Mean	Median	<0.02	<0.05	<0.10
Poisson	our model	**0.033**	0.019	50.5%	79.0%	**93.1%**
	PEVA14	0.034	0.019	**52.2%**	**79.2%**	92.9%
Real trace	our model	0.027	0.022	46.5%	86.6%	**97.5%**
	PEVA14	0.027	**0.019**	**52.2%**	86.6%	96.6%

small: 0.019 in case of Poisson traffic and 0.022 in case of the real trace.

In addition to the deviation on u_v's, we evaluate the solutions ability to classify APs as either congested or satisfied. From an operational point of view, the ability to forecast which APs in a WLAN are likely to be congested may be crucial for an effective network planning and resource management. As defined in Section III-A, an AP v is considered **congested** if $u_v < \rho_v$, and **satisfied** if $u_v = \rho_v$. To account for the randomness in traffic generation (the actual load may slightly deviate from the target ρ_v), we introduce a tolerance threshold. In our experiments, we set $\varepsilon = 0.05$. Therefore, an AP is considered congested if $u_v < \rho_v - \varepsilon$, and satisfied otherwise.

To evaluate the classification performance of each solution, we compute the standard metrics of precision, recall, and F1 score. In the following, we denote True Positives as TP, False Positives as FP and False Negatives as FN. The **precision** measures the proportion of APs predicted to be in a given state (e.g., congested) that are actually in that state. More formally, Precision $= \frac{TP}{TP+FP}$. The **recall** measures the proportion of APs actually in a given state that are correctly predicted as such. It is computed as follows: Recall $= \frac{TP}{TP+FN}$. The **F1 score** is the harmonic mean of precision and recall and hence balances both components: $F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$.

Table V reports the obtained results for identifying satisfied and congested APs. Our model achieves high accuracy in predicting congested APs, reaching up to 95.3% for Poisson traffic and 94.4% for real-trace traffic. We also observe that the F1 score of our solution slightly exceeds that of PEVA14.

Overall, the results of Table V confirms the ability of our model to predict whether an AP is likely to be satisfied or congested, based on its location in the conflict graph and its assigned load.

TABLE V: Scenario 3 - Classification performance metrics.

Traffic	AP state	Algorithm	Precision	Recall	F_1
Poisson	Satisfied	our model	**0.778**	0.907	**0.838**
		PEVA14	0.740	**0.940**	0.828
Poisson	Congested	our model	0.953	**0.880**	**0.915**
		PEVA14	**0.968**	0.847	0.903
Real trace	Satisfied	our model	**0.801**	0.892	**0.844**
		PEVA14	0.755	**0.918**	0.829
Real trace	Congested	our model	0.944	**0.890**	**0.916**
		PEVA14	**0.954**	0.853	0.901

F. Online implementation of our model

Finally, we have implemented and deployed an online version of our model, which is publicly accessible at [15]. This web-based tool enables anyone to specify the conflict graph of a WLAN and the load at each AP and, in return, to obtain the corresponding predicted radio channel access rates u_v. Figure 5 is a snapshot of the web page hosting our tool. The graphical interface (front-end) was coded using the specialized JavaScript library D3 [3], while the computation of the performance metrics by the model was coded using C++ (code available at [2]).

The tool enables the end-user to easily add/remove APs (vertices) and conflicts (edges) between APs to describe the WLAN topology (conflict graph) as well as the normalized load at each AP. It also features preset examples conflict graph (e.g., paths, cliques, stars) and the ability to save examples for a later utilization.

We believe that this tool can be of interest for a WLAN administrator to better understand the strong relationships between the APs locations and the WLAN performance.

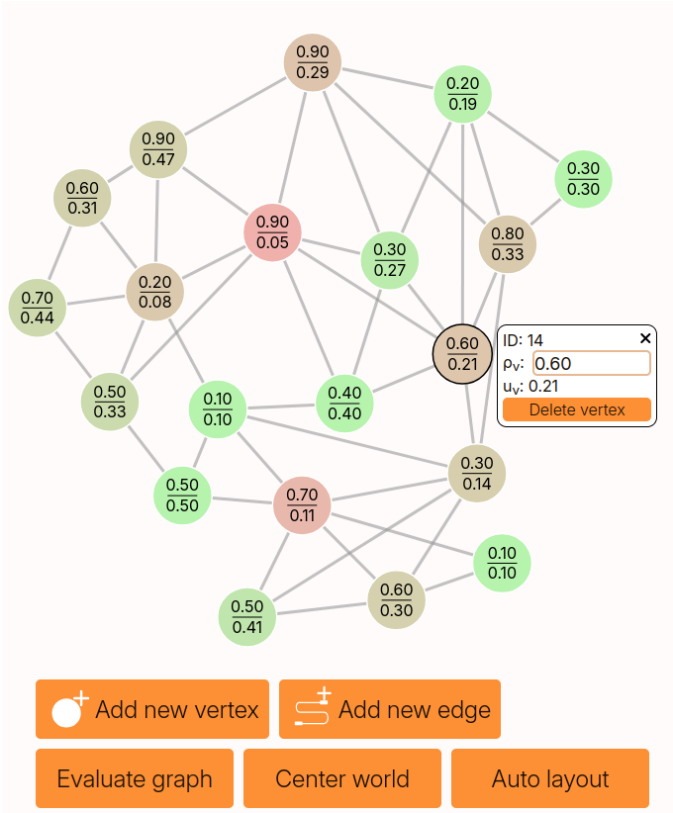


Fig. 5: Snapshot of the online implementation of our proposed performance model.

V. Conclusions

In this paper, we address the problem of modeling the performance of multi-AP WLANs in unsaturated traffic conditions. While the saturated case has been extensively studied and formally resolved, proposing an analytical model to address the case of an unsaturated WLAN remains largely an open problem. Our model estimates the radio channel access rates of each AP based on their traffic load and the conflict graphs among APs. The model is scalable and can handle WLANs with several tens of APs. We evaluated its accuracy on WLANs of varying sizes and load configurations and the corresponding results show that our model is generally accurate.

Our model leverages the intrinsic topology of the conflict graph among APs, particularly focusing on the existence of independent sets. In terms of accuracy and scalability, our approach significantly outperforms existing analytical models that also utilize the conflict graph structure. Other solutions, which employ large Markov chains, achieve comparable accuracy to our model. However, these alternatives typically require simulations that emulate the CSMA/CA protocol of WLANs to determine the radio channel access rates of each AP, thereby neglecting the critical role of the conflict graph and the structure of its independent sets.

Our model can fin natural application to complement a radio channel allocation algorithm. Its ability to accurately detect APs that are likely to experience congestion, based on their position in the conflict graph or the load of adjacent APs, makes it a valuable asset to these algorithms.

For future work, we plan to expand our work to support the case of dynamic channel bonding. Under this setting, APs would be able to transmit on channels with widths that adapt based on the activity of adjacent APs.

Acknowledgment

We thank the Centre Inter-établissement pour les Services Réseaux (CISR) of University Claude Bernard Lyon 1 for collecting and providing us the Wi-Fi traces of real traffic from UCBL's library.

References

[1] ns-3.43. The Network Simulator ns-3. https://www.nsnam.org/, 2025.
[2] Source code : Performance modeling of radio access in unsaturated WLANs. https://github.com/WLAN-performance/Performance-Modeling-of-Radio-Access-in-Unsaturated-WLAN, 2025.
[3] D3. The D3 JavaScript library. https://d3js.org/, 2026.
[4] G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on selected areas in communications*, 18(3):535–547, 2000.
[5] R. R. Boorstyn, A. Kershenbaum, B. Maglaris, and V. Sahin. Throughput analysis in multihop CSMA packet radio networks. *IEEE Transactions on Communications*, 35(3):267–274, 1987.
[6] F. Cali, M. Conti, and E. Gregori. IEEE 802.11 wireless LAN: Capacity analysis and protocol enhancement. In *Proceedings. IEEE INFOCOM'98*, volume 1, pages 142–149. IEEE, 1998.
[7] Fabio Cecchi, Sem C Borst, and JSH Van Leeuwaarden. Throughput of csma networks with buffer dynamics. *Performance Evaluation*, 79:216–234, 2014.
[8] M. Garetto, T. Salonidis, and E. W. Knightly. Modeling per-flow throughput and capturing starvation in CSMA multi-hop wireless networks. *IEEE/ACM Transactions on Networking*, 16(4):864–877, 2008.
[9] Caihong Kai and Shengli Zhang. Throughput analysis of csma wireless networks with finite offered-load. In *2013 IEEE International Conference on Communications (ICC)*, pages 6101–6106. IEEE, 2013.
[10] J.H Kim and J.K. Lee. Performance of carrier sense multiple access with collision avoidance protocols in wireless LANs. *Wireless Personal Communications*, 11:161–183, 1999.
[11] L. Kleinrock and F. Tobagi. Packet switching in radio channels: Part I-carrier sense multiple-access modes and their throughput-delay characteristics. *IEEE Transactions on Communications*, 23(12), 1975.
[12] R. Laufer and L. Kleinrock. The capacity of wireless CSMA/CA networks. *IEEE/ACM Transactions on Networking*, 24(3), 2015.
[13] S.C. Liew, C.H. Kai, H.C. Leung, and P. Wong. Back-of-the-envelope computation of throughput distributions in CSMA wireless networks. *IEEE Transactions on Mobile Computing*, 9(9):1319–1331, 2010.
[14] Malory Marin. Subexponential and parameterized mixing times of glauber dynamics on independent sets. In *International Conference and Workshops on Algorithms and Computation*, pages 276–290. Springer, 2026.
[15] Jean Pantaleon, Loïc Chassin de Kergommeaux, Malory Marin, Rémi Watrigant, Thomas Begin, and Anthony Busson. Performance Evaluation of WLANs. https://wlan-performance.ens-lyon.fr/, 2026.
[16] M. Stojanova, T. Begin, and A. Busson. Conflict graph-based model for IEEE 802.11 networks: A divide-and-conquer approach. *Performance Evaluation*, 130:64–85, 2019.
[17] H. Takagi and L. Kleinrock. Throughput analysis for persistent CSMA systems. *IEEE transactions on communications*, 33(7):627–638, 2003.
[18] Y.C. Tay and K.C. Chua. A capacity analysis for the IEEE 802.11 MAC protocol. *Wireless networks*, 7:159–171, 2001.
[19] X. Wang and K. Kar. Throughput modelling and fairness issues in CSMA/CA based ad-hoc networks. In *Proceedings IEEE INFOCOM '05*, volume 1, pages 23–34. Ieee, 2005.