Conflict graph-based model for IEEE 802.11 networks: A Divide-and-Conquer approach

M. Stojanova, Thomas Begin, A Busson

Conflict graph-based model for IEEE 802.11 networks: A Divide-and-Conquer approach

M. Stojanova[a,*], T. Begin[a], A. Busson[a]

[a] *LIP, ENS Lyon, 46, allée d'Italie, 69364 Lyon, France*

Abstract

WLANs (Wireless Local Area Networks) based on the IEEE 802.11 standard have become ubiquitous in our daily lives. We typically augment the number of APs (Access Points) within a WLAN to extend its coverage and transmission capacity. This leads to network densification, which in turn demands some form of coordination between APs so as to avoid potential misconfigurations. In this paper, we describe a performance modeling method that can provide guidance for configuring WLANs and be used as a decision-support tool by a network architect or as an algorithm embedded within a WLAN controller. The proposed approach estimates the attained throughput of each AP, as a function of the WLAN's conflict graph, the AP loads, the frame sizes, and the link transmission rates. Our modeling approach employs a Divide-and-Conquer strategy which breaks down the original problem into multiple sub-problems, whose solutions are then combined to provide the solution to the original problem. We conducted extensive simulation experiments using the ns-3 simulator that show the model's accuracy is generally good with relative errors typically less than 10%. We then explore two issues of WLAN configuration: choosing a channel allocation for the APs and enabling frame aggregation on APs.

Keywords: WLAN, IEEE 802.11, Performance, Throughput, Conflict graph, Markovian, Divide-and-Conquer.

1. Introduction

WLANs (Wireless Local Area Networks) have become ubiquitous and part of our daily lives. They are frequently offered in public places such as cafes, restaurants, hotels, shopping malls, museums, metro and train stations, airports, and often available in places like trains, planes, workplaces, domestic houses, educational institutions, etc. As for the devices connected to an Access Point

*Corresponding author

Email addresses: marija.stojanova@ens-lyon.fr (M. Stojanova), thomas.begin@ens-lyon.fr (T. Begin), anthony.busson@ens-lyon.fr (A. Busson)

(AP) of a WLAN, their variety has greatly expanded, comprising desktop and laptop computers, IP phones, smartphones, digital media players, etc.

WLANs are typically based on the IEEE 802.11 standard [1]. In order to meet the increasing needs of WLAN users, IEEE 802.11 has undergone several amendments, mostly aimed at strengthening its performance and security. In particular MAC (Medium Access Control) and PHY (Physical) functions have been enhanced. Indeed, transmission technologies, defining the PHY layer of IEEE 802.11, have significantly evolved over the years using e.g., wider channels, higher-order modulations, multiple-input multiple-output antennas (MIMO). Maybe to a lesser extent, the MAC layer has also undergone some transformations with the possibility of using the Request to Send / Clear to Send mechanism (RTS/CTS), smaller mandatory waiting periods before transmissions, as well as frame aggregation and block acknowledgment in the latest amendments of IEEE 802.11.

In order to extend the coverage and the available transmission capacity of WLANs, network architects may augment the number of APs within a WLAN. This network densification comes with a growing complexity in the WLAN management. Indeed, a WLAN with several APs requires some form of coordination between its APs so as to avoid potential misconfigurations that could lead to an inefficient use of radio resources, poor performance and/or unfairness between users. For instance, coordination efforts can pertain to the selection of a radio channel for each AP (for mitigating interferences from neighboring APs) as well as to the association of user devices with the APs (for balancing the load among APs). Some proprietary and commercial solutions implement such mechanisms. Among others, CAPWAP and 802.11v protocols, issued by IETF and IEEE respectively, enable APs (within the same WLAN) to exchange information about the network topology and radio environment to a central controller. However, the algorithms run by the controller and exploiting this knowledge are yet to be designed. Indeed, unlike PHY and MAC layers, coordinating the APs of a WLAN has attracted little attention so far.

In this paper, we describe a performance modeling method that can provide guidance for configuring an IEEE 802.11-based WLAN composed of multiple APs. The method can be used as a decision-support tool by a network architect or as an algorithm embedded within a WLAN controller. The proposed approach offers estimates of the attained throughput of each AP. These estimates are obtained in return for a WLAN description including its conflict graph, the AP loads, the frame sizes, and the link transmission rates. Our modeling approach employs a Divide-and-Conquer strategy in which we break down the complexity of the original problem by considering multiple sub-problems, whose solutions are then combined to provide the solution to the original problem. The proposed solution is conceptually simple, easily implementable, and can be fully automated. We conducted extensive simulation experiments using the ns-3 simulator to evaluate the accuracy of our solution. Numerical results show that its accuracy is generally good with relative errors typically less than 10%.

The remainder of the paper is organized as follows. Section 2 provides a review of existing related works. In Section 3, we describe the considered system

and the performance metrics of interest. Our modeling framework for IEEE 802.11-based WLANs is detailed in Section 4 and shown in an algorithmic form in Section 5. In Section 6, we present the numerical evaluation of our model as well as two possible applications to the sizing and tuning of a WLAN. Section 7 concludes this paper.

2. State of the Art

The different models that evaluate the performance of IEEE 802.11-based WLANs range over a wide spectrum of levels of abstraction. Bianchi [2] as well as Cali, Conti, and Gregori [3] model the network at a very fine level of abstraction. Both models take into account the behavior of every single frame transmission. In [3] the authors analyze the ratio of the average frame size and its average transmission time in order to study the utilization of the network's capacity. Bianchi's seminal work [2] introduced a model based on a two dimensional Markov chain. The Markov chain models the backoff process that takes place before every Distributed Coordination Function (DCF) frame transmission while the network is assumed to be fully-connected, i.e., all nodes are neighbors. A property shared by both models is that the networks they consider are saturated, meaning all nodes constantly have frames waiting to be sent.

Because the saturation assumption can be deemed too restrictive in some cases, many subsequent works are centered on relaxing it. Kosek-Szott [4] as well as Gupta and Rai [5] circumvent this barrier by adding one more state to the Markov chain proposed by Bianchi [2]. This new state represents a node that has no frames to be sent. Note that both works deal only with fully-connected networks.

Another solution is proposed by Felemban and Ekici [6], who have removed the condition of saturation by introducing the probability that a node has a frame waiting to be sent. They do so by creating a second Markov chain, embedded into Bianchi's original Markov chain. The embedded chain describes the current state of the channel, which can be either idle, in collision, or in successful transmission. The solution to their model is found by successively iterating between the two chains. Upon convergence, the found solution delivers the steady state transmission probability for each node, which can then be used to evaluate the network's performance. However, like Bianchi's original model [2], the focus of this work is restricted to fully-connected networks.

Aside from performance evaluation, a fine-level modeling can also help refine networking protocols in order to improve the overall network performance. To achieve this goal, the authors of [7] introduce the so-called Optimal DCF (O-DCF). O-DCF adapts a node's MAC parameters, such as the backoff period and transmission length, in order to improve the network's utilization or fairness. The adaptation depends on the current length of a node's MAC queue and tends to favor nodes that have more queue buildup. Even though the adaptation is calculated in a distributed manner for every node, an estimation of global performance metrics is required. This requirement is removed in [8] where Fitzgerald,

Körner, and Landfeldt introduce the Throughput Optimal DCF (TO-DCF) that needs only local measurements. TO-DCF also favors nodes with larger queue buildup, but it uses node weights to express the different priorities. Nodes with higher weights decrement their backoffs faster, which gives them a higher transmission priority. While O-DCF can be directly implemented in an existing 802.11 chipset as a driver update, TO-DCF modifies the DCF procedure and represents a new DCF-based protocol.

To overcome the inherent complexity tied to a fine level of abstraction when the network grows in size, other works have developed modeling approaches that incorporate both a fine-level and a high-level of abstraction. Two such models are given in [9] and [10]. Both models analyze non-saturated multi-hop networks. In a multi-hop network, a packet from node A travels across relay nodes before arriving at its destination node B (as opposed to single-hop networks, where A and B directly exchange packets). Both papers present two-level modeling approaches of unsaturated multi-hop wireless networks, in which the low-level model is a version of Bianchi's original Markov chain, while the high-level model aims at capturing the inter-node dependencies in the network. The solution to the overall model is found using a fixed-point iteration between the high and low level. In [10], the high-level model consists of a set of $M/M/1/K$ queues, where each queue represents a given node of the network. Although their modeling framework was designed to handle any number of nodes, examples shown in their paper involve multi-hop wireless paths with at most 4 nodes. In [9], the high-level model is a separate Markov chain describing the channel's behavior depending on the current states of neighboring nodes, with nodes being either idle, transmitting, or in backoff. Because of the three possible states for each node and the added complexity brought by multi-hop networks, the analytical model of [9] leads to a large state space as the number of nodes increases, making it intractable for networks with more than 7 or 8 nodes, for which a decomposition into smaller networks is necessary.

Finally, at the other extreme of the spectrum, there are the modeling approaches that analyze the network from a high level of abstraction. These models do not take into account the behavior of every frame transmission, and instead, deal with the behavior of the entire network as a whole. In [11, 12, 13], Markov chains are used to model a network based on its topology. The states of the chain describe the set of nodes that are transmitting in the current network state. Nardeli and Knightly [11] rely on their proposed Markov chain to derive a model that takes into account the errors due to collisions and hidden terminals for a single-hop network. The authors derive a closed-form multi-parameter expression of throughput, which is subsequently used for evaluating the performance of the considered network. Although the model accurately captures the behavior of CSMA/CA networks, it only deals with saturated networks and introduces some complexity due to the calculation of successful transmission probability. In [12], a similar Markov chain is used to evaluate the fairness and spatial reuse in multi-hop, saturated networks with different carrier sensing and reception zones. More particularly, the authors study the spatial reuse in line-networks to show that CSMA/CA achieves maximal spatial reutilization

as traffic intensity increases, at the cost of creating starvation in certain links. In [14, 15] CSMA/CA networks are modeled as continuous time Markov chains and the model is then used to study the fairness of the network. Jiang and Walrand [13] extend the usage of this model by proposing an adaptive solution that changes the nodes' backoff periods in the goal of maximizing the network's throughput and utilization.

A significant number of models have been developed for specific network topologies, such as chain networks. Chaudet et al. [16] study the behavior of the three-node chain network known as the Flow In the Middle (FIM) topology of Fig. 1. FIM networks are well-known for exhibiting high levels of inequality because, when placed in saturation, the edge nodes 1 and 3 experience a high throughput while the middle node is in starvation. The authors model the network as a Markov chain in which every state contains, among other values, the idle time experienced by the middle node. They show that as this idle time increases, the inequality of channel access decreases, i.e., the middle node gets more channel access. They conclude that shorter transmission times favor equality at the expense of utility. Recently, Ducourthial et al. [17] developed a model that is also based on the idle time experienced by a node. However, it can be used on chain networks of arbitrary lengths. They use a set of equations where the variables describe the transmission probability of every node in the network. They show that chains with an even number of nodes manifest more equality, and that for very large chains the inequality of channel access vanishes around the 15th node. The authors prove analytically and in simulation that modifying either the transmission rate or the frame length of the edge nodes can drastically increase the fairness in the network.

A novel approach in the modeling of non-saturated networks is introduced in [18] and [19], where the authors have chosen to map the idle time of a node to a longer backoff period. This approach keeps the simplicity of a saturated network model by not explicitly representing idle states, and yet allows the study of unsaturated nodes. Kai and Zhang [18] propose a model that calculates the throughput of non-saturated CSMA/CA networks with arbitrary topologies. Laufer and Kleinrock [19] use a similar model to estimate the throughput of a node in a fully-connected CSMA/CA network using the ratio between the transmitting and the backoff periods of that node, its probability of successful transmission, and the channel capacity. The result is then used in the analysis of a network's capacity region, based on nodes' throughputs, under stability conditions. Bonald and Feuillet [20] also characterize both the capacity region and the stability of a wireless network. However, their work focuses on multi-channel networks in either ad-hoc or infrastructure mode, and they propose a refinement to CSMA to achieve a more efficient and fair access to the channel in the infrastructure mode.

In this paper, we study unsaturated, not fully-connected IEEE 802.11 wireless networks. We present a conflict graph-based modeling approach to discover the attainable throughput of each node. We apply a Divide-and-Conquer approach resulting in a series of Markov chains that together describe, at a high-level of abstraction, the current state of the entire wireless network. The

conceptual simplicity of our model allowed us to fully automatize the procedure and to test it on networks of different sizes and topologies.

3. System Description

The system we consider is a wireless local area network (WLAN). In this paper, WLAN refers to any wireless network that implements the IEEE 802.11 standard in the Physical (PHY) and Medium Access Control (MAC) layers. IEEE 802.11 standards are accompanied by a series of amendments. Each amendment serves as an addition to the IEEE 802.11 standard and is developed to either modify the standard's PHY and MAC characteristics or offer additional functionalities not implemented in the basic standard. Examples of these are the IEEE 802.11g that enhances the physical layer of the standard (and preceding amendments), and the IEEE 802.11i which offers additional security features.

In terms of the physical layer, every IEEE 802.11 standard amendment has a set of *transmission rates* that represent the physical rate at which a node can send data over the channel. The wireless channel is generally imperfect and highly affected by its environment. Therefore, when a transmission rate is chosen for a communication between two nodes the goal is to have the highest possible transmission rate while keeping a low error rate [21]. A node can also choose which wireless channel it wishes to use. IEEE 802.11 standard amendments generally use two distinct frequency bands: the 2.4GHz and the 5GHz [22]. In the 2.4GHz band, a node can choose from up to 14 wireless channels. Out of those 14 channels, only three occupy non-overlapping frequencies i.e., can be used simultaneously without collisions. In the 5GHz band, there can be up to 24 non-overlapping channels, meaning that as many as 24 transmissions can happen simultaneously in close proximity.

WLANs use the Distributed Coordination Function DCF in the MAC layer. DCF makes sure that a node that wishes to start a transmission senses an idle medium before the beginning of that transmission. This procedure is employed with the help of the Carrier-Sense Multiple Access with Collision Avoidance (CSMA/CA) mechanism in two distinct steps. First, before every transmission a node must sense the medium idle for the duration of a DIFS period whose length depends on the IEEE 802.11 standard amendment. Then, the node starts a countdown timer called the *backoff* period. The backoff has a random duration that helps desynchronize the beginnings of transmissions of neighboring nodes. Nevertheless, its average duration depends on the IEEE 802.11 amendment in use and we denote it by $T_{backoff}$. Additionally, for unicast frames only, DCF uses a short silent period called SIFS followed by an acknowledgment frame sent by the destination, serving as a confirmation that a frame was correctly received. In the interest of brevity, we postpone additional explanations of DCF to Appendix A.

Each WLAN is composed of nodes representing the Access Points (APs) and the user stations. A node's *Carrier-Sensing* (CS) zone contains all other

nodes whose transmissions can be detected. DCF, through its carrier sensing and backoff mechanisms, attempts to ensure that two nodes that belong to each others' CS zones do not simultaneously transmit, and instead, have to share the available transmission capacity. Nevertheless, collisions occur whenever two such nodes are simultaneously transmitting, potentially resulting in the loss of one or both transmissions. More precisely, a collision happens whenever the backoff countdowns of the two nodes finish (approximately) at the same time. CS zones play an important role in evaluating the WLAN's global transmission capacity as they determine with whom and how nodes have to share the medium/channel capacity. Therefore, we can use the CS zones to represent a WLAN as a *conflict graph* in which two vertices share an edge if the corresponding network nodes belong to each others' CS zones. Although nodes can interfere even beyond their CS zones, Padhye *et al.* [23] showed that the major source of collisions (interference) are nodes that belong to the CS zone. Note that the same authors also developed a pairwise interference measurement that can be used to discover a WLAN's conflict graph.

We refer to nodes that share a link in the conflict graph as *neighbors*. We denote by v_n the set of n's neighboring nodes (excluding n), referred to as n's neighborhood. In our work, we assume that the CS zones are symmetrical, meaning that if node m belongs to node n's CS zone, then node n belongs to node m's CS zone. Figure 1 shows the conflict graph of a three-node network, known as the Flow In the Middle (FIM) topology. We notice that in this network nodes 1 and 3 can simultaneously transmit as they do not detect each others' transmissions. On the other hand, nodes 1 and 2 cannot transmit at the same time without causing a collision resulting in a potential loss of frame(s).

Note that, like in [24], only the network's APs are considered in the conflict graphs (user stations are not). Because of the traffic asymmetry where downloads from APs vastly outweigh uploads [25], the set of APs provides a convenient, though approximate, description of a WLAN. We tested the validity of this assumption through simulations. Our results suggest that user stations can be disregarded in the conflict graph at the cost of a limited loss of accuracy.

For example, Fig. 1 may describe a network in which three APs are transmitting traffic to some user stations, which are mostly receiving traffic rather than generating traffic so that they can be disregarded in the graph. With regard to the wireless channels, all the nodes that belong to the same channel belong to the same conflict graph as well. As a result, an average WLAN working in the 2.4GHz band (with three non-overlapping channels) can be represented by three conflict graphs, each containing roughly a third of the total number of APs in the WLAN. With a maximal number of 24 non-overlapping channels in the 5GHz band, the corresponding conflict graphs would be even smaller.

We use x_n to denote the normalized *input rate* of node n. The higher x_n, the larger the demand of node n for throughput. On the other hand, we let y_n indicate the normalized *output rate* of node n. By definition, we have: $x_n \leq 1$ and $y_n \leq 1$, and, because in the long run the output rate of a node cannot exceed its input rate, it follows: $y_n \leq x_n$. Note that $y_n = 0$ indicates that node n never

gets access to the channel (i.e., a starving node), while $y_n = 1$ signifies that the node permanently occupies the channel (be it in active transmission or with DCF overhead). Note also that y_n can be easily derived from the actual average throughput of a node (typically expressed in Mbps) by simply normalizing the latter by the maximum throughput achievable by the node (i.e., when all its neighbor nodes are silent). More precisely, we have:

$$y_n = \frac{t_n}{t_{n,max}} , \tag{1}$$

where t_n denotes the throughput achieved by node n, and $t_{n,max}$ is the maximum throughput node n can achieve, calculated as:

$$t_{n,max} = \frac{L \times 8}{T_{backoff} + T_{DIFS} + T_{PHY} + T_{FRAME} + T_{SIFS} + T_{PHY} + T_{ACK}} . \tag{2}$$

In Eq. (2) T_{PHY} is the duration of the physical layer's header, L is the mean payload length, T_{FRAME} is the total frame transmission time, including all headers brought by the MAC, Network, and Transport layers, $T_{backoff}$, T_{DIFS}, T_{SIFS}, T_{ACK} are all overhead times present in DCF. Note that the maximum throughput depends on the standard amendment, mean payload length, and transmission rate, while being independent of the network's topology or nodes' input rates.

Figure 1: Conflict graph of a three-node network known as the Flow In the Middle (FIM) topology.

We now illustrate through an example how the nodes' output rates, y_n, are influenced by parameters such as the mean frame length, the transmission rate over the channel as well as internal parameters of each IEEE 802.11 amendment. Let us consider the FIM network (see Fig. 1) and assume that all nodes are saturated, i.e., $x_1 = x_2 = x_3 = 1$. Clearly, the middle node is in an unenviable position as it competes for medium access with two neighbors that are mutually independent. More generally, Durvy et al. [12] have shown that CSMA/CA tends to maximize the number of concurrent transmissions in the network at the cost of creating starvation in some nodes. However, the severity of the starvation largely depends upon the length of the backoff period relative to the length of frame transmissions. In fact, the shorter the frame transmissions, the more chances node 2 has to access the channel (for more details, see [16, 17]). To capture this property, we introduce a *backoff factor*, denoted by α, that is defined as the ratio between the average backoff period duration and the duration of a full frame transmission:

$$\alpha = \frac{T_{backoff}}{T_{DIFS} + T_{PHY} + T_{FRAME} + T_{SIFS} + T_{PHY} + T_{ACK}} . \tag{3}$$

As shown in Appendix B, the output rate of node 2 varies significantly with the value of α, mostly in a quadratic manner. Although the exact values of this quadratic pattern will vary from one network to another, it is our experience that for growing values of α the pattern tends to also occur in other network topologies.

System	
N	total number of nodes
v_n	node n's neighborhood
x_n	input rate of n-th node, $x_n \in [0,1]$
y_n	output rate of n-th node, $y_n \in [0,1]$
α	backoff factor
R	transmission rate (in Mbps)
L	mean payload length (in bytes)
H	amount of headers brought by the MAC, Network, and Transport layers (in bytes)
$t_{n,max}$	maximal throughput node n can achieve if all its neighbors are silent (in Mbps)
t_n	achieved throughput of node n (in Mbps)
Model	
B	set of possible subnetworks, $B = \{1,0\}^N$
b_i	ith subnetwork, $b_i \in B$
$b_i(n)$	regime of the nth node in subnetwork b_i, $b_i(n) \in \{ON, OFF\}$
β_i	occurrence probability of the ith subnetwork
S	set of possible sending states, $S \subseteq \{\boldsymbol{0,1}\}^N$
s_k	kth sending state, $s_k \in S$
$s_k(n)$	state of the nth node in sending state s_k, $s_k(n) \in \{\boldsymbol{1,0}\}$
S_i	set of sending states associated to subnetwork b_i, $S_i \subseteq S$
c_i^m	mth irreducible Markov chain of subnetwork b_i
S_i^m	set of sending states associated to c_i^m, $S_i^m \subseteq S_i$
$\sigma_i(k)$	probability that sending state s_k of subnetwork b_i is initially chosen
$P_{k,\ell}$	probability of the transition from sending state s_k to s_ℓ
w_n	restricted set of neighbors of node n with blocked nodes removed
M_i	number of irreducible Markov chains for the subnetwork b_i
π_i^m	steady-state probability distribution of c_i^m
π_i	steady-state probability distribution of subnetwork b_i
ω_i^m	occurrence probability of c_i^m
D_i	set of dominant chains in subnetwork b_i
d_i	set of dominated chains in subnetwork b_i
Q_n	set of cliques that contain node n
q_j	jth clique

Table 1: Principal notation.

4. Model and its Solution

For the sake of clarity, when presenting our modeling framework, we resort to the sample network depicted in Fig. 2 to show its step by step execution.

4.1. Decomposing into subnetworks

In any network, nodes typically alternate their activity between ON and OFF periods. When in the ON regime, a given node n has at least one frame

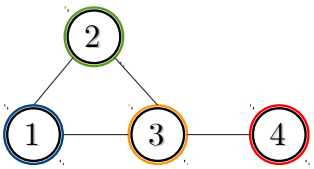


Figure 2: Conflict graph of a four-node network.

to send, and thus has a non-empty buffer. In other words, an ON node is either transmitting or willing to start a transmission. In the OFF regime, a node's buffer is empty. We consider that the nodes' regimes, and consequently their input rates, are independent of each other. In practice, a node may postpone the transmission of a frame because of the activity of its neighbors, thus extending its ON period. In order to keep the model tractable, we decided to omit the potential dependencies among the nodes' ON periods.

At any time, the state of the network activity can be described by a vector of length N, where N is the number of nodes in the network and the nth element expresses the current regime of node n (be it ON or OFF). Thus, for a network with N nodes, there are 2^N such vectors that correspond to all the possible combinations of the two regimes over the N nodes.

In our work, we apply a Divide-and-Conquer approach by choosing to analyze the WLAN not as a single complex network in which any node can alternate between ON and OFF, but rather as a collection of 2^N simpler networks in which every node is either ON or OFF. We refer to these new networks as the *subnetworks* and we denote them by $b_1, b_2, ..., b_{2^N}$. Hence $b_i(n)$ indicates the regime of node n in subnetwork b_i. We use B to designate the set that contains all subnetworks.

For the sample network of Fig. 2, as well as for any other four-node network, there is a total of 16 such subnetworks:

$$B = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{16} \end{bmatrix} = \begin{bmatrix} OFF & OFF & OFF & OFF \\ OFF & OFF & OFF & ON \\ \vdots & \vdots & \vdots & \vdots \\ ON & ON & ON & ON \end{bmatrix} \tag{4}$$

We refer to the probability that the current state of the network is subnetwork b_i as the *occurrence probability* of b_i and we denote it by β_i ($i = 1, \ldots, 2^N$). Note that a subnetwork's occurrence probability depends only on the nodes' input rates and can be calculated as:

$$\beta_i = \prod_{n|b_i(n)=ON} x_n \prod_{m|b_i(m)=OFF} (1 - x_m) \ . \tag{5}$$

For example, in our four-node network, one of the possible subnetworks is $b_{14} = [ON\ ON\ OFF\ ON]$. This subnetwork represents the case when nodes 1,

2, and 4 are in transmission or have a frame waiting to be sent, while node 3 has an empty buffer. Its occurrence probability is calculated as:

$$\beta_{14} = x_1 x_2 (1 - x_3) x_4. \tag{6}$$

Figure 3 shows a schematic representation of the entire solution where Stage 1 corresponds to breaking down the network into several subnetworks. We will

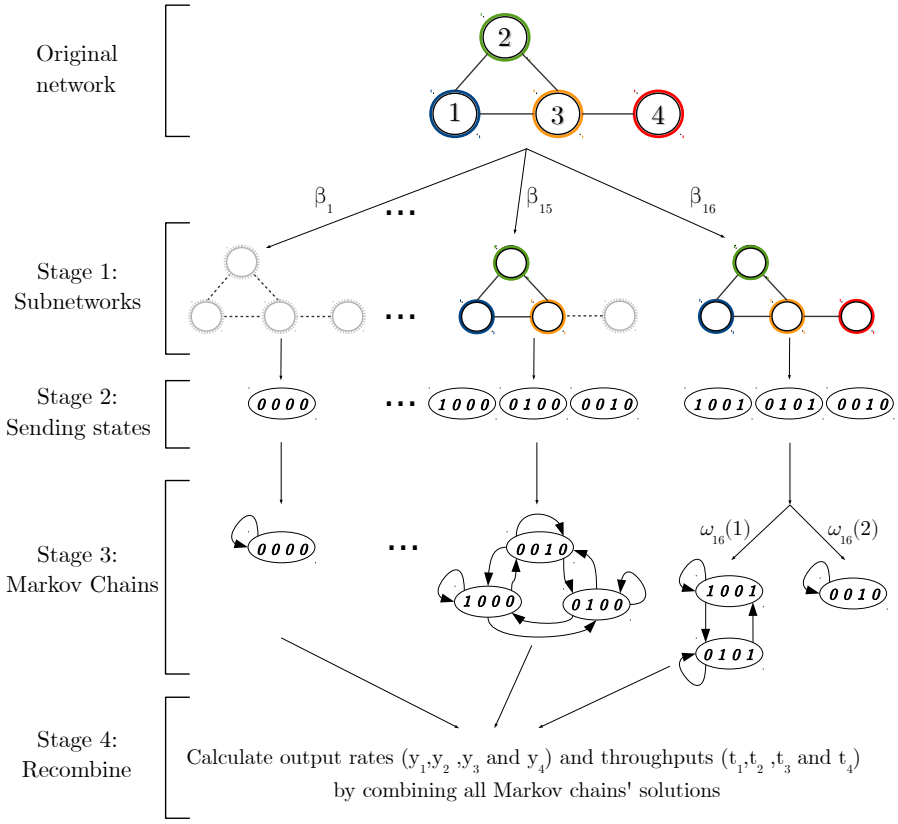


Figure 3: Schematic representation of the proposed solution.

now show how to solve each of the subnetworks separately and independently of the rest of the subnetworks.

4.2. Solving each subnetwork as one or more Markov chain(s)

We now detail how we analyze the behavior of each subnetwork using Markov chains. We start by defining the possible states and transitions of the corresponding Markov chains. Note that, in this subsection, the subject of study is any of the subnetworks b_i $(i = 1, \ldots, 2^N)$ resulting from the network decomposition (see Section 4.1).

4.2.1. Defining the possible states for the subnetwork

While the regime (ON or OFF) of each node is set and fixed (for the considered subnetwork b_i), knowing the regime is not sufficient to determine if a node is currently sending a frame or not. Indeed, an ON node can be either transmitting or waiting for the medium to become idle. We eliminate this ambiguity by introducing the notion of sending states.

Like a subnetwork, *a sending state* is a vector of length N whose nth element refers to the activity of the nth node. However, unlike a subnetwork, a sending state indicates for each node n if the node is transmitting (marked $\boldsymbol{1}$) or not (marked $\boldsymbol{0}$). Let s_k denote the kth sending state (with $k = 1, \ldots,$). Thus, if node n is currently transmitting we have $s_k(n) = \boldsymbol{1}$, and $s_k(n) = \boldsymbol{0}$ otherwise, for $n = 1, \ldots, N$. Note that $s_k(n) = \boldsymbol{0}$ means that node n is either OFF, or ON but waiting access for transmission. While in theory, the total number of sending states for each subnetwork is equal to 2^N, in practice this number is much smaller as we consider only a fraction of them to be possible. Let S denote the set of all possible sending states over all existing subnetworks. Each possible sending state must comply with a common property of CSMA/CA: neighboring nodes cannot transmit successfully at the same time, i.e., if the conflict graph contains an edge between nodes n and $n + 1$, then $s_k(n)$ and $s_k(n + 1)$ cannot both be equal to $\boldsymbol{1}$. Next, we designate by S_i the set of possible sending states associated to the subnetwork b_i. Note that we can easily determine S_i since S_i is a subset of S whose elements satisfy the following properties: (i) if $b_i(n) = ON$ and node n has no transmitting neighbors, then $s_k(n) = \boldsymbol{1}$; (ii) if $b_i(n) = OFF$, then $s_k(n) = \boldsymbol{0}$. Note that the rationale behind the second property is quite straightforward: a node that has no frames to be sent cannot be sending. The first property is derived from a phenomenon studied in [12]: CSMA/CA networks tend to increase the spacial reutilization of the medium by maximizing the number of simultaneous transmissions. As a result, in our model, we enforce any node that is ON and senses an idle medium to be in transmission.

In the case of our sample network, the subnetwork $b_{16} = [ON\ ON\ ON\ ON]$ has three possible sending states, $s_1 = [\boldsymbol{1\ 0\ 0\ 1}]$, $s_2 = [\boldsymbol{0\ 1\ 0\ 1}]$, and $s_3 = [\boldsymbol{0\ 0\ 1\ 0}]$. Note that other sending states may exist but we consider them to be negligible in b_{16}. For example, the sending state $[\boldsymbol{1\ 1\ 0\ 1}]$ breaks the CSMA/CA condition, as nodes 1 and 2 are neighbors and cannot be simultaneously transmitting. The sending state $[\boldsymbol{1\ 0\ 0\ 0}]$ is deemed not possible since node 4 breaks the first condition. Indeed, b_{16} indicates that node 4 is ON, and because it has no sending neighbors, it should be sending its frames.

This step of determining the sending state is illustrated by Stage 2 of Fig. 3.

4.2.2. Determining the possible transitions

The set of sending states found for the subnetwork b_i, namely S_i, will serve as the Markov chain's set of states. We now detail how we decide which transitions are possible between those sending states. Our reasoning is based on the idea that, in a CSMA/CA network, the probability of two nodes starting (or ending) their transmission at the exact same time is negligibly small. We translate this

CSMA/CA property into the following rule for our modeling purpose. Let s_k and s_ℓ be two possible sending states of S_i. The transition from sending state s_k to s_ℓ is deemed possible if and only if s_k and s_ℓ both verify that:

1. no more than one node alters from ***1*** in s_k to ***0*** in s_ℓ, and
2. no more than one node alters from ***0*** in s_k to ***1*** in s_ℓ.

Note that a self-transition on a given sending state s_k is always possible, as it implies no changes in the sending state.

For example, in our four-node network it is possible to go from sending state [***1 0 0 1***] to [***0 1 0 1***], as in this transition node 1 ends and node 2 starts a transmission. However, it is not possible to go from network state [***1 0 0 1***] to [***0 0 1 0***], as it implies both nodes 1 and 4 ending their transmissions at the exact same time. Figure 4 shows the existing transitions in our modeling framework between the possible sending states associated to the subnetwork b_{16}.

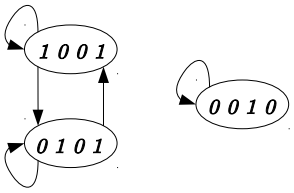


Figure 4: Possible sending states and corresponding existing transitions associated to the subnetwork $b_{16} = [ON\ ON\ ON\ ON]$.

4.2.3. Calculating the transition probabilities

We now explain how we determine the probability of the transitions between the possible sending states s_k composing our Markov chain. Note that impossible transitions have zero probability. To evaluate the non-zero transition probabilities, we need to introduce our definition of a *blocked node*. A node having at least one of its neighbors currently transmitting is said to be blocked as it is unable to start a collision-free transmission. For example, in the four-node network, node 3 can be blocked by the transmissions of any of the other three nodes.

We can now calculate $P_{k,\ell}$, the probability of the transition from sending state s_k to s_ℓ, as:

$$P_{k,\ell} = C \prod_{n|s_\ell(n)=\mathbf{1}} \frac{1}{1 + \sum_{m \in w_n} \mathbb{1}_{b_i(m)=ON}}, \tag{7}$$

where C is a normalizing constant such that $\sum_{\ell \geq 1} P_{k,\ell} = 1$, and w_n defined as $w_n = \{m \in v_n \backslash \{n\} \mid m$ is not blocked in s_ℓ by a node $\in v_m / \{n\}\}$ is the restricted neighborhood of node n, i.e., w_n contains all neighbors of n that are not blocked by some node different from node n. As an example, in the subnetwork $b_{16} = [ON\ ON\ ON\ ON]$ and the sending state [***1 0 0 1***], the

restricted neighborhood of node 1 contains only node 2, as node 3 is blocked by node 4.

Note that the indicator function $\mathbb{1}_{b_i(m)=ON}$ returns 1 if $b_i(m) = ON$, and 0 otherwise. The underlying logic behind Eq. (7) is that all nodes that are ON (whether they are sending or not) compete with their neighbors for accessing the medium. We also consider them equally likely to gain the medium access. On the other hand, nodes that are OFF do not affect the transition probability because they do not compete for medium access.

For instance, when node 3 of the subnetwork b_{16} in Fig. 3 competes with nodes 1, 2, and 4, it has a $\frac{1}{4}$ chance of gaining the medium. However, in this same scenario node 4 competes with only one neighbor, so its chance of gaining access would be $\frac{1}{2}$.

4.2.4. Calculating the steady-state probabilities

At this stage, the Markov chain associated to subnetwork b_i is fully characterized and we can calculate its steady-state probabilities. Let us remind that a Markov chain is *irreducible* if from any of its states there is a way to reach any other state. Depending on the setting of the subnetwork under study, the corresponding Markov chain may or may not be irreducible. Should the Markov chain not be irreducible, we consider each irreducible Markov chain separately. We denote by M_i the number of irreducible Markov chains in subnetwork b_i. For example, as shown by Fig. 4, the subnetwork $b_{16} = [ON\ ON\ ON\ ON]$ contains two irreducible chains, i.e., $M_{16} = 2$ (since it is not possible to go from the sending state $[\boldsymbol{0\ 0\ 1\ 0}]$ to the other two sending states). We use c_i^m, $m \in [1, ..., M_i]$, to denote the mth irreducible chain of subnetwork b_i. Hence the left-hand chain of Fig. 4 is denoted by c_{16}^1 and the right-hand chain is c_{16}^2.

We compute the steady-state probabilities of each irreducible chain c_i^m for the subnetwork b_i and we denote by π_i^m the vector containing the corresponding values. Note that we use S_i^m to refer to the set of sending states in chain c_i^m (while, as defined previously, S_i denotes the set of possible sending states associated to b_i). If the subnetwork has a single irreducible Markov chain ($M_i = 1$), then it follows that $S_i^1 = S_i$. Thus, the steady-state probabilities of the subnetwork's sending states are equivalent to those of the Markov chain c_i^1 and we have $\pi_i = \pi_i^1$, where π_i is the vector containing the steady-state probabilities of subnetwork b_i. In this case, we can skip subsections 4.2.5 and 4.2.6 and proceed to Section 4.3.

4.2.5. Combining several irreducible Markov Chains

For subnetworks that contain more than one irreducible Markov chain, we need to combine the steady-state probabilities found for each Markov chain into the steady-state probabilities for the whole subnetwork b_i. To do so, our approach consists in evaluating the odds of entering each irreducible chain. Let us denote by $\sigma_i(s_k)$ the probability that the sending state s_k of subnetwork b_i is initially chosen. Clearly, we must have: $\sum_{k=1}^{|S_i|} \sigma_i(s_k) = 1$. Our way to evaluate

$\sigma_i(s_k)$ is to consider all possible direct paths of reaching s_k starting from the empty sending state where all nodes are not transmitting, namely [***0 0 ... 0***], and then to sum their probabilities. For example, in the case of the subnetwork b_{16}, we can itemize two paths leading to $s_1 = [\boldsymbol{1\ 0\ 0\ 1}]$ from [***0 0 0 0***], namely (a) node 1 starts transmitting, followed by node 4, and (b) vice versa. For path (a), the probability that node 1 gains medium access is $\frac{1}{4}$, since a total of 4 nodes are competing for the access. Once node 1 starts its transmission, its neighbors nodes 2 and 3 are blocked. This leaves node 4 alone to compete for the medium meaning that node 4 gains access to the medium with a probability of 1. Thus, the overall probability of path (a) is $\frac{1}{4}$. Turning to path (b), the probability that node 4 is the first to gain access to the medium is $\frac{1}{4}$. Once node 4 transmits, node 3 becomes blocked, while nodes 1 and 2 are still competing. Node 1's chance of transmitting can then be approximated to $\frac{1}{2}$. Therefore the overall probability of path (b) is $\frac{1}{4}.\frac{1}{2} = \frac{1}{8}$. It follows that $\sigma_{16}(\boldsymbol{1\ 0\ 0\ 1}) = \frac{1}{4} + \frac{1}{8} = \frac{3}{8}$. With the same reasoning, we obtain $\sigma_{16}(\boldsymbol{0\ 1\ 0\ 1}) = \frac{3}{8}$ and $\sigma_{16}(\boldsymbol{0\ 0\ 1\ 0}) = \frac{1}{4}$.

Having calculated the probabilities of entering each sending state of b_i, we now introduce a *weighting factor*, denoted by ω_i^m, to express the probability that this particular irreducible Markov chain c_i^m is initially chosen. Keeping in mind that S_i^m denotes the set of sending states in the mth irreducible Markov chain of b_i, we compute ω_i^m as follows:

$$\omega_i^m = \sum_{k|s_k \in S_i^m} \sigma_i(k). \tag{8}$$

In Fig. 5 we show the entry probabilities as well as the weighting factors for the subnetwork $b_{16} = [ON\ ON\ ON\ ON]$ of our sample network of Fig. 2. The values found for the weighting factors are $\omega_{16}^1 = \sigma_{16}(\boldsymbol{1\ 0\ 0\ 1}) + \sigma_{16}(\boldsymbol{0\ 1\ 0\ 1}) = \frac{3}{4}$ and $\omega_{16}^2 = \sigma_{16}(\boldsymbol{0\ 0\ 1\ 0}) = \frac{1}{4}$.

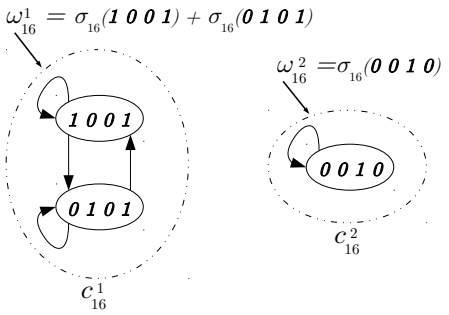


Figure 5: Probabilities of entering each sending state, i.e., $\sigma_i(k)$, and the corresponding weighting factors, i.e., ω_i^m for the subnetwork $b_{16} = [ON\ ON\ ON\ ON]$.

Finally, to calculate π_i (steady-state probabilities of the subnetwork b_i) from π_i^m's (steady-state probabilities found for each of the M_i irreducible Markov chains associated with b_i) we simply proceed as follows:

$$\pi_i = [\pi_i^1 \times \omega_i^1,\ ...,\ \pi_i^{M_i} \times \omega_i^{M_i}]. \tag{9}$$

In other words, π_i is obtained as a weighted sum of π_i^m's.

4.2.6. Adjusting to the IEEE 802.11 parameters

In this section, we refine the computation of the weighting factors ω_i^m to incorporate the value of the backoff factor α (see Eq. (3) in Section 3), which accounts for the mean length of frames sent over the network, the transmission rates of wireless channels, as well as the particular amendment of IEEE 802.11 in use. In networks that contain different backoff factors, for example when not all nodes use the same transmission rate, we simply calculate the average α over all nodes. We have shown that, in the case of an FIM network topology, α correlates highly with the severity of the starvation experienced by the node in the middle (also shown in [16] and [17]). To put it simply, the larger α, the more the nodes experience a fair sharing of the medium. More precisely, in the case of the FIM topology, the output rate of the node in the middle grows approximately quadratically with α (see Fig. B.15 in Appendix B) and we denote by $f(\alpha)$ the corresponding quadratic function found using the least squares method (see Eq. (B.2) also in Appendix B). In fact, it is our experience that the value of α tends to significantly affect the performance of many other network topologies beyond the FIM example. Although we are fully aware that the exact quadratic function of α varies from one network to another, we believe that it will be difficult (if not impossible) to discover a general expression for α that applies equally to all networks. For this reason, we choose to extrapolate the knowledge we obtain on the FIM topology and reuse it on other networks.

We now detail how we incorporate the value found for α into our model by adjusting the weighting factors ω_i^m of the irreducible Markov chains c_i^m. To begin with, we introduce the notion of *dominant* and *dominated* Markov chains. The dominant Markov chains of a given subnetwork b_i are those containing the highest number of transmitting nodes. Conversely, irreducible Markov chains of b_i with a lower number of sending nodes than the dominant chain(s) are called dominated chains. We denote by D_i (resp. d_i) the set of all dominant (resp. dominated) Markov chains for the subnetwork b_i, while $|D_i| \geq 1$ (resp. $|d_i| \geq 0$) refers to the number of dominant (resp. dominated) Markov chains. In our example shown for the subnetwork b_{16} (see Fig. 5), the Markov chain on the left is a dominant chain while the Markov chain on the right is a dominated chain so that we have $D_{16} = \{c_{16}^1\}$, $|D_{16}| = 1$, $d_{16} = \{c_{16}^2\}$ and $|d_{16}| = 1$. To capture the increasing fairness between nodes for growing values of α, we modify the values of the weighting factor of all chains as follows:

$$\widetilde{\omega}_i^m = \begin{cases} \omega_i^m \times f(\alpha), \text{ if } c_i^m \in d_i \ , \\ \frac{1-\Omega_i}{|D_i|}, \text{ if } c_i^m \in D_i \ , \end{cases} \tag{10}$$

where Ω_i the sum of the modified weighting factors of all the dominated irreducible Markov chains in b_i (i.e., $\Omega_i = \sum_{m|c_i^m \in d_i} \widetilde{\omega}_i^m$). It is clear that if $M_i = 1$ then $\omega_i^1 = \widetilde{\omega}_i^1 = 1$, as the subnetwork has a single irreducible Markov chain and it will always be initialized in that chain.

Let us assume that the network of Fig. 5 uses the IEEE 802.11g standard amendment with frames of size $L = 1064$ bytes of which $H = 64$ bytes are headers, and a transmission rate of $R = 54$ Mbps. With these values we obtain an $\alpha = 0.268$ from Eq. (3) and the chains' weighting factors are adjusted as follows: $\widetilde{\omega}_{16}^2 = \omega_{16}^2 \times f(\alpha) = 0.175$ and $\widetilde{\omega}_{16}^1 = 1 - \omega_{16}^2 \times f(\alpha) = 0.825$.

4.3. Combining subnetwork solutions

So far we have divided the network into subnetworks and solved each one of them separately. The last phase of the model consists in combining the results obtained for different subnetworks and calculating the nodes' output rates. A node's output rate represents the portion of time when the node is occupying the medium, including the frame transmission itself and all the necessary DCF overhead. Thus, we calculate node n's output rate, y_n, as:

$$y_n = \sum_{i=1}^{|B|} \left\{ \mathbb{1}_{b_i(n)=ON} \times \beta_i \times \sum_{m=1}^{M_i} \left(\widetilde{\omega}_i^m \times \sum_{k|s_k \in S_i^m} \left(\mathbb{1}_{s_k(n)=\mathbf{1}} \times \pi_i^m(k) \right) \right) \right\}, \tag{11}$$

Equation (11) gives the sum of the stationary probabilities of all the sending states in which node n is sending, times the occurrence probabilities of all the irreducible Markov chains in which those states appear, times the occurrence probabilities of the subnetwork to which those chains belong. Otherwise stated, it is simply the sum product of the probabilities of all the *subnetwork* $\times$ *Markov chain* $\times$ *sending state* combinations in which node n is sending. Note that because of the nodes' input rates, every subnetwork occurs with a given occurrence probability calculated in Eq. (5). Then for every subnetwork, Eq. (11) involves the steady-state probabilities of all its irreducible Markov chains. The normalization of the steady-state probabilities is ensured by the β_i and $\widetilde{\omega}_i^m$ terms.

We can now transform node n's output rate into its obtained throughput, t_n. When all the network nodes use the same standard amendment, transmission rate, and mean payload length, then all nodes have an equal maximum achievable throughput, $t_{n,max}$, and we can calculate the throughput of node n as:

$$t_n = y_n \times t_{n,max} \ . \tag{12}$$

However, when these parameters vary per node, nodes will have different maximum achievable throughputs, $t_{n,max}$, resulting in different node throughputs, t_n. First, we calculate the maximum achievable throughput of every node using Eq. (2). Next, we consider all the maximal cliques of the network and estimate their global clique throughput. We denote by q_j ($j \geq 1$), the jth maximal clique of the network, and by Q_n ($n \geq 1$) the set of maximal cliques that contain the node n. For example, our four-node network in Fig. 2 contains two maximal cliques, the first one containing nodes 1, 2, and 3, denoted q_1, and the second containing nodes 3 and 4, denoted q_2. Node 3 is the only node that belongs to both cliques, thus $Q_3 = \{q_1, q_2\}$. We calculate the global throughput

of clique q_j, t_{q_j} as:

$$t_{q_j} = \frac{\sum\limits_{m \in q_j} y_m \times L_m}{\sum\limits_{m \in q_j} y_m \times \frac{L_m}{t_{m,max}}} \ , \tag{13}$$

where L_m denotes the mean payload length of node m.

The reasoning behind Eq. (13) is that every node m that belongs to the clique q_j will gain a portion of the medium access that is proportional to its output rate, y_m. If node m belongs to several cliques, we calculate the average throughput over all those cliques. Finally, node m's throughput is calculated as the product of its output rate and its average clique throughput:

$$t_n = y_n \times \frac{\sum\limits_{j | q_j \in Q_n} t_{q_j}}{|Q_n|} \ . \tag{14}$$

In the following section we provide a summarized version of the proposed model in an algorithmic form.

5. Algorithm

Decomposing into subnetworks:
  Find the set of all possible subnetworks B as given in Eq. (4) of Section 4.1.
  Calculate the occurrence probability β_i of each subnetwork b_i of B as described in Eq. (5).
for all $b_i \in B$ **do**
  States:
    Create the set S_i, $S_i \subseteq S$, containing all the possible sending states for the subnetwork b_i using the method described in Section 4.2.1.
  Transitions:
    Determine the possible transitions between the sending states with the method described in Section 4.2.2.
  Transition probabilities:
    Calculate the probability of each transition:
      (i) non-possible transitions have a zero probability,
      (ii) possible transitions have a non-zero probability calculated with Eq. (7) of Section 4.2.3.
  Steady-state probabilities:
    Create the Markov chain of b_i with the states of step 6, the transitions of step 8, and the transition probabilities of step 12. If the Markov chain is not irreducible, divide it into its M_i corresponding irreducible Markov chains, otherwise $M_i = 1$.
    Calculate the steady state probability for each of the M_i chains.
  Weighting factors:
    Calculate the weighting factor for all the M_i Markov chains using Eq. (8) of Section 4.2.5.
  Adjusting to the IEEE 802.11 parameters:
    Calculate the backoff factor α using Eq. (3) of Section 3.
    Determine the set of dominant and dominated chains, D_i and d_i, respectively, as described in Section 4.2.6.
    Calculate the modified weighting factors using Eq. (10).
end
Combining network solutions:
  Combine the solutions of the different subnetworks in order to calculate the output rate y_n of node n using Eq. (11) of Section 4.3.
  Calculate the throughput t_n of node n using Eq. (12) or (14).

Algorithm 1: Complete algorithm.

6. Numerical Results

We start this section by assessing the accuracy of the proposed modeling approach by comparing the model outcomes with those delivered by a discrete-event simulator under various scenarios. Then, we study the computational

complexity of the modeling approach as a function of the network topology. At last, we explore two possible applications of the model relating to the configuration and performance improvement of a WLAN.

6.1. Model validation

To evaluate the accuracy of the proposed model, we explore several scenarios with different values of various network parameters, such as the IEEE 802.11 standard, the mean frame length, the transmission rate, the topology and size of the network, and we compare the model's estimations to the simulation results delivered by the discrete-event network simulator ns-3 [26].

6.1.1. Various network topologies and standard amendments

We begin by examining the proposed model's accuracy under different topologies and IEEE standard amendments. We consider three topologies: the four-node network of Fig. 2, the larger six-node network depicted in Fig. 7, and the ten-node network of Fig. 9. Recall that the nodes of a conflict graph represent only the access points (APs) that belong to the same communication channel. Typically, the original WLAN contains several other APs operating on other channels. Besides, as discussed in Section 3, only APs (and not user stations receiving traffic from APs) appear in conflict graphs. Nonetheless, in the simulator, each AP transmits traffic to an associated user station. The four-node network uses the IEEE 802.11g standard amendment while the six-node and the ten-node networks use the IEEE 802.11n standard amendment.

Table 2 sums up the parameters used in our scenarios as well as the simulation parameters. To account for the intrinsic uncertainty of the measured quantities in a simulator, we replicate each simulation 20 times and we calculate the 95% confidence intervals. However, given the length of the simulation runs and the number of replications, the computed confidence intervals are virtually indistinguishable from their mean values and we decided not to represent them in the following figures.

Parameter	Four-node	Six-node	Ten-node
Standard amendment	IEEE 802.11g	IEEE 802.11n	IEEE 802.11n
Simulation runs / Replications	20	20	20
Run duration [sec]	60	90	120
Payload length [bytes]	1000	1000	1000
Transmission rate [Mbps]	54	65	65

Table 2: Simulation parameters used in scenarios.

Four-node network. In our first scenario, we consider the four-node topology (depicted in Fig. 2) with the IEEE 802.11g standard amendment and a transmission rate of 54Mbps (maximum speed). Figure 6 shows the throughput evolution of all four nodes, named N1 through N4, as a function of the input rate of node 2, x_2, that gradually varies from 0 to 1 by step of 0.05. Let us

Scenario	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
Four-node	0.3	0.5	1	0.5	/	/	/	/	/	/
Six-node	0.5	0.4	0.6	0.3	0.7	0.9	/	/	/	/
Ten-node	0.3	0.6	0.8	0.4	0.7	0.3	0.5	0.4	0.9	0.7

Table 3: Input rates of nodes by default in scenarios.

recall that $x_2 = 0$ indicates that node 2 never has a frame to be sent while $x_2 = 1$ denotes that the node is always willing to transmit frames. The input rates for the other nodes are given in Table C.12. We observe that as x_2 grows, so does the throughput of node 4, while nodes 1 and 3's throughputs decrease. Indeed, larger values of x_2 imply more competition between nodes 1, 2, and 3 in accessing the medium. On the other hand, the gain in node 4 throughput is not directly caused by the increased throughput of node 2, but rather as a by-product of the decrease of node 3's throughput (with whom node 4 compete for accessing the medium). In fact, Fig. 6 suggests that the amount of throughput lost by node 3 is gained by node 4. Finally, we observe that our model was able to accurately capture all these behaviors.

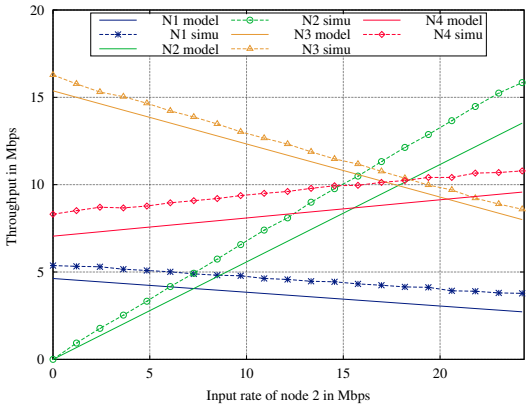


Figure 6: Four-node network: varying the input rate of node 2, x_2.

For the sake of completeness, we repeat the same scenario three other times but each time we keep x_2 steady to its default value (see Table C.12) and we let the input rates of one of the other nodes vary from 0 to 1 by step of 0.05. This gives us a total of $21 \times 4 \times 4 = 336$ points, out of which we derived the statistics on the relative error shown in Table 4. We notice that in over 90% of the samples the relative error is less than 20%.

Six-node network. Our second scenario deals with a larger network composed of 6 nodes and a different standard amendment, namely IEEE 802.11n whose transmission rate is set to 65Mbps. The network topology is depicted in Fig. 7. Figure 8 shows the throughputs attained by each of the six nodes as the input

Scenario	**Mean**	**Median**	**<5%**	**<10%**	**<20%**	**<30%**	**>30%**
Four-node	12.67%	13.43%	10.00%	35.00%	91.25%	100%	0.00%
Six-node	9.80%	9.77%	21.10%	54.91%	97.75%	99.20%	0.80%
Ten-node	9.11%	7.47%	27.78%	68.12%	88.77%	99.36%	0.64%

Table 4: Distribution of the relative error for the throughput, t_n.

rate of node 6 gradually increases from 0 to 1. Note that the input rates of the other nodes are given in Table C.12.

Not surprisingly, we observe in Fig. 8 that the throughputs of nodes 4, 5, and 6 are the most affected by the increasing input rate of node 6, as all three belong to the same clique. Node 6 increases its throughput mostly at the expense of node 5, that loses more than a third of its original throughput. Node 4's throughput decays to a lesser extent, however its already small throughput is even further decreased. We also notice that nodes 1, 2, and 3 are not directly affected by node 6 and they keep an almost steady throughput regardless of the value of x_6. Again, as in the previous case, we repeat the same experience but letting other nodes than node 6 vary their input rate from 0 to 1 by step of 0.05. Because it is a network of 6 nodes, this gives us a total of $21 \times 6 \times 6 = 756$ samples that are used to derive the statistics shown in Table 4. The table shows the predictions made by our model fit well those delivered by the simulator with a mean relative error of less than 10%.

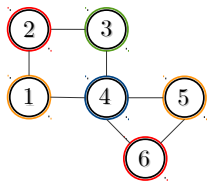


Figure 7: Conflict graph of the six-node network.

Ten-node network. Our third scenario involves a network of 10 nodes (see Fig. 9) using IEEE 802.11n. We study the throughput attained by all nodes as a function of node 4's input rate.

Figure 10 shows the corresponding results when we use the input rates of Table C.12. In order to keep the figure legible, we represent the attained throughput only for a subset of nodes. First, we observe that the variation of node 4's input rate causes its throughput to increase from 0 to approximately 20Mbps. On the other hand, as x_4 grows, the throughput of node 3 decays significantly (nearly halved). This agrees with the fact that node 3 is the only neighbor of node 4 (see Fig. 9). Because of node 3's declining throughput, nodes 1 and 2 experience a slight gain in their throughput as x_4 grows. As for the nodes far from node 4 such as nodes 8 and 10, their attained throughput is almost not influenced by the variations in x_4. Finally, Fig. 10 shows that our model man-

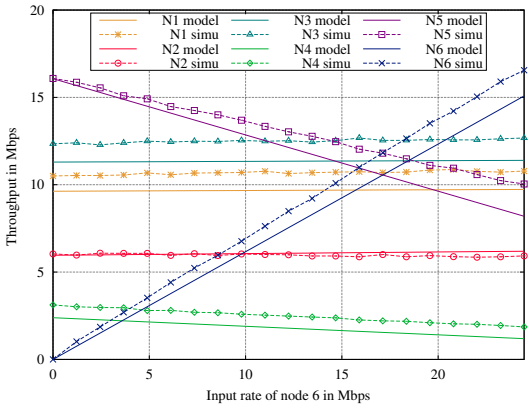


Figure 8: Six-node network: varying the input rate of node 6, x_6.

ages to capture all these behaviors with a good level of precision. Like in the two former scenarios, we repeat the same experiences letting the input rate of each node in lieu of x_4 vary from 0 to 1. This leads to a total of 2100 samples that we use to compute the statistics shown in Table 4.

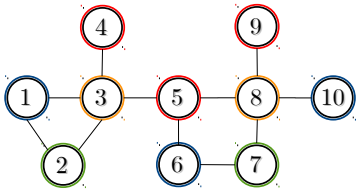


Figure 9: Conflict graph of the ten-node network.

Overall, in each of these three scenarios, we explore from dozens to hundreds of examples that differ by the nodes' input rates. To provide a broader overview of the accuracy reached by our model, we computed the mean and the median of the relative error as well as the distribution of the relative error attained on each scenario. Table 4 presents the corresponding results. The typical mean relative error is usually close to 10% and so is the median relative error. We also observe that in the vast majority of examples (around 90% of cases), the relative error made by our model is less than 20%. We obtain similar results on other common network topologies. In the interest of brevity we include the results in Appendix C.

6.1.2. Heterogeneous transmission rates

We now study the case where network nodes are heterogeneous with regard to their transmission rates. To do that, we reconsider the six-node node network but we assign a different transmission rate to every node as indicated by Table 5.

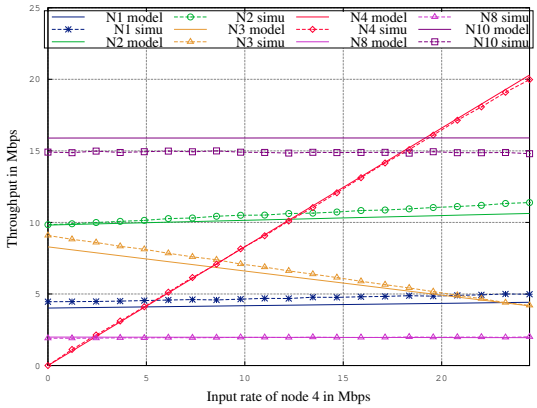


Figure 10: Ten-node network: varying the input rate of node 4, x_4.

Note that under these settings, node 2 has a transmission rate that is five times that of node 5. Due to the heterogeneity in the transmission rates, each node has a different backoff factor, α (see Section 4.2.6) and we derive the throughputs of nodes using Eq. (14). Analogously to the former scenarios, we let the input rate of each node vary from 0 to 1 while keeping the input rates of the other nodes to their default values (see Table C.12). This gives us a total of 756 cases on which we calculated the estimated throughputs using our model and compare these values to those delivered by the simulator.

Scenario	**N1**	**N2**	**N3**	**N4**	**N5**	**N6**
Six-node	18 Mbps	54 Mbps	24 Mbps	12 Mbps	9 Mbps	12 Mbps

Table 5: Transmission rates for the nodes of the six-node network in Fig. 7.

Table 6 presents the corresponding results. We notice that despite having nodes with significantly different transmission rates, our model is still able to deliver accurate estimations for the throughput. More precisely, the mean relative error of the model is 9% with 94% of the samples having an error less than 20%.

Scenario	**Mean**	**Median**	**<5%**	**<10%**	**<20%**	**<30%**	**>30%**
Six-node	9.31%	6.98%	20.29%	68.49%	94.13%	97.73%	2.27%

Table 6: Heteregenous transmission rates: distribution of the relative error for the throughput, t_n.

6.1.3. Frame aggregation in IEEE 802.11n

In our last scenario, we study the model's precision when the nodes implement the aggregation feature. When frame aggregation is enabled, multiples frames are concatenated into a single large frame before being transmitted. This

tends to diminish the cost of the overhead introduced by the MAC protocol, thereby increasing the maximal achievable throughput.

We consider again the six-node network of Fig. 7 with the input rates given in Table C.12 and the simulation setup of Table 2. However, all six nodes now aggregate four MAC service data units (MSDUs) into a single frame at each transmission. While the simulator actually implements the aggregation features, in our model we simply extended by a factor of 4 the length of frames.

Figure 11 shows the attained throughputs of all nodes as a function of the input rate of node 6. We can assess the influence of the frame aggregation feature on this scenario by comparing Fig. 8 and Fig. 11. Although the trends exhibited by the throughputs are still comparable, we observe that the frame aggregation feature significantly increases (almost doubles) the attained throughput. Finally, we included in Table 7 the mean, median, and distribution of the relative error when we let another node than node 6 vary its input rate. Figure 11 along with Table 7 show that our modeling approach can successfully handle the frame aggregation and capture its effects.

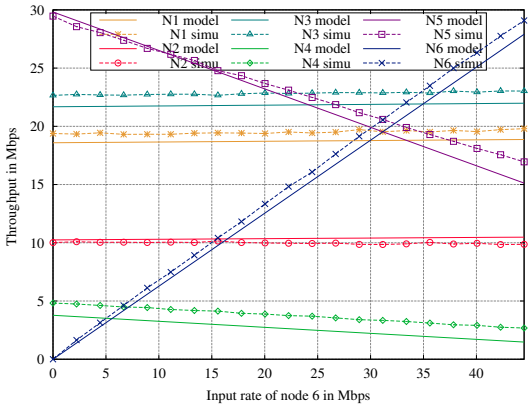


Figure 11: Frame aggregation on six-node network: varying the input rate of node 6.

Scenario	**Mean**	**Median**	**<5%**	**<10%**	**<20%**	**<30%**	**>30%**
Six-node	6.04%	5.07%	49.06%	92.03%	98.28%	99.22%	0.78%

Table 7: Frame aggregation: distribution of the relative error for the throughput, t_n.

6.2. Modeling complexity

In this section, we explore how the computational complexity of our modeling framework increases as the size of the WLAN under study grows. Unlike many existing modeling approaches [11, 12, 18, 24] that make use of a single Markov chain to describe the whole network behavior, ours revolves around a Divide-and-Conquer approach. Indeed, our approach breaks the original problem into

a set of smaller problems, each being solved individually thanks to the solution of a smaller Markov chain.

Unfortunately, we were not able to derive a closed-form expression (nor a tight upper bound) for the number of states in the Markov chains involved in our modeling approach. This exercise is made difficult as the exact values depend significantly not only on the number of nodes in the network, N, but also on the network's density, aka the average node degree. We nonetheless provide an empirical study.

We randomly generate thousands of conflict graphs with size varying from $N = 5$ up to $N = 14$. We sort them into five groups based on their density: average node degree of less than 3, between 3 and 4, between 4 and 5, between 5 and 6, and between 6 and 7. Then, for each interval of network density, we calculate the mean number of (sending) states per Markov chain. Figure 12 shows the corresponding results for a number of nodes in the network varying from $N = 5$ to $N = 14$. As expected, the average number of sending states per subnetwork grows with increasing values of N. However, even for $N = 14$, the mean number of states per Markov chain tends to lie around 8, meaning that most involved Markov chains are very small. On a side note, Fig. 12 suggests that networks with higher density tend to result in slightly larger Markov chain. Indeed, consider the subnetwork $b_{16} = [ON\ ON\ ON\ ON]$ for the four-node network of Fig. 2. Three possible (sending) states exist: $[\boldsymbol{1\ 0\ 0\ 1}]$, $[\boldsymbol{0\ 1\ 0\ 1}]$, and $[\boldsymbol{0\ 0\ 1\ 0}]$. However, if we remove the link between nodes 1 and 2, then only two (sending) states are possible: $[\boldsymbol{1\ 1\ 0\ 1}]$ and $[\boldsymbol{0\ 0\ 1\ 0}]$. More generally, the fewer links in a conflict graph, the smaller the density and the smaller the mean number of states per Markov chain.

Finally, for the sake of comparison, we included as a subplot in Fig. 12 the number of states in the Markov chain if one uses a classical description such as [11, 12, 18, 24]. The actual number values were found using a previous work of ours [27] that relies on a single large Markov chain to describe the whole network behavior. As expected, the mean number of states for the Markov chain is significantly larger (say two orders of magnitude) when using a single Markov chain as opposed to a series of smaller Markov chains, and can lead up to several hundreds of states when the number of nodes closes 14. Hence, we chose to have a large number of smaller Markov chains, keeping in mind that the last stage of our approach, aiming to combine the solutions found for each subnetwork, is a simple summation of the stationary probability distributions over all the subnetworks using the law of total probability [28].

Overall, by splitting the original problem into many smaller problems, whose solutions can be easily parallelized, our Divide-and-Conquer strategy circumvents the dimensionality curse associated to large Markov chain for conflict graphs having up to a dozen or so nodes. In practice, with a non-optimized implementation, models are typically solved at a click-speed for N around 4 or 5, and within a couple of seconds for N near to 10. We remind that our conflict graphs contain only APs belonging to the same channel, and that, depending on the IEEE 802.11 standard amendment in use, there can be from three to 24 non-overlapping channels.

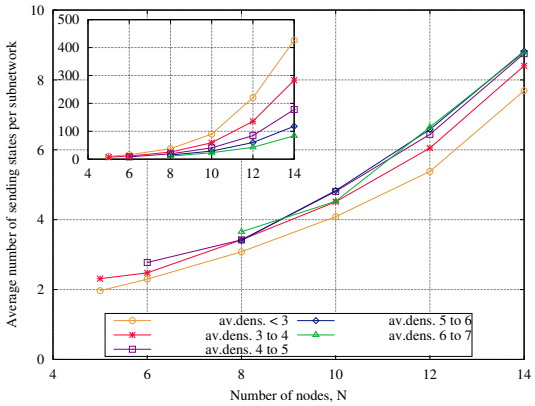


Figure 12: Number of (sending) states per Markov chain (subnetwork) as a function of the network's size and density.

6.3. Possible applications for the models

We provide two practical examples to illustrate how the proposed modeling framework can help in the deployment and configuration of an IEEE 802.11 WLAN.

6.3.1. Channel assignment

In our first example, we consider the well-known issue of channel assignment. In IEEE 802.11n and 802.11g networks, each AP can choose its channel among 14 different wireless channels in the 2.4GHz frequency range. However, out of these 14 channels, at most three can be chosen in a manner that no two channels have overlapping frequencies [22]. Obviously, given the way APs share the channel, the choice of channel assignment considerably affects the network's performance.

We consider the 12-node network ($N = 12$) depicted in Fig. 13a with three non-overlapping channels. The input rates of nodes are given in Table 8. For the sake of convenience, we classify nodes into two categories: high-demanding nodes whose input rates are higher than 0.5, and low-demanding nodes whose input rates are below 0.5. Let a be a vector of length N that represents one possible allocation of the three channels among the N APs. We denote by $y(a) = \{y_1, y_2, \ldots, y_N\}$ the set of output rates obtained when implementing the channel assignment a. Remind that y_i can be viewed as a measure of the normalized throughput attained by node i.

We consider four different performance metrics to evaluate the performance of the network:

1. The global satisfaction rate, GSR, or the proportion of the network's

general throughput demand that has been met, calculated as:

$$GSR(y(a)) = \frac{\sum_{n=1}^{N} y_n}{\sum_{n=1}^{N} x_n} \ . \tag{15}$$

2. The Jain's fairness index [29], J, that measures how fairly the throughput was divided among the nodes. Jain's index is a quantity in the interval $[0, 1]$, where 1 represents the highest fairness, meaning all nodes get an equal share. It is calculated as:

$$J(y(a)) = \frac{\left(\sum_{n=1}^{N} y_n\right)^2}{\sum_{n=1}^{N} y_n^2} \tag{16}$$

Additionally, we can calculate the Normalized Jain's index, NJ. The normalization refers to accounting for the nodes' input rates when calculating Jain's index:

$$NJ(y(a)) = \frac{\left(\sum_{n=1}^{N} \frac{y_n}{x_n}\right)^2}{\sum_{n=1}^{N} \frac{y_n}{x_n}^2}. \tag{17}$$

3. The proportional fairness, PF, that is a trade-off between GSR and J as it tries to maximize both fairness and throughput by giving more throughput to nodes with higher demands:

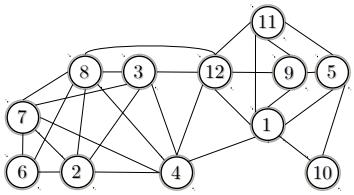
$$PF(y(a)) = \sum_{n=1}^{N} log\frac{y_n}{x_n} \ . \tag{18}$$

Scenario	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
12-node	0.2	0.4	0.9	0.7	0.8	0.9	0.1	0.3	0.2	0.6	0.8	0.3

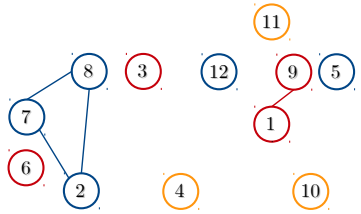
Table 8: Input rates of the 12-node network in Fig. 13a.

In practice, our model could be used jointly with existing solutions in the field of channel allocation, such as [30, 31]. A classical way of finding (sub)optimal channel allocations is to start from a given allocation, and then iteratively improve it with regard to some network performance parameters until convergence is found. In this regard, our model could be used to quickly evaluate the performance parameters of interest at each iteration (rather than relying on long simulations). However, for the sake of simplicity and given the size of the network, we choose to explore all of the $3^{12} \simeq 530,000$ possible allocations and

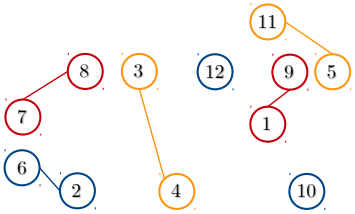
retain the ones maximizing one of the criteria given above. Figures 13b, 13c, and 13d illustrate the channel assignment that maximize GSR, J, and PF, respectively. For each of these three channel assignments, we also indicated in Table 9 their score over the other performance metrics.



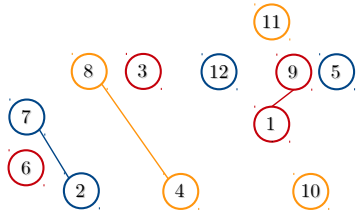
(a) Original



(b) Maximize global satisfaction rate



(c) Maximize Jain's fairness index



(d) Maximize proportional fairness

Figure 13: Different channel allocations for a randomly-generated 12-node network.

Performance metric	GSR	J	NJ	PF
Fig. 13b Maximize GSR	96%	0.725	0.983	-1.27
Fig. 13c Maximize J	73%	0.796	0.955	-3.30
Fig. 13d Maximize PF	95%	0.735	0.987	-1.08

Table 9: Evaluation of the proposed channel allocations.

When maximizing the global satisfaction rate, GSR, the retained solution maximizes the overall throughput obtained in the network and leads a GSR of 96%. Interestingly, we notice in Fig. 13b that all the high-demanding nodes (whose input rate is over 0.5) do not share the channel with any other node, thereby enabling them to obtain the highest possible throughput. On the other hand, when maximizing Jain's index, we observe in Fig. 13c that almost all nodes have a neighbor with whom they share the medium. In fact, with the exception of the pair of nodes 6 and 2, all the other pairs involves two nodes belonging to the same class (be it low-demanding or high-demanding nodes). As a consequence high-demanding nodes get lower output rates, as they have to share the medium with other high-demanding nodes. The optimal solution for the Jain's index increases its score from 0.725 to 0.796, at the expense of over 20% loss in the GSR. The last optimal solution maximizes the propor-

tional fairness, PF. In Fig. 13d we observe that the only difference between the PF solution and the GSR solution lies in the selected channel of node 8. This similarity can be understood by the fact that proportional fairness, unlike Jain's fairness index, takes into consideration not only the output rate of each node but also its input rate. Overall, in this example, the optimal solution for Proportional fairness coincides with the optimal solution for Normalized Jain's index, and appears as a good trade-off between maximizing throughput or fairness, as it offers both a GSR value and J value that are remarkably close to their optimal values.

6.3.2. Upgrading from IEEE 802.11g to 802.11n

Our second example illustrates how our model can help when considering an upgrade of the IEEE 802.11 standard amendment deployed over the APs of a WLAN. More specifically, while the 802.11g amendment is widely used, its maximum transmission rate of 54Mbps can be viewed as insufficient in some cases. Upgrading to 802.11n can be an attractive solution as it enables higher transmission rates and also implements the frame aggregation feature. However, by aggregating frames, there is a potential risk in deepening the effect of starvation that some nodes may already face. Therefore, a thorough analysis of an upgrade to 802.11n must include the benefits both in terms of overall throughput and fairness.

We consider the four, six, and ten-node networks presented in Section 6.1.1 together with the input rates given in Table C.12. Let us denote by k the number of frames aggregated in each transmission. First, we run our model using IEEE 802.11g at 54 Mbps (without aggregation, $k = 1$). Then, we rerun our model on the same network but using IEEE 802.11n at 65Mbps, while considering two possible sizes for the frame aggregation, $k = 4$ and $k = 16$. We calculate the throughput gain the network experiences with aggregation, as opposed to without, as well as the Normalized Jain's index.

Table 10 shows the associated results. We observe that the gain in throughput is typically around 85% when aggregating four frames, while it reaches nearly 230% if frames are aggregated by batches of 16. We include in Table 10 the values found for the normalized Jain's index. It appears that frame aggregation has very little effect on Jain's index suggesting that the medium sharing between the nodes remains fair, regardless of the aggregation features. Based on these results, upgrading from IEEE 802.11g to 802.11n appears as an attractive option.

Scenario	Throughput gain $k = 4$	Throughput gain $k = 16$	NJ $k = 1$	NJ $k = 4$	NJ $k = 16$
four-node network	86%	239%	0.981	0.965	0.953
six-node network	84%	233%	0.892	0.865	0.848
ten-node network	87%	238%	0.890	0.857	0.838

Table 10: Evaluating the gain in upgrading to IEEE 802.11n.

7. Conclusion

We have presented a modeling framework for IEEE 802.11-based WLANs. Our approach accounts for WLANs composed of multiple APs assuming their conflict graph is known. Our framework assumes any levels of load in the APs, arbitrary sizes for frames and arbitrary transmission rates for links, as well as recent amendments to IEEE 802.11 such as 802.11n. The proposed solution revolves around a Divide-and-Conquer approach to split the initial problem into many sub-problems, each being of much lower complexity.

We studied several hundreds of examples to assess the accuracy of our modeling framework comparing its results with those delivered by the ns-3 simulator. We considered several network topologies with the number of APs ranging from 3 to 10, different amendments of IEEE 802.11, various levels of the load on each AP, different transmission rates on the APs, as well as examples where APs implement the aggregation feature so that multiples frames are concatenated into a single large frame before being transmitted. Overall, in our examples, our model was able to forecast with a reasonable degree of precision (typically within 10% of relative errors) the mean throughput attained by each AP of the network.

To illustrate potential uses of our modeling framework, we explore two issues related to WLAN configuration: choosing the optimal AP channel allocation and enabling the aggregation of frames on APs. We show how our model outcomes can quickly help find an efficient configuration of the network.

A possible extension of our work would be to attempt to apply it in the context of mobile nodes (e.g. vehicular networks).

Acknowledgments

The authors would like to express their sincere thanks to the anonymous referees for their remarks and comments.

References

[1] IEEE Computer Society LAN MAN Standards Committee and others, Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, IEEE Standard 802.11-2016.

[2] G. Bianchi, Performance analysis of the IEEE 802.11 distributed coordination function, IEEE Journal on selected areas in communications 18 (3) (2000) 535–547.

[3] F. Cali, M. Conti, E. Gregori, IEEE 802.11 wireless LAN: Capacity analysis and protocol enhancement, in: INFOCOM'98. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, Vol. 1, IEEE, 1998, pp. 142–149.

[4] K. Kosek-Szott, A comprehensive analysis of IEEE 802.11 DCF heterogeneous traffic sources, Ad Hoc Networks 16 (2014) 165–181.

[5] N. Gupta, C. Rai, New analytical model for non-saturation throughput analysis of IEEE 802.11 DCF, in: International Conference on Advances in Communication, Network and Computing, 2014, pp. 66–76.

[6] E. Felemban, E. Ekici, Single hop IEEE 802.11 DCF analysis revisited: Accurate modeling of channel access delay and throughput for saturated and unsaturated traffic cases, IEEE Transactions on Wireless Communications 10 (10) (2011) 3256–3266.

[7] J. Lee, H. Lee, Y. Yi, S. Chong, E. W. Knightly, M. Chiang, Making 802.11 DCF near-optimal: Design, implementation, and evaluation, IEEE/ACM Transactions on Networking 24 (3) (2016) 1745–1758.

[8] E. Fitzgerald, U. Körner, B. Landfeldt, An analytic model for throughput optimal distributed coordination function (TO-DCF), Telecommunication Systems (2017) 1–19.

[9] Z. Shi, C. Beard, K. Mitchell, Analytical models for understanding space, backoff, and flow correlation in CSMA wireless networks, Wireless networks 19 (3) (2013) 393–409.

[10] T. Begin, B. Baynat, I. Guérin Lassous, T. Abreu, Performance analysis of multi-hop flows in IEEE 802.11 networks: A flexible and accurate modeling framework, Performance Evaluation 96 (2016) 12–32.

[11] B. Nardelli, E. W. Knightly, Closed-form throughput expressions for CSMA networks with collisions and hidden terminals, in: INFOCOM, 2012 Proceedings IEEE, IEEE, 2012, pp. 2309–2317.

[12] M. Durvy, O. Dousse, P. Thiran, Self-organization properties of CSMA/CA systems and their consequences on fairness, IEEE Transactions on Information Theory 55 (3) (2009) 931–943.

[13] L. Jiang, J. Walrand, A distributed CSMA algorithm for throughput and utility maximization in wireless networks, IEEE/ACM Transactions on Networking (ToN) 18 (3) (2010) 960–972.

[14] R. Boorstyn, A. Kershenbaum, B. Maglaris, V. Sahin, Throughput analysis in multihop CSMA packet radio networks, IEEE Transactions on Communications 35 (3) (1987) 267–274.

[15] X. Wang, K. Kar, Throughput modelling and fairness issues in CSMA/CA based ad-hoc networks, in: INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE, Vol. 1, IEEE, 2005, pp. 23–34.

[16] C. Chaudet, I. Guérin Lassous, E. Thierry, B. Gaujal, Study of the impact of asymmetry and carrier sense mechanism in IEEE 802.11 multi-hops networks through a basic case, in: Proceedings of the 1st ACM international workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks, ACM, 2004, pp. 1–7.

[17] B. Ducourthial, S. Mottelet, A. Busson, Improving fairness between close Wi-Fi access points, Journal of Network and Computer Applications 87 (2017) 87–99.

[18] C. Kai, S. Zhang, Throughput analysis of CSMA wireless networks with finite offered-load, in: IEEE International Conference on Communications (ICC), IEEE, 2013, pp. 6101–6106.

[19] R. Laufer, L. Kleinrock, The capacity of wireless CSMA/CA networks, IEEE transactions on Networking.

[20] T. Bonald, M. Feuillet, Performance of CSMA in multi-channel wireless networks, Queueing Systems 72 (1-2) (2012) 139–160.

[21] S. Biaz, S. Wu, Rate adaptation algorithms for IEEE 802.11 networks: A survey and comparison, in: Computers and Communications, 2008. ISCC 2008. IEEE Symposium on, IEEE, 2008, pp. 130–136.

[22] L. Verma, M. Fakharzadeh, S. Choi, WiFi on steroids: 802.11 ac and 802.11 ad, IEEE Wireless Communications 20 (6) (2013) 30–35.

[23] J. Padhye, S. Agarwal, V. N. Padmanabhan, L. Qiu, A. Rao, B. Zill, Estimation of link interference in static multi-hop wireless networks, in: Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement, USENIX Association, 2005, pp. 28–28.

[24] M. Garetto, T. Salonidis, E. W. Knightly, Modeling per-flow throughput and capturing starvation in CSMA multi-hop wireless networks., in: INFOCOM, 2006.

[25] Cisco Visual Netwotking Index, `https://www.cisco.com/c/en/us/solutions/service-provider/visua`

[26] The Network Simulator ns-3, https://www.nsnam.org/.

[27] M. Stojanova, T. Begin, A. Busson, Conflict graph-based Markovian model to estimate throughput in unsaturated IEEE 802.11 networks, in: IEEE International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, WiOpt 17, 2017.

[28] M. Harchol-Balter, Performance modeling and design of computer systems: queueing theory in action, Cambridge University Press, 2013.

[29] R. Jain, D. M. Chiu, W. R. Hawe, A quantitative measure of fairness and discrimination for resource allocation in shared computer system, Digital Equipment Corporation.
URL https://books.google.fr/books?id=M2QLGwAACAAJ

[30] S. Chieochan, E. Hossain, J. Diamond, Channel assignment schemes for infrastructure-based 802.11 WLANs: A survey., IEEE Communications Surveys & Tutorials 12 (1) (2010) 124–136.

[31] C. Peng, H. Zheng, B. Y. Zhao, Utilization and fairness in spectrum assignment for opportunistic spectrum access, Mobile Networks and Applications 11 (4) (2006) 555–576.

[32] IEEE Computer Society LAN MAN Standards Committee and others, Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, IEEE Standard 802.11-1997.

Appendix A. DCF and IEEE 802.11 parameters

IEEE 802.11 WLANs use DCF to manage the medium access. We briefly review DCF and use Fig. A.14 to schematically represent the procedure. For a more detailed explanation, we refer the reader to the IEEE 802.11 standard [32]. It should be noted that the described procedure only applies to unicast frames. In Table. A.11 we show the DCF parameters for the two amendments we use in the ns-3 simulations, the IEEE 802.11g and 802.11n.

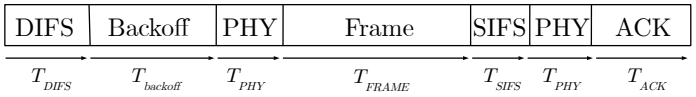


Figure A.14: The DCF procedure for medium access.

Parameter	802.11g	802.11n
CW_{min}	15	15
CW_{max}	1023	1023
T_{slot} (μs)	9	9
$T_{backoff}$ (μs)	67.5	67.5
T_{DIFS} (μs)	28	34
T_{SIFS} (μs)	10	16
T_{PHY}	20	36
Payload, L (bytes)	1000	1000
Headers H (bytes)	64	66
ACK (bytes)	14	14
R_{frame} (Mbps)	54	65
R_{ack} (Mbps)	24	24

Table A.11: The DCF parameters for IEEE 802.11g and 802.11n, corresponding to the simulation results of Section 6.

Before a node starts a frame transmission, it first needs to make sure that the channel is continuously sensed idle for the duration of one DIFS period, so as to avoid a collision with an ongoing transmission. Next, the node starts the $backoff$ period whose goal is to desynchronize the beginnings of transmissions of neighboring nodes. Unlike the DIFS period, the backoff can be frozen if the medium is sensed busy and then resumed when the medium becomes idle again. The duration of each backoff period is random and is calculated as the product of an integer value randomly generated in the interval $[0, CW]$ and the slot time T_{slot}. Initially, the contention window is set as $CW = CW_{min}$. For every retransmission of the same frame, the CW is doubled until it reaches CW_{max}. Thus, for frames without retransmissions, the mean duration of the backoff period is:

$$T_{backoff} = \frac{CW_{min} \times T_{slot}}{2} \ . \tag{A.1}$$

Once the backoff countdown has finished, the physical header and then the frame transmission (payload and headers) begin. The duration of the frame transmission, T_{FRAME} is calculated as:

$$T_{FRAME} = \frac{(L+H)\times 8}{R_{frame}} \ . \tag{A.2}$$

The last mechanism is the acknowledgement frame. The destination acknowledges every successfully received frame by sending an ACK frame to the source. As all other frames, the ACK frame is preceded by a physical header. The duration of the ACK transmission is:

$$T_{ACK} = \frac{ACK\times 8}{R_{ack}} \ . \tag{A.3}$$

Finally, the total transmission time, T, can be calculated:

$$T = T_{backoff} + T_{DIFS} + T_{PHY} + T_{FRAME} + T_{SIFS} + T_{PHY} + T_{ACK} \ . \tag{A.4}$$

Appendix B. The backoff factor α

Chaudet et al. [16] and Ducourthial et al. [17] study the impact of a frame's transmission duration in the three-node FIM topology of Fig. 1 and in larger chain networks, respectively. They find that when transmissions are kept short the duration of the backoff becomes comparable to the duration of the frame transmission. As a result, it is more likely that both nodes 1 and 3 of the FIM network are simultaneously in backoff, leaving the channel idle for a potential transmission of the starving node 2. It follows that shorter transmissions increase the fairness of resource sharing.

We wish to quantify the impact of the transmission duration on fairness. We do so by introducing the backoff factor α, as the ratio between the average backoff period duration and the duration of the entire transmission:

$$\alpha = \frac{T_{backoff}}{T_{DIFS} + T_{PHY} + T_{FRAME} + T_{SIFS} + T_{PHY} + T_{ACK}} \ . \tag{B.1}$$

In our study of resource sharing we choose the FIM topology and the IEEE 802.11g standard amendment. We begin by varying the backoff factor α and tracing the evolution of the middle node's output rate, y_2, as a function of α. In Fig. B.15 we show this evolution for 11 different α values in the interval $[0.03, 0.50]$, obtained by changing the network's transmission rate and frame size. More specifically, the first eight values use the IEEE 802.11g's eight mandatory transmission rates of 6, 9, 12, 18, 24, 36, 48, and 54Mbps, respectively, and a fixed payload length of 1500B. The last three values use a transmission rate of 54 Mbps and payloads of 1000B, 500B, and 200B, respectively. Higher values of α are theoretically possible, however we do not consider them to be representative of real-world examples. Next, we perform a quadratic fit (using the least squares

method) of node 2's output rate as a function of α and discovered it is closely matched by the function:

$$f(\alpha) = -0.66\alpha^2 + 0.88\alpha + 0.01 \ . \tag{B.2}$$

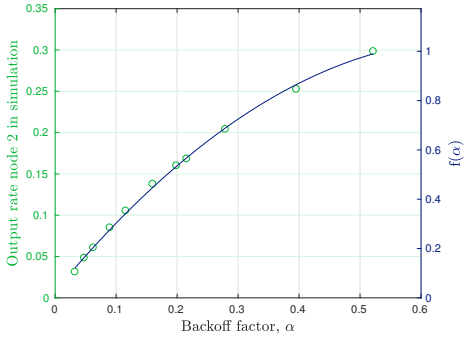


Figure B.15: Influence of the backoff factor α on node 2's output rate.

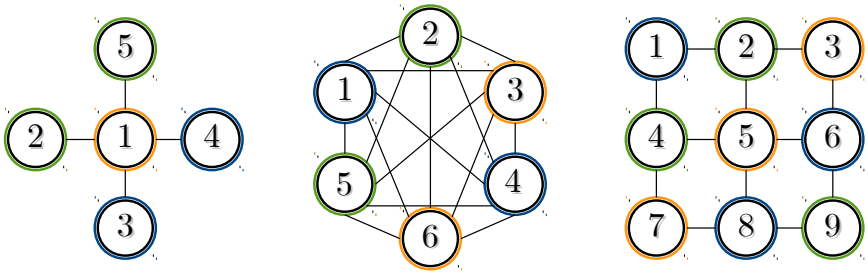
For the purpose of our modeling framework, we normalized the $f(\alpha)$ function so that $0 \leq f(\alpha) \leq 1$, as shown by the right-hand side y-axis of Fig. B.15. Thus, the best-case scenario for the middle node is when $\alpha = 0.5$. In this case in Eq. (10) we have $f(\alpha) = 1$, and the middle mode has the highest achievable output rate. For any lower α values we obtain an $f(\alpha) < 1$, which leads to decreasing the weighting factor of all dominated chains in Eq. (10), and subsequently decreasing the output rate of the middle node.

Appendix C. Additional simulation results

For the sake of completeness, we also ran our model (along with the ns-3 simulator) on three well-known topologies: a five-node star, a six-node full-mesh, and a nine-node grid network (see Fig. C.16). Note that we used the IEEE 802.11n standard amendment, with 1000B payload length and a transmission rate of 65Mbps for these three scenarios. The nodes' input rates are given in Table C.12. Similarly to our previous scenarios, we set and keep constant the input rates of all nodes but one whose input rate will vary in the interval [0,1]. Figures C.17 and C.18 shows the throughput of nodes as given by our model against that delivered by the simulator. Table C.13 reports the distribution of the relative error for the throughput. We observe that the mean relative error lies around 10 % for the star and grid networks, while it is below 4% for the full-mesh network. We also notice that in the vast majority of examples, the relative error is less than 20%.
Overall, the levels of accuracy found in these new scenarios seem to be in line with the ones presented in Section 6.

Scenario	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
Five-node star	0.7	0.3	0.4	0.9	0.2	/	/	/	/
Six-node mesh	0.7	0.3	0.4	0.9	0.2	0.8	/	/	/
Nine-node grid	0.7	0.3	0.4	0.9	0.2	0.8	0.6	0.5	0.1

Table C.12: Default input rates in scenarios.

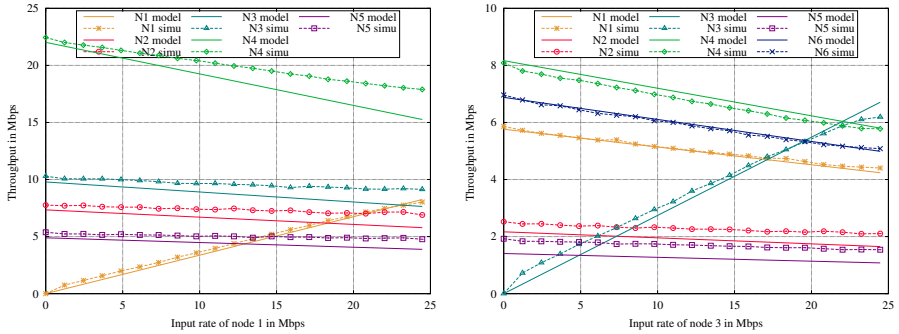


(a) five-node star (b) six-node mesh (c) nine-node grid

Figure C.16: Conflict graphs

Scenario	**Mean**	**Median**	**<5%**	**<10%**	**<20%**	**<30%**	**>30%**
Five-node star	10.96%	11.31%	11.83%	41.82%	98.62%	99.61%	0.39%
Six-node complete	3.46%	2.62%	74.58%	95.51%	100.00%	100.00%	0.00%
Nine-node grid	10.46%	8.24%	36.10%	54.63%	83.26%	99.33%	0.64

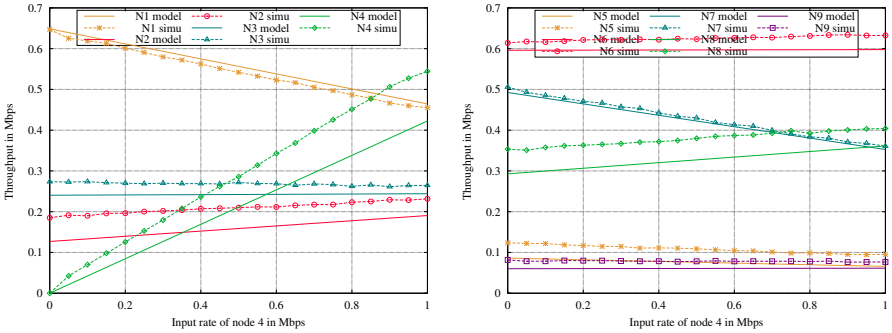
Table C.13: Distribution of the relative error for the output rates, y_n.



(a) Five-node star, varying the input rate of node 1, x_1.

(b) Six-node mesh, varying the input rate of node 3, x_3.

Figure C.17: Results for the star and mesh topologies.



(a) Nodes 1-4

(b) Nodes 5-9

Figure C.18: Results for the Grid topology, varying the input rate of node 4, x_4.