A Markov Model for Performance Evaluation of Channel Bonding in IEEE 802.11

Marija Stojanova, Thomas Begin, Anthony Busson

A Markov Model for Performance Evaluation of Channel Bonding in IEEE 802.11

Marija Stojanova, Thomas Begin, Anthony Busson

Univ Lyon, UCB Lyon 1, ENS Lyon, Inria, CNRS, LIP UMR 5668 - Lyon, France

Abstract

The ever-growing popularity of Wireless Local Area Networks (WLANs) in home, public, and work environments is fuelling the need for WLANs that can accommodate more stations, each with higher throughput. This typically results in WLANs containing a larger number of heterogeneous devices, making the prediction of the network's behavior and its efficient configuration an even more elaborate problem. In this paper, we propose a Markovian model that predicts the throughput achieved by each Access Point (AP) of the WLAN as a function of the network's topology and the AP's throughput demands. By means of simulation, we show that our model achieves mean relative errors of around 10% for networks of different sizes and with diverse node configurations. The model is adapted to the specification of IEEE 802.11 standards that implement channel bonding, namely 802.11n/ac/ax, and as such it can be used to provide insight into issues of channel assignment when using channel bonding. We derive guidelines on the best practice in static channel bonding given a performance metric and for different node characteristics such as the Modulation and Coding Scheme (MCS) indexes, frame aggregation rates, saturation levels, and network topologies. We then put our findings to the test by identifying the optimal channel bonding combination in an 802.11ac WLAN containing nodes with diverse characteristics. We conclude that the optimal solution is highly dependent on the particular network configuration. However, we find that, in general, larger channels are better suited for throughput maximization and smaller (and separate) channels render higher fairness.

1. Introduction

Current Internet users largely depend on Wireless Local Area Networks (WLANs) to provide them with a reliable and efficient access to online services. The possibility of connecting mobile and fixed devices using the same network in a seamless manner has allowed WLANs to penetrate the home, work, and public network markets. WLANs grow in popularity in each of these environments, resulting in constantly increasing demands for wireless coverage and capacity. Cisco's Visual Networking Index (VNI) predicts that 50% of all IP traffic will

be wireless or cellular by 2022 [1] Despite cellular technologies becoming more widespread, an increasing amount of cellular traffic will be offloaded to WLANs, with an estimate of 71% of traffic being offloaded from 5G networks by 2022. Moreover, a four-fold increase is expected between 2017 and 2022 in both the number of Access Points (APs) available worldwide and in the traffic they generate. Existing WLANs need to be enhanced in order to efficiently handle the increasing volume of traffic they will experience in the following years.

Since its first release in 1997, the IEEE 802.11 standard, which specifies the Physical and Medium Access Control (MAC) layers for most WLANs, has been amended many times in order to address, among others, issues of capacity, security, or to adapt to a particular network environment. Subsequent amendments often enrich the available physical data rates using denser modulations than previously. In the recent 802.11n and 802.11ac amendments and the upcoming 802.11ax, the notions of frame aggregation and channel bonding were introduced and enhanced to increase the maximum data rates by either sending several frames at a time (frame aggregation), or by using several radio channels at a time (channel bonding). Two different channel bonding modes were proposed, namely static [2, 3, 4] and dynamic channel bonding [4, 5, 6]. While these techniques have been shown to provide significantly higher throughputs, they also increase the complexity of the network by creating more heterogeneity among devices. As the devices are more complex and their number increases, predicting how the network's resources are shared becomes an even more elaborate problem. Thus, having predictive WLAN models can help to move from inefficient to more advisable configurations.

In this paper, we propose a Markovian model that aims at predicting the throughput achieved by each AP of the WLAN as a function of the network's topology and the AP's throughput demands. The model is adapted to and designed for standard amendments that implement channel bonding, such as 802.11n/ac/ax. The use of Continuous Time Markov Chains (CTMC) allows us to incorporate a detailed account of each node's average frame aggregation rate, chosen channel width, data rate, and demanded throughput. By means of simulation, we show that our model achieves mean relative errors of less than 10% for networks of different sizes and with diverse node configurations. As the model's solution can be automatized and can typically be found within seconds for a network of a dozen nodes, we can use it to easily explore a large space of different configurations and study the impact of the network's topology, the channel bonding, the frame aggregation rate, the physical data rates, and the throughput demands on the overall performance. This exploration allows us to derive guidelines on the best practice in applications such as channel bonding or channel assignment. We then put our findings to the test by identifying the optimal channel bonding combination given a performance metric in an IEEE 802.11ac WLAN containing a diverse set of nodes.

This paper is organized as follows. Section 2 reviews the existing related work on the topic of WLAN performance evaluation. Section 3 describes the physical system in which we are interested, followed by an explanation of the proposed CTMC modeling approach. In Section 4, we validate the model's predicted throughputs using the ns-3 discrete-event network simulator. Section 5 illustrates how the proposed model can be used to study the behavior of a WLAN and then optimize the number of channels bonded per AP. Finally, Section 6 concludes this paper.

2. State of the art

The performance evaluation of IEEE 802.11 WLANs is a rich research area where analytical models, experimental work, and simulation studies coexist and evolve with each new standard amendment. We review a portion of the works that study three generations of 802.11 standard amendments: *i*) legacy 802.11 and a/b/g, *ii*) 802.11n, and *iii*) 802.11ac and 802.11ax. We conclude this section with a review of the existing work in performance evaluation of channel bonding, a technique introduced in IEEE 802.11n and then reused (and enhanced) in 802.11ac/ax/be.

2.1. IEEE 802.11 legacy/a/b/g

Markovian modeling of 802.11 networks began with the seminal work of Bianchi [7]. Bianchi models the backoff process happening before every frame transmission using a two-dimensional Markov chain to represent saturated and fully-connected WLANs. Many following works extended Bianchi's work and focus on alleviating the saturation constraint. Kosek-Szott [8] adds one more state to the Markov chain to represent a node that occasionally has no frames to be sent and thus is able to model non-saturated networks. Felemban and Ekici [9] choose a different approach by introducing the probability that a node has a frame waiting to be sent. They do so by embedding a second Markov chain into Bianchi's original chain. This new chain represents the current state of the medium as being idle, in transmission, or in collision. They then iterate between the two chains until convergence.

The two-level approach is often encountered in 802.11 modeling. Shi et al. [10] and Begin et al. [11] propose two such models that handle non-saturated multi-hop networks. The low-level model in both works is a modification of Bianchi's original chain. The high-level model of [10] is a separate Markov chain describing the current states of neighboring nodes as being either idle, transmitting, or in backoff. In [11] the high-level model consists of a set of $M/M/1/K$ queues, where each queue represents a given node of the network. Even though both models circumvent the saturation and connectivity issues of Bianchi's original work, their computational complexity limits their usage to networks of less than ten nodes.

Other authors have chosen to model only specific node topologies. Chaudet et al. [12] study the behavior of the three-node chain network using a Markovian model where the states contain, among other parameters, the idle time experienced by a node. They conclude that shorter transmissions of the edge nodes result in higher equality of resource sharing and avoids starvation in the middle node. Following up on the work of Chaudet et al. [12], Ducourthial et al. [13] present an approach for modeling chain networks of arbitrary sizes and test it on the 802.11a/b standard amendments, as well as the 802.11n. They show that chains with an even number of nodes manifest more equality, and that for very large chains the inequality of channel access disappears around the 15th node. They also show that modifying the transmission duration of the edge nodes can significantly improve the overall equality of resource sharing. While their work is easily generalizable to different standards, it is still limited to chain topologies and does not inspect into detail the impact of all 802.11n parameters.

2.2. IEEE 802.11n

As opposed to older standards, IEEE 802.11n allows for the use of frame aggregations and the bonding of two 20 MHz channels into a single 40 MHz channel in the 2.4 GHz band. Combining these two techniques with MIMO transmissions results in data rates of up to 600 Mbps. IEEE 802.11n proposes two types of frame aggregation: Aggregate MAC Service Data Unit (A-MSDU) and Aggregate MAC Protocol Data Unit (A-MPDU). By aggregating several frames at a time, the same DCF overhead is shared among several frames resulting in higher transmission rates. By default, all frames in the IEEE 802.11ac standard are sent as aggregate MPDUs (even if they contain a single MPDU). The 802.11n standard amendment also defines new Modulation and Coding Schemes (MCS) that are combinations of a given modulation type (BPSK, QPSK, 16-QAM...) and a coding rate (1/3, 2/3, 3/4...) that result in a given data rate.

Ginzburg and Kesselman [14] propose an analytical model to study the maximum throughput experienced in a WLAN of a single Access Point (AP) and its station. They quantify the impact of different types of frame aggregation when using either UDP or TCP as transport layer protocols, and while having different levels of noise. They conclude that A-MPDU aggregation largely outperforms A-MSDU aggregation as the noise level increases. Lin and Wong [15] extend Bianchi's original model to study the performance of A-MSDU and A-MPDU aggregation in WLANs prone to errors. They compare the results delivered by their analytical model to those provided by the ns-2 simulator and propose a frame aggregation adaption algorithm that chooses the optimal size of an A-MSDU for a transmitter with a given bit error rate.

In our previous work [16] we present a discrete time Markovian model for IEEE 802.11 networks. We test the model on 802.11g/n WLANs with no channel bonding and find that it can accurately predict the achieved throughputs of all nodes. However, when the links are heterogeneous it is a fairly complex task to calculate the nodes' throughputs as our DTMC provides the proportion of medium accesses (and not the proportion of time) a node obtains. This quantity is then normalized clique-wise in order to calculate the throughput.

2.3. IEEE 802.11ac/ax

Ong et al. [17] provide a detailed description of the enhancements and mandatory/optional features of 802.11ac: channel bonding of up to eight 20 MHz channels (two in 802.11n), MIMO enhancement allowing up to eight spatial streams (four in 802.11n), as well as an increased highest order modulation (from 64-QAM to 256-QAM). They then perform an in-depth theoretical analysis of 802.11ac and 802.11n in a saturated, error-free network and show how the enhancements of 802.11ac (channel width and MIMO) can result in an 84% increase from 802.11n's throughput. The authors also show that a combination of A-MSDU and A-MPDU aggregation can result in a higher throughput that using either aggregation alone. Cha et al. [18] analytically compare Multi-User MIMO (MU-MIMO) to Single and Multi-User Frame Aggregation (SU-FA and MU-FA) in a network consisting of a two-antenna AP transmitting to two single-antenna stations. They conclude that when both stations have similar frame lengths MU-MIMO renders higher throughputs as it better utilizes the channel. When frame lengths are heterogeneous or the channel is fast-varying it is better to use MU-FA. Both MU-MIMO and MU-FA always outperform SU-FA due to the fact that SU-FA experiences a much higher MAC overhead.

4

As the 802.11ac equipment became widely available, many works focused on experimental performance evaluation of 802.11ac WLANs. Dianu et al. [19] study the performance of 802.11ac networks in an office environment. They test the maximum achieved throughput when the transmitter-receiver distance and the interference change and when the 802.11ac network cohabitates with an 802.11n network. They show that the network can easily achieve a throughput of 700 Mbps when the transmitter-receiver distance is small. However, as any WLAN operating in the 5GHz frequency band, the network suffers from low throughput and bad connectivity when obstacles are present, such as load-bearing walls. Kriara et al. [20] empirically study the impact of MIMO, channel bonding, MCS, and guard intervals on the performance of 802.11ac networks in a home and office environment. They perform extensive tests in a environment with/without interference, using different link qualities and topologies. The authors use multiple linear regression to show how the throughput and jitter are highly impacted not only by the choice made for a single parameter, but also by the mutual influence of several parameters.

Recently, Khorov et al. published an extensive tutorial on the upcoming 802.11ax standard amendment [21], with a spotlight on the Orthogonal Frequency-Division Multiple Access (OFDMA) technique as the key feature of the amendment, as well as an update on the current status of 802.11be's development [22]. The authors provide a detailed list of all the modifications introduced in 802.11ax as well as a comparison with existing techniques in previous amendments. Bellalta and Kosek-Szott [23] review the 802.11ax amendment and then propose an adaptation of Bianchi's original model [7] to the context of AP-initiated Multi User (MU) transmissions scheme. The authors use their model to compare the performance of 802.11ac and ax in a fully-connected network of one AP and several stations. Khorov et al. [24] study the coexistence of 802.11ax and legacy devices and examine fairness issues between the two types of devices. The authors propose a modification to the Enhanced Distributed Channel Access (EDCA) parameters in order to tackle the fairness issues and a mathematical model to show the improvement achieved when those modifications are implemented. Wilhelmi et al. [25] study the performance of the Spatial Reuse (SR) feature of 802.11ax amendment in simulation and show that SR can significantly improve the performance of a WLAN, especially in high density and traffic load scenarios.

2.4. Channel bonding

In legacy IEEE 802.11 standard amendments, frames are sent over 20 MHz channels. Then, in IEEE 802.11n *channel bonding* was introduced, meaning two contiguous channels could be used as a single 40 MHz channel. This results in over two times larger data rates, as the guard interval that was separating the channels can now also be used for transmissions. A novelty introduced in IEEE 802.11ac is the channel bonding of up to eight channels resulting in 160 MHz channels and over eight times larger data rates. Since its introduction, many works have been focused specifically on analytical models of the impact of channel bonding on the performance of 802.11n/ac WLANs. Kim et al. [26] study the achieved throughput of 802.11ac WLANs using a Markovian model similar to Bianchi's original work [7]. Like Kosek [8], they use additional states to represent an empty buffer. Their model takes into account channel bonding and it handles unsaturated networks with collisions, however the model is

limited to fully-connected WLANs. Barrachina-Muñoz et al. developed Markovian models for predicting throughput in saturated [5] and unsaturated [27] WLANs using different Dynamic Channel Bonding (DCB) policies in dense networks. They find that the DCB policy has a high impact on the networks performance and that inter-node dependencies are important even outside carrier sensing zones. They conclude that using the widest available channel can lead to maximizing the throughput of several nodes while creating starvation in others. Recently, Peng et al. [28] studied dense WLAN deployments. They propose a game-theoretic alternative to the classic DCB scheme with the goal of improving WLAN performance by better choosing the set of bonded channels. In a separate work, Bellalta et al. [3] use a Continuous Time Markov Network (CTMN) model to study the performance of channel bonding in dense WLAN deployments. The authors also conclude that using wider channels can increase the overall performance at the expense of creating localized starvation. Bellalta et al. [4] derive an analytical model for the performance evaluation of channel bonding in short-range WLANs. They show that channel bonding, especially DCB, significantly enhances the performance when the interference in the network is kept at a low or medium level. Several works have focused on channel bonding when a coexistence with legacy stations is present [29, 30, 6]. Sree Vasthav et al. [29] use two 20 MHz channels that can be bonded into a single channel for the 802.11ac users. The authors present a six-state, discrete-time Markov chain that explicitly models the 40 MHz and 20 MHz channels. Using the model, they derive the saturation throughput for a fully-connected network operating in an error-free environment. Han et al. [30] develop an analytical model for opportunistic channel bonding for WLANs containing 802.11ac and legacy stations. Khairy et al. [6] propose a renewal theory based analytical model. They show that the throughput of legacy stations is greatly degraded by the presence of stations with channel bonding and that 802.11ac offers a slightly friendlier resource sharing, while 802.11ax benefits from higher spectrum utilization.

As a result of the increased availability of 802.11n/ac devices, many authors focus on a purely experimental approach to the study of channel bonding. Deek et al. [2, 31] study how the efficiency of channel bonding is impacted by the interference, RSSI, physical obstacles and channel leakage, as well as data rate and the transport protocol in 802.11n networks. They perform an extensive experimental study to conclude that both the network and environmental conditions play an important role when choosing an intelligent channel bonding solution and that a node's knowledge about its environment can greatly help in choosing the right channel width. In a large testbed of 12 AP-station couples, Simić et al. [32] test the benefits of using larger channels in high-density, saturated WLANs. They find that the 80 MHz channels are best suited for highly-dense networks, while lower-density networks can take better advantage of separate 40 MHz or even 20 MHz channels by providing independent simultaneous transmissions. Using an 18-node 802.11ac testbed, Zeng et al. [33] study the energy efficiency and interference of channel bonding and compare it with MIMO setups that provide similar throughput gains. Among other findings, the authors conclude that when energy efficiency is a key factor in the network performance, the high energy consumption in idle mode on wider channels makes them a less efficient choice than using several Spatial Streams (SS) on a narrower channel (while still keeping the same throughput).

Lastly, there are numerous works that use simulation to evaluate the performance of channel bonding. Malekmohammadi [34] studies dense, saturated WLANs with the help of the ns-3 discrete-event simulator. The author concludes that using shared wider channels with lower transmission powers results in higher throughputs than using separate narrower channels. Moreover, the author shows how the starvation experienced by some nodes can be lessened by finely tuning the transmission powers and carrier sensing thresholds of those nodes. Park [35] studies the performance of DCB and Static Channel Bonding (SCB) using simulation. The author shows that DCB can outperform SCB when 802.11ac stations that use 80 MHz channels share the available bandwidth with 802.11a stations that use 20 MHz secondary channels. Gong et al. [36] use extensive simulation results to propose a MAC protection mechanism (now included as a mandatory feature of the IEEE 802.11ac standard) that aims at avoiding the hidden node problem, thus enhancing the throughput fairness on the application level.

In this paper, we propose an analytical model for the throughput achieved by each of the WLAN's APs. Our approach covers features that are seldom combined in a single model: *i*) we study unsaturated APs that have different traffic characteristics (saturation level, MCS index, channel width, frame aggregation rate), *ii*) we look at infrastructure-mode WLANs with arbitrary topologies (not fully-connected), and *iii*) we show that the model is robust to changes in the considered system (such as the incorporation of uplink traffic and hidden stations). Given the possibility to finely tune each AP's traffic, the proposed Markovian model is adapted to the performance evaluation of the current 802.11ac standard amendment, to the older 802.11n and to the upcoming 802.11ax amendment. More specifically, we focus on the performance evaluation of static channel bonding and show that, because of the high number of parameters that impact the efficiency of channel bonding, having a single guideline for optimal configuration remains an open issue. Using a series of examples, we show how the proposed model can be used to explore, and possibly improve, the efficiency of different channel bonding schemes.

3. WLAN performance modeling

Our modeling framework is based on the idea that any unsaturated WLAN with an arbitrary topology can be modeled as a collection of simpler saturated WLANs. We first provide several definitions for the considered WLAN and then a summary of the proposed model that implements CTMCs to represent the described WLAN.

3.1. System description

In this paper, we use WLAN to designate any network that uses an IEEE 802.11 standard amendment in the infrastructure mode. The WLAN is a collection of N nodes that are either APs or user stations (STAs). We assume that STAs are stationary. We define the *neighbors* of node n, $n \in [1, ..., N]$, as all the nodes whose transmissions can be detected by n. It follows that neighbors share the same radio resources and the access to those resources needs to be properly managed. Thus, WLANs implement the Distributed Coordination Function (DCF) that defines the procedure taking place before every frame transmission

in order to avoid collisions. Before starting a transmission, the node first listens to the medium to make sure it is not occupied by one of its neighbors. Should the medium be occupied, the node defers its transmission until it senses an idle medium. Nevertheless, neighbors can sometimes begin transmitting simultaneously leading to collisions and the potential loss of one or both frames (note that collisions can also occur in other scenarios such as topologies containing hidden nodes). As a result, knowing the neighbors of all the nodes in a given WLAN provides helpful information about how the available capacity is shared among the nodes. For this reason, we choose to represent the WLAN as a *conflict graph* in which an edge exists between any two nodes that are neighbors in the WLAN. Figure 1 depicts the conflict graph of a nine-node network. From looking at the graph, it is clear that some nodes have a more advantageous position as they have less neighbors with whom they have to share the medium, such as nodes 4 and 9. While the conflict graph is a convenient abstraction, clearly there are effects it does not capture, such as the second-order interference. The potential inaccuracies brought by this abstraction, among others, are discussed in the next section.

Most of the time, when a STA is connected to an AP, the vast majority of the traffic is downloaded from the AP to the STA [37, 38]. As a consequence, similarly to the works of [39, 40], our conflict graphs only show the APs of the WLAN, as they will generate the majority of the traffic. As long as the traffic generated by the STAs remains marginal, our ns-3 simulations have shown that we can exclude this traffic at a reasonably small loss in the accuracy of WLAN representation. On the other hand, if the traffic uploaded by certain STAs becomes significant, we can simply extend the conflict graph by incorporating these STAs.
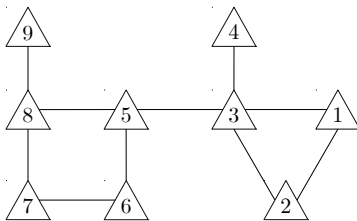

Figure 1: Conflict graph of a nine-node network.

The amount of medium resources a node can access depends not only on the number of neighbors it has, but also on their traffic demands as well as its own traffic demand. If node n has no neighbors, then it can potentially occupy the medium at all times and achieve its maximum throughput, calculated as:

$$\gamma_{n,\max} = \frac{L}{d_n} = \frac{L}{T_{DCF} + \frac{L+H_{\text{MAC}}}{R} + T_{ACK}} \ , \qquad (1)$$

where L is the payload length, d_n is the frame's total transmission time, T_{DCF} is the mean DCF overhead time (including average backoff time, physical header, SIFS, and DIFS), T_{ACK} is the transmission time of an acknowledgement frame, H_{MAC} is the MAC header length, and R is the data rate (i.e., the physical rate at which data is sent over the medium). Note that most of the values used in Eq. (1) are fixed in the IEEE 802.11 standard. The only parameters left

unspecified are the frame length and the data rate, that depend on mechanisms such as the frame aggregation rate, the MCS index, and the channel width. In this paper, we assume that the nodes are static (i.e., no mobility) resulting in constant data rates. We also consider fixed average frame length and static channel bonding. For a more detailed description of those features and the internal functioning of IEEE 802.11ac, we refer the reader to [41, 17].

3.2. Network decomposition

We use the rest of this section to detail our proposed CTMC modeling approach.

We begin by first defining the behavior of the network nodes and then show how we can use it to decompose our network. Any node that is not saturated experiences periods of activity where it occupies or wishes to occupy the channel, and periods when it is idle because of the absence of frames to be sent. We refer to these two periods as the node's ON and OFF regimes. Over a period of time, the evolution of the node's ON and OFF regimes will result in the node's *input rate*, calculated as $x_n = T_{ON}/(T_{ON} + T_{OFF})$, where T_{ON} and T_{OFF} are the proportions of time the node spent in the ON and OFF regimes in a given period, respectively. The node's input rate can be seen as the percentage of time it wishes to occupy the channel: if $x_n = 0$ then n never has any frames to be sent, if $x_n = 1$ then n wishes to permanently occupy the channel, i.e., n is saturated. Analogously, the node's *output rate*, y_n, is the proportion of time the node is occupying the channel. From the input and output rate of a node we can derive its demanded and achieved throughput, denoted γ_n and v_n, respectively. We calculate them as:

$$\gamma_n = x_n \times \gamma_{n,max} \quad \text{and} \quad v_n = y_n \times \gamma_{n,max} \ . \tag{2}$$

In a network of N nodes, there can be at most 2^N different combinations of nodes being in the ON or OFF regime. These combinations represent our *subnetworks*. In every subnetwork, a node is either saturated (if it is ON) or is completely removed from the subnetwork (if it is OFF). Thus, the subnetworks only have saturated nodes and usually containing less nodes than the original network, making them easier to solve. Depending on the nodes' T_{ON} and T_{OFF} distributions, some subnetworks are much more likely to occur than others. For example, in a network where all $x_n = 1$, all nodes have a frame to send at all times and only a single subnetwork exists. This leads us to calculate the subnetwork's *occurrence probability*. Let us denote by b_i ($i = 1, \ldots, 2^N$) the i-th subnetwork whose occurrence probability is calculated as:

$$\beta_i = \prod_{n|b_i(n)=ON} x_n \prod_{m|b_i(m)=OFF} (1 - x_m) \ , \tag{3}$$

where $b_i(n)$ is the regime of node n (ON or OFF) in subnetwork b_i under the assumption that the nodes' ON and OFF regimes are independent.

3.3. Representing a subnetwork as a Markov chain

The next step is to solve all subnetworks independently of each other. We choose to model the subnetworks as CTMCs and in this section we detail how we calculate the states and transition probabilities of those Markov chains.

3.3.1. Finding the states of the Markov chain, their possible transitions and transition probabilities

Each subnetwork defines whether a node has a frame to be sent (the node is ON) or it has none (the node is OFF). However, this does not inform us whether the node is currently transmitting or not, as an ON node can be either in transmission or waiting to access the medium. For this reason, we define the set of the network's *sending states*, denoted by S. S contains all combinations of sending states $s_k \in \{\mathbf{0}, \mathbf{1}\}^N$ that satisfy the rule that two neighboring nodes cannot be simultaneously sending. Note that $s_k(n) = \mathbf{1}$ means node n is sending in sending state s_k. We denote by S_i the subset of S containing all the sending states associated to the subnetwork b_i such that they satisfy:

1. if $b_i(n) = OFF$ then node n is never sending ($s_k(n) = 0, \ \forall s_k \in S_i$)

2. if $b_i(n) = ON$ and n has no sending neighbors, then n is sending.

The first rule is fairly easy to deduce, since a node that has no frames to send cannot be in transmission. The second rule is inspired by CSMA/CA's functioning and the fact that CSMA/CA tends to maximize the number of concurrent transmissions in order to achieve higher spatial reutilization [42]. For example, in a three-node fully-connected network and its subnetwork where all three nodes are ON, S_i contains three sending states: $s_1 = (\mathbf{1 \ 0 \ 0})$, $s_2 = (\mathbf{0 \ 1 \ 0})$, and $s_3 = (\mathbf{0 \ 0 \ 1})$.

Once we have the set of sending states for every subnetwork, we proceed by calculating the probabilities of the transitions between them. We consider that a transition from sending state s_k to s_ℓ is possible if:

1. one node n changes from $s_k(n) = \mathbf{1}$ to $s_\ell(n) = \mathbf{0}$ and one node n changes from $s_k(n) = \mathbf{0}$ to $s_\ell(n) = \mathbf{1}$, or;

2. $k = \ell$, i.e., self-transition,

and all other transitions are considered to have a probability of zero. These two rules capture the fact that it is highly unlikely for two nodes to independently start (or finish) transmitting at the exact same time.

It should be noted that in real-life WLANs, as well as in the simulations used to validate our modeling approach, other sending states are possible. A case in point are collision states where all three nodes of the fully-connected network transmit at the same time. By not incorporating these states in our model, we aim at keeping the latter simple at a small expense in accuracy, thus resulting in a fidelity versus simplicity trade-off. Moreover, we model changes between sending states as immediate, while in WLANs and in simulation there are often transitory sending states that can occur. Our validation section will however show that our simplifying hypotheses result in only minor deviations from the results obtained in simulation.

If the transition from state s_k to s_ℓ has a non-zero probability, then we approximate that probability as:

$$P_{k,\ell} = C \prod_{n|s_\ell(n)=\mathbf{1}} \frac{1}{1 + \sum\limits_{m \in w_n} \mathbb{1}_{b_i(m)=ON}}, \tag{4}$$

where C is a normalizing constant that ensures $\sum_{\ell \geq 1} P_{k,\ell} = 1$. We denote by w_n the set of neighbors of n that compete with n for medium access, i.e., nodes that are ON and that do not have a currently transmitting neighbor. In short, we consider that n and its competing neighbors have an equal chance of gaining medium access. An interesting feature of Eq. (4) is that the probability of the transition $P_{k,\ell}$ depends only on s_ℓ and it will be the same for any departing state s_k.

3.3.2. Calculating the steady state probabilities and introducing holding times

So far, for every subnetwork b_i we have a DTMC that is fully defined by its states and their transition probabilities. We now proceed by describing how we solve each DTMC and how we then transition to a CTMC.

Because some transitions between sending states are impossible, it may happen that the DTMC is not irreducible. Should that be the case, we simply divide the DTMC into its irreducible components and solve them separately. We denote by M_i the number of irreducible DTMCs in subnetwork b_i. We enumerate the irreducible DTMCs by c_i^m, $m \in [1, ..., M_i]$ and we use π_i^m to denote their corresponding steady state probabilities.

Next, we transition from discrete to continuous time by introducing the *holding times* of sending states. In the case of a WLAN, it is logical that the holding times of the sending states represent the durations of the frame transmissions. As several nodes are potentially transmitting in a given state s_k in the subnetwork b_i, we calculate holding time of s_k, $h_i(k)$ as:

$$h_i(k) = \frac{1}{\sum\limits_{n \mid s_k(n) = \mathbf{1}} \frac{1}{d_n}} \; , \tag{5}$$

where d_n is the transmission duration of node n as defined in Eq. (1). In simple terms, the holding time of a sending state is simply the average of the transmission times of the nodes sending in that state. As a result, on average, the modeled network spends more time in sending states whose transmission times are longer. Then, we obtain the solution to each irreducible CTMC simply by applying:

$$\mu_i(k) = \frac{\pi_k h_k}{\sum\limits_{s_\ell \in S_i^m} \pi_\ell h_\ell} \tag{6}$$

where $S_i^m \in S_i$ is the set of sending states associated to the m-th irreducible Markov chain (clearly, $S_i = S_i^m$ when the subnetwork has a single irreducible Markov chain).

It is important to note that in the case of multiple irreducible CTMCs, we need to calculate the entry probability of each chain. In such cases, we refer once again to the work of Durvy et al. [42] and try to replicate the idea that CSMA/CA networks tend to maximize the number of concurrent transmissions. Our reasoning is that we assign a larger entry probabilities to the irreducible CTMCs with a larger number of transmitting nodes. We denote the entry probability of the m-th irreducible CTMC of subnetwork b_i by $\widetilde{\omega}_i^m$ and we refer the reader to [16] for a detailed explanation of its calculation.

3.4. Calculating output rates and achieved throughputs

Finally, we can calculate node n's output rate using Equation (7) that is simply the sum product of the probabilities of all the *subnetwork × Markov chain × sending state* combinations in which node n is sending. In simple terms, we first decompose the original network and then look at all the subnetworks one by one to obtain the output rate y_n as the sum of the probabilities of all the sending states where node n is sending. It should be noted that the passage from DTMCs to CTMCs allows us to largely simplify the computation of a node's achieved throughput. When using CTMCs, the node's output rate y_n is the percentage of time the node occupies the channel and obtaining its achieved throughput is as simple as multiplying y_n by the node's maximum achievable throughput: $v_n = y_n \times \gamma_{n,max}$.

$$
y_n = \sum_{i=1}^{2^N} \left\{ \mathbb{1}_{b_i(n)=ON} \times \beta_i \times \sum_{m=1}^{M_i} \left(\widetilde{\omega}_i^m \times \sum_{k|s_k \in S_i^m} \left(\mathbb{1}_{s_k(n)=\boldsymbol{1}} \times \mu_i^m(k) \right) \right) \right\} .
$$
(7)

4. Model validation

We use this section to validate the accuracy of our model by comparing its predicted throughput to simulation results. The model validation consists in comparing the results delivered by a realistic simulator with those estimated by the proposed model. All our simulations are performed in the ns-3 discrete-event network simulator [43] using the IEEE 802.11ac standard amendment. Note that the model supports 802.11n, ac, and ax, and we choose 802.11ac as it is currently the most recent ratified 802.11 standard amendment that is also widely available in consumer products. In our simulations, APs send traffic to their associated stations using different MCS indexes, channel widths, frame aggregation rates, and input rates. For the sake of completeness, our validations also include scenarios in which the STAs are generating a portion of the traffic, i.e., uplink traffic, and the topology contains several hidden nodes.

The presented numerical results correspond only to a fraction of the networks we have tested. We found that the two topologies shown in this section are representative of the mean results we obtain when validating our model. Moreover, these networks contain certain features, such as diverse node degrees and complex inter-clique configurations, that make them interesting examples to be used in validation.

4.1. Simulation setup and relative error definition

All nodes generate datagrams of 1500 bytes using different ON and OFF periods that result in different input rates. In our examples, we consider that nodes are sending only Best Effort traffic and use the default MAC parameters set in the ns-3 simulator and they are using static channel bonding. We remind that collisions are not explicitly represented in our model, yet they are present both in real-life WLANs and in the simulation results presented throughout Section 4. Each AP has a single associated station. The specific parameters regarding the rate of frame aggregation, channel bonding, MCS, and the input rates of all nodes are summarized in Tables 1 and 2 for the networks in Fig. 1

and Fig. 2, respectively. Since it is our experience that having homogeneous parameters values tends to simplify the WLAN performance modeling effort, we purposely chose highly heterogeneous values.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
MCS	9	8	7	1	4	6	5	2	3
Channel width (MHz)	40	20	80	40	20	80	40	20	80
Frame aggr.	4	4	8	2	8	4	2	4	8
Input rate	0.5	0.2	0.7	0.4	0.9	0.3	0.8	0.6	0.9

Table 1: Traffic parameters for the nine-node network.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
MCS	9	8	5	1	3	5	7	2	4	6
Channel width (MHz)	40	80	20	40	20	80	40	80	20	40
Frame aggr.	2	4	8	8	8	4	4	2	2	8
Input rate	0.5	0.6	0.2	0.4	0.9	0.3	0.8	0.7	0.9	0.1

Table 2: Traffic parameters for the ten-node network.

The proposed model, the ns-3 code, and the rest of the scripts necessary to reproduce our simulations are made available for download online [44]. In order to compare the model to the simulation data, we need a dataset of measurements and a definition of the relative error we use. We obtain our dataset by fixing the input rates of all nodes except for one, and we then vary the input rate of that one node in the interval [0, 1] (recall that $x_n = 0$ and $x_n = 1$ mean that node n has no frames to send and node n is saturated, respectively). The varying input rate takes 11 different values (0, 0.1, 0.2, ..., 1). To make sure we obtain a representative value of the nodes' output rates we execute 20 simulation runs for every scenario. In total, we obtain $N \times 11 \times 20$ different samples for a network of N nodes. We then run our model using the same parameters as the simulation scenario and we obtain our predicted output rate, denoted by $\widetilde{y}_n$. Finally, we calculate the relative error as:

$$e = \sum_{n=1}^{N} \frac{|\widetilde{y}_n - y_n|}{y_n} \ , \tag{8}$$

where y_n is output rate of node n provided by the simulator, i.e., our ground truth. Note that when y_n and $\widetilde{y}_n$ have a value lower than 0.1 we remove that data point from the relative error dataset, as even small differences between the two values result in abnormally large relative errors. This removal concerns only a small fraction of the samples and does not heavily impact the accuracy of the calculated error as the number of samples for each network is sufficiently large.

4.2. Downlink traffic

This section presents the results for the nine-node network of Fig. 1 and the ten-node network of Fig. 2 in the scenario when only the APs are generating traffic. The cumulative distribution function (CDF) of the relative errors for the nine and ten-node network is shown in Fig. 3. In general, we obtain a mean

relative error of 9.03% for the nine-node network and 6.48% for the ten-node network, with a median of 7.09% and 4.86%, respectively. In both networks the relative error is rarely greater that 20% and it never exceeds 50%. Table 3 shows the distribution of the relative error for the nine and the ten-node network.
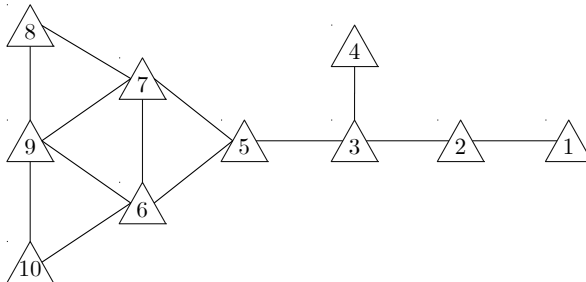

Figure 2: Conflict graph of a ten-node network.

Network	Mean	Median	≤ 5%	≤ 10%	≤ 20%	≤ 30%	> 30%
Nine-node	9.03%	7.09%	33.62%	68.82%	92.24%	95.98%	4.02%
Ten-node	6.48%	4.86%	51.39%	80.44%	97.81%	98.83%	1.17%

Table 3: Distribution of the relative error for the nine and ten-node networks.

Figures 4a and 4b show the evolution of the achieved throughputs of several nodes when node 1 and node 10 vary their demanded throughput, i.e., input rate, in the nine and ten-node network, respectively. Note that, in the interest of clarity, only a part of the nodes' throughputs are shown in the figures. We chose these nodes as they are representative of the general behavior of the entire network, i.e., neither the best nor the worst accuracy, and also they exhibit the most interesting behavior as other nodes tend to be less affected by the occurring changes. Both figures show node n's estimated throughput in solid lines (named Model n), the average simulation scenario throughput in dashed lines (Mean simu n), and the throughput for the 20 scenario runs in discrete points (Samples simu n).

In the nine-node network, the varying demanded throughput of node 1 mostly impacts its one-hop neighbors. Nodes 2 and 3 need to forgo a part of the resources they use in order to accommodate the newly increased demand of their neighbor node 1. Nodes located at several hops from node 1, such as node 7, are only slightly affected by the dynamics happening at the right-hand side of the network. As node 1's demand varies, node 7 keeps an almost constant throughput both in the simulation and in our estimation. We notice similar behavior in nodes 5 to 9.

The results for the ten-node network show similar tendencies, presented in Fig. 4b. This time we vary the input rate of node 10. As node 10 approaches saturation, we notice the behavior of a flow-in-the-middle topology composed of nodes 8, 9, and 10. In these topologies the edge nodes ally in order to maximize their throughput, at the expense of a starving node in the middle [12]. As node 10 goes from 0 Mbps to almost 80 Mbps, node 8 nearly doubles its initial throughput and node 9 slowly starts to experience starvation. Remote nodes show almost no change in their achieved throughputs, such as nodes 1 to 5. As
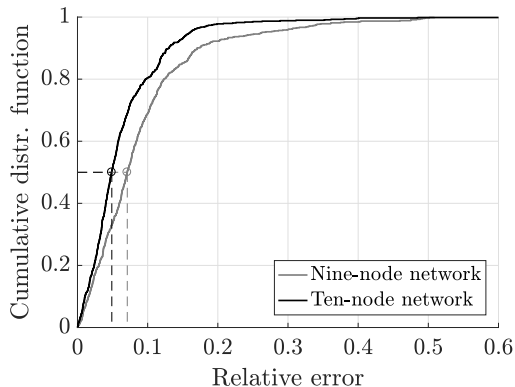
Figure 3: CDF of the relative error.

in the nine-node network, the proposed model manages to accurately capture the tendencies of all nodes. Finally, it should be noted that a single 60 seconds simulation run of the ten-node network with the parameters in Table 2 in ns-3 takes, on average, 21 minutes to execute. The unoptimized implementation of the proposed model takes roughly 3 seconds for the same network. We used our laboratory's Dell Inc. PowerEdge R630 workstation with 2 Intel$^{(R)}$ Xeon$^{(R)}$ CPUs E5-2697 2.60GHz, 28 cores, and 252 GB RAM.

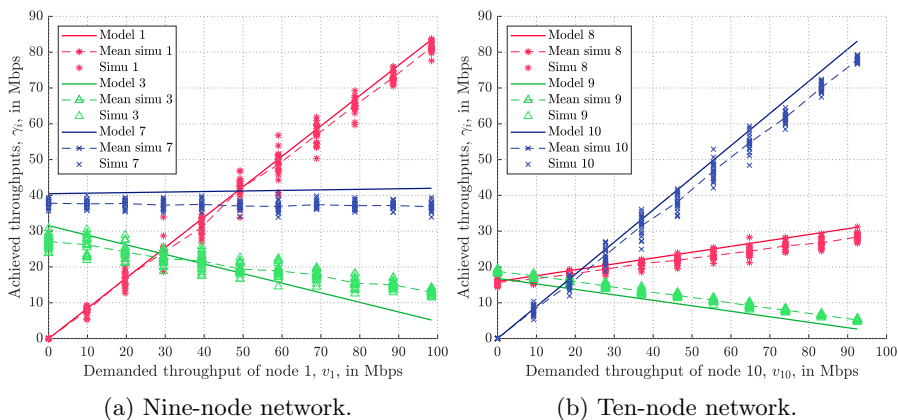(a) Nine-node network.

(b) Ten-node network.

Figure 4: Simulation and model results for the nine and ten-node networks. "Model N" denotes the achieved throughput of node N as predicted by our model, "Mean simu N" is the average throughput obtained in simulation, while "Simu N" shows all the data points for the 20 simulation replications.

In summary, we found that our model provides relative errors generally smaller than 10% for networks of different topologies and different levels of node heterogeneity. Moreover, we obtain our model's solution in a fraction of the time needed by an ns-3 simulation. The following subsection explores the precision of the model when hidden stations and uplink traffic are added to the network.

15

4.3. Downlink and uplink traffic with hidden nodes

The hypothesis that only downlink traffic should be modeled may be perceived as restrictive in some scenarios and we use this section to present a validation example that includes uplink traffic. It is worth keeping in mind that Internet traffic is highly asymmetrical [37, 38], i.e., stations download much more data than they upload. We did our own measurements by sniffing the traffic generated in our graduate school, in a residential building, and in Lyon's busiest Part Dieu train station. In all the locations, we found that uplink traffic consistently represents less than 10% of the total generated traffic. According to the Cisco Visual Networking Index [1], this trend will only magnify in the future. Thus, in our simulations, when STAs are generating traffic, they use the same MCS index, channel width, and rate of frame aggregation as the corresponding AP, and they generate a tenth of the AP's traffic. Nevertheless, adding uplink traffic helps in testing the robustness of the model on more complex networks and allows us to better assess its impact on the general behavior of the network. Moreover, given that adding uplink traffic increases the number of nodes competing for medium access, certain parts of the nine-node network will have up to 10 competing neighbors. As a result, more collisions are expected to happen than in scenarios with only downlink traffic.

In this section, we present a scenario where the stations are generating 10% of the traffic originally generated by the APs. This uplink traffic uses the parameters defined for the downlink traffic in Table 1 with an input rate $x_j = \frac{x_i}{10}$, where x_j and x_i are the input rates of station j and AP i, and station j is associated with AP i. Furthermore, we move some of the stations away from the APs so that they become hidden terminals for other stations and APs in the network[1]. The resulting network is shown in Fig. 5. This new conflict graph contains the APs (depicted with triangles) and their associated stations (depicted with squares). The blue dotted lines between the stations and the APs indicate association and the black continuous lines indicate conflict, as they did before. The reader will notice that the conflicts between APs remain exactly the same as defined in Fig. 1. However, we need to introduce special notation for the cases where a station becomes a hidden terminal. As a general rule, all stations have the same conflicts as their corresponding AP. Exceptionally, when a station is far away from its AP, it may become a hidden terminal for some APs and stations. We use the red dotted lines to represent such connections. For example, the station A has moved away from AP1 so that it becomes a hidden terminal for AP2 and station B. Station A is still within the detection zone of AP3 and station C, therefore we add no special notation. We notice that there are three occurrences of hidden terminals: *i*) station A is a hidden terminal for AP2 and station B, *ii*) station D is a hidden terminal for AP3 and station C, and *iii*) station I is a hidden terminal for AP8 and station H.

The results we obtain when we vary the throughput of node 1 are shown in Fig. 6. As expected, the increase in demanded throughput of AP 1 (and in consequence station A) mostly impacts its one-hop neighbors, APs 2 and 3, and to a lesser extent APs further away, such as node 7. We first notice that the

---

[1]Two transmitting nodes become hidden terminals when they cannot detect each others' transmissions, but one (or both) of their corresponding receivers can detect both transmissions. As a result, collisions can potentially occur at the receiver's end.
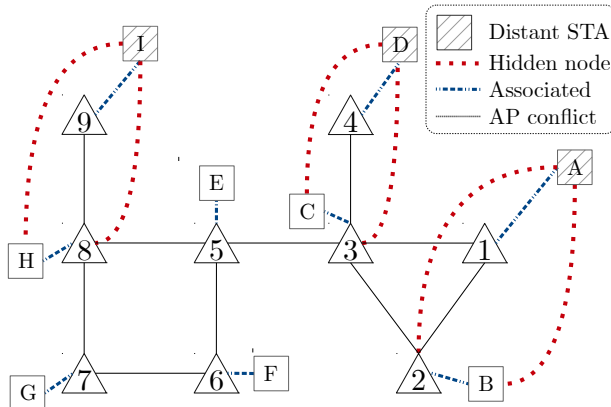
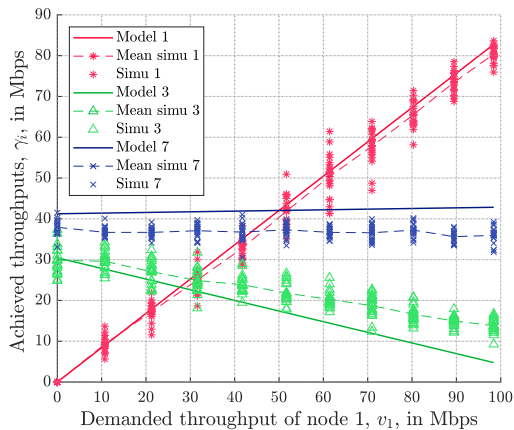Figure 5: Conflict graph of a nine-node network with hidden nodes.

Figure 6: Simulation and model results for the nine-node network of Fig. 5.

obtained throughput of all APs is remarkably similar to the one in the scenario with no hidden terminals or uplink traffic. In general, the 10% of uplink traffic only have an average of 5% impact on the overall AP throughput. In order to incorporate the uplink traffic, we consider that every node n in our model is composed of AP i and its associated station j. The input rate of that node, assuming independence of the input rates of the AP and the station, is calculated as:

$$X_{i,j} = x_i + x_j - x_i x_j \ . \tag{9}$$

The output rate is simply the sum of the AP's and the station's output rates. This approach allows us to omit explicitly representing the hidden stations and uplink traffic in our model and still obtain accurate predictions. The predictions made by our model remain accurate with relative errors of the same order as for the scenario with no hidden terminals or uplink traffic. We obtain a mean relative error of 11.94% and a median of 9.18%.

These results support the idea that, for the sake of our study, both uplink traffic and hidden terminals can be omitted from the model, provided that

17

the uplink traffic is negligible compared to the downlink traffic. Given our current usage of the Internet, and the predictions for its future development, we believe that the hypothesis on traffic asymmetry is acceptable and continue to concentrate our work on the sharing of resources between APs. However, should uplink traffic become non-negligible, one possible way to incorporate it in our model is to add the station generating this traffic as a node in the conflict graph.

5. Guidelines for channel bonding

Given the complexity of channel bonding in 802.11 WLANs, we propose a series of three scenarios that gradually increase in complexity. Similarly to the previous section, the results we present here focus on the 802.11ac standard amendment, but the findings apply to all amendments that offer channel bonding. We begin by studying how well channel bonding performs when the nodes' MCS indexes and frame aggregation rates are varying. Next, we investigate the impact of the network's topology on the performance of channel bonding. Finally, we present how our proposed modeling approach can help to find the optimal channel bonding configuration for a larger network.

5.1. Influence of MCS and frame aggregation on channel bonding

We consider a WLAN containing four APs in close proximity, as shown in the conflict graph in Fig. 7a. We consider a WLAN whose topology consists of four APs in close proximity. Note that, depending on the channel (bonding) configuration, different conflict graphs can be obtained for this WLAN. For example, if the four APs of Fig. 7a share a single 80 MHz channel, the corresponding conflict graph is the one in Fig. 7a. On the other hand, if they use four non-overlapping 20 MHz channels, we obtain the edgeless graph in Fig. 7b[2]

(a) Fully-connected graph (80 MHz).

(b) Edgeless graph (20MHz).
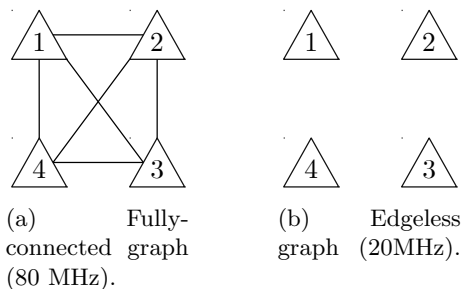
Figure 7: Conflict graphs for the four-node network of Section 5.1.

All APs are saturated and generate datagrams of 1500 bytes on an error-free channel. We obtain the AP's achieved throughputs through simulations in the ns-3 network simulator [43]. By symmetry, the four APs, on average, obtain the same throughput, meaning that the only quantity we need to monitor is the

---

[2]In order to keep the figure readable, we exclude results for channels of 40 MHz.

aggregate throughput, V, calculated here as:

$$V = \sum_{n=1}^{4} v_i \, , \tag{10}$$

where v_i is the achieved throughput of AP i.

Figure 8 shows the obtained aggregate throughput as a function of the selected MCS indexes and frame aggregation rates, for the two conflict graphs of Fig. 7. We test MCS indexes 0 to 8 and nodes use MPDU aggregation with aggregation rates (i.e., maximum number of MPDUs in an aggregated frame) of 2, 8, 16, or the maximum allowed number of MPDUs. We refer the reader to Appendix A for a detailed explanation of IEEE 802.11 regulation regarding MPDU aggregation and the maximum number of aggregate MPDUs.
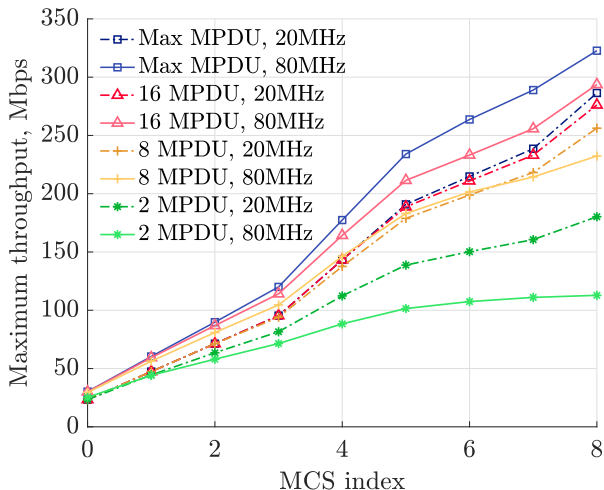

Figure 8: The maximum aggregate throughput as a function of MCS with varying channel width and frame aggregation rate. Full and dashed lines correspond to the 80 MHz and 20 MHz configurations, respectively.

In agreement with previous works on the performance of frame aggregation [14, 17], Fig. 8 shows that the aggregate throughput always increases with higher frame aggregation rates. The same is true for increasing the nodes' MCS indexes since the stations and APs are close enough to support the denser modulations. Moreover, we notice that the higher-order MCS indexes are more impacted by the changing frame aggregation rates and channel widths. Specifically, the best and worst aggregate throughputs for MCS0 are around 30 Mbps and 23 Mbps. In contrast, when using MCS8 we see an almost three-fold increase in throughput between the lowest (112 Mbps) and the highest available throughput (313 Mbps).

Perhaps more interestingly, Figure 8 shows that for any MCS index the highest and the lowest values for the aggregate throughputs are achieved using a single 80 MHz channel. The lowest throughput is achieved with the aggregation rate of 2 MPDUs, and the highest with the maximum possible aggregation rate. Thus, when we have no information about the frame aggregation rate, using 20 MHz channels appears to be the safer option.

We identified several reasons that explain the described behavior and conclude that when choosing an 80 MHz or a 20 MHz channel there is a trade-off to be made between four factors: the capture effect, the maximum aggregation rate, the guard intervals, and the DCF overhead. When all four APs share the 80 MHz channel, sometimes transmissions will happen simultaneously and benefit from the capture effect. Moreover, the 80 MHz channel includes the guard intervals separating the different 20 MHz channels, resulting in data rates slightly over four times higher that a single 20 MHz channel. Because of the higher data rates, transmissions are shorter and the maximum number of MPDUs we can aggregate tends to be higher on an 80 MHz channel. However, another side effect of short transmissions is that the DCF overhead on the 80 MHz channel becomes too important relative to the data transmission time. This gives the 20 MHz channels an advantage as, on average, we send four frames on every DCF overhead.

In summary, the analysis of this network of four APs in close proximity indicates that channel bonding effects are most visible when APs use MCS indexes higher than 3. Moreover, using separate smaller channels appears to be a more suited option when APs aggregate less than 8 MPDU whereas a single wider channel tends to become more efficient when more MPDUs are aggregated. Lastly, this first example demonstrates the difficulty of finding the optimal channel configuration even for an elementary network.

5.2. Influence of network topology on channel bonding

We are now interested in how introducing the notion of topology affects the performance of channel bonding. In the previous section, all APs were statistically identical and obtained, on average, the same throughput. We now investigate the impact of channel bonding on issues of fairness in channel access in networks with arbitrary topologies, i.e., not fully-connected or edgeless conflict graphs. We use *proportional fairness* to quantify the degree of equity in resource sharing experienced by the APs. We calculate the proportional fairness for a given demanded/achieved throughput combination as:

$$PF = \sum_{n=1}^{N} \log \frac{v_n}{\gamma_n} \ , \tag{11}$$

where γ_n and v_n are the demanded and achieved throughput of AP n, respectively. When maximizing proportional fairness we are looking for the channel allocation that renders the PF value closest to zero, i.e., allow APs with higher throughput demands to have higher obtained throughputs without creating starvation in low-demanding APs.

We study the three four-node network topologies depicted in Fig. 10, and we consider the scenario where every AP can choose one of the seven radio channels depicted in Fig. 9. The parts of the channels in Fig. 9 that are vertically aligned use the same frequency, e.g., the 80 MHz channel **g** contains the four 20 MHz channels **a** to **d**, as well as channels **e** and **f**. We use a quadruplet to denote the current channel allocation, e.g., {**a, d, f, e**} denotes the allocation of channel **a** to AP 1, channel **d** to AP 2, channel **f** to AP 3, and channel **e** to AP 4. We assume that all APs are saturated, and they have an aggregate rate of 8 MPDUs using MCS0 or MCS8. Because there are 4 APs and 7 possible channels, the
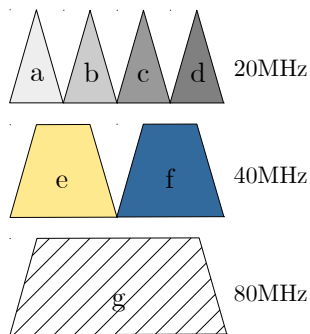
Figure 9: Enumeration of available channels for the example of Section 5.2.

total number of combinations amounts to $7^4 = 2,401$ on which we run our model. Note that given the speediness of the model, all the channel allocations can be individually run to obtain the resulting throughputs. Clearly in real-life applications, such brute-force approaches may quickly be rendered inefficient as the network grows in size and should be replaced by heuristics..

Once a given channel allocation is chosen for a network topology, we obtain that allocation's conflict graph in which we keep the edges between the APs whose channels are at least partially overlapping. For example, if the allocation {**a, b, c, d**} is chosen for the network in Fig. 10a, then all APs use separate channels which renders an edgeless graph, like the one in Fig. 7b. If the current channel allocation is {**g, a, b, g**} then we only keep the edges between APs 1 and 2 and between APs 3 and 4. In reality, the obtained conflict graph may also depend on the chosen channel width, as when the same power is spread over a wider channel some APs may not be neighbors anymore or might become hidden nodes [2]. Without loss of generality, we choose in this example to keep the edges from the network's topology whenever (partially) overlapping channels are used.
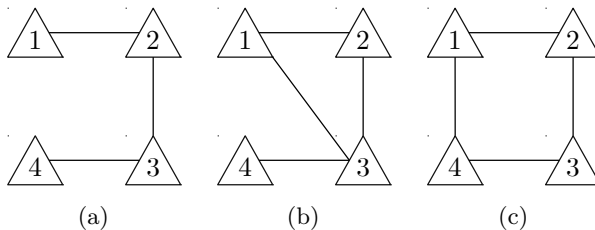
Figure 10: Four-node network topologies of Section 5.2.

We begin with the four-node chain network in Fig. 10a in which all APs are using MCS8. For this scenario, we obtain the same channel allocation {**e, f, e, f**} as the optimal solution for maximizing both V and PF. The conflict graph arising from this allocation is denoted (a1) in Fig. 11. Note that we have tested this example in simulation, i.e., we ran the 2,401 possible allocations, and found that the solution proposed by our model does belong to the set of several equivalent solutions that maximize both V and PF. Our modeling results also show that if one of the edge APs falls back to the MCS0, then the optimal

Create starvation by replicating the FIM topology

Figure 11: Conflict graphs for several commonly recurring channel allocations of Section 5.2.

allocation does not change. However, should AP 2 (or 3) switch to MCS0, the model proposes us to switch to the channel allocation {**g, g, e, f**} (or {**e, f, g, g**}) if we wish to optimize V, shown as (a2) in Fig. 11. This allocation creates starvation in AP 2 (or 3) as this AP is now in a flow-in-the-middle (FIM) configuration [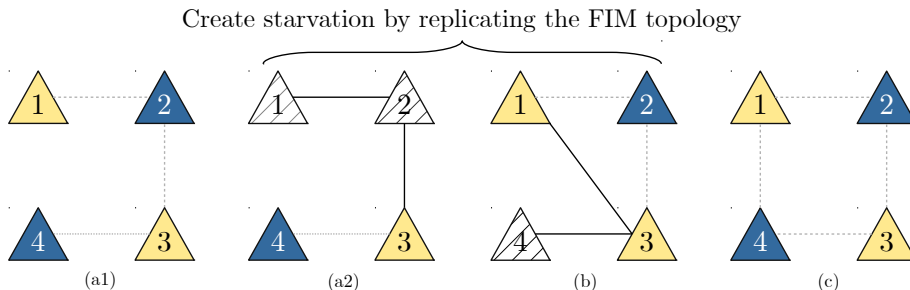18]. The edge AP takes advantage of having only one neighbor that never gains medium access and increases its obtained throughput to almost its maximum achievable throughput. While this solution maximizes throughput, it does so by creating starvation in one AP. If we wish to maximize proportional fairness, it is still better to keep the nodes on separate channels and use the allocation {**e, f, e, f**}.

Next, we test the network topology in Fig. 10b. Once again we find that many of the maximum throughput solutions consist in creating starvation in the middle AP 3. In fact, whenever AP 3 and AP 1 (or node 2) both use MCS8, then the maximum throughput solution is always {**e, f, e, g**}, i.e., AP 3 shares the channel with AP 1 and AP 4 and thus experiences FIM starvation. The corresponding conflict graph is denoted (b) in Fig. 11. In this topology, it is logical that maximizing throughput often means creating starvation in AP 3, as it will, on average, allow two other APs to transmit simultaneously. In general, we find that the solutions that maximize throughput tend to use larger channels (channels **e, f**, and **g**) than those maximizing fairness. When maximizing fairness, we often have to separate the APs on different channels, so that the unfavorable topological position of an AP, such as AP 3, does not prevent it from achieving a higher throughput.

Our last network topology is a cyclic network of four APs depicted in Fig. 10c. Except for the configurations when all APs or all nodes but one use MCS0, the solution that maximizes both aggregate throughput and fairness is to use the {**e, f, e, f**} channel allocation, as shown by the conflict graph (c) in Fig. 11. Interestingly, in this network, the maximum aggregate throughput is never achieved by creating starvation. Because of the circular nature of the network, it is impossible to recreate the FIM topology without forcing starvation in two APs. However, in that case the two high-throughput APs cannot compensate for the low throughput of the two others, and as a result the solution is to always have a more fair resource sharing.

From the study of these four-node networks, we conclude that the network's topology can highly influence the optimal choice of channel allocation, even in the simplified conditions of a small saturated network. In the next section we extend this study by applying our model to a larger unsaturated network.

5.3. A practical case study

Having studied several simplified network configurations, we are now interested in the impact of channel bonding on a larger network where APs are unsaturated and have more diverse sets of MCS indexes and aggregation rates. We study the network in Fig. 2 using the parameters defined in Table 4. Each of the ten APs can choose to use one of the three channels in Fig. 13, where channels **a** and **b** are 80 MHz wide and channel **c** is 160 MHz wide. Analogously to the four-node scenarios, we define the new conflict graph for every channel allocation by keeping the conflicts between nodes belonging to at least partially overlapping channels. This setup results in $3^{10} = 59,049$ possible channel allocations.

(a) Maximizing aggregate throughput, V.

(b) Maximizing proportional fairness, PF.

Figure 12: Channel allocations for the ten-node network of Fig. 2 considered in Section 5.3.

Figure 13: Enumeration of available channels for the example of Section 5.3.

Figure 12 shows the results we obtained when we maximize the aggregate throughput and the proportional fairness, respectively. We first notice that the two metrics provide the same configuration for the first four APs. This is the low-density part of the network, where it is possible to completely separate the four APs even with only two non-overlapping channels. Although only AP 1

can fully obtain its demanded throughput on a 80 MHz channel, all four APs achieve at least 30% of their demanded throughputs and none of them are in starvation. On the left-hand side where the network becomes denser, we can no longer provide a separate channel for every AP and the two performance metrics provide very different solutions. If we maximize the aggregate throughput V, we still obtain a connected component containing APs 5 to 10. In this component, APs 6 and 9 are analogous to the middle node of an FIM topology and they experience starvation by obtaining a throughput of 0.7 Mbps and 1.1 Mbps while demanding a throughput of 45 Mbps and 50 Mbps, respectively. Their neighbors, however, obtain throughputs as high as 79 Mbps and the aggregate throughput is $V = 262$ Mbps. When maximizing the proportional fairness PF, we notice that a chain network equivalent to the one in Fig. 10b is created using APs 6, 7, 8, and 10. Characteristically for the four-node chain, the throughput is shared fairly as long as the node's throughput demands are somewhat similar as it is too costly to create starvation in the two middle nodes for the benefit of the edge nodes. However, the aggregate throughput in this case is almost 60 Mbps lower than the maximum throughput configuration.

In general, we conclude that the optimal solution is highly dependent on the AP's demanded throughputs and their levels of saturation by testing several sets of configurations. However, we find that when optimizing the aggregate throughput V, the solution often involves using larger channels at the expense of creating starvation in some APs. The use of smaller (and separate) channels when maximizing proportional fairness PF leads to smaller aggregate throughputs, as we avoid the network's inherent working that maximizes the number of concurrent transmissions.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	V	PF
MCS	8	7	4	0	2	4	6	1	3	5		
Aggr. rate	2	4	8	8	8	4	4	2	2	8		
γ_n	40	60	50	10	25	45	70	20	30	100		
v_n: max V	40	48.7	34.4	5.8	17.5	1.1	44.0	10.7	0.7	79.7	282.6	-4.6
v_n: max PF	40	48.7	34.4	5.8	17.8	12.2	9.5	8.8	20.4	27.7	225.3	-3.3

Table 4: Input parameters, demanded and obtained throughputs for the ten-node network considered in Section 5.3.

6. Conclusions

In this paper, we presented a continuous time Markovian model for IEEE 802.11 networks that estimates the individual throughput of every AP. Our modeling approach handles networks with arbitrary topologies, MCS indexes, frame aggregation rates, channel widths, and saturation levels. By means of simulation, we show that our model achieves mean relative errors of around 10% for networks of about ten nodes and with diverse node configurations. The fast execution of our model for networks of a dozen nodes allows us to consider applications such as channel bonding or channel assignment. We study the impact of the MCS indexes, the frame aggregation rates, the saturation levels, and the network topology on the efficiency of channel bonding. The model helps us develop some insights into the best practice in channel bonding given a performance metric such as the maximization of throughput or fairness. We test these

insights by identifying the optimal channel bonding combination in a WLAN containing a diverse set of nodes and for different network performance metrics. We conclude that the optimal solution is highly dependent on the particular network configuration and, above all, on the network's topology. However, we find that, in general, larger channels are better suited for throughput maximization and smaller (and separate) channels render higher fairness. The speed and accuracy of the proposed model lead us to believe that its implementation (be it in centralized WLAN controllers or by network administrators) could help to test and improve the performance of 802.11-based WLANs offering channel bonding. Future extensions of this work may include station mobility resulting in a distribution of possible MCS indexes per node.

Acknowledgements

Appendix A. A-MPDU in IEEE 802.11ac

The 802.11 standard imposes new regulation on the maximum frame length when using aggregation [41]: i) at most 64 MPDUs can be aggregated in a single frame, ii) the duration of the Physical Protocol Data Unit (PPDU) must not exceed 5,484 μs, and iii) the maximum length of the aggregate frame cannot be more than 1,048,575 bytes. Thus, we consider that the aggregation rate is limited either by the wireless network card's memory size or by the three conditions defined in the standard. In Fig. 8, we use the notation "Max MPDU" for the case when the card's memory is unlimited and only the standard defines the aggregation rate, and "k MPDU" for the case when the card's memory can hold at most k frames. Note that the standard's aggregation limits always override the limits defined by the card's memory size. This means that for some MCS and channel width combinations the actual number of aggregated MPDUs will be smaller than our proposed aggregation rate. For example, when using MCS0 and a 20 MHz channel, we can at most aggregate two MPDUs of 1500 bytes due to the limit on the maximum transmission duration.

[1] VNI Cisco. Cisco visual networking index: Forecast and trends, 2017–2022. White Paper, 2018.

[2] Lara Deek, Eduard Garcia-Villegas, Elizabeth Belding, Sung-Ju Lee, and Kevin Almeroth. The impact of channel bonding on 802.11 n network management. In Proceedings of the Seventh Conference on Emerging Networking Experiments and Technologies, 2011.

[3] Boris Bellalta, Alessandro Checco, Alessandro Zocca, and Jaume Barcelo. On the interactions between multiple overlapping WLANs using channel bonding. IEEE Transactions on Vehicular Technology, 2015.

[4] Boris Bellalta, Azadeh Faridi, Jaume Barcelo, Alessandro Checco, and Periklis Chatzimisios. Channel bonding in short-range WLANs. In European Wireless 2014; 20th European Wireless Conference. VDE, 2014.

[5] Sergio Barrachina-Muñoz, Francesc Wilhelmi, and Boris Bellalta. Dynamic channel bonding in spatially distributed high-density wlans. IEEE Transactions on Mobile Computing, 2019.

[6] Sami Khairy, Mengqi Han, Lin X Cai, Yu Cheng, and Zhu Han. A renewal theory based analytical model for multi-channel random access in IEEE 802.11 ac/ax. IEEE Transactions on Mobile Computing, 2018.

[7] Giuseppe Bianchi. Perf. analysis of the IEEE 802.11 distributed coordination function. IEEE Journal on selected areas in communications, 2000.

[8] Katarzyna Kosek-Szott. A comprehensive analysis of IEEE 802.11 DCF heterogeneous traffic sources. Ad Hoc Networks, 2014.

[9] Emad Felemban and Eylem Ekici. Single hop IEEE 802.11 DCF analysis revisited: Accurate modeling of channel access delay and throughput for saturated and unsaturated traffic cases. IEEE Trans. on Wireless Communications, 2011.

[10] Zhefu Shi, Cory Beard, and Ken Mitchell. Analytical models for understanding space, backoff, and flow correlation in CSMA wireless networks. Wireless networks, 2013.

[11] Thomas Begin, Bruno Baynat, Isabelle Guérin Lassous, and Thiago Abreu. Perf. analysis of multi-hop flows in IEEE 802.11 networks: A flexible and accurate modeling framework. Performance Evaluation, 2016.

[12] Claude Chaudet, Isabelle Guérin Lassous, Eric Thierry, and Bruno 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 Perf. eval. of wireless ad hoc, sensor, and ubiquitous networks. ACM, 2004.

[13] Bertrand Ducourthial, Stéphane Mottelet, and Anthony Busson. Improving fairness between close Wi-Fi access points. Journal of Network and Computer Applications, 2017.

[14] Boris Ginzburg and Alex Kesselman. Perf. analysis of A-MPDU and A-MSDU aggregation in IEEE 802.11 n. In IEEE Sarnoff symposium, 2007.

[15] Yuxia Lin and Vincent WS Wong. Frame aggregation and optimal frame size adaptation for IEEE 802.11 n WLANs. In IEEE Globecom, 2006.

[16] Marija Stojanova, Thomas Begin, and Anthony Busson. Conflict graph-based model for IEEE 802.11 networks: A Divide-and-Conquer approach. Performance Evaluation, 2019.

[17] Eng Hwee Ong, Jarkko Kneckt, Olli Alanen, Zheng Chang, Toni Huovinen, and Timo Nihtilä. IEEE 802.11 ac: Enhancements for very high throughput WLANs. In IEEE 22nd International Symposium on Personal, Indoor and Mobile Radio Communications, 2011.

[18] Jiyoung Cha, Hu Jin, Bang Chul Jung, and Dan Keun Sung. Perf. comparison of downlink user multiplexing schemes in IEEE 802.11 ac: Multi-user MIMO vs. frame aggregation. In IEEE WCNC, 2012.

[19] Mihaela-Diana Dianu, Janne Riihijärvi, and Marina Petrova. Measurement-based study of the perf. of IEEE 802.11 ac in an indoor environment. In IEEE ICC, 2014.

[20] Lito Kriara, Edgar Costa Molero, and Thomas R Gross. Evaluating 802.11 ac features in indoor WLAN: an empirical study of performance and fairness. In Proceedings of the Tenth ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation, and Characterization. ACM, 2016.

[21] Evgeny Khorov, Anton Kiryanov, Andrey Lyakhov, and Giuseppe Bianchi. A tutorial on IEEE 802.11 ax high efficiency WLANs. IEEE Communications Surveys & Tutorials, 2018.

[22] Evgeny Khorov, Ilya Levitsky, and Ian F Akyildiz. Current Status and Directions of IEEE 802.11 be, the Future Wi-Fi 7. IEEE Access, 2020.

[23] Boris Bellalta and Katarzyna Kosek-Szott. AP-initiated multi-user transmissions in IEEE 802.11 ax WLANs. Ad Hoc Networks, 2019.

[24] Evgeny Khorov, Vyacheslav Loginov, and Andrey Lyakhov. Several EDCA parameter sets for improving channel access in IEEE 802.11 ax networks. In 2016 International Symposium on Wireless Communication Systems (ISWCS). IEEE, 2016.

[25] Francesc Wilhelmi, Sergio Barrachina-Muñoz, and Boris Bellalta. On the Performance of the Spatial Reuse Operation in IEEE 802.11 ax WLANs. In 2019 IEEE Conference on Standards for Communications and Networking (CSCN). IEEE.

[26] Mun-Suk Kim, Tanguy Ropitault, Sukyoung Lee, and Nada Golmie. A throughput study for channel bonding in IEEE 802.11 ac networks. IEEE Communications Letters, 2017.

[27] Sergio Barrachina-Muñoz, Francesc Wilhelmi, and Boris Bellalta. To overlap or not to overlap: Enabling channel bonding in high-density WLANs. Computer Networks, 2019.

[28] Min Peng, Kai Zhang, and Caihong Kai. Channel Bonding Based on Game Theory in Dense WLANs. In 2019 IEEE 11th International Conference on Communication Software and Networks (ICCSN). IEEE, 2019.

[29] Sree Vasthav SV, S Srikanth, and Venkatesh Ramaiyan. Performance analysis of an IEEE 802.11 ac WLAN with dynamic bandwidth channel access. In National Conference on Communication (NCC). IEEE, 2016.

[30] Mengqi Han, Sami Khairy, Lin X Cai, and Yu Cheng. Performance analysis of opportunistic channel bonding in multi-channel WLANs. In 2016 IEEE Global Communications Conference (GLOBECOM). IEEE, 2016.

[31] Lara Deek, Eduard Garcia-Villegas, Elizabeth Belding, Sung-Ju Lee, and Kevin Almeroth. Intelligent channel bonding in 802.11 n WLANs. IEEE Trans. on Mobile Computing, 2014.

27

[32] Ljiljana Simić, Janne Riihijärvi, and Petri Mähönen. Measurement study of IEEE 802.11 ac Wi-Fi perf. in high density indoor deployments: Are wider channels always better? In IEEE WoWMoM, 2017.

[33] Yunze Zeng, Parth H Pathak, and Prasant Mohapatra. A first look at 802.11 ac in action: Energy efficiency and interference characterization. In 2014 IFIP Networking Conference. IEEE, 2014.

[34] Saber Malekmohammadi. Parameterizing enterprise wifi networks: The use of wide channels. Master's thesis, University of Waterloo, 2019.

[35] Minyoung Park. IEEE 802.11 ac: Dynamic bandwidth channel access. In 2011 IEEE international conference on communications (ICC). IEEE, 2011.

[36] Michelle X Gong, Brian Hart, Liangfu Xia, and Roy Want. Channel bounding and MAC protection mechanisms for 802.11 ac. In 2011 IEEE Global Telecommunications Conference-GLOBECOM 2011. IEEE, 2011.

[37] Arpit Gupta, Jeongki Min, and Injong Rhee. WiFox: Scaling WiFi perf. for large audience environments. In Proceedings of the 8th international conference on Emerging networking experiments and technologies. ACM, 2012.

[38] Francesco Malandrino, Carla-Fabiana Chiasserini, and Scott Kirkpatrick. Cellular Network Traces Towards 5G: Usage, Analysis and Generation. IEEE Trans. on Mobile Computing, 2018.

[39] Michele Garetto, Theodoros Salonidis, Edward W Knightly, et al. Modeling per-flow throughput and capturing starvation in csma multi-hop wireless networks. In Infocom, 2006.

[40] Jitendra Padhye, Sharad Agarwal, Venkata N Padmanabhan, Lili Qiu, Ananth Rao, and Brian Zill. Estimation of link interference in static multi-hop wireless networks. In Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement, pages 28–28. USENIX Association, 2005.

[41] IEEE Computer Society LAN MAN Standards Committee and others. IEEE P802.11ac. Specification framework for TGac. IEEE 802.11-09/0992r21, January 2011.

[42] Mathilde Durvy, Olivier Dousse, and Patrick Thiran. Self-organization properties of CSMA/CA systems and their consequences on fairness. IEEE Trans. on Information Theory, 2009.

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

[44] Ns3 simulation code and MATLAB model. http://perso.ens-lyon.fr/thomas.begin/support_data_CTMC.zip.