Responsive Spectrum Management for Wireless Local Area Networks: from Heuristic-based Policies to Model-Free Reinforcement Learning

Sergio Barrachina-Muñoz

TESI DOCTORAL UPF / ANY 2020

Director de la tesi
Boris Bellalta
Departament of Information and Communication Technologies

A l'Iris.

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my advisor, Dr. Boris Bellalta, for trusting me and letting me be part of his research group. His guidance throughout the PhD years has provided the best research environment to grow both as a researcher and a person.

A special thanks goes to Dr. Francesc Wilhelmi, my preeminent colleague, for his invaluable help and encouragement to overcome those moments of panic one shall encounter in the PhD path. A lot of what I have accomplished in this work is thanks to him. Let the Komondor simulator be the vestige of our work together.

Because of the relaxed (and sometimes bottomless) chats and the inestimable help with intricate research issues, I am thankful to the rest of the members of the Wireless Networking group: Kostis Dovelos, Álvaro López, and Marc Carrascosa. Besides, I could never forget how Albert Bel, Toni Adame, and Verónica Moreno embraced me when I started working as a research intern back in 2015. It has been a pleasure to take this journey with you all, guys.

Of course, the PhD would not have been the same without the undergraduate classes at Universitat Pompeu Fabra. So, thanks to all the students I had the opportunity to teach something. One thing is sure: you have taught me so much.

I am also sincerely thankful to Prof. Edward Knightly and the members of the Rice Networks Group for giving me such a warm welcoming when I joined the group as a visitor in 2018. Those were four fruitful months in Houston where I learned a lot from incredible researchers. It was an excellent opportunity to broaden my thesis scope towards hardware-related topics and design WACA, which has become one of the main assets of this thesis.

My friends outside the academic world were, are, and will always be a precious give. Albert, Edgar, Sergi, Jorge, Jacob, you are simply the best. Having you all around means a lot. Likewise, I want to express my deepest gratitude to my high school teacher Víctor. He has been a gentle and sage oracle throughout these years and made me understand that life, art, and science are and should be all the same thing.

I wish to thank my parents, Enrique and Cati, and my brother Carlos, for raising me and making me always feel the family had my back. My academic career could not have been possible without their endless support.

Lastly, I would like to dedicate this thesis to Iris (and our dog Wilco) for always standing by my side. You are all one could ask for having a good life.

Abstract

In this thesis, we focus on the so-called spectrum management's joint problem: efficient allocation of primary and secondary channels in channel bonding wireless local area networks (WLANs). From IEEE 802.11n to more recent standards like 802.11ax and 802.11be, bonding channels together is permitted to increase transmissions' bandwidth. While such an increase favors the potential network capacity and the activation of higher transmission rates, it comes at the price of reduced power per Hertz and accentuated issues on contention and interference with neighboring nodes. So, if WLANs were per se complex deployments, they are becoming even more complicated due to the increasing node density and the new technical features required by novel highly bandwidth-demanding applications. This dissertation provides an in-depth study of channel allocation and channel bonding in WLANs and discusses the suitability of solutions ranging from heuristic-based to reinforcement learning (RL)-based.

To characterize channel bonding in saturated WLANs, we first propose an analytical model based on continuous-time Markov networks (CTMNs). This model relies on a novel, purpose-designed algorithm that generates CTMNs from spatially distributed scenarios, where nodes are not required to be within the carrier sense range of each other. We identify the key factors affecting the throughput and fairness of different channel bonding policies and expose critical interrelations among nodes in the spatial domain. By extending the analytical model to support unsaturated regimes, we highlight the benefits of allocating channels as wide as possible all together with adaptive policies to cope with unfair situations.

Apart from the analytical model, this thesis relies on simulations to generalize channel bonding in dense scenarios while avoiding costly, sometimes unfeasible, experimental testbeds. Unfortunately, existing wireless network simulators tend to be too simplistic or too computational demanding. That is why we develop the Komondor wireless network simulator, with the essential advantage over other well-known simulators lying in its high event processing rate.

We then deviate from analytical models and simulations and tackle real measurements through the Wi-Fi All-Channel Analyzer (WACA), the first system specifically designed to simultaneously measure the energy in all the 24 bondable Wi-Fi channels at the 5 GHz band. With WACA, we perform a first-of-its-kind spectrum measurement in areas including urban hotspots, residential neighborhoods, universities, and even a football match in Futbol Club Barcelona's Camp Nou stadium. Our experimental findings reveal the underpinning factors controlling throughput gain, from which we highlight the inter-channel correlation.

As for solution proposals, we first cover heuristic-based approaches to find satisfactory configurations quickly. In this regard, we propose dynamic-wise (DyWi), a lightweight, decentralized, online primary channel selection algorithm for dynamic channel bonding. DyWi improves the expected WLAN throughput by considering not only the occupancy of the target primary channel but also the activity in the secondary channels. Even when assuming significant delays due to primary channel switching, simulations reveal important throughput and delay improvements.

Finally, we identify machine learning (ML) approaches applicable to the spectrum management problem in WLANs and justify why model-free RL suits it the most. In particular, we put the focus on the adequate performance of stateless variations of RL and anticipate multi-armed bandits as the right solution since *i*) we need fast adaptability to suit user experience in dynamic Wi-Fi scenarios and *ii*) the number of multichannel configurations a network can adopt is limited; thus, agents can fully explore the action space in a reasonable time.

Resum

En aquesta tesi ens centrem en el problema conjunt de la gestió de l'espectre: assignació de canals primaris i secundaris a xarxes d'àrea local sense fils (WLAN) amb channel bonding. Des de l'estàndard IEEE 802.11n fins a estàndards més recents com el 802.11ac, el 802.11ax i el 802.11be, s'han anat proposant amplades de banda més grans per permetre agrupar canals, augmentant així l'amplada de banda total per transmissió. Tot i que aquest augment en l'amplada de banda afavoreix la capacitat potencial de les xarxes, suportant així els requeriments de les noves aplicacions Wi-Fi, també redueix la potència per Hertz i accentua els problemes de contenció i interferència entre nodes veïns. En resum, si les xarxes WLANs ja eren complexes per se, s'estan tornant encara més complexes a causa de l'augment de la densitat de nodes i de les noves prestacions incloses als darrers estàndards.

Primer proposem un model analític basat en xarxes Markov en temps continu (CTMN) per caracteritzar channel bonding en WLANs saturades. Aquest model es basa en un nou algorisme que genera CTMNs a partir d'escenaris distribuïts espaialment, on no és necessari que els nodes estiguin dins del rang de contenció de la resta. Identifiquem els factors claus que afecten el rendiment i l'equitat de les diferents polítiques de channel bonding i mostrem l'existència d'interrelacions crítiques entre nodes en forma de reacció en cadena. D'això se'n desprèn que no hi ha una política channel bonding òptima única per a cada escenari. En ampliar el model analític per donar suport a règims no saturats, destaquem els avantatges d'assignar els canals tan amplis com sigui possible a les WLAN i implementar polítiques d'accés adaptatiu per fer front a les situacions que poden aparèixer tant en termes de rendiment com d'equitat.

A part dels models analítics, aquesta tesi es basa en simulacions per generalitzar escenaris evitant costosos bancs de proves experimentals, de vegades inviables. Malauradament, els simuladors de xarxes sense fils existents solen ser massa simplistes o molt costosos computacionalment. És per això que desenvolupem el simulador de xarxes sense fils Komondor, concebut com una eina de codi obert accessible (llesta per utilitzar) per a la investigació de xarxes sense fils. L'avantatge essencial de Komondor respecte d'altres simuladors sense fils coneguts rau en la seva elevada velocitat de processament d'esdeveniments.

A continuació ens desviem de models analítics i simulacions i abordem mesures reals a través del Wi-Fi All-Channel Analyzer (WACA), el primer sistema que mesura simultàniament l'energia de tots els 24 canals que permeten channel bonding a la banda Wi-Fi dels 5 GHz. Amb WACA, realitzem un estudi únic de localitzacions que inclouen nuclis urbans, barris residencials, universitats i fins i tot un partit a al Camp Nou, un estadi ple amb 98.000 aficionats i 12.000 connexions Wi-Fi simultànies. Les dades experimentals revelen els factors fonamentals que controlen el guany de rendiment, a partir dels quals ressaltem la correlació entre canals. També mostrem la importància del conjunt de dades recopilades per trobar nous factors claus, que d'una altra manera no seria possible, atès que els models d'ocupació de canals simples subestimen els guanys potencials.

Pel que fa a solucions, primer discutim propostes basades en heurístiques per trobar configuracions satisfactòries ràpidament. En aquest sentit, proposem dinàmicament (DyWi), un algorisme de selecció de canal primari en línia, descentralitzat i eficient per xarxes channel bonding. DyWi millora el rendiment esperat tenint en compte no només l'ocupació del canal primari objectiu, sinó també l'activitat dels canals secundaris. Fins i tot quan suposem retards significatius a causa del canvi de canal primari, observem millores importants en termes de rendiment i retard.

Finalment, identifiquem els enfocaments d'aprenentatge automàtic (o machine learning) aplicables al problema de la gestió de l'espectre a les WLAN i justifiquem per què l'aprenentatge del tipus reinforcement learning (RL) és el més adient. En particular, ens centrem en el rendiment adequat de les variacions d'RL sense estats i proposem multi-armed bandits com la solució adequada, ja que *i*) necessitem una adaptabilitat ràpida per millorar l'experiència d'usuari en escenaris Wi-Fi dinàmics i *ii*) el nombre de configuracions multicanal que una xarxa pot adoptar és limitat; per tant, els agents poden explorar completament l'espai d'acció en un temps raonable.

Contents

List of abbreviations

AM (or DCB): always-max
AP: access point
BSS: basic service set
CA: channel allocation
CB: channel bonding
CCA: clear channel assessment
CDF: cumulative distribution function
CE: capture effect
CSMA/CA: carrier sense multiple access with collision avoidance
CTMN: continuous time Markov network
CTS: clear to send
DCA: dynamic channel allocation
DCB (or AM): dynamic channel bonding
DCF: distributed coordination function
DIFS: DCF interframe space
DFS: dynamic frequency selection
DRL: deep reinforcement learning
DiWy: dynamic wise
DF: dynamic free
DR: dynamic random
FP: fixed primary
IEEE: Institute of Electrical and Electronics Engineers
ISM: industrial, scientific and medical
MAB: multi-armed bandit
MAC: medium access control
MCS: modulation and coding scheme
ML: machine learning
NAV: network allocation vector
OBSS: overlapping basic service set
OPS: online primary selection
PCF: point coordination function
PER: packet error rate
PHY: physical (layer)
PIFS: PCF interframe space
PU: probabilistic uniform
RF: radio frequency
RL: reinforcement learning
RSSI: received signal strength indicator
RTS: request to send
SC: single channel
SCB: static channel bonding
SIFS: short interframe space
SNR: signal-to-noise ratio
SINR: signal-to-noise-plus-interference ratio
SDR: software defined radio
TD: temporal difference
TPC: transmit power control
WACA: Wi-Fi all-channel analyzer
WLAN: wireless local area network
WN: wireless network
WSN: wireless sensor network

List of variables and terms

a: action (or arm) identifier
$\mathcal{A}$: action space
b: maximum bandwidth
B: bandwidth of a basic channel
$\mathcal{B}$: band of basic channels
BO: pending backoff time
BW: bandwidth usage
c: basic channel identifier
C: channel allocation
C^{tx}: transmission channel
$\mathcal{C}$: channelization
CE: capture effect threshold
d: packet delay
d^*: minimum achievable delay
$\mathcal{D}$: channel bonding policy
$\mathbb{E}$: expected value
G: cumulative reward
h: hypothesis function
$\mathcal{H}$: WLAN status (in RL)
$\mathcal{I}_w$: world of BSS w (in RL)
J: Jain's fairness index
Q: action value
K: number of actions (or arms)
L: size of a data packet
ℓ: mean traffic load
$\mathcal{L}$: quantized mean traffic load
m: maximum backoff stage
M: matrix of transition rates
n_C: number of basic channels in the system
n_{STA}: number of STAs
$\bar{o}_{\mathcal{B}}$: mean occupancy at band $\mathcal{B}$
p: primary channel identifier
$\mathbb{P}$: probability
q: backoff stage
r: instantaneous reward
R: reward function
s: feasible state (in CTMN) or state (in RL)
$\mathcal{S}$: feasible state space (in CTMN) or state space (in RL)
t: iteration identifier or time
T: number of iterations
T_{m}: duration of a measurement (in WACA)
T_{per}: duration of a period (or epoch)
T_{slot}: duration of a slot
$\hat{U}$: UCB1 upper confidence bound
w: BSS identifier
W: number of BSS's
W': number of agent-empowered BSS's
α: learning rate
$\vec{\alpha}$: transition probabilities
β: set of configurable max. bandwidths
γ: discount factor
Γ: throughput
$\hat{\Gamma}$: normalized Γ wrt single-channel
δ: bounded delay ratio
ζ: throughput satisfaction
η: throughput satisfaction ratio
θ: reward probability distribution
λ: packet transmission attempt rate
μ: departure rate
$\vec{\nu}$: equilibrium distribution
ξ: inter-channel correlation
π: policy (in RL)
ρ: long-term probability
ϱ: Pearson's correlation coefficient
σ: standard deviation
ψ: global state identifier
Ψ: global state space
ω: bandwidth deprivation
Ω: context (in RL)

List of main publications

The following publications constitute the backbone of this thesis. We refer to them in the dissertation as **thesis papers #1** to **#6**.

1. S. Barrachina-Muñoz, F. Wilhelmi, and B. Bellalta. Dynamic channel bonding in spatially distributed high-density WLANs. *IEEE Transactions on Mobile Computing*, 19(4): 821–835, 2019.

2. S. Barrachina-Muñoz, F. Wilhelmi, and B. Bellalta. To overlap or not to overlap: enabling channel bonding in high-density WLANs. *Computer Networks*, 152:40–53, 2019.

3. S. Barrachina-Muñoz, F. Wilhelmi, I. Selinis, and B. Bellalta. Komondor: A wireless network simulator for next-generation high-density WLANs. *In 2019 Wireless Days (WD)*, pages 1–8. IEEE, 2019.

4. S. Barrachina-Muñoz, F. Wilhelmi, and B. Bellalta. Online primary channel selection fordynamic channel bonding in high-density WLANs. *IEEE Wireless Communications Letters*, 9(2):258–262, 2019.

5. S. Barrachina-Muñoz, B. Bellalta, and E. Knightly. Wi-Fi all channel analyzer. *In The 14th ACM Workshop on Wireless Network Testbeds, Experimental evaluation & CHaracterization (WINTECH). ACM.*, 2020. **Runner up, best paper award**.

6. S. Barrachina-Muñoz, B. Bellalta, and E. Knightly. Wi-Fi channel bonding: an all-channel system and experimental study from urban hotspots to a sold-out stadium. *Submitted to IEEE Transactions on Networking (under major revision)*, 2020.

List of other publications

While this dissertation does not directly include contributions from the following papers, they have significantly influenced the thesis's development.

1. S. Barrachina-Muñoz, B. Bellalta, T. Adame, and A. Bel. Multi-hop communication in the uplink for LPWANs. *Computer Networks, 123*, 153-168, 2017.

2. S. Barrachina-Muñoz and B. Bellalta. Learning optimal routing for the uplink in LPWANs using similarity-enhanced e-greedy. *In 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 1–5. IEEE, 2017.

3. T. Adame, S. Barrachina-Muñoz, B. Bellalta, and A. Bel. HARE: Supporting efficient uplink multi-hop communications in self-organizing LPWANs. *Sensors, 18*(1), 115, 2019.

4. S. Barrachina-Muñoz, T. Adame, A. Bel, and B. Bellalta. Towards energy efficient LPWANs through learning-based multi-hop routing. I*n IEEE 5th World Forum on Internet of Things (WF-IoT)* 2019, 2019.

5. F. Wilhelmi, S. Barrachina-Muñoz, and B. 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)* (pp. 1-6). IEEE, 2019.

6. F. Wilhelmi, S. Barrachina-Muñoz, B. Bellalta, C. Cano, A. Jonsson, G. Neu. Potential and pitfalls of multi-armed bandits for decentralized spatial reuse in WLANs. *Journal of Network and Computer Applications, 127*, 26-42, 2019.

7. F. Wilhelmi, B. Bellalta, C. Cano, A. Jonsson, and S. Barrachina-Muñoz. Collaborative spatial reuse in wireless networks via selfish multi-armed bandits. *Ad Hoc Networks, 88*, 129-141, 2019.

8. F. Wilhelmi, S. Barrachina-Muñoz, B. Bellalta, C. Cano, A. Jonsson, and Ram, V. A flexible machine-learning-aware architecture for future WLANs. *IEEE Communications Magazine, 58*(3), 25-31, 2020.

Chapter 1

INTRODUCTION

Wireless local area networks (WLANs), with IEEE 802.11 as the most widely used standard, are a cost-efficient solution for wireless Internet access that can satisfy most of the current communication requirements in domestic, public, and business scenarios. Modern applications like augmented reality, virtual reality, or real-time 8K video are pushing next-generation WLANs to support ever-increasing performance demands along with 5G systems and beyond. Wi-Fi relies on basic service sets (BSS's)[1] composed of an access point (AP) and one or multiple stations (STAs). The AP provides access to the Internet to the STAs in downlink and uplink manner. This simple and flexible architecture has been a success since the first commercial deployments of Wi-Fi back in 1999.

Wi-Fi[2] works primarily with the carrier sense multiple access with collision avoidance (CSMA/CA) method,[3] where nodes attempt to avoid collisions by initiating transmissions only after the channel is sensed idle during a random backoff time. This listen-before-talk scheme is critical to ensure fairness in the industrial, scientific, and medical (ISM) bands, license-free bands that everyone can use. So, before transmitting in a given channel, a Wi-Fi device must wait to sense that channel free during a time indicated by the backoff counter. Once the backoff expires, frame transmissions can be initiated. Since the introduction of channel bonding in the IEEE 802.11n [4] amendment, multichannel transmissions were also permitted to support larger bandwidths and reach higher capacities. Current and future amendments like 802.11ac/ax [5, 7] and 802.11be [6], respectively, extend channel bonding capabilities to support higher bandwidths and more channel combinations.

CSMA/CA is Wi-Fi's cornerstone and has been performing extraordinarily well for single and multichannel transmissions in previous years, when channels were vacated most of the time, so one could typically transmit without significant delays nor losses due to interference. However, the random nature of CSMA/CA altogether with the scarce shared frequency spectrum at the ISM bands and the emergence of new Wi-Fi features mandates adequate spectrum management to handle today's and future's Wi-Fi complexity.

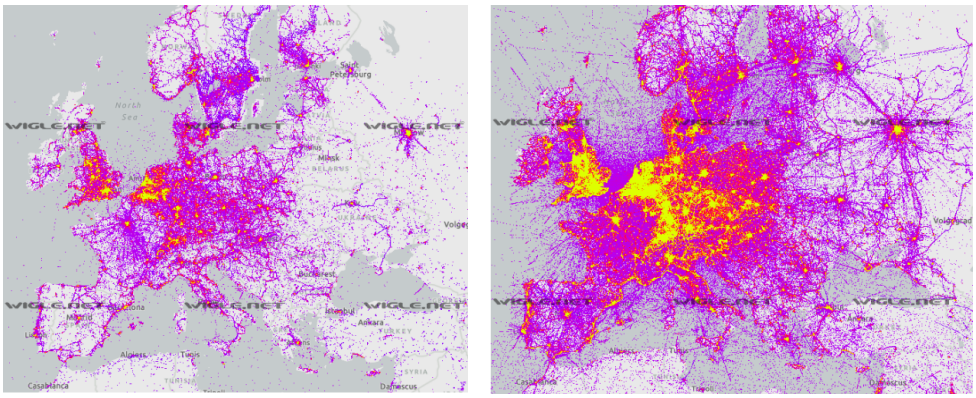
Indeed, the number of hungry-bandwidth devices accessing the Internet through WLAN APs such as laptops, smart-phones, tablets, and wearables is increasing drastically at the same

[1] Some works in the literature use the term WLAN to refer to a BSS. However, it is more formal to separate such concepts as we do in this dissertation.

[2] The terms WLAN and Wi-Fi are often used interchangeably, as in this thesis. Nonetheless, a WLAN can be built on various wireless technologies, including Wi-Fi (i.e., the IEEE 802.11 standards).

[3] New IEEE 802.11 amendments like 802.11ax introduce other medium access methods based on scheduling. However, CSMA/CA is still the most common access method due to its simplicity.

time that users' performance requirements do. The cause of such increasing demand is accentuated by the explosion of new high-definition, tactile-like, and real-time interacting multimedia applications. In short, more devices per area lead to more contention, and better performance requirements lead to a major use of bandwidth. Further, the non-coordinated Wi-Fi nature, where any user can instantly deploy a new BSS on its own, hinders the problem even more. Figure 1.1 shows how the hot-spot (or AP) density evolution has remarkably risen in the past few years.



(a) Density map in 2012. (b) Density map in 2020.

Figure 1.1: Wireless hot-spots density map: approximated location of APs in (a) in 2012, and (b) in 2020. [Source: `www.wigle.net`]

In dense WLANs scenarios like home apartments or sports stadiums – where there may be 1 user/m^2 and multiple neighboring BSS's – providing users with such high performance is even a more arduous task. When Wi-Fi nodes are close to each other, they often overlap if they share any channel during simultaneous transmissions. This increases packet losses as a consequence of the co-channel and adjacent channel interference. Therefore, efficient resource allocation is necessary to relieve the performance degradation caused by overlapping networks and the problems arising from it (i.e., hidden and exposed nodes, among others).

As for the spectrum management, a new paradigm is required to overcome the challenges arising from the heterogeneous and highly dynamic next-generation WLANs, where channel bonding BSS's may have demanding and varying traffic load needs. That is, apart from widening the Wi-Fi bands to support more channels, it is becoming critical to raise the spectrum efficiency of the current channelization. In this regard, as we show later in this dissertation through on-the-field measurements, there is still plenty of room for increasing efficiency.

1.1 Spectrum management in uncoordinated Wi-Fi networks

This thesis focuses on the spectrum management problem in uncoordinated channel bonding WLANs. That is, distributively assigning chunks of the band to channel bonding BSS's. Channel allocation determines the set of *basic* channels – of $B = 20$ MHz width in the 5 GHz band – a BSS may use, composed of the primary and secondary channels. It is especially convenient to appropriately allocate the primary channel in order to avoid contention with neighboring nodes. When channel access is gained, the final combination of channels to transmit depends on the

channel allocation, the channel bonding policy, and the frequency spectrum status. In particular, channel allocation is configuration-dependent, and channel bonding acts on a per-frame basis according to the sensed spectrum activity.

Channel bonding is a key mechanism for increasing Wi-Fi data rates as the maximum data rate increases in proportion to the total channel bandwidth. The Shannon-Hartley theorem for n channels of bandwidth B establishes that the information-theoretic capacity of a link is defined (in bits per second) as

$$C = nB \log_2 \left(1 + \frac{S/n}{N}\right), \tag{1.1}$$

where S/N is the signal-to-noise-ratio (SNR) that would be perceived at the receiver when the transmitters use single-channel ($n = 1$). So, in regimes where the SNR is kept sufficiently high, leading to a constant modulation and coding scheme (MCS) regardless of the bandwidth, the capacity grows linearly with the number of channels. However, the theoretical capacity is sub-linear with the bandwidth (or the number of channels) as expressed by the term $\frac{S/n}{N}$ in the logarithm of (1.1). The reason lies in the fact that for a given transmit power, the power per unit-Hertz halves when doubling the bandwidth, thus reducing the SNR and potential MCS as a consequence.

Notice that channel bonding is practical only in the 5 GHz band rather than in the 2.4 GHz: while the former has 25 non-overlapping channels, the latter only provides 3. Figure 1.2 and Figure 1.3 show Wi-Fi's 2.4 and 5 GHz channelization, respectively.

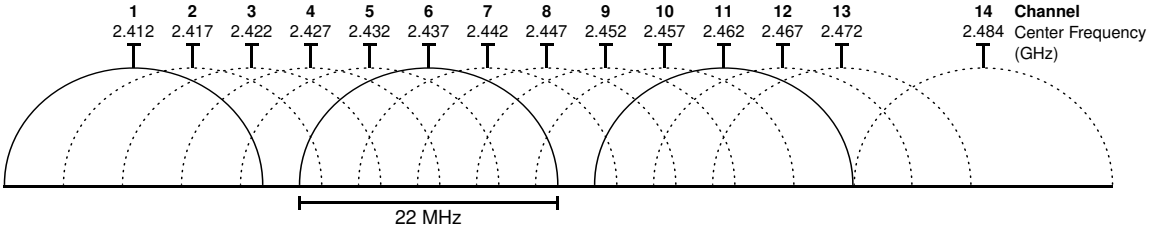


Figure 1.2: Channelization within the 2.4 GHz band (image retrieved from [8]).

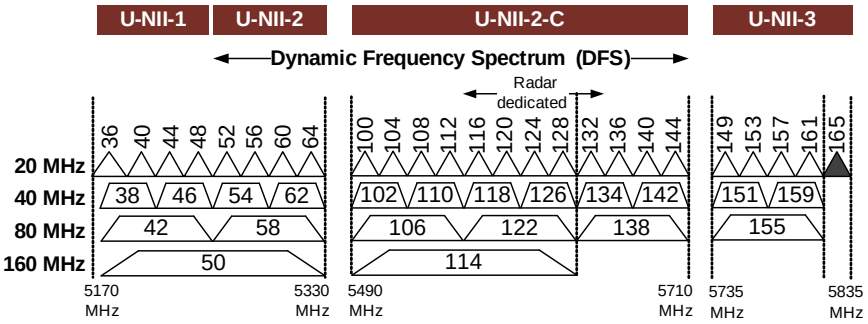


Figure 1.3: Channelization within the 5 GHz band (-A region).

In brief, WLAN **spectrum management** entails channel allocation and channel bonding. The main objective of **spectrum allocation** is to avoid interference between potentially overlapping nodes, and the main objective of **channel bonding** is to maximize the network capacity.

Unfortunately, it is unavoidable per definition to transmit in higher bandwidths while reducing the interference with neighboring nodes: the objectives are in principle in contradiction. However, the trade-off between lowering interference and maximizing capacity has aspects that we can leverage. For instance, BSS's are not always saturated, which means that two overlapping BSS's may share a channel and support moderate traffic loads. At the same time, as mandated by (1.1), by increasing the bandwidth, we reduce the power per unit-Hertz, reducing the absolute interference per channel as well, which may ultimately favor the contention of large bandwidth transmissions.

Implementing efficient spectrum allocation and channel bonding is not straightforward because WLANs often operate in permanent transitory systems with varying traffic loads. Hence, performance estimations computed through optimization models are not accurate enough. Due to such a lack of certainty and required flexibility, a new paradigm seems necessary to endure spectrum management in Wi-Fi networks. In this context, there is a trend in the literature towards the use of artificial intelligence and machine learning (ML) for addressing the challenges raised by next-generation high-density WLANs.

1.2 Contributions

In light of spectrum management's importance to present and future WLANs, in this thesis, we characterize channel allocation and channel bonding and study different approaches to tackle them, including the use of responsive solutions aided with heuristics and reinforcement learning. We categorize the contributions of this thesis into conceptual and methodological/tools:

- Conceptual:

  C1. First, we conduct an in-depth study of how channel bonding policies perform in saturated and unsaturated WLANs. Among the gathered findings, we observe that while being greedy with dynamic channel bonding – transmitting over the maximum possible bandwidth – usually raises the individual throughput; there are scenarios where more conservative policies reach both higher individual throughput and fairness. It follows that different policies per BSS may be required to maximize the system's metric of interest (e.g., throughput, delay, or fairness). We also find that it is more convenient in terms of individual delay to use conservative approaches under low traffic loads because of the *chain-reactions* resulting in a reduced contention.

  C2. Next, we analyze channel bonding policies' feasibility through on-the-field measurements of the Wi-Fi 5 GHz spectrum at different locations of interest. Our experimental findings reveal the underpinning factors controlling throughput gain, from which we highlight the inter-channel correlation. Also, the Camp Nou stadium's measurement campaign reveals that, even under extremely high average load, short durations of lower load can be exploited to yield significant throughput gains.

  C3. Then, we discuss heuristic-based spectrum management solutions and propose a lightweight, decentralized, online primary channel selection algorithm for dynamic channel bonding. Dynamic-Wise (DyWi) improves the expected WLAN throughput by considering the occupancy of the target primary channel and the activity of the secondary channels.

C4. Lastly, we envision the need for machine learning to cope with the spectrum management problem in high-density and dynamic settings. We justify the suitability of a stateless variation of reinforcement learning (RL) through multi-armed bandits (MABs) suits the task of adapting fast in uncoordinated deployments. Results from extensive experiments show the responsiveness of MABs in front of other RL approaches.

- Methodological/tools:

M1. First, we design an analytical model based on continuous-time Markov networks (CTMNs) to capture the behavior of spatially-distributed WLANs implementing channel bonding policies under saturated and unsaturated regimes. This is the first CTMN model to capture partial overlaps in channel bonding WLANs. Part of contribution C1 is derived from analyses through this model.

M2. Second, we develop Komondor, a wireless network simulator for assessing the novel IEEE 802.11 features in high-density WLAN deployments. Komondor provides reliable simulations with much lower execution times than other well-known simulators such as ns-3. Findings on high-density deployments in contribution C1 are gathered through Komondor. Besides, our wireless network simulator is also used for contributions C3 and C4.

M3. Finally, we design and build WACA, a novel spectrum analyzer covering all the Wi-Fi's 5 GHz band. We use it to gather a unique dataset covering multiple places of interest, including a sold-out game in the Camp Nou stadium. The key novelty of WACA is its ability to measure all the basic channels simultaneously. Through a trace-driven framework, we discover novel insights into channel bonding otherwise not possible to get. These insights compose contribution C2.

To make our research results more accessible to the community, all the work made in this thesis has been disclosed in open access. To that purpose, we have made publicly available all the resources developed to undertake our research, including results, code, and datasets. The tools used to enable open access are GitHub and Zenodo. We provide links to such sources through the chapters of this dissertation.

1.3 Document structure

This thesis is a compendium of articles resulting from the characterization and benchmarking of channel bonding and channel allocation in Wi-Fi deployments. We refer to the thesis publications attached at the end of this document (§8) as **paper #1** to **paper #6**. Apart from the the list of publications, a monograph is provided to introduce the research topic and emphasize the main findings. This document is structured as follows. Chapter §2 introduces the problem of spectrum management in Wi-Fi networks. In Chapter §3, we depict the three main enabler tools of this thesis. Next, in Chapter §4 we employ the presented enablers to assess spectrum management performance in a plethora of scenarios. The use of heuristics to cope with the problem at issue is discussed in Chapter §5. Then, Chapter §6 treats the convenience of stateless reinforcement learning and benchmarks different solutions. Finally, Chapter §7 concludes with the summary and future work.

Chapter 2

SPECTRUM MANAGEMENT IN IEEE 802.11 WLANS

In this chapter, we briefly depict the Wi-Fi CSMA/CA operation and the features related to the spectrum management in IEEE 802.11 WLANs. Spectrum management techniques can be mainly divided in spectrum allocation (or channel allocation) and channel bonding.

2.1 Channel allocation

The WLAN spectrum covers different frequency bands inside the ISM bands. The most significant are the 2.4 GHz (802.11b/g/n/ax) [4] and 5 GHz (802.11a/h/j/n/ac/ax) [5] bands, but there are other bands like 3.65 GHz (802.11y) or 60 GHz (802.11ad/ay). Future amendments like the 802.11be [6] will also use the 6 GHz band. In the 2.4 GHz band, there are 14 channels of 22 MHz, from which just 3 of them (channels 1, 6, and 11) do not overlap. Instead, in the 5 GHz band, there are 25 non-overlapping basic channels of 20 MHz, from which 24 are bondable, as shown in Figure 1.3. Nonetheless, different limitations exist depending on the country. For instance, in 2007, the Federal Communications Commission (FCC) began requiring that devices operating in the United States on basic channels 52 to 144 must employ dynamic frequency selection (DFS)[1] and transmission power control capabilities to avoid interference with weather-radar and military applications [1].

As with any wireless communication technology, Wi-Fi nodes use electromagnetism to communicate through the channels mentioned above. What channels to use in a given transmission depend on two factors: the channel allocation and the channel bonding policy. **Channel allocation** (CA), also known as spectrum allocation, channel assignment, or channel selection, refers to the process of allocating the potential transmission channels of a BSS or a group of BSS's. Such allocation contemplates (mandatorily) the *primary* channel and (optionally) one or multiple *secondary* channels. Naturally, proper channel allocation planning should contribute to reducing interference while assigning sufficient bandwidth to each BSS.

> **Channel allocation** (or spectrum allocation) refers to assigning the primary and secondary channels to one or multiple BSS's.

[1] DFS is the process by which the AP must detect the signature of existing government weather radar and other priority radio systems and vacate the channel accordingly during the time specified by the corresponding regulator.

For the sake of clarity, let us define below the concepts related to spectrum management we use throughout this dissertation:

- **Basic channel**: the Wi-Fi 5-GHz frequency spectrum is split into basic channels of width $B = 20$ MHz.
- **Primary channel**: the primary channel p_w of a BSS w is a basic channel with different roles depending on the node status. It is used to sense the medium's occupancy i) for decrementing the backoff when the primary channel is found free, and ii) for selecting the transmission channel once the backoff expires. The primary channel must be the same in all the nodes (AP and STAs) in the BSS.
- **Channelization**: the channelization $\mathcal{C}$ is the set of possible combinations of basic channels to transmit. Depending on the channelization $\mathcal{C}$, different restrictions may be taken into consideration when transmitting. For instance, only channel combinations whose number of basic channels is a power of two are considered for contiguous channel bonding in the 802.11ac/ax standard (see Figure 1.3). In particular, any 160 MHz sub-band (composed of 8 basic channels) is *channelized* as follows by the 802.11ac/ax standards,

  $$\mathcal{C} = \{\{1\}, \{2\}, ..., \{8\}, \{1,2\}, \{3,4\}, \{5,6\}, \{7,8\}, \{1,2,3,4\}, \{5,6,7,8\}, \{1, ..., 8\}\}.$$

  So, for instance, bonding basic channels $\{3, 4\}$ and $\{5, 6, 7, 8\}$ is allowed, whereas bonding basic channels $\{4, 5\}$ and $\{4, 5, 6, 7\}$ is not permitted.
- **Channel allocation**: a channel allocation $C_w \in \mathcal{C}$ consists of a contiguous set of basic channels containing the primary channel, i.e., $p_w \in C_w$. The rest of the channels in C_w are called secondary channels. The channel allocation C_w limits the possible channels to use for transmitting, C^{tx}.
- **Channel bonding policy**: the channel bonding policy $\mathcal{D}$ establishes the set of rules for selecting those channels in C_w found free at the backoff termination.
- **Transmission channel**: when the backoff of a transmitter in BSS w expires, it first detects the basic channels found free in C_w. Then, it selects the range of basic channels to transmit $C^{\text{tx}} \subseteq C_w$ according to the channel bonding policy $\mathcal{D}$.

Channel allocation in the literature

Traditional channel allocation relied on a fixed assignment of channels to each BSS. However, with the appearance of dynamic channel allocation (DCA), channel selection tried to adapt to the environment in different ways by monitoring the available channels for the BSS's and tracking the changing conditions. DCA usually develops a cost metric that is used to evaluate various channel plan options. The cost metric is often associated with RSSI values comprised of interference, noise, user sensitivity thresholds, and load (if enabled). Changing the channel of an AP is potentially disruptive, and care must be taken to evaluate apparent improvements. This is where next-generation DCA excels.

Several DCA approaches can be found in the wireless communications literature tackling specific goals. While some approaches aim to provide fairness [89], some others try to grant more resources to nodes with higher traffic demands [127]. A survey on CA is provided in [46].

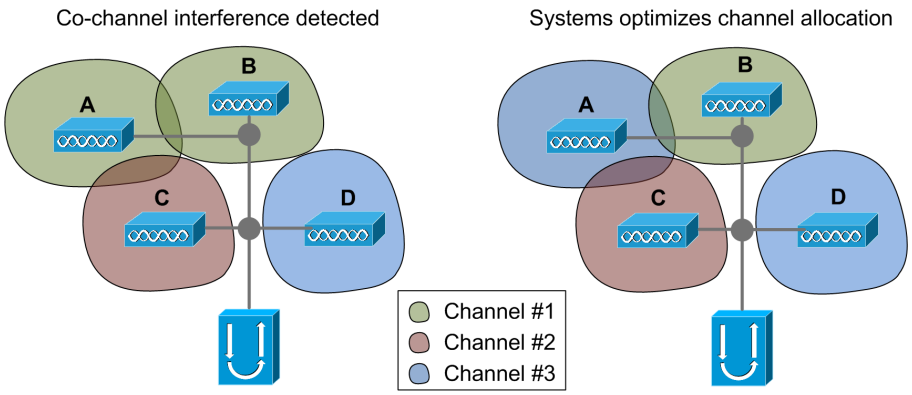


Figure 2.1: Coordinated DCA system. BSS's A and B suffer from co-channel interference, reduced by switching A's channel from #1 to #3.

Technique	Description	Ref.
Graph coloring	By assuming that APs are vertices and non-overlapping channels are colors, graph coloring is applied to minimize adjacent interference.	[39, 64] [94, 110] [111, 121]
ILP	Integer Linear Programming (ILP) is used to place APs and assign channels by considering load balancing.	[85]
Priority-map	A floor plan is divided into pixels, which are assigned different priorities. Then, both CA and AP placement is solved according to built priorities.	[127]
Patching algorithm	An algorithm for joint AP placement and channel assignment is proposed to maximize throughput and maintain fairness through a time-efficient local searching heuristic called patching algorithm.	[89]
Coverage-oriented	Linear programming is used to optimize CA and AP placement via two objective functions.	[55]
Conflict-free coloring	Channel assignment and load balancing based on a conflict set coloring formulation that explicitly captures interference effects at clients and intrinsically exposes opportunities for better channel re-use.	[96]
Local coordination	Both APs and clients need to measure the interference sensed in all the frequency channels in order to derive an optimal solution for CA.	[44]

Table 2.1: Coordinated approaches for channel allocation

Many of the reviewed techniques imply a central node that makes the channel assignment, which is infeasible for uncoordinated networks such as in residential scenarios. In contrast, decentralized or uncoordinated mechanisms allow DCA based only on local information. However, many of them require inter-AP communication entailing non-negligible overheads and may lead to counterproductive solutions.

Coordinated approaches allow to effectively provide optimal (or close-to-optimal) solutions to the channel allocation problem since complete knowledge and control on the network is granted. Moreover, in [23], it is shown that coordinated approaches allow to significantly improve a network's performance, even with the existence of independent devices that operate with random channel selection. Table 2.1 collects some representative solution proposals for coordinated CA.

Technique	Description	Ref.
Least congested channel search	APs listen to all the channels and select the one with the lowest sensed interference. Other parameters, such as traffic information, are also considered.	[9]
MinMax	Based on the assumption that heavily loaded APs degrade the network performance, MinMax aims to minimize the maximum effective channel utilization of those devices.	[86, 138] [137]
Weighted coloring	Uses the sensed interference and the number of overlapping devices to minimize an objective function.	[95]
Pick-rand & pick-first	Based on the sensed interference, a new channel is chosen randomly or according to a ranking list (also based on the power sensed).	[12, 60] [13]
Channel hopping	With MAXchop, APs first obtain the hopping sequences of other interfering APs, and then compute the hopping sequence that maximizes their throughput.	[97]

Table 2.2: Uncoordinated approaches for channel allocation

Many limitations arise in the real world regarding coordinated approaches. First, overlapping BSS's are usually independent, so they cannot be managed by a central entity. Besides, the required underlying communication adds a higher degree of complexity, which solutions may lead to obtaining counter-productive overhead. Table 2.2 collects some representative works on uncoordinated approaches for CA.

2.2 Channel bonding

Channel bonding (CB) is the technique whereby nodes are allowed to use a set of idle basic channels for transmitting in larger bandwidths, thus potentially achieving higher throughput.[2] In Wi-Fi, while the basic channel width remains 20 MHz, the maximum bonded channel width has increased from 40 MHz in 802.11n [4] to 160 MHz in 802.11ac/ax [5, 7], and 320 MHz in 802.11be [6]. During this time, the standard has evolved to not only support wider bandwidths, but also to enable more sophisticated channel bonding policies: in 802.11n, only static channel bonding was allowed in which a fixed group of pre-configured basic channels must always be bonded. Today, the standard enables a far richer set of capabilities including dynamic selection of channel width as well as bonding both contiguous and non-contiguous channels through preamble puncturing.

> **Channel bonding** refers to selecting, on a per-frame basis, the basic channels to bond (or aggregate) within the channel allocation.

CSMA/CA operation in channel bonding WLANs

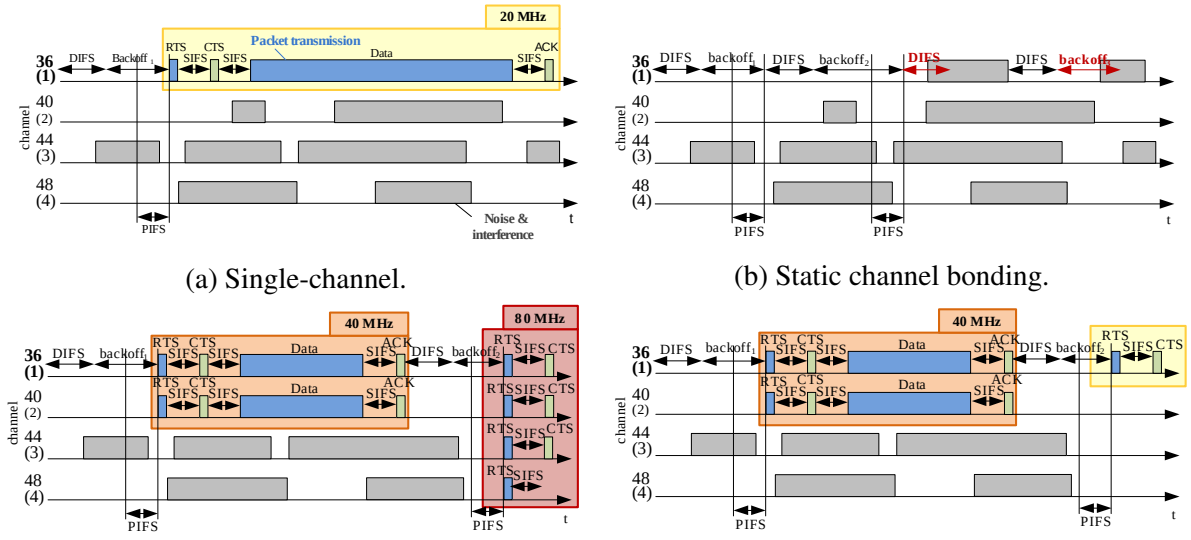
How does channel bonding work? According to the CSMA/CA operation, when a transmitter node belonging to BSS w has a data packet ready for transmission, it measures the power sensed

[2] In this dissertation, we refer to channel bonding as any technique that combines multiple basic channels in the transmission bandwidth. However, in the literature, *channel bonding* is sometimes referred exclusively to contiguous channels, whereas *channel aggregation* also comprises non-contiguous channels.

in primary channel p_w. Once the primary channel has been detected free, i.e., the power sensed at p_w is smaller than the clear channel assessment (CCA) threshold, the node starts the backoff procedure by selecting a random initial value of BO $\in [0, \text{CW} - 1]$ time slots of duration T_{slot}. The contention window is defined as $\text{CW} = 2^q \text{CW}_{\text{min}}$, where $q \in \{0, 1, 2, ..., m\}$ is the backoff stage with maximum value m, and CW_{min} is the minimum contention window. When a frame transmission fails, q is increased by one unit, and reset to 0 when the frame is acknowledged.

After computing BO, the node starts decreasing the backoff counter while sensing the primary channel. Whenever the power sensed at p_w is higher than its CCA, the backoff is paused and set to the nearest higher time slot until p_w is detected free again, at which point the countdown is resumed. When the backoff counter reaches zero, the node selects the transmission channel C^{tx} based on the set of basic channels found free and on the channel bonding policy $\mathcal{D}$.

The selected transmission channel C^{tx} is normally kept throughout the whole frame exchanges between the transmitter and receiver involved in the transmission of a data frame, which may aggregate multiple data packets. Namely, a request to send (RTS) – used for notifying the selected transmission channel – a clear to send (CTS), and an acknowledgment (ACK) or block ACK frame are also transmitted in C^{tx}. Any other node that receives an RTS in its primary channel with enough power to be decoded will enter in network allocation vector (NAV) state, used for deferring channel access and avoiding packet collisions. Notice that with the introduction of the adaptive RTS/CTS mechanism for dynamic bandwidth in the 802.11ac [5], the receiver can modify the original transmission channel of the RTS if it detects any secondary channel in C^{tx} busy. If so, the transmission bandwidth is reduced, and the receiver responds with a CTS packet in a subset of C^{tx}.



(a) Single-channel.

(b) Static channel bonding.

(c) Dynamic channel bonding.

(d) Stochastic channel bonding.

Figure 2.2: CSMA/CA temporal evolution of a node operating under (a) single-channel, (b) static channel bonding, (c) dynamic channel bonding, and (d) stochastic channel bonding, respectively, in an IEEE 802.11ax channelization scheme. While legacy packets (RTS, CTS, and ACK) duration is the same no matter the bandwidth, the data duration is reduced when transmitted in higher bandwidths.

Channel bonding policies

When the backoff terminates, the transmitter in w must identify the secondary channels free within its channel allocation C_w.[3] Then, its channel bonding policy $\mathcal{D}$ determines the transmission channel $C^{\text{tx}} \subseteq C_w$. We propose a generalization of contiguous channel bonding policies restricted to the 802.11ac/ax channelization,[4] which select the transmission channel as follows:

- **Only-primary (OP) or single-channel**: pick just the primary channel p_w for transmitting. It is the traditional channel access where no bonding is performed.
- **Static channel bonding (SCB)**: exclusively pick all the channels in C_w when found free. If any channel $c \in C_w$ is found busy, the transmission is aborted, and a new backoff counter is generated. This was the first type of bonding introduced in the 802.11n [4] amendment.
- **Dynamic channel bonding (DCB) or always-max (AM)**: pick the widest possible channel found free in C_w for transmitting including p_w and fulfilling channelization $\mathcal{C}$.
- **Stochastic channel bonding**: pick different subsets in C_w permitted by $\mathcal{C}$ at random according to a certain distribution. In this thesis, we study probabilistic uniform (PU) [26], which assigns the same probability to all the permitted channel ranges found idle at the backoff termination.

For the sake of illustration, let us consider the example shown in Figure 2.2 showing the evolution of a node implementing different CB policies. In this example, a node can transmit in the set of basic channels $C_w = \{1(p), 2, 3, 4\}$, where $p_w = 1$ is the primary channel. While OP picks just the primary channel, the rest of the policies try to bond channels differently. In this regard, SCB is highly inefficient in scenarios with partial interference. In fact, no packets can be transmitted with SCB in this example since the basic channel $\{3\} \in C_w$ is busy during the PIFS duration previous to the backoff terminations. However, more flexible approaches like AM and PU can transmit more than one frame in the same period. On the one hand, AM adapts in an aggressive (or greedy) way to the channel state. Specifically, it is able to transmit in 40 and 80 MHz channels at the end of the first and second backoff, respectively. On the other hand, the stochastic nature of PU makes it more conservative than AM. In the example, the node could transmit in 1 or 2 basic channels with the same probability (1/2) at the end of the first backoff. Likewise, after the second backoff, a channel composed of 1, 2, or 4 basic channels could be selected with probability (1/3).

Channel bonding poses several challenges [19, 40]. For instance, while RSSI provides information about signal quality and usually corresponds to the distance between transmitter and receiver, authors in [54] show that throughput is not always monotonic with RSSI. Besides, governmental regulations mandate that devices transmit below a maximum transmission power. Therefore, by doubling the channel bandwidth, the SNR is effectively decreased by 3dB, and thus, the transmission rate may decrease, whereas reception errors tend to increase [105]. Further, it is possible that increasing the bond size reduces the overall throughput of the network

[3] The 802.11ac/ax amendments mandate that secondary channels must be idle during at least a PIFS duration before the backoff terminates in order to use them for transmitting.

[4] While non-contiguous channel bonding is defined in the 802.11ax [3] in the form of preamble puncturing, it is yet not present in commercial devices. We study channelization-unrestricted policies in §4.3.

due to the fact that the bond may cause harmful co-channel and adjacent channel interference. As for energy efficiency, any increase in channel bandwidth implies that more power will be required to transmit if the coverage range is kept. Besides, extra power is wasted when nodes exchange control messages for multichannel sensing and operation.

Literature on channel bonding

Throughput gains of channel bonding have been demonstrated previously in testbeds. In particular, IEEE 802.11n static channel bonding has been shown to be affected not only by link signal quality, but also by the power and rates of neighboring links [52]. Likewise, intelligent channel bonding management was shown to benefit from identifying the signal strength of neighboring links and interference patterns [54]. High bandwidths were shown to be vulnerable to increased thermal noise or the power per Hertz reduction [18, 19]. Finally, an experimental study of IEEE 802.11ac channel bonding (both static and dynamic) showed that unplanned primary channel selection and bandwidth allocation may severely degrade the throughput of links operating at wider channels [141].

Type	Description	Ref.
Survey	Complete survey of CB schemes for wireless networks.	[40]
	Survey of bandwidth aggregation solutions in heterogeneous wireless networks.	[108]
Research	Discusses the impact of CB as well as the effects of both co-channel and adjacent channel interference on network performance.	[52]
	MAC protection mechanism to combat the hidden node problem on non-primary channels.	[59]
	Overviews 802.11 channel bonding. Simulation results show that DCB outperforms SCB MHz scheme when the secondary channels are occupied with moderate traffic loads.	[104]
	Provides a network detector that identifies interference conditions for predicting the impact on performance and make CB decisions accordingly.	[54]
	Assesses the suitability of CB for enhancing the performance of short-range WLANs, which are highly susceptible to external interference.	[33]
	Analyzes the interactions between groups of neighboring WLANs that use CB and evaluates the impact of those interactions.	[32]
	An analytical framework showing that CB is generally beneficial, though the extent of the benefits depend on features like the packet size and the total number of channels available for bonding.	[72]
	Analyzes via CTMN models the performance of channel bonding/aggregation strategies when primary channels are not time slotted and the time scale of primary activities is at the same level as the secondary users.	[71]
	The system model of NC-OFDM transmission for non-contiguous CB is presented and the system performance is studied by simulation. Results show that non-contiguous CB achieves better symbol error rate performance and results in slightly higher peak to average power ratio compared with contiguous CB.	[123]

Table 2.3: Representative works on channel bonding.

Simulation studies and analytical models have also been employed to study channel bonding, e.g., early simulation studies demonstrated throughput gains of channel bonding compared to single-channel transmission [59, 104]. Likewise, analytical models have been proposed to study channel bonding, especially through Markov chains [31, 33, 56, 78, 135]. For instance, high-density deployments are evaluated in [31], showing exposure to unfairness in groups of

overlapping BSS's. Analytical models for unsaturated traffic have been also proposed [78]. Table 2.3 collects some relevant works on channel bonding.

The joint problem of spectrum management

The joint problem of spectrum management refers to dealing with channel allocation and channel bonding at the same time. That is, selecting the primary and secondary channels to boost the performance of channel bonding policies.

> **The joint problem of spectrum management** refers to performing channel allocation in channel bonding WLANs to raise performance.

As discussed before, there are many valuable works in the literature dealing with channel allocation and channel bonding in wireless networks. However, only a few of them treat the joint problem altogether in the context of WLANs. We cover heuristic-based and ML approaches for the joint spectrum management problem in chapter §5 in chapter §6, respectively.

Chapter summary

This chapter described channel allocation and channel bonding and reviewed relevant works on the matter. We have seen that channel allocation is for assigning the primary and secondary channels, and channel bonding for selecting the transmission bandwidth on a per-frame basis. We also discussed the challenges posed by these techniques and proposed a generalization of channel bonding policies to be later studied. Next, we depict the enablers and tools we have used for getting the main results of this dissertation.

Chapter 3

METHODOLOGY AND ENABLERS

This chapter presents the methodology and the different enablers we used throughout the thesis, including the modeling of channel bonding WLANs through CTMNs, the Komondor wireless network simulator, and the WACA spectrum analyzer.

3.1 Modeling spectrum management through continuous time Markov networks

We introduce below our model for channel bonding WLANs based on CTMNs. The thesis papers of reference are **#1** [26] and **#2** [28].

3.1.1 CTMNs for channel bonding in spatially-distributed WLANs

A Markov chain or Markov network is a stochastic process that satisfies the Markov property, meaning a process that depends only on the present state to make predictions of future states, not on the sequence of events that preceded it. This Markov assumption allows for a significant reduction in the number of parameters when studying stochastic processes. Markov chains are used to compute the probabilities of events occurring by viewing them as states transitioning into other states or transitioning into the same state as before. A *continuous-time* Markov chain (CTMC) or network (CTMN) is a Markov chain in which, for each state, the process will change the state according to an exponential random variable and then move to a different state as specified by the transition probabilities.

The analysis of CSMA/CA networks through CTMN models for saturated WLANs was firstly introduced in [38]. Such modelling approach was later applied to IEEE 802.11 networks in [30, 32, 33, 35, 56, 73, 78], among others. Experimental results in [88, 103] demonstrate that CTMN models, while idealized, provide remarkably accurate throughput estimates for current IEEE 802.11 systems. A comprehensible example-based tutorial of CTMN models applied to different wireless networking scenarios can be found in [34]. Nevertheless, to the best of our knowledge, the works that model channel bonding through CTMNs study standard channel bonding policies or assume fully overlapping scenarios. Therefore, there is an essential lack of insights on more general Wi-Fi scenarios, where such conditions usually do not hold, and interdependencies among nodes have a critical impact on their performance.

In this section, we depict our extended version of the algorithm introduced in [56] for generating the CTMNs corresponding to spatially distributed WLAN scenarios implementing any channel bonding policy. The corresponding thesis paper is **#1** [26]. Notice that the CTMN model considers additive interference, which results from the combination of different simultaneous interfering transmissions. With this extension, as fully overlapping networks are no longer required for constructing the corresponding CTMNs, we can make more factual observations from spatially distributed deployments.

Implications of the use of CTMNs to model WLAN dynamics

It is worth pointing out some assumptions made by the CTMN model. First, only downlink traffic is assumed, and the interference produced in the uplink (i.e., CTS and ACK control packets) is not considered. Second, modeling WLAN scenarios with CTMNs requires the backoff and transmission times to be exponentially distributed. It follows that, because of the negligible propagation delay, the probability of packet collisions between two or more nodes within the carrier sense range of the others is zero. The reason is that two BSS's will never end their backoff at the same time, and therefore they will never start a transmission at the same time either. Also, it is shown that the state probabilities are insensitive to the backoff and transmission time distributions [88, 114]. However, even though the authors in [56] prove that the insensitivity property does not hold for channel bonding networks, the sensitivity to the backoff and transmission time distributions is minimal. Therefore, the analytical results obtained using the exponential assumption offer a good approximation for deterministic distributions of the backoff, data rate, and packet length. Despite all those are unrealistic assumptions, the model is particularly useful to depict inter-BSS interactions.

A crucial disadvantage of the CTMN model is its computational cost when characterizing crowded deployments. Indeed, modeling dense scenarios becomes intractable since the number of feasible states increases in a combinatorial manner with the number of BSS's and basic channels. However, the CTMN model is very useful to understand the new kind of inter-BSS interactions resulting from channel bonding in spatially-distributed deployments. Besides, it allows us to validate the implementation of the spectrum management techniques developed in the Komondor wireless simulator we present later in §3.2.

3.1.2 Constructing the CTMN

A state in the CTMN is defined by the set of active BSS's and the basic channels on which they are transmitting. Essentially, we say that a BSS is active if it is transmitting in some channel and inactive otherwise. For simplicity, we consider that each BSS is composed by one AP and one STA. Hence, we simply refer to the BSS activity as a single entity.[1] We define two types of state spaces: the global state space (Ψ) and the feasible state space ($\mathcal{S}$). A global state $\psi \in \Psi$ is a state that accomplishes two conditions: *i*) the channels in which the active BSS's are transmitting comply with the channelization scheme $\mathcal{C}$, and *ii*) all active BSS's transmit inside their allocated channels. That is, Ψ only depends on the particular channelization scheme $\mathcal{C}$ in use and on the channel allocation of the BSS's in the system. In contrast, a feasible state $s \in \mathcal{S} \subseteq \Psi$ exists only if each of the active BSS's in such state started their transmissions by

[1] Notice that the model could also estimate the long-run mean performance of multiple STAs by generating new states indicating the receiver STA corresponding parameters like the SINR or data rate.

accomplishing the CCA requirement (or contention) derived from the assigned channel bonding policy. Namely, given a global state space, $\mathcal{S}$ depends only on the spatial distribution and on the channel bonding policies assigned to each BSS.

The first step for constructing the CTMN is to identify the global state space Ψ, which is simply composed of all the possible combinations given by the channelization under consideration and the channel allocations of the BSS's. The feasible states in $\mathcal{S}$ are later identified by exploring the states in Ψ. The transitions among feasible states are represented by the transition rate matrix M. Essentially, as long as there are discovered states in $\mathcal{S}$ that have not been explored yet, for any feasible state $s_k \in \mathcal{S}$ not explored, and for each BSS w in the system, we determine if w is active or not. If w is active, we then set possible *backward* transitions to already known and unknown states. To do so, it is required to fully explore Ψ by looking for states where: *i)* other active BSS's in the state remain transmitting in the same transmission channel, and *ii)* BSS w is not active.

On the other hand, if BSS w is inactive in state s_k, we try to find *forward* transitions to other states by fully exploring Ψ looking for states where *i*) other active BSS's in the state remain transmitting in the same transmission channel, and *ii*) w is active in the new state as a result of applying its channel bonding policy $\mathcal{D}_w$. It is important to remark that in order to apply such a policy, the set of idle basic channels in state s_k must be identified according to the power sensed in each of the basic channels allocated to w and on its CCA threshold. Each transition between two states s and s' has a corresponding transition rate $M_{s,s'}$. For *forward* transitions, the packet transmission attempt rate (or simply backoff rate) has an average duration $\lambda = 1/(\mathbb{E}[\text{BO}]T_{\text{slot}})$, where $\mathbb{E}[\text{BO}]$ is the expected backoff duration in time slots, determined by the minimum contention window, i.e., $\mathbb{E}[B] = (\text{CW}_{\text{min}} - 1)/2$. Furthermore, for *backward* transitions, the departure rate (μ) depends on the duration of a successful transmission, i.e., $\mu = 1/T_{\text{suc}}$,[2] which in turn depends on both the data rate given by the selected MCS and transmission channel width, and on the expected frame length. In this regard, the data rate and packet error rate of a BSS w depends on the state of the system, which collects such information, i.e., $\mu_{\text{w}}(s)$.

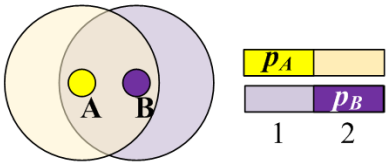


Figure 3.1: Deployment and channel allocation of *toy scenario I*.

In order to depict how to generate CTMNs given a WLAN deployment and channel allocation, let us consider *toy scenario I* shown in Figure 3.1, composed of two fully overlapping BSS's, A and B, with primary channel $p_A = 1$ and $p_B = 2$, respectively. Both BSS's are allocated the same two channels, i.e., $C_A = C_B = \{1, 2\}$.[3] So, the set of valid transmission channels according to the 802.11ac/ax channelization is $\mathcal{C} = \{\{1\}, \{2\}, \{1, 2\}\}$. Due to the fact that both BSS's are inside the carrier sense range of each other, their APs could

[2]The duration of a successful transmissions is the sum of the different delays involved in a packet exchange, $T_{\text{suc}} = T_{\text{RTS}} + T_{\text{SIFS}} + T_{\text{CTS}} + T_{\text{SIFS}} + T_{\text{DATA}} + T_{\text{SIFS}} + T_{\text{BACK}} + T_{\text{DIFS}} + T_{\text{slot}}$

[3]This toy scenario is selected for conveniently depicting the algorithm. CTMNs corresponding to non-fully overlapping scenarios can be also generated with the very same algorithm.

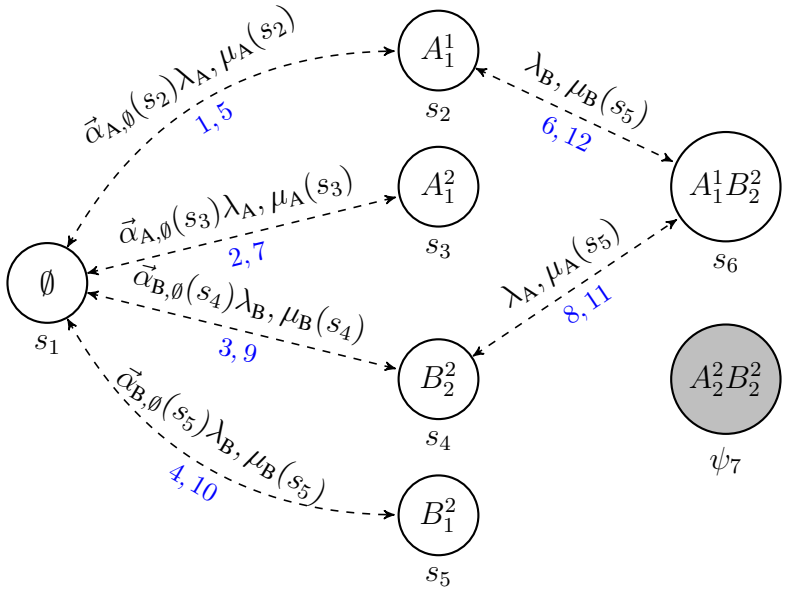


Figure 3.2: CTMN corresponding to *toy scenario I*. Transitions edges are dashed for referring to those that may be given or not depending on the channel bonding policy. For instance, state s_6 is only reachable for single channel and stochastic policies. The discovery order of the states and transitions in the algorithm (displayed in blue) corresponds to the stochastic channel bonding policy.

transmit simultaneously at any time t only if their transmission channels do not overlap, i.e., $C_{\text{A}}^{\text{tx}}(t) \cap C_{\text{B}}^{\text{tx}}(t) = \emptyset$. Notice that slotted backoff collisions cannot occur because their counters decrease continuously in time, and therefore two transmissions can be neither started nor finished at the same time.

The CTMN corresponding to *toy scenario I* is shown in Figure 3.2. Regarding the notation, we represent the states by the most left and most right basic channels used in the transmission channels of each of the active BSS's. For instance, state $s_6 = \text{A}_1^1\text{B}_2^2$ refers to the state where A and B are simultaneously transmitting in channels $C_{\text{A}}^{\text{tx}} = \{1\}$ and $C_{\text{B}}^{\text{tx}} = \{2\}$, respectively, whereas state $s_3 = \text{A}_1^2$ refers to the state where only A is transmitting in channels $C_{\text{A}}^{\text{tx}} = \{1, 2\}$. Concerning the state spaces, state $\psi_7 \notin \mathcal{S}$ is not reachable (i.e., it is global but not feasible) for any policy because of the overlapping channels involved. Further, state s_6 is only reachable when single-channel or stochastic channel bonding is used. So, it would be not feasible when using static channel bonding or dynamic channel bonding given that, at any time t that BSS A(B) finishes its backoff and B(A) is not active, A(B) picks the widest available channel, i.e., $C_{\text{A}}^{\text{tx}}(t) = \{1, 2\}$ or $C_{\text{B}}^{\text{tx}}(t) = \{1, 2\}$, respectively.

Every feasible *forward* transition rate from state s to s' is weighted by a transition probability vector $\vec{\alpha}_{w,s}(s')$ – dictated by the channel bonding policy of w – whose elements determine the probability of transiting to each of the possible states. As a consequence, $\vec{\alpha}$ must follow the normalization condition $\sum_{s'} \vec{\alpha}_{w,s}(s') = 1$. Table 3.1 shows the transition rate probabilities for the different channel bonding policies in *toy scenario I*. Notice that both the reachable states and transition rates may vary depending on the channel bonding policy.

Since there are a limited number of possible channels to transmit in, the constructed CTMN will always be finite. Furthermore, it will be irreducible because backward transitions between

$\mathcal{D}$	$\lvert\mathcal{S}\rvert$	$\vec{\alpha}_{\mathrm{A},\emptyset}(s_2)$	$\vec{\alpha}_{\mathrm{A},\emptyset}(s_3)$	$\vec{\alpha}_{\mathrm{B},\emptyset}(s_4)$	$\vec{\alpha}_{\mathrm{B},\emptyset}(s_5)$
OP	4	1.0	0.0	1.0	0.0
SCB	3	0.0	1.0	0.0	1.0
AM	3	0.0	1.0	0.0	1.0
PU	6	0.5	0.5	0.5	0.5

Table 3.1: Transition probabilities of A and B in *toy scenario I* for different channel bonding policies.

neighboring states are always feasible. Therefore, a steady-state solution to the CTMN always exists. However, due to the possible existence of one-way transitions between states, the CTMN is not always time-reversible, and the local balance may not hold [75]. Accordingly, it prevents to find simple product-from solutions to compute the equilibrium distribution of the CTMNs. Nonetheless, we can rely on the equilibrium distribution vector $\vec{\nu}$, which represents the fraction of time the system spends in each feasible state. Hence, we define $\vec{\nu}(s)$ as the probability of finding the system at state s. In order to obtain $\vec{\nu}$ we can use the transition rate matrix M given the system of equations $\vec{\nu}M = 0$. Once $\vec{\nu}$ is computed, estimating the average throughput experienced by a BSS w is straightforward,

$$\Gamma_w := \mathbb{E}[L]\left(\sum_{s\in\mathcal{S}}\{\text{SINR}_w(s) > \text{CE} : 0, 1\}\mu_w(s)\vec{\nu}(s)\right), \tag{3.1}$$

where $\mathbb{E}[L]$ is the expected data packet length, $\text{SINR}_w(s)$ is the SINR perceived by the STA in BSS w in state s and CE is the capture effect. The system aggregate throughput is therefore the sum of the throughputs of all the BSS's in the WLAN, i.e., $\Gamma := \sum_{w=1}^{M}\Gamma_w$.

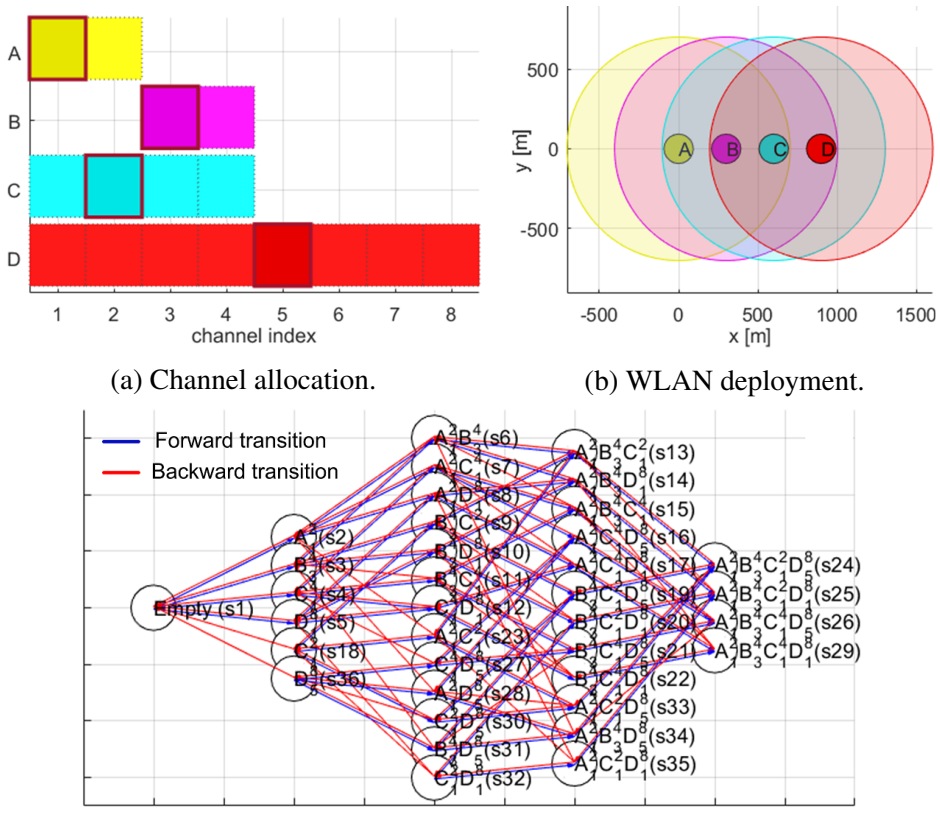
3.1.3 Extending CTMNs to unsaturated regimes

In thesis paper **#2** [28], we propose an extension of the algorithm for constructing the CTMNs to support traffic loads. That is, we extended the model presented in paper **#1** [26] by considering unsaturated regimes as proposed in [83]. As discussed before, a transition between two states s and s' in the CTMN has a corresponding transition rate $M_{s,s'}$. Now, when considering unsaturated regimes, for *forward* transitions, the average packet transmission attempt rate is $\rho_w\lambda$, where $\lambda = 1/(\mathbb{E}[\text{BO}]\cdot T_{\text{slot}})$.

Parameter ρ_w is the long-run stationary probability that BSS w has packets ready for transmission when the primary channel is sensed idle and so the backoff counter is active. Consequently, ρ_w depends on the traffic load ℓ_w of BSS w. A BSS becomes saturated ($\rho_w = 1$) whenever it is not able to deliver its traffic load, i.e., whenever it generates more packets than it transmits. As for *backward* transitions, the departure rate μ is independent on the load, so it is still determined by the duration of a successful transmission T_{suc}. Note that the unknown ρ parameters must be obtained by solving a non-linear system of equations, which in general does not have a closed-form. As done in [32], we use an iterative fixed-point approach for updating the ρ values until the throughputs of all the BSS's converge to their corresponding traffic load or the BSS's become saturated.

3.1.4 A Matlab framework for constructing WLAN CTMNs

The spatial-flexible CTMN (SFCTMN) framework[4] is the tool we have developed for automatically constructing the CTMNs of spatially-distributed WLANs with channel bonding capabilities. Namely, SFCTMN generates the CTMN corresponding to any spatial distribution and channel allocation of any WLAN scenario. Therefore, both the data and carrier sense ranges given by the power settings (i.e., transmission power, CCA level, sensitivity, path loss model, etc.) are entirely configurable.



(a) Channel allocation.

(b) WLAN deployment.

(c) Resulting CTMN.

Figure 3.3: Generation of the CTMN for a set of BSS's implementing dynamic channel bonding.

The procedure followed to generate the CTMN corresponding to a given WLAN scenario can be summarized in 3 general steps:

1. **Generate the global state space Ψ**: only some channel ranges for transmitting will be permitted depending on the input channel allocation and on the channelization considered. In this regard, a global state $\psi \in \Psi$, is a state that could be given if all BSS's operated in isolation (i.e., without contention). For instance, in Figure 3.3a, if a 802.11ac/ax channelization was selected, D_5^6 would be a global state (i.e., $D_5^6 \in \Psi$), while D_4^5 would not (i.e., $D_4^5 \notin \Psi$) since it is not compliant with the standard.

[4]All of the source code of SFCTMN is open, encouraging sharing of algorithms between contributors and providing the ability for people to improve on the work of others under the GNU General Public License v3.0. The repository can be found at `https://github.com/sergiobarra/SFCTMN`.

2. **Compute sensed power**: after the global state space Ψ is generated, the power sensed by each BSS's AP in each of the assigned channels (i.e., primary and secondaries) is computed for $\forall \psi \in \Psi$. Notice that the coverage areas in Figure 3.3b are represented for 20 MHz transmissions, but the power is actually reduced whenever doubling the bandwidth.
3. **Generate the feasible state space S**: a feasible state $s \in S$ is a state that considers carrier sense restrictions. Namely, active APs in s started transmitting in their corresponding channel ranges because *i*) the interference power in such channels was low enough to start them, and *ii*) their channel bonding policies determined to use such channels. For instance, in Figure 3.3c, we see a forward transition from $A_1^2 \in S$ to $A_1^2 D_1^8 \in S$ as APs A and D do not overlap because of their spatial separation and because dynamic channel bonding is implemented in all the BSS's. Moreover, we note that the CTMN also captures the backward transitions that occur unidirectionally. For instance, it is possible to transit from state D_5^8 to $\emptyset$, but not the other way around because D will always pick the widest channel combination whenever the rest of BSS's are inactive as occurs in state $\emptyset$.

3.2 The Komondor wireless network simulator

In this Section, we present Komondor,[5] an open-source, event-driven simulator based on the CompC++ COST library [43]. For details, please, refer to thesis paper **#3** [29].

3.2.1 The need of a new simulator

Komondor aims at fulfilling the need for assessing the novel features introduced in recent and future IEEE 802.11 amendments. The motivation for developing Komondor arises from *i*) the lack of next-generation WLAN-oriented simulators, *ii*) the complexity of extending current simulators comprising an exhaustive implementation of the physical (PHY) layer, *iii*) the large – and sometimes intractable – execution time required by other simulators to simulate high-density deployments, and *iv*) the need for conveniently incorporating ML features in the simulations. In short, Komondor is designed to efficiently implement new functionalities by relying on flexible and simplified PHY layer dependencies, to be faster than most off-the-shelf simulators, and to provide reliable simulations and a gentle learning curve to new users.

Wireless network simulators can be categorized into continuous-time and discrete-event. On the one hand, continuous-time simulators continuously keep track of the system dynamics by dividing the simulation time into minimal periods. On the other hand, in discrete-event simulators, events are used to characterize changes in the system. Accordingly, events are ordered in time and typically allow running faster simulations than continuous-time simulators. Besides, discrete-event simulators allow tracing events with higher precision. From the family of discrete-event driven network simulators, only a few ones are publicly available. Table 3.2 highlights the essential characteristics of the overviewed network simulators and Komondor.

Among the family of discrete-event simulators, we use ns-3 as a baseline for comparing against Komondor because of its popularity. Despite the plethora of features supported in

[5] All of the source code of Komondor, under the GNU General Public License v3.0., is open, and potential contributors are encouraged to participate. The repository can be found at `https://github.com/wn-upf/Komondor`

Simulator	Open-source	Source lang.	Complexity	GUI	11ax features	ML/based module
ns-3 [112]	Yes	C++	High	No	Partial	No
ns-2 [68]	Yes	C++/OTcl	Low	No	No	No
OMNET++ [120]	No	C/C++	Medium	Yes	No	No
OPNET [42]	No	C++	Medium	Yes	No	No
NetSim [109]	No	Java	Low	Yes	No	No
Komondor [29]	**Yes**	**C/C++**	**Low**	**No**	**Partial**	**Yes**

Table 3.2: Comparison of discrete-event driven wireless network simulators.

ns-3, it has some inherent limitations, such as the high complexity for developing new features/models as an extension of the simulator core. In particular, compatibility with the already existing/supported models is required and must be carefully ensured. Moreover, the integration of new features strongly depends on the community's willingness to contribute to development.

3.2.2 Design principles

Komondor aims at realistically capturing the operation of WLANs at the MAC layer. Henceforth, it reproduces actual transmissions on a per-packet basis. To that purpose, Komondor is based on the COST library, which allows building interactions between components (e.g., wireless nodes, buffers, packets) through synchronous and asynchronous events. While the former are messages explicitly exchanged between components through input/output ports, the latter are based on timers. In practice, components perform a set of operations until a significant event occurs. For instance, a node decreasing its backoff may freeze it when an overlapping node occupies the channel. The beginning and end of such a transmission are significant events, whereas decreasing the backoff counter is not. Nevertheless, events may be triggered by different timers.

Komondor entails a long-term project initiated and supervised by Sergio Barrachina-Muñoz and Dr. Francesc Wilhelmi. Currently, several contributors are involved in the extension of new modules. That is, the simulator is continuously evolving to include novel techniques and generally improve performance. The current version of Komondor (v2.0) includes, among others, the following thoroughly tested 802.11ax features: distributed coordination function, buffering and dynamic packet aggregation, channel bonding policies, dynamic MCS, bandwidth adaptive RTS/CTS, and NAV states. Future development stages are under progress, including features such as custom spatial reuse, MU-MIMO transmissions, and beamforming.

The current version of Komondor supports intelligent agents embedded to APs to monitor BSS's performance, run ML methods, and suggest new configurations to be applied by the BSS or the whole WLAN. The application of intelligent agents with Komondor has been previously studied in [128, 131], where decentralized learning is employed to both transmit power control (TPC) and carrier sense threshold (CST) adjustment. Also, results in this dissertation we use Komondor for reinforcement learning (see §6.5.3).

Komondor's core operation relies on states representing the status (or situation) in which a node can be involved. A state diagram summarizing both states and transitions is shown in Fig. 3.4. Roughly, a given node starts in the SENSING state, where multiple events can occur (e.g., a new packet is buffered or a new transmission is detected). Then, according to the noticed event, the node transits to the corresponding state.

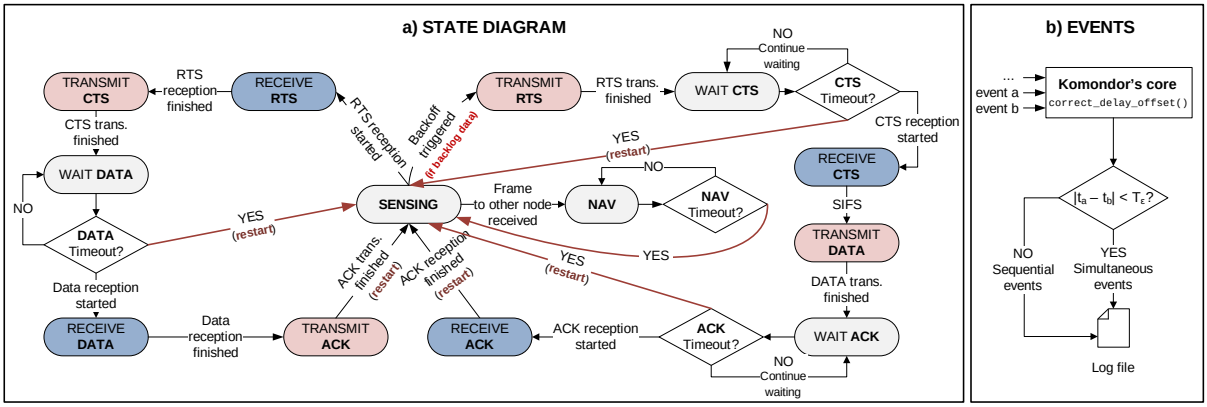


Figure 3.4: Komondor's state diagram and events. a) States are reachable by different transitions. b) Simultaneous events are properly processed through delay offset correction.

As for the events, each time a node performs an action that can affect the system (e.g., it starts transmitting a frame), an event is announced. Events in Komondor are lined up on the time axis and handled by the core entity. Events' management is similar in ns-3. However, the latter exhibits a significant limitation since events scheduled simultaneously can be executed in any order, leading to unpredictable results. That is, inconsistencies may occur if the execution order affects multiple simultaneous events (e.g., two packets arriving at the same time). To solve this, Komondor employs temporal variables to compare the exact timestamps at which two or more events were generated. As a result, Komondor can successfully simulate simultaneous events' behavior while keeping the states' logic.

In summary, Komondor is a wireless network simulator that stems from the need to provide a reliable and low-complexity simulation tool able to capture the operation of novel WLAN mechanisms like channel bonding or spatial reuse. Komondor's operation has been validated against the ns-3 simulator and analytical tools such as CTMNs and Bianchi's DCF model. In this regard, we have shown in [29] its effectiveness when dealing with high-density scenarios, thereby outperforming ns-3 with respect to the simulation time. Finally, we emphasize the potential of Komondor regarding complex scenarios and ML integration. In particular, a preliminary ML-based architecture is already implemented [132], so that intelligent agents can rule self-configuring operations at different communication levels.

3.3 Trace-driven evaluation of channel bonding

The Wi-Fi All-Channel Analyzer (WACA) is the first platform to simultaneously measure the energy in all the 24 bondable basic channels in the 5 GHz band with microsecond scale granularity. With WACA, we perform a first-of-its-kind measurement campaign in areas including urban hotspots, residential neighborhoods, universities, and a sold-out stadium with 98,000 fans and 12,000 simultaneous Wi-Fi connections. The gathered dataset is a unique asset to find insights otherwise not possible in the context of multi-channel technologies like Wi-Fi. The thesis papers of reference are **#5** [24] and **#6** [25].

3.3.1 Why a custom Wi-Fi spectrum analyzer?

Prior work performed spectrum measurements for Wi-Fi traffic, e.g., [36, 50, 65, 80, 81, 92, 107, 113, 116, 118, 126]. Example objectives include creating interference maps [65], assessing interference behavior [62], surveying Wi-Fi usage [50], quantifying spectrum occupancy in outdoor testbeds [107, 118], designing efficient scanning methods [113, 116], modeling spectrum use [126], opportunistic spectrum access [80, 81], dynamic channel selection [61], and assessing real-world network behavior by examining data from thousands APs [36]. Unfortunately, no such prior work provided simultaneous measurements across the entire 5 GHz band, which we require for our channel bonding study. While some papers do provide multi-channel measurements, e.g., [80,81,107,136], they do so via sequential scanning, thus taking on the order of seconds to change from one channel to the next, orders of magnitude beyond the transmission time scale for channel bonding. Namely, WACA measures all channels simultaneously using software defined radios (SDRs) having a sampling rate of 10μsec. Moreover, prior work does not consider (for example) stadiums, the potential costs of other BSS's deferring or colliding, the impact of fine-grained channel correlation, nor a diverse set of channel bonding policies.

Our objective is to capture all Wi-Fi channels simultaneously, i.e., all 24 basic (non aggregated) 20 MHz channels in the 5 GHz band that permit channel bonding. In principle, this could be achieved with an off-the-shelf spectrum analyzer. However, most spectrum analyzers cannot deal with the required bandwidth of this objective, i.e., they cannot simultaneously measure the entire Wi-Fi 5-GHz band: 645 MHz ranging from channel 36 to 161 (i.e., from 5170 to 5815 MHz). Moreover, wide-band spectrum analyzers covering this bandwidth lack resolution to analyze basic channels within the band. Likewise, one could envision a system comprising 24 off-the-shelf Wi-Fi cards as sniffers, one per basic channel. Unfortunately, such a system would be quite unwieldy and would introduce a challenge of ensuring synchronicity among wireless cards: restricting the delay between channel measurements to the order of nano/microseconds is unfeasible due to the hardware interrupt latency and jitter from the different peripherals [41, 57].

Thus, we design WACA to simultaneously measure power (and I/Q signals if required) on all the bondable basic channels in the 5 GHz band. Key benefits of WACA include the simplicity of experimental procedures (from deployment to post-processing), a dedicated RF chain per channel (covering the whole band and easing hardware failure detection), and the ease of configuration empowered by the WARPLAb framework [17].[6]

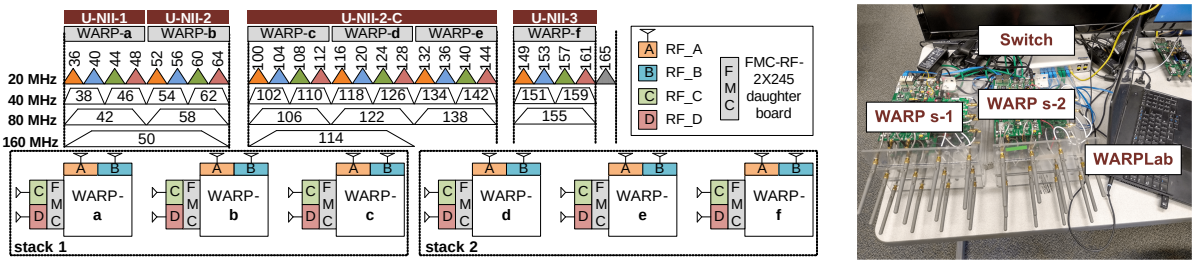
3.3.2 The WACA platform

The key building blocks of WACA are *i*) six WARP v3 programmable wireless SDRs [2], *ii*) six FMC-RF-2X245 dual-radio FMC daughterboards,[7] *iii*) 24 5-GHz antennas (one per RF chain), and *iv*) one Ethernet switch to enable communication between the WARPLab host (e.g., a PC) and the WARP boards. The preeminent building block is WARP, a scalable and extensible programmable wireless platform to prototype advanced wireless networks. The FMC-RF-2X245 module dual-radio FPGA Mezzanine Card (FMC) daughterboard extends the ca-

[6] All of the source code of WACA is open, encouraging sharing of algorithms between contributors and providing the ability for people to improve on the work of others under the GNU General Public License v3.0. The repository can be found at `https://github.com/sergiobarra/WACA_WiFiAnalyzer`.

[7] FMC-RF-2X245 datasheet: `https://mangocomm.com/products/modules/fmc-rf-2x245`, retrieved January 30, 2020.

pability of WARP v3 boards from 2 to 4 RF chains. Therefore, by combining 2 stacks of 3 WARP boards, each with their corresponding FMC-RF-2X245 daughterboards, we realize 24 RF chains (with one 5-GHz antenna each), enabling us to assign a single RF chain per basic channel. Finally, the Ethernet switch enables the communication from the WARP nodes to the WARPLab host. Fig. 3.5a shows the assignation of the RF chains to each basic channel allowed for bonding, and Fig. 3.5b depicts the physical realization of WACA.



(a) IEEE 802.11ax channelization at the 5-GHz and assignation per RF. (b) Deployment schematic.

Figure 3.5: The WACA spectrum analyzer.

WACA follows an iterative procedure for collecting power samples. Namely, in each iteration, WACA first simultaneously measures the power at each basic channel during T_{m} and then takes T_{proc} to process and forward the measurements to the WARPLab host. Both tasks are sequentially performed until the end of the measurement campaign. WARP boards install the MAX2829 transceiver, which has a fixed 10 Msps received signal strength sampling rate. Accordingly, since the measurement duration in an iteration is T_{m} = 1 second, the number of consecutive samples captured per basic channel per iteration is $n_{\text{s}}^{*} = 10^{7}$. Then, in each iteration, we store $|\mathcal{R}| \times n_{\text{s}}^{*} = 24 \times 10^{7}$ samples. Nonetheless, to decrease the amount of required memory, we downsample the gathered samples in each iteration by a 100$\times$ factor, thus reducing the data size per iteration from 60 MB to 600 KB. Essentially, while the transceiver measures one power sample every 100 ns by default, we keep just one sample every $T_s = 10$ µs. Notice that the resulting time scale is also suitable given Wi-Fi timings. Indeed, the short interframe space (SIFS) is the smallest interframe space and takes 16 µs ($> T_s$). Once initiated, WACA operates by itself, and no human intervention is required.

3.3.3 On-field dataset

Using WACA, we perform extensive measurement campaigns[8] covering two continents, dense urban areas, and multiple hours of samples in places of interest such as university campuses, apartment buildings, shopping malls, hotels and the Futbol Club Barcelona (a.k.a Barça) stadium (Camp Nou), one of the largest sports stadiums in the world. The measurements were taken from February to August 2019 in Houston, TX, USA, and Barcelona, Spain. The shortest campaign took 20 minutes and the longest covered more than one week. Table 3.3 lists the campaign locations. The complete dataset contains 153,033 1-second iterations accounting for 42 hours, 30 minutes, and 3 seconds of actual measurements in the 24 bondable channels of the 5 GHz Wi-Fi band. Figure 3.6 shows the deployment of WACA during two measurement campaigns.

[8]The WACA dataset v1 can be found at `https://www.upf.edu/web/wnrg/wn-datasets`.

Id	Location	Type	Duration
1_RVA	Rice Village Apart., HOU	Apartment	1 day
2_RNG	RNG lab at Rice, HOU	Campus	1 day
3_TFA	Technology for All, HOU	Com. center	1 day
4_FLO	Flo Paris, HOU	Cafe	1 hour
5_VIL	Rice Village, HOU	Shopping mall	20 min
6_FEL	La Sagrera, BCN	Apartment	1 week
7_WNO	WN group office, BCN	Campus	1 day
8_22@	22@ area, BCN	Office area	1 day
9_GAL	Hotel Gallery, BCN	Hotel	1 day
10_SAG	Sagrada Familia, BCN	Apartment	3 days
11_FCB	Camp Nou, BCN	Stadium	5 hours

Table 3.3: Measurement locations.

(a) Press box in the Camp Nou stadium. (b) Shopping mall parking lot.

Figure 3.6: On-field measurement campaigns with WACA: (a) press box in the Camp Nou stadium (Barcelona, Spain), and (b) parking lot in Rice Village shopping mall (Houston, TX, United States).

3.3.4 Trace-driven framework

Our objective is to study the performance that a fully backlogged channel bonding BSS, w, would obtain if it encountered the channels recorded in the measurement campaigns described above (see Fig. 3.7). The performance is a function of several factors such as which primary channel the bonding BSS selects, which channel bonding policy it employs, and the spectrum occupancy. Such factors are explored in §4.3. Here, we describe the overall methodology for all the experiments.

The datasets captured by WACA at scenario k are represented by a 2-dimensional matrix Y^k of size $(N_{\text{it}}^k \times n_{\text{k}}) \times |\mathcal{R}|$, where N_{it}^k is the number of iterations of scenario k, and any element $y_{t,c}^k$ represents the power value at temporal sample t in basic channel c. From Y^k, we generate a binary matrix X^k of same size through an occupancy indicator function, where any

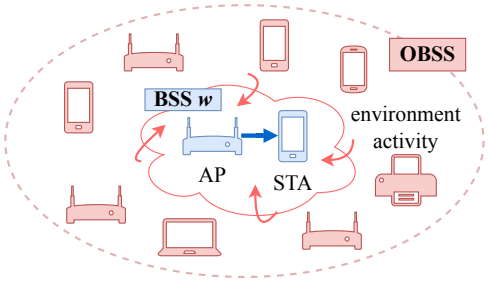


Figure 3.7: Diagram of the system model.

element $x^k_{t,c}$ represents whether channel c was occupied at temporal sample t (1) or not (0). Formally, $x^k_{t,c} = (y^k_{t,c} > \text{CCA} : 1, 0), \forall t, c$, where the CCA is set to -83.5 dBm (or 150 10-bit RSSI units), corresponding to the common CCA threshold -82 dBm plus a safety margin of -1.5 dBm. While 802.11ac/11ax introduce different CCA levels for the primary and secondary channels, in this work, we consider a more restrictive approach by assuming the same threshold in order to fairly compare different channel bonding policies. The mean occupancy at band $\mathcal{B}$ in scenario k is simply defined as

$$\bar{o}^k_{\mathcal{B}} = \frac{\sum_t \sum_{c \in \mathcal{B}} x^k_{t,c}}{N^k_{\text{it}} n_{\text{k}} |\mathcal{B}|}. \tag{3.2}$$

We develop a discrete state machine that characterizes how the channel bonding BSS responds to each power sample (or temporal sample) t according to the current state $S(t)$, and channel bonding policy $\mathcal{D}$, following the 802.11 standard. Given that the channel bonding BSS is fully backlogged, the set of possible states is $\mathcal{S} = \{\textit{Busy}, \textit{DIFS}, \textit{BO}, \textit{TX/RX}\}$. State *Busy* indicates that the primary channel is busy, *DIFS* represents the period before initiating the backoff process, the backoff counter is decreased during *BO* state, and *TX/RX* represents the actual frame transmission-reception (including the control frames RTS, CTS, and ACK, the DATA frame, and the SIFS periods in between). We represent the channel bonding BSS w as an AP and one or multiple clients that would perceive exactly the same spectrum activity as WACA captured in the measurement campaigns and must contend accordingly. To focus on channel bonding effects, we do not consider collisions within the channel bonding BSS, but only collisions that can occur due to other BSS's.

The set of basic channels selected for transmitting a frame depends both on the spectrum occupancy and on the selected channel bonding policy $\mathcal{D}$. Empty slots have a duration $T_{\text{slot}} = 10\,\mu\text{s}$ rather than $9\,\mu\text{s}$ (802.11's default value) to align the duration of an idle backoff slot with the sample duration. Hence, whenever the channel bonding BSS is in the backoff process at state *BO*, every idle sample at the primary channel p results in a backoff counter decrease of one empty slot. We use Wi-Fi parameters according to IEEE 802.11ax. After running the state-machine through all the temporal samples in the epoch, we compute the throughput Γ as the number of bytes in the successfully transmitted data packets n_{d} divided by the duration of the epoch, i.e., $\Gamma = (n_{\text{d}} L_{\text{d}})/T_{\text{per}}$, where L_d is the size of a data packet.

We emphasize the significance of the gathered dataset for finding insights, which would not be possible otherwise, given that simple channel occupancy models severely underestimate the available gains. An in-depth study on the dataset is conducted in §4.3.

Chapter summary

This chapter depicted the three key enablers we used throughout this thesis: a CTMN model for spatially-distributed WLANs, a wireless network simulator for 802.11ax (and beyond), and a custom spectrum analyzer for simultaneously measuring the whole 5 GHz band. The next chapter shows the main results and derived findings we got from using such enablers.

Chapter 4

PERFORMANCE EVALUATION OF CHANNEL BONDING

We depict next the principal results and derived findings on the performance of channel bonding in WLANs. These results were gathered in thesis papers **#1** [26], **#2** [28], **#5** [24], and **#6** [25].

4.1 Analytical characterization of channel bonding

In paper **#1** [26], we use CTMNs to characterize channel bonding policies in saturated WLANs.

4.1.1 States and transitions depend on the policy

Through CTMN modeling (see §3.1), we can get insights into channel bonding WLANs by observing what states are generated and their transitions. Let us consider now *toy scenario II* to discuss further the behavior of the selected channel bonding policies in the system. In other words, let us see how the CTMN changes according to the selected policies in terms of reachable states and generated transitions. In particular, *toy scenario II* is composed of $W = 2$ overlapping BSS's, A and B, with channel allocation $C_A = \{1, 2, 3, 4\}$ with primary channel $p_A = 2$, and $C_B = \{3, 4\}$ with $p_B = 3$ for A and B, respectively. Figure 4.1 shows an schematic of the deployment and channel allocation. Due to the fact that both BSS's are inside the carrier sense range of each other, their APs could transmit simultaneously at time t only if their transmission channels do not overlap, i.e., $C_A^{\text{tx}}(t) \cap C_B^{\text{tx}}(t) = \emptyset$.

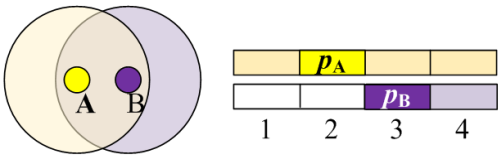
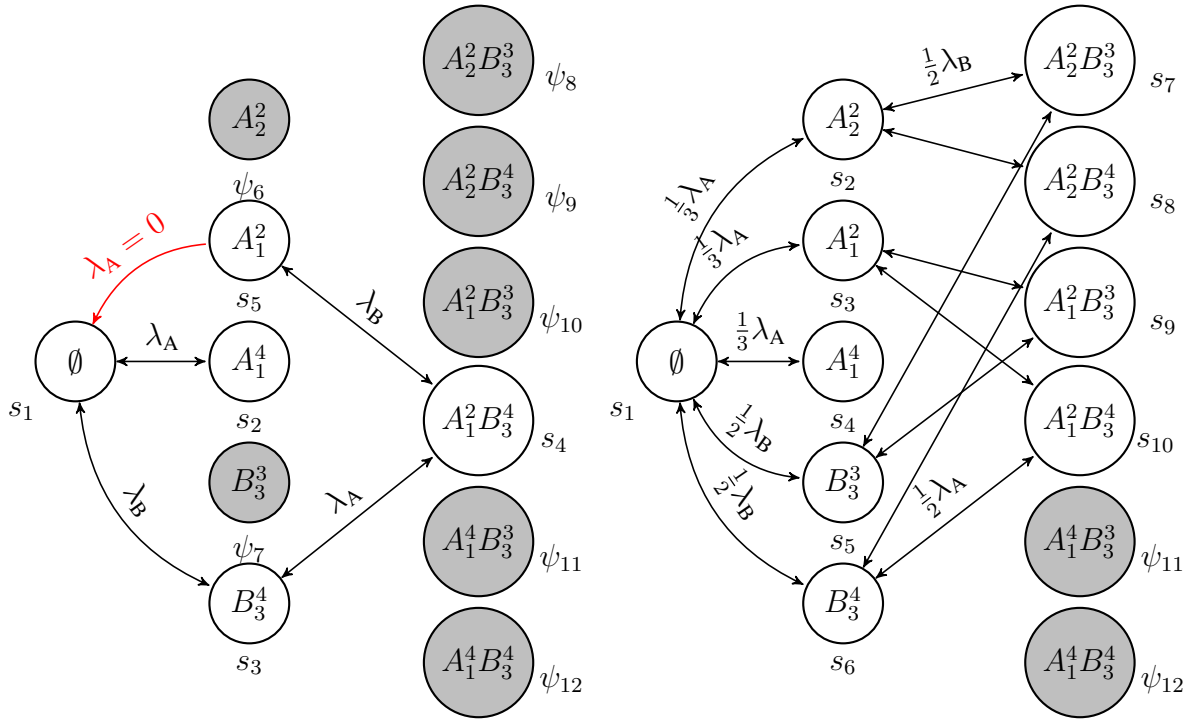


Figure 4.1: Deployment and channel allocation of *toy scenario II*.

Now, let us generate the WLAN CTMNs when applying two different channel bonding policies: dynamic channel bonding (or AM) and stochastic channel bonding with uniform probabilities (i.e., PU). Figure 4.2a and Figure 4.2b show the resulting CTMNs for the former and the

latter policies, respectively. We observe three main differences: *i*) there are states only reachable with stochastic channel bonding, generating a larger set of reachable states $\mathcal{S} = \{s\}$, *ii*) transitions rates are weighted for stochastic channel bonding according to the transition probabilities (in this case, we assume uniform probabilities, e.g., 1/3 λ from $\emptyset$ to $\{A_1^4\}$ since A has three bonding combinations to choose when all its allocated spectrum is free), and *iii*) there is a unidirectional transition from A_1^2 to $\emptyset$ just for dynamic channel bonding. That is, while it is possible to reach state A_1^2 from $A_1^2B_3^4$ when B finishes its transmission, and then reach $\emptyset$ from A_1^2 when A finishes, it is not possible to go from $\emptyset$ to A_1^2 because dynamic channel bonding will always select the maximum allowed bandwidth when possible (4 channels, in this case, resulting always in a transition from $\emptyset$ to A_1^4).



(a) Dynamic channel bonding.

(b) Stochastic channel bonding (uniform).

Figure 4.2: CTMNs of *toy scenario II*: (a) dynamic channel bonding and (b) stochastic channel bonding in the form of PU. For the sake of lightweighting the CTMN representation, we indicate only some forward transition rates (λ) and obviate backward transition rates (μ).

Finding #1: States and transitions in the CTMN are generated by the WLAN deployment and the channel bonding policies of the BSS's, which are determined by their transition probabilities. Accordingly, while different channel bonding policies share the same global states, we observe that feasible states may be different, with some policies never visiting some states. Likewise, unidirectional (backward) transitions from one state to another can also appear, generating non-time-reversible CTMNs.

4.1.2 Being greedy is not always convenient

Most often, the best channel bonding policy for increasing the individual throughput, no matter what policies the rest of BSS's implement, is dynamic channel bonding. Nonetheless, there are exceptions like the one we present below. Besides, if achieving throughput fairness between all BSS's in the WLAN is part of the goal, more conservative policies are generally preferable. Therefore, there is not always an optimal standard policy to be implemented by all the BSS's. Indeed, there are cases where different policies must be assigned to different BSS's to increase both fairness and individual throughput.

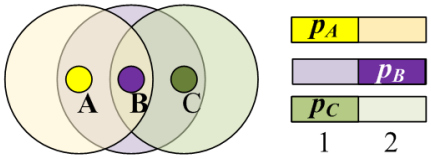


Figure 4.3: Deployment and channel allocation of *toy scenario III*.

Let us consider *toy scenario III* where $W = 3$ BSS's are located in a line in such a way that they are in the carrier sense range of the immediate neighbor. So, BSS A and C overlap only with B, and B overlaps with both. The proposed channel allocation is as follows: $C_{\mathrm{A}} = C_{\mathrm{B}} = C_{\mathrm{C}} = \{1, 2\}$ and $p_{\mathrm{A}} = p_{\mathrm{C}} = 1$, $p_{\mathrm{B}} = 2$. Figure 4.3 illustrates the corresponding deployment and channel allocation. Table 4.1 shows, for different combinations of channel bonding policies, the individual and aggregated saturation throughput, Γ_w and Γ, respectively, and the Jain's fairness index,

$$J = \frac{(\sum_{w=1}^{W} \Gamma_w)^2}{W \sum_{w=1}^{W} \Gamma_w^2}. \tag{4.1}$$

We note that, while implementing AM in all the BSS's the WLAN's aggregated throughput is the highest (i.e., $\Gamma = 403.49$ Mbps), the throughput experienced by B is the lowest (i.e., $\Gamma_{\mathrm{B}} = 3.58$ Mbps), leading to a very unfair flow-in-the-middle situation as indicated by $J \approx 0.67$. We also find that it may be much more convenient for B to use stochastic channel bonding than dynamic channel bonding to maximize its individual throughput. Namely, when A and C implement dynamic channel bonding (i.e., $\mathcal{D}_A = \mathcal{D}_C =$ AM), it is preferable for B to implement stochastic channel bonding (i.e., $\mathcal{D}_B =$PU) and make the WLAN reach states in the CTMN where A and C transmit only in their primary channels, thus increasing both the throughput of B and the fairness accordingly.

Policy			**States**	**Throughput [Mbps]**				**Fair.**
$\mathcal{D}_A$	$\mathcal{D}_B$	$\mathcal{D}_C$	$\lvert S\rvert$	Γ_{A}	Γ_{B}	Γ_{C}	Γ	J
AM	AM	AM	5	199.96	3.58	199.96	**403.49**	0.67
AM	PU	AM	10	149.41	62.45	149.41	361.27	0.89
PU	AM	PU	25	109.84	108.44	109.84	328.12	**0.99**
AM	AM	PU	9	111.31	106.91	110.33	328.55	**0.99**

Table 4.1: Policy combinations effect on throughput and WLAN fairness.

Looking at the fairest combinations, we notice that A, C, or both must implement PU to let B transmit with a similar amount of opportunities. This is achieved by the stochastic nature

of PU, which lets the CTMN explore more states. Accordingly, B experiences the highest throughput, and the system achieves complete fairness (i.e., $J \approx 1$). Nonetheless, the price to pay is to decrease the throughput of A and C significantly. In essence, this toy scenario is a paradigmatic example showing that with less aggressive policies like PU (of probabilistic nature), not only more states in the CTMN can be potentially explored with respect to AM, but also the probability of staying in states providing higher throughput (or fairness) may increase. Therefore, since it generally does not exist a global policy that satisfies all the BSS's in the system, different policies should be adopted depending on the target performance metric.

Finding #2: Being greedy with dynamic channel bonding does not always maximize the individual throughput. There are scenarios where more conservative policies like stochastic channel bonding are preferable since they tend to force the WLAN to reach more favorable states in the CTMNs. Besides, we find that there is not a unique policy for maximizing the performance metrics (e.g., aggregate throughput or fairness). Instead, the assignation of different policies to different BSS is sometimes required.

4.2 Assessment of high-density deployments

Apart from modeling WLANs through CTMNs, in the thesis papers **#1** [26] and **#2** [28], we simulate high-density WLANs deployments under a variety of channel bonding policies, node densities, and traffic loads. We rely on the Komondor wireless network simulator [29] to simulate thousands of scenarios in different deployments and configurations.

4.2.1 Type of scenarios

The type of high-density Wi-Fi deployments under evaluation consists of randomly spread nodes in square maps of different areas. Random primary channel, spectrum allocation, and, sometimes, channel bonding policy, is assigned to the BSS's in the WLAN. We deploy W BSS's each with one AP and one or multiples STAs. We use typical IEEE 802.11ax parameters.[1] In this section, we first assess full-buffer traffic loads and then a Poisson process where BSS w generates a data packet every $t_w \sim \text{Exponential}(1/\ell_w)$, being ℓ_w the mean traffic load in packets per second. We focus on studying a particular BSS in the WLAN, normally named A, as well as the whole WLAN. An example of the scenarios considered is shown in Figure 4.4.

4.2.2 On the individual throughput and fairness

In thesis paper **#1** [26], we study the throughput of saturated BSS's under full-buffer traffic.

On the individual throughput

We aim to identify the optimal policy that a particular BSS should implement to increase its own throughput in a high-density, uncoordinated Wi-Fi deployment. We consider three rectangular

[1] We refer the reader to [26–28] for more details

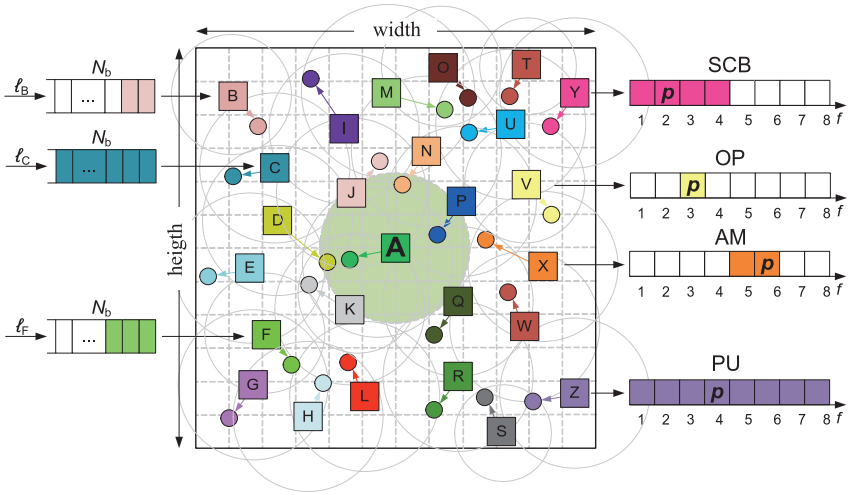


Figure 4.4: Example of a high-density deployment consisting of multiple BSS's with different traffic loads, channel allocations, and channel bonding policies in a spatially distributed WLAN.

maps of decreasing areas (sparse, semi-dense and high-dense) with one BSS (A) located at the center and $W - 1 = 24$ BSS's spread uniformly at random in the area. We assume BSS A has $n_{\text{STA,A}} \sim U[1, 20]$ STAs. Channel allocation (including the primary channel) is set uniformly at random to all the BSS's, except A. While the central BSS is also set with a random primary channel, it is allocated the widest channel (i.e., $C_{\text{A}} = \{1, ..., 8\}$) to provide more flexibility and capture complex effects. The the rest of BSS's select their channel bonding policy (OP, SCB, AM, or PU) uniformly at random and A's policy is set deterministically to each of them.

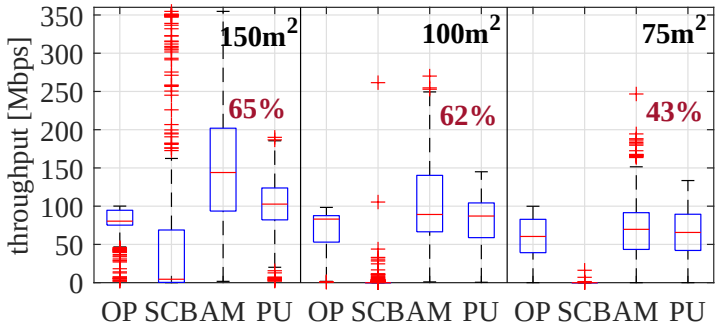


Figure 4.5: Distribution of the individual throughput experienced by BSS A for different policies in areas of increasing density.

Figure 4.5 shows the average throughput experienced by A in the considered maps. The first noticeable result is that, in dense scenarios, SCB is non-viable for BSS's with wide allocated channels because they are most likely prevented from initiating transmissions. Regarding the rest of the policies, on average, A's throughput is higher when implementing AM in all the maps. Especially, AM (and SCB in some cases) stands out in sparse deployment. Nevertheless, there is a clear trend for dense deployments to pick just one channel when implementing AM or PU. That is why OP provides an average throughput relatively close to the ones achieved by these policies. Nonetheless, as the throughput's high standard deviation indicates, there are important differences regarding Γ_{A} among the evaluated scenarios. The number in red

inside the boxplot of Figure 4.5 indicates the share of scenarios where AM provides the highest individual throughput for A: 65%, 62%, and 43% for sparse, dense, and high-density maps, respectively.

We see that in most cases, AM performs better than PU in terms of individual throughput. However, in a significant share of scenarios, PU outperforms AM. Also, there are scenarios where the throughput experienced by PU with respect to AM is extremely higher (up to hundreds of Mbps). This mainly occurs when the neighboring nodes occupy A's primary channel through complex interactions caused by network asymmetries, keeping its backoff counter frozen for long periods. These are clear cases where adaptive policies could significantly improve performance.

Finding #3: In saturated WLANs, dynamic channel bonding normally reaches higher individual throughput than the rest of channel bonding policies, especially in sparse deployments. However, we find a significant number of scenarios where more conservative approaches like stochastic channel bonding are preferable in terms of individual throughput, thus corroborating **Finding #2** for denser deployments.

On the bandwidth efficiency and fairness

In order to asses the use of the spectrum, we define the average bandwidth usage of a BSS w as

$$\text{BW}_w = \frac{1}{T_{\text{obs}}} \sum_{c=1}^{N_c} t_w^{\text{tx}}(c) B, \tag{4.2}$$

where T_{obs} is the observation (or simulation) duration, $N_c = 8$ is the number of basic channels in the system, $t_w^{\text{tx}}(c)$ is the duration that BSS w is transmitting in a channel containing at least the basic channel c, and $B = 20$ MHz is the bandwidth of a basic channel. The mean spectrum used by all the BSS's, i.e., $\text{BW} = \sum_{w=1}^{W} \text{BW}_w$ is shown in Figure 4.6 for the different channel bonding policies.

Similarly to the throughput results, while OP and PU do not leverage the free spectrum in low-density scenarios, SCB and AM do so by exploiting the most bandwidth. Instead, when the number of nodes per area increases, SCB suffers from heavy contention periods, reiterating the need for flexibility to adapt to the channel state. In this regard, we note that AM exploits

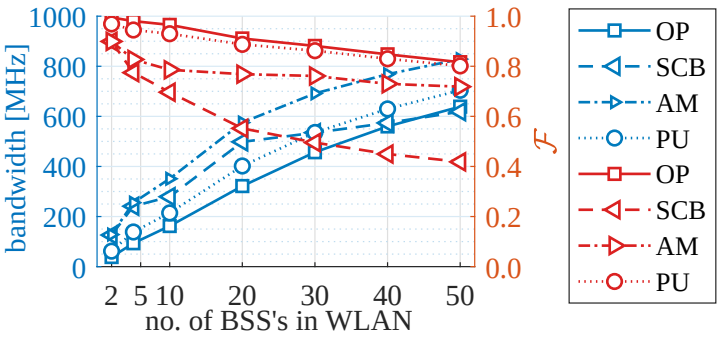


Figure 4.6: Total bandwidth usage and throughput Jain's fairness index vs. node density.

the most bandwidth on average for any number of WLANs. Nonetheless, neither the average throughput per BSS nor the spectrum utilization may be a proper metric when assessing the whole WLAN's performance. Namely, having some BSS's experiencing high throughput when some others starve is often a situation preferable to be avoided. In that sense, we focus on the fairness, which is suggested by the *boxes* and outliers in Figure 4.5, and more clearly represented by the expected Jain's fairness (4.1) index shown in Figure 4.6.

As expected, the policy providing the highest fairness is OP. No matter the channel allocation, BSS's only pick their primary channel for transmitting when implementing OP; hence the fairness is always maximized at the cost of probably wasting part of the frequency spectrum, especially when the node density is low. In this regard, PU also provides high fairness while exploiting the spectrum to a larger extent, which increases the average throughput per BSS accordingly. Regarding the aggressive policies, SCB is the most unfair policy due to its *all or nothing* strategy. Therefore, it seems preferable to prevent BSS's from applying SCB in dense scenarios. However, even though being aggressive, AM can adapt its transmission channel to the spectrum's activity, thus providing both higher throughput and fairness. Still, as indicated by the *boxes* and *outliers* of Figure 4.5, AM is not *per se* the optimal policy. There are scenarios where PU performs better in terms of both fairness and throughput. Consequently, there is room to improve the presented policies with some smarter adaptation or learning approaches (e.g., tuning properly the transition probabilities $\vec{\alpha}$ when implementing stochastic channel bonding).

There are also some phenomena that are worth to be mentioned. Regarding backoff decreasing slowness, it can be the case that a BSS w is forced to decrease its backoff counter very slowly due to the fact that neighboring BSS's operate in a channel including the primary channel of w. That is why more fairness is achieved with PU in dense networks as such neighboring BSS's do not always pick the whole allocated channel. Thus, they let w decrease its backoff more often and transmit accordingly. Finally, concerning the transmission power and channel width, we have observed that transmitting just in the primary channel can also be harmful to other BSS's because of the higher transmission power used per 20 MHz channel. While this may allow using higher MCS, it may also cause packet losses in neighboring BSS's operating with the same primary channel due to heavy interference.

Finding #4: Dynamic channel bonding provides the higher bandwidth usage at the expense of being the most detrimental in terms of fairness.

4.2.3 Traffic load and delay

While saturated regimes offer valuable insights on worst-case scenarios, BSS's are characteristically unsaturated with traffic patterns that deeply depend on the application/s being supported. That is, unsaturated traffic patterns fit better to real-world problems and allow us to assess critical performance metrics like the delay. In such scenarios, overlapping approaches seem to be even more convenient since the sensed channels usually remain free during larger periods. Unsaturated BSS's are studied in thesis paper **#2** [28].

Successful traffic delivery

Figure 4.7 shows the probability of BSS A to successfully transmit its traffic load ℓ_A, i.e., $P_{\mathrm{A}} = \mathbb{P}\big(\Gamma_{\mathrm{A}} \geq (1-\epsilon_\Gamma)\ell_{\mathrm{A}}\big)$. Note that we use an error margin $\epsilon_\Gamma = 0.05$ to cope with the stochastic packet generation of the performed simulations. So, we say that a BSS w is satisfied if its throughput satisfaction $\zeta_w = \Gamma_w/\ell_w$ is close to 1. As expected, SCB is viable only for a few scenarios when the traffic load is minimal. This is because the rest of BSS's most likely prevent A to initiate any transmission by occupying part of its allocated channel C_{A}. Instead, the other policies perform much better – especially AM – since they avoid high probability saturation even for high traffic loads. While A avoids saturation in some scenarios for $\ell_{\mathrm{A}} < 92.16$ Mbps with OP and $\ell_{\mathrm{A}} < 122.88$ Mbps with PU, respectively, the aggressive adaptability nature of AM allows avoiding saturation in scenarios even where $\ell_{\mathrm{A}} = 184.32$ Mbps. Nevertheless, we note that there are scenarios where AM heavily suffers from the hidden node problem since the SINR at the receiver is notably reduced. Accordingly, even though the AP may find the whole spectrum free, the STA cannot decode most of the RTS or data frames due to interference. This effect can be seen in the P_{A} improvement of OP or PU for low traffic loads.

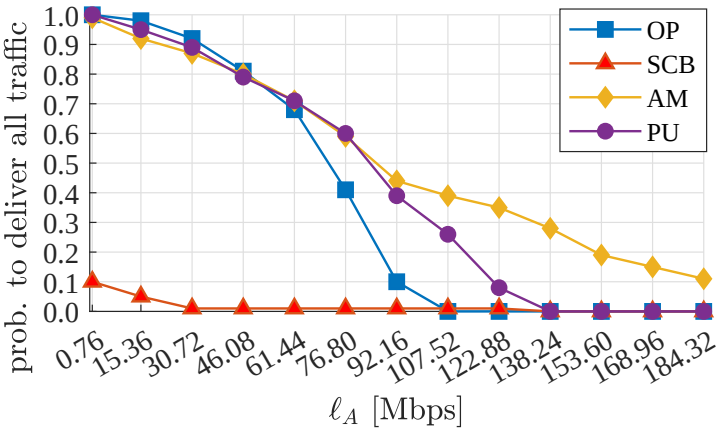


Figure 4.7: Probability of successfully delivering all the generated traffic load.

Looking for the lowest delay

We now assess the share of scenarios where each policy provides the smallest average packet delay for BSS A d_A under different traffic loads ℓ_A. Figure 4.8 compares the share of scenarios where AM is better than the best combination of OP and PU, i.e., AM is compared against $\mathcal{D} = \mathrm{argmin}_{\mathrm{OP,PU}}\, d_A$ for each of the simulated scenarios. We observe that in most of the cases, AM outperforms OP and PU, especially for scenarios with mid-high traffic loads. Nonetheless, for low loads, we note that there is always a better choice than AM for reducing the delay, corroborating the outcomes from the previous sections. In addition, there is a significant share of scenarios where OP and, especially, PU provide similar or even smaller delays than AM for all traffic loads. This mainly occurs when A and its neighboring nodes are able to concurrently transmit in different channels through interactions that are not given when implementing AM.

Essentially, when A transmits in its whole allocated bandwidth, neighboring BSS's with primary channels overlapping with A's transmission must wait until it finishes. Afterward, such BSS's can terminate their backoffs and select a transmission channel including A's primary in

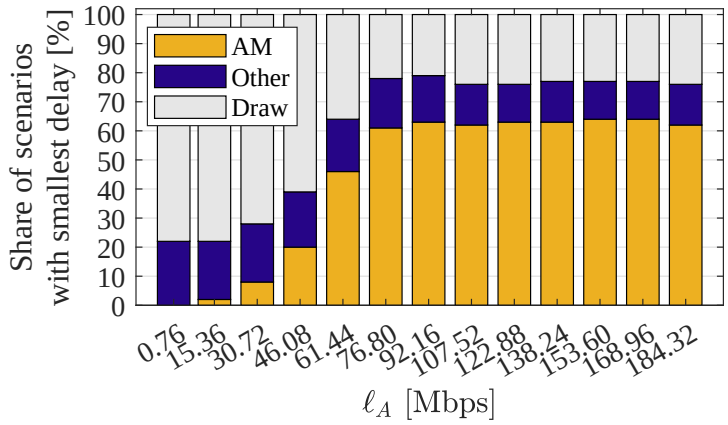


Figure 4.8: Share of scenarios providing the lowest delays: AM vs. others. Scenarios where the delay difference between policies is less than 1 ms are labeled as draw.

turn. This generates *all or nothing* states that keep A's backoff frozen for larger periods. Instead, if A transmits in narrower channels by implementing OP or PU, such BSS's could transmit at the same time in non-overlapping channels and enable more successful parallel transmissions. This effect can also be seen in the average number of packets aggregated per frame: AM aggregates more packets on average for low traffic loads since the buffer of one BSS can be filled with more packets during the transmission of the other. In turn, when the backoff expires, larger frames are sent.

To better illustrate this phenomenon about the smaller delays of OP for low loads, let us consider a conceptual example with two overlapping BSS's, A and B, with primary channels $p_A = 1$ and $p_B = 2$, respectively. Both share a channel allocation containing basic channels 1 and 2. What is better in terms of delay under low loads? To reduce channel access delay by letting them transmit in parallel, i.e., let A transmit in channel 1 and B transmit in channel 2. By transmitting in one channel, the other is forced to do the same even if it implements dynamic channel bonding. So, when A finishes its transmissions, most probably, just the second channel will be occupied by B, so A can transmit in the first channel again, reducing the contention accordingly.

In summary, we see that overlapping approaches can significantly enhance traditional single-channel performance in terms of delay and throughput in uncontrolled high-density deployments. Still, there are cases when an overlapping approach that always selects the maximum available bandwidth can be counterproductive. Despite the intrinsic uncertainty of spatially distributed WLAN deployments, we can state as a rule of thumb that channel bonding is convenient when applied through spectrum-adapting policies. Nonetheless, as indicated by the scenarios where OP or PU outperformed AM, there is room for improvement through smarter adaptation by adopting policies on a per-BSS basis. Hence, we envision that the most effective way of using channel bonding is to allocate all the nodes with the whole available frequency spectrum and assign the primary channel smartly. Moreover, more significant improvements could be achieved by endowing the nodes with the capability to recover from lousy situations like flow-in-the-middle, which are more likely to happen when neighboring BSS's implement greedy policies.

Finding #5: While dynamic channel bonding is usually the best in terms of individual throughput for any traffic load, it is more convenient in terms of delay to use conservative approaches under low loads. The reason lies in the *chain-reaction* generated by selecting (conservative) smaller bandwidths, which tends to allow neighboring BSS's to pick other chunks of the spectrum, thus avoiding contention between them and reducing the channel access delay consequently.

4.3 Trace-driven simulations from real-world measurements

This section collects the main results and findings on the trace-driven analysis of the dataset gathered with WACA. Details on the experiments and further results can be found in thesis papers **#5** [24] and **#6** [25].

4.3.1 A look into the world's Wi-Fi spectrum

We first assess each location's entire activity record at band $\mathcal{B}$, where $\mathcal{B}$ is a predefined set of adjacent channels. For instance, U-NII-1 band is defined as $\mathcal{B} = \{1, 2, 3, 4\}$, corresponding to basic channels 36, 40, 44, and 48 (see Figure 3.5a).[2] Figure 4.9 shows the normalized mean idleness of each band, i.e., the mean number of samples that were found idle in each channel of band $\mathcal{B}$, including night hours (if any). We observe that the spectrum is idle most of the time in all scenarios except the stadium, indicating that the 5 GHz band is still profoundly underutilized even in densely populated areas.

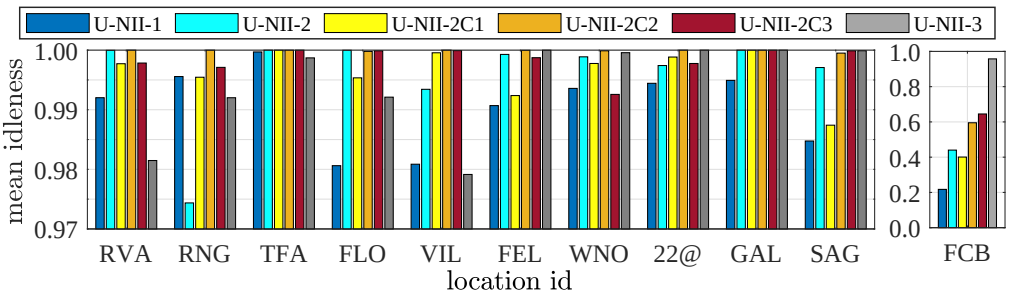


Figure 4.9: Mean idleness per band.

Figure 4.10 shows the daily temporal evolution of 4 example locations (2 apartments in the upper subplots and 2 university campuses in the lower subplots). For the sake of representation, we plot the normalized occupancy of the whole band averaged in periods of 10 minutes. Concretely, we normalize with respect to the highest 10-minutes average occupancy encountered in each location. We observe higher activity at working hours in the campus locations and a much less variable pattern in the apartment locations.

Regardless of the low mean occupancy at all bands, we find peaks of activity in the generic scenarios (those excluding Camp Nou stadium) yielding maximum occupancy values near 45%

[2]The rest of bands are sequentially composed of groups of four consecutive 20-MHz channels. So, the next band is U-NII-2 with channels $\{5, 6, 7, 8\}$ (basic channels 52, 56, 60, and 64), and the last one is U-NII-3 with channels $\{21, 22, 23, 24\}$ (basic channels 149, 153, 157, 161).

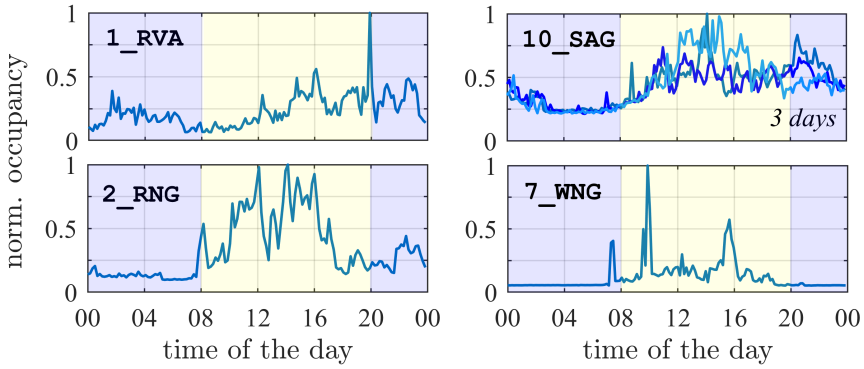


Figure 4.10: Run chart of the normalized occupation for apartments (upper) and universities (lower) highlighting day (yellow) and night (blue).

over 100 ms periods. As for the stadium, the peaks of activity reach occupancies as high as 99%. So, while significant opportunities for channel bonding are expected under such low mean occupancy, we focus our study on those challenging peaks of activity to evaluate channel bonding performance in current and next-generation high-density WLANs.

Finding #6: Overall, the Wi-Fi's 5-GHz band is currently underutilized even in dense urban scenarios. However, we find periods of high activity that impose challenges to channel bonding performance.

4.3.2 Characterization of channel bonding in the real-world spectrum

To provide meaningful experiments, we separately consider two 160-MHz bands composed of 8 basic channels: the U-NII-1&2 and part of the U-NII-2c sub-bands, $\mathcal{B}_{1\&2} = \{1, 2, 3, ..., 8\}$ and $\mathcal{B}_{2c} = \{9, 10, 11, ..., 16\}$, respectively. These sub-bands cover from channel 36 to 64 and from channel 100 to 128, respectively (see Figure 3.5a). Notice that these are the only sub-bands that allow to perform 160-MHz transmissions in the IEEE 802.11ac/ax channelization. Moreover, we focus on epochs (or periods) of duration $T_{\text{per}} = 100$ ms (containing 10^4 temporal samples each) for which the mean occupancy at such sub-bands is at least 5%, i.e., $\bar{o}_{\mathcal{B}} \geq 0.05$, where $\mathcal{B} \in \{\mathcal{B}_{1\&2}, \mathcal{B}_{2c}\}$.

How will other BSS's respond to the simulated channel bonding BSS? For most experiments, we consider that they will defer their transmissions. Namely, the channel bonding BSS needs the channels to be available only when its countdown timer expires. If the bonding BSS does transmit but the trace indicates that a channel would have been occupied at some point during the transmission, we consider that such other BSS's will sense the bonding BSS and defer. The exception is in the experiments on hidden nodes, in which we consider that other BSS's are hidden, do not sense the channel bonder, and cause a collision.

Contiguous vs. non-contiguous channel bonding

As we have seen in the previous sections, channel bonding is a mechanism to increase the instantaneous throughput. We now assess its performance limits by considering unrestricted,

non-standard compliant policies as for the channelization. In particular, we consider contiguous channel bonding to bond any combination of consecutive *basic* channels, including the primary, and non-contiguous channel bonding, which can utilize additional channels compared to contiguous, by "skipping over" the busy channels to find the next unused one. Here, we explore the gains of this flexibility as well as (in rare cases) the losses by comparing the throughput of contiguous and non-contiguous channel bonding in three load regimes: low ($\bar{o}_{\mathcal{B}} \leq 0.1$), medium ($0.1 < \bar{o}_{\mathcal{B}} \leq 0.2$) and high ($\bar{o}_{\mathcal{B}} > 0.2$), respectively.

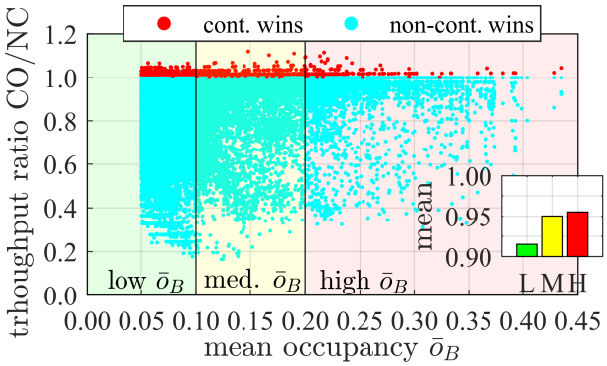


Figure 4.11: Throughput ratio of contiguous vs. non-contiguous channel bonding. The bar chart inset depicts the mean aggregated ratio for low (L), medium (M), and high (H) occupancy regimes.

Figure 4.11 shows the throughput ratio of contiguous to non-contiguous channel bonding $\Gamma_{\text{CO}}(p)/\Gamma_{\text{NC}}(p)$, where $\Gamma_{\pi}(p)$ is the throughput achieved by policy π when selecting primary p in a given period. We plot the ratio for all possible primaries in $\mathcal{B}_{1\&2}$ and $\mathcal{B}_{2c}$. The data reveals two remarkable phenomena. First, contiguous outperforms non-contiguous in 2.5% of the cases (albeit with a modest throughput difference of 1.9%). But since non-contiguous is more flexible, how can it ever do worse? The answer is that the two policies result in different instants for transmission attempts. The contiguous policy occasionally (and quite randomly since the traces are the same) ends with more favorable attempt instants.

Nonetheless, in most cases, non-contiguous obtains higher throughput. For example, in many periods, at least one 20-MHz channel is idle during the whole period, which will always yield a gain for non-contiguous, but only sometimes yields a gain for contiguous bonding. In some cases, the difference can be quite high (e.g., a ratio of approximately 0.2 observed in low load). The origins of such extreme cases are the selection of the primary channel, which we explore next. Second, the bar chart inside Figure 4.11 reveals that both contiguous and non-contiguous channel bonding perform quite close on average for all occupancy regimes (low, medium, and high), and especially for the latter, as high load results in far fewer bonding opportunities overall.

Finding #7: Non-contiguous almost always outperforms contiguous channel bonding, and their throughput differences are occasionally over a factor of 5. Nonetheless, their average throughputs are quite similar, which may ultimately favor contiguous channel bonding, since it is simpler to implement.

Primary channel selection

The primary channel selection is critical for channel bonding, as it is where the backoff procedure runs. Namely, the transmitter must wait for a transmission opportunity on the primary channel and then explore adding channels. For contiguous channel bonding, there is an additional "edge effect". For example, having the first channel as primary only allows bonding higher-numbered channels, whereas having a central channel allows both higher and lower, provided they are contiguous in both cases. To explore this issue, we define the best-throughput $\Gamma^*(\pi)$ of policy π as the throughput achieved when selecting the best primary channel $p^* \in \mathcal{B}$, i.e., the primary channel that maximizes throughput in each period when implementing π. So, apart from assessing the throughput when considering all possible primary channels, we also evaluate an upper bound on performance by focusing only on the best primary channel.

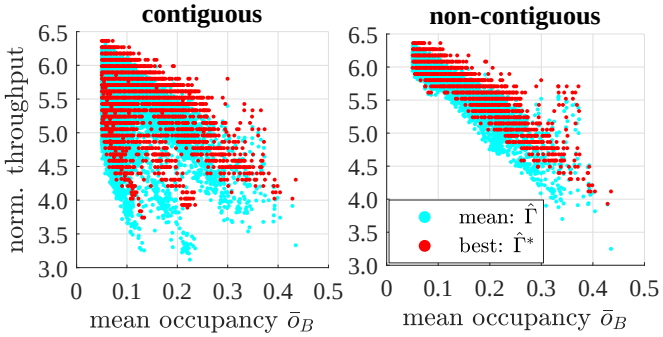


Figure 4.12: Normalized throughput of the best primary channel (*best*) and mean throughput for every primary channel (*mean*).

Figure 4.12 shows the throughput gain with respect to the throughput achieved by single-channel transmission, i.e., $\hat{\Gamma}_\pi = \Gamma_\pi/\Gamma_{\text{SC}}$, where Γ_{SC} and Γ_π are the throughput achieved by single-channel transmission and channel bonding policy π, respectively. Such normalized throughput is presented in two ways for each period: the *mean* value $\hat{\Gamma}$ in the legend refers to the mean normalized throughput achieved by selecting each of the possible 8 primary channels per sub-band, and the *best* value $\hat{\Gamma}^*$, to the normalized throughput achieved by the primary p^* providing the highest throughput.

First, observe that both contiguous and non-contiguous channel bonding outperform single-channel by a factor of at least 3 to over 6. Second, observe the region where contiguous channel bonding has the lowest gains. Strikingly, while selecting the best primary channel boosts throughput, these lower-gain cases cannot be entirely eliminated with better primary channel selection. Thus, in these cases, the available spectrum does not have a consistent structure and only non-contiguous bonding can exploit the gaps (and there are indeed many gaps as the load is low here). Next, observe the high variance of the throughput due to the spectrum activity distribution. In particular, throughput differences for similar occupancy values can be up to $1.9\times$ and $1.7\times$ for contiguous and non-contiguous, respectively. Lastly, despite these first two findings, selecting the best primary channel indeed provides substantial gains and can raise throughput by up to 68% and 64% for contiguous and non-contiguous, respectively. While non-contiguous channel bonding might seem impervious to primary channel selection as it can bond any channels, recall that even non-contiguous requires an idle primary channel to initiate a transmission.

Finding #8: The selection of the best primary channel improves average throughput by over 60%. But for contiguous channel bonding, it cannot overcome high variance and lower gain scenarios due to the channel occupancy dynamics of the spectrum band.

Inter-channel occupancy correlation

When a transmitter finds its primary channel available for transmission and attempts to bond channels, it aims to find secondary channels available. Thus, the correlation among channel occupancies can be expected to help channel bonding performance. Here we study the spectral correlation of the best primary channel with other channels and compute the mean correlation coefficient as

$$\begin{aligned} \xi &= \sum_{p\in\mathcal{B}, p\neq p^*} \varrho(p^*, p)/\big(|\mathcal{B}| - 1\big), \text{ with} \\ \varrho(p^*, p) &= \frac{\mathbb{E}[(p^* - \bar{p^*})(p - \bar{p})]}{\sigma_{p^*}\sigma_p} \end{aligned} \tag{4.3}$$

where $\mathcal{B} \in \{\mathcal{B}_{1\&2}, \mathcal{B}_{2c}\}$, thus $|\mathcal{B}| = 8$, $\varrho(p^*, p)$ is the Pearson correlation coefficient of the temporal occupancy of channels p^* and p, as a function of the means, $\bar{p^*}$ and $\bar{p}$, and standard deviations, σ_{p^*} and σ_p, of channel p^* and p, respectively. Notice that this assessment of spectral correlation by simultaneously measuring all channels is achieved for the first time by WACA.

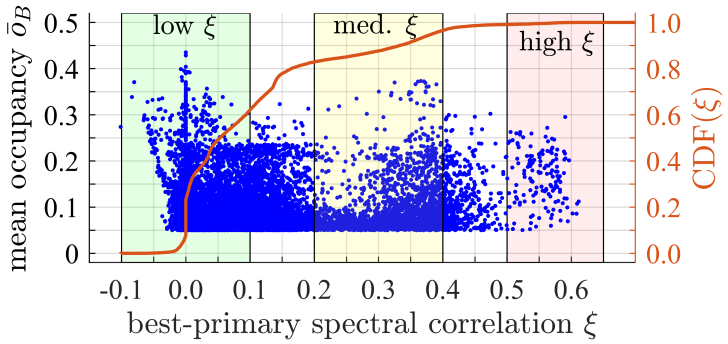
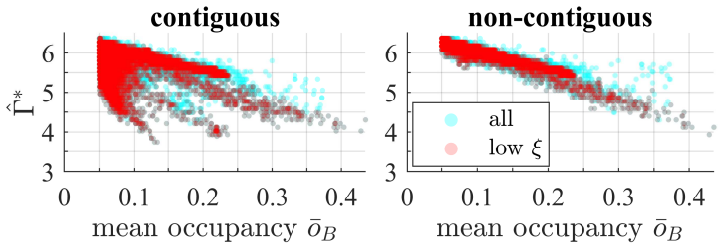


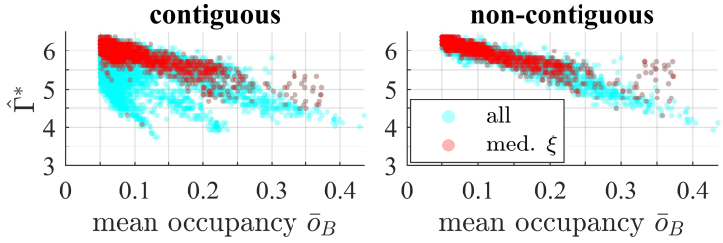
Figure 4.13: Distribution and empirical CDF of the best primary channel correlation ξ.

Figure 4.13 shows both the distribution of ξ vs. mean occupancy $\bar{o}_{\mathcal{B}}$ and the empirical cumulative distribution function (CDF) of ξ. We observe that ξ ranges approximately from -0.10 to 0.60, so we define three correlation levels for ease of analysis: low ($\xi \leq 0.1$), medium ($0.2 \leq \xi < 0.4$), and high ($\xi \geq 0.5$). While we find that most of the periods (62%) fall inside the low correlation range, 17% of them present medium or higher correlation, indicating a significant amount of correlated epochs. A key origin of correlation among channels in the data set may be that the APs are already using channel bonding. Indeed, even though the dataset does not provide header information, some subsets of traces have multiple contiguous channels having almost identical busy/idle evolution. This suggests that channel bonding was employed during some portions of the measurement campaign.

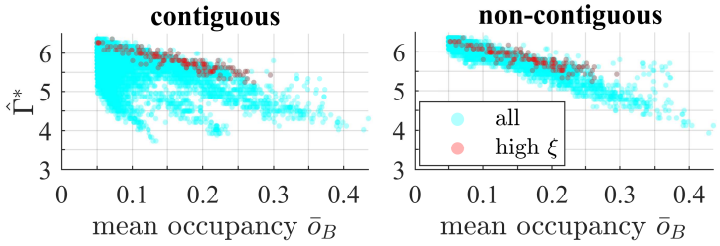
Figure 4.14 shows throughput given selection of the best primary channel, with the sub-figures highlighting intervals with low, medium, and high correlation. We observe that most



(a) Low-correlation periods.



(b) Low-correlation periods.



(c) High-correlation periods.

Figure 4.14: Evolution of the best-primary normalized throughput vs. correlation. We use transparencies at the points to highlight denser regions.

of the measured channel occupancies show minimal inter-channel correlation. However, the figures for periods of medium and high correlation indicate that with higher correlation among channel occupancy, the lower performance periods are increasingly avoided, especially for contiguous channel bonding. Interestingly, this effect holds regardless of the mean occupancy $\bar{o}_{\mathcal{B}}$.

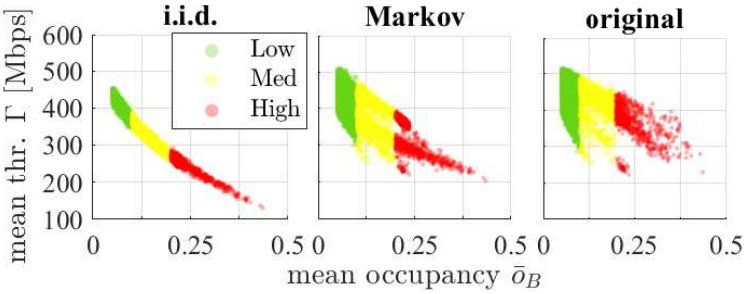
Finding #9: Significant periods present medium to high correlation among channel occupancies. Such periods improve the performance of channel bonding regardless of the load. This can provide a hint as to what ideally would happen if multiple BSS's employ channel bonding since channel bonding itself creates channel-occupancy correlation.

A Markov model for channel occupancy?

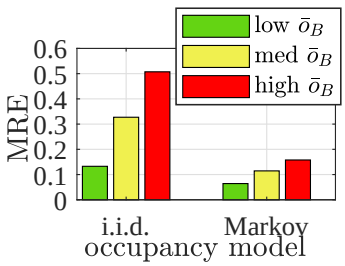
In §4.3.2, we found that even though channels could be considered to be uncorrelated on average since the activity is expected to be generally from separate BSS's, traces exhibited a significant amount of epochs of inter-channel occupancy correlation. With this experiment,

we aim at pointing out the importance of capturing inter-channel correlation by comparing the original traces against two simple models.

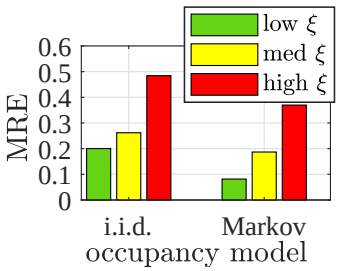
In particular, we first consider a Markov model that characterizes each channel as an independent two-state (occupied, not occupied) Markov chain with exponential holding times for each state. We compute the mean transition rates (and hence mean occupied and not-occupied time) for each channel from the traces. We then compare the performance between the parameter-matched Markov model and the actual traces. As a baseline, we also consider a *uniform i.i.d.* model in which each temporal sample is occupied or not according to an i.i.d. Bernoulli distribution, with each channel, again having the mean band occupancy matched from data. Our objective is not to develop a statistical occupancy model but rather to study the extent to which simple models can or cannot characterize the behavior we observe. We study below downlink throughput for these two models as compared to the traces when applying contiguous channel bonding. Since the primary channel selection impacts the throughput, we evaluate the selection of the 16 possible primary channels: 8 in sub-band $\mathcal{B}_{1\&2}$ and 8 in $\mathcal{B}_{2c}$.



(a) Throughput vs. occupancy



(b) Error vs. occupancy.



(c) Error vs. correlation.

Figure 4.15: Original vs. synthetic traces.

Figure 4.15a shows the raw throughput (considering every possible primary) of the original traces and the synthetic model-generated occupancy. While both models capture the general trend of throughput decaying with increasing occupancy, the difference between datasets is evident in the scatter plots. Figure 4.15b and Figure 4.15c depict the mean relative error (MRE) of the models' throughput with respect to the original traces grouping by load and correlation regime, defined by the thresholds presented in §4.3.2 and §4.3.2, respectively. We observe that the i.i.d model is completely misleading, with a mean error of around 50% for high loads and high correlation. Unfortunately, although the Markov model can capture the mean on and off times for each channel, we observe that the mean error is still significant (up to 16% and 37% on average for high load and high correlated periods, respectively). Worse, particular periods assessed through the Markov model lead to large errors up to 33%, 58%, and 62% for low, medium, and high occupancy regimes, respectively.

While we expected the higher relative errors to arise for high load periods, it is surprisingly critical the impact of not capturing correlation on both models. In fact, in contrast to the correlation distribution of the original traces shown in Figure 4.13, both models present a distribution similar to a delta function centered at $\xi = 0$. Given that the original mean occupancy is kept for all the periods generated by the models, results suggest that missing correlation is the main cause of outraging the models' accuracy.

Finding #10: Simple channel occupancy models severely underestimate the available gains for channel bonding. Thus, modeling occupancy behavior by introducing inter-channel correlation remains an important avenue for future work.

How much are others hindered?

When other BSS's defer to a channel bonder, it will add to their channel access latency and decrease their throughput. Here, we study how much all other BSS's are hindered due to channel bonding and introduce a bandwidth deprivation metric ω as the aggregate bandwidth that would have been active (i.e., was being used by neighbor BSS's) during the transmissions of the channel bonding BSS. This metric can serve as an upper bound to how much others are hindered by viewing that all of these attempts by other BSS's would be lost vs. deferred. More formally, let

$$\omega = \frac{\left(\sum_{f=1}^{n_f} \sum_{t \in \mathcal{T}_f} \sum_{c \in \mathcal{B}_{\text{tx},f}} x_{t,c}\right) B T_{\text{s}}}{T_{\text{per}}}, \tag{4.4}$$

where n_f is the number of frames transmitted, $\mathcal{T}_f$ is the set of consecutive temporal samples used for transmitting frame f, $\mathcal{B}_{\text{tx},f}$ is the set of channels used for transmitting frame f, T_{s} is the duration of a temporal sample ($10\,\mu\text{s}$), $B = 20$ MHz is the bandwidth of a basic channel, and $T_{\text{per}} = 100\,\text{ms}$ is the observation duration. We normalize by $T_{\text{per}} = 100\,\text{ms}$ rather than the transmission time of each policy to provide a fair comparison. That is, ω represents how much raw bandwidth is deprived to surrounding BSSs in absolute terms [MHz], without considering how many transmissions are performed by each policy.

Figure 4.16 shows the distribution (boxplot) of the bandwidth deprivation in MHz and Mbps for every primary channel using the occupancy categories defined in §4.3.2. To estimate the corresponding data rate, we assume that all occupied samples are comprised of 20-MHz data frames also transmitted at MCS 9. The figure illustrates the intrinsic consequences of bonding channels without considering how others might use the spectrum during the transmission. Namely, a channel bonder may find that most channels are free at backoff termination, subsequently occupying most (or all) of them. However, it may occur that right after starting the transmission, external activity appears in some of the channels being used, thus hindering surrounding BSS's.

Second, we observe that the differences in bandwidth deprivation for non-contiguous vs. contiguous channel bonding are quite modest, even when non-contiguous channel bonding achieves higher throughput. The key reason is that most high occupancy periods concentrate activity in some 20 MHz channels, often leaving at least one 20 MHz channel idle during nearly the entire period. In contrast to contiguous, non-contiguous channel bonding can always achieve small throughput gains by bonding those idle channels regardless of the primary

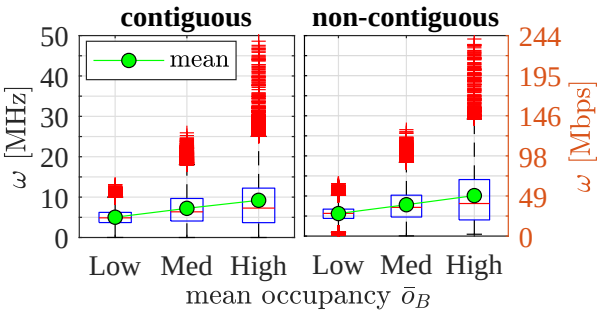


Figure 4.16: Distribution of bandwidth and data rate lost for other BSS's for every primary channel.

channel allocation. Also, despite the low average impact, there are multiple outliers in which the bandwidth lost by other BSS's is significantly higher than average, thus manifesting the worst-case external effect.

Finally, we can ask if channel bonding is simply a zero-sum-game: do the gains of a bonding BSS subtract from other BSS's throughput? To address this question, we assess whether the average throughput gained by channel bonding exceeds the worst-case prevented transmissions by other BSS's, using single-channel transmission as a baseline. Thus, we define the ratio

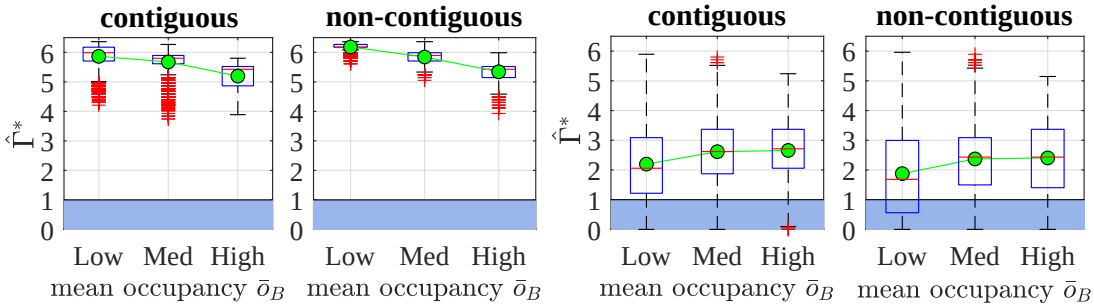
$$\kappa_\pi = \frac{\bar{\Gamma}_\pi - \bar{\Gamma}_{\text{SC}}}{\bar{\omega}_\pi - \bar{\omega}_{\text{SC}}}, \tag{4.5}$$

where $\bar{\Gamma}_\pi$ and $\bar{\Gamma}_{\text{SC}}$ are the mean throughput of policy π and SC, and $\bar{\omega}_\pi$ and $\bar{\omega}_{\text{SC}}$ are the corresponding mean throughput deprivation. A value $\kappa_\pi = 1$ would indicate that the trade-off between the bonding gain and external throughput deprivation is actually a zero-sum-game, whereas $\kappa_\pi > 1$ would indicate net gains. The results indicate that both contiguous and non-contiguous channel bonding yield substantial net gains over all occupancy regimes. Specifically, $\kappa_{\text{CO}} = 95.2, 68.9, 49.4$ for contiguous, and $\kappa_{\text{NC}} = 91.9, 65.7, 45.5$ for non-contiguous, under low, medium, and high occupancy regimes, respectively.

Finding #11: In the worst case, a channel bonding BSS can hinder neighbors by 100's of Mb/sec. However, the *average* hindrance is quite low, and neighbors can defer instead of not transmitting to reduce the impact. Moreover, the data rate gained by bonding far exceeds the worst-case data rate lost by neighbors since neighbors are not always backlogged during bonding epochs.

The hidden cost of hidden nodes

Here, we continue our study of potential detrimental effects of channel bonding by considering the case of hidden nodes. We again impose a worst-case scenario on the measurements as follows: channel access occurs as previously with contention occurring on the primary channel and channel bonding adding channels according to availability and the policy, contiguous or non-contiguous. However, here we reconsider the measured activity that occurs during the bonded transmission. While in the previous subsection we considered that the other BSS's



(a) Hindering. (b) Hidden terminals.

Figure 4.17: Distribution of the norm. best-primary-throughput.

would have deferred, here, we consider that other BSS's are hidden from the bonded transmission and still transmit and result in frame loss. Thus, this scenario considers that the RTS/CTS mechanism is not used or fails and disallows capture effects in which the higher signal-to-noise ratio receiver correctly decodes, e.g., [84]. Thus, in these two ways, this experiment provides an empirical upper bound as to the damage that hidden terminals could do to a channel bonded transmission.

In particular, we assume that a frame f is lost if temporal samples are found active in any of the channels used for transmitting f. Specifically, f is lost if the following expression holds,

$$\sum_{t \in \mathcal{T}_f} \Big(\big(\sum_{c \in \mathcal{B}_{\text{tx},f}} x_{t,c} \big) \geq 1 : 1, 0 \Big) \geq \epsilon |\mathcal{T}_f|, \tag{4.6}$$

where $\mathcal{T}_f$ is the set of consecutive temporal samples required for transmitting frame f, and the term $\sum_{c \in \mathcal{B}_{\text{tx},f}} x_{t,c}$ denotes the number of occupied channels at time t. We apply a factor $\epsilon = 0.01$ to exclude spurious noise as cause of loss. Thus, we avoid for instance losing an entire frame when just a single temporal sample was found active during the transmission.

Fig. 4.17a and Fig. 4.17b depict the throughput distribution under hindering (without hidden terminals) and hidden-terminal scenarios, respectively. First, observe that throughput is significantly reduced in comparison to the case with no hidden terminals. For example, for contiguous channel bonding in high load, mean throughput gains over single-channel transmission are reduced from $5.2\times$ to $2.6\times$ (50%).

Second, there is significant amount of periods where channel bonding results in extremely low performance, even for low occupancy values. In fact, there are scenarios in which single-channel access outperforms bonding (blue areas where $\hat{\Gamma}^* < 1$ in the figure). In particular, single channel outperforms contiguous channel bonding in 18%, 8% and 8% of the cases for low, medium and high occupancy, respectively. Even worse, single channel outperforms non-contiguous channel bonding in 38%, 14%, and 15% of cases respectively. This illustrates that external traffic patterns may lead to accentuated chances of incurring collisions, thus impairing the performance of the channel bonder.

Third, due to its more aggressive nature, non-contiguous channel bonding performs slightly worse than contiguous both in terms of average throughput and how often it is outperformed by single channel transmission. This, together with the fact that contiguous performs similarly on average to non-contiguous without hidden terminals, indicates that gains provided by non-contiguous can be relatively small considering its greater risk and more complex design.

Finally, also in contrast to Fig. 4.17a, Fig. 4.17b indicates that the decaying trend of mean normalized best-throughput with load when neighbors defer does not hold with hidden terminals. Indeed, we observe a slight increase of the normalized best-throughput as the occupancy increases as fewer bonding opportunities also diminish the risk of collisions.

Finding #12: While channel bonding outperforms single-channel transmission in most cases, it is vulnerable to hidden node interference, with non-contiguous channel bonding being impacted the most. To avoid the aforementioned worst-case throughput penalties, nodes can use RTS/CTS, implement capture (to better receive even while a hidden terminal transmission overlaps in time), or increase sensitivity, i.e., reduce the threshold for sensing channel activity to reduce the frequency of hidden terminal transmissions.

The Camp Nou: a sold out stadium

Finally, we assess if channel bonding can provide throughput gains in areas with persistently high load over all channels. Figure 4.18 shows normalized throughput under the best primary channel selection when applying contiguous and non-contiguous channel bonding. Under loads exceeding 90%, both policies have throughput close to single-channel transmission, indicating minimal remaining margin for gain. Nonetheless, in transient epochs, where the load is low, both policies exploit throughput gains. For example, in the stadium during epochs of 20-30% load, the gain over single-channel throughput is $3.69\times$ and $5.30\times$ for contiguous and non-contiguous, respectively. In fact, we observe that non-contiguous significantly outperforms contiguous in a broad regime of "non-extreme" loads given the heterogeneous activity distribution over channels. For example, at 40-50% average occupancy, contiguous channel bonding has an average throughput gain of $2.38\times$ whereas non-contiguous has $3.97\times$, a gain of 67%.

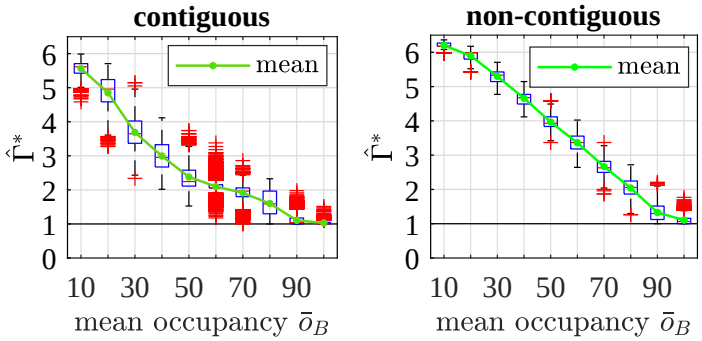


Figure 4.18: Gains for the bonding BSS assuming others defer.

Lastly, the story can turn quite negative if competing BSS's do not defer to the channel bonder, i.e., if other BSS's are hidden. In such a worst-case, all bonded transmissions with subsequent activity are considered lost. Figure 4.19 shows the results of this scenario for the stadium traces. As indicated by the blue shadowed areas, most of the periods achieve higher throughput with single-channel transmission rather than channel bonding. Indeed, single-channel transmission outperforms both channel bonding policies in at least 80% of the epochs over all the occupancy regimes.

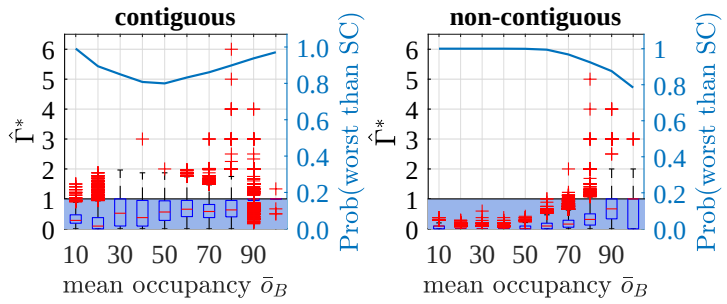


Figure 4.19: Hidden terminal scenario with the ratio of epochs that single-channel outperforms bonding shown by the blue curve.

Moreover, comparing similar loads to the other data sets (e.g., 20-30% occupancy), the stadium performs far worse with hidden terminals. For instance, we find that for 20-30% average occupancy, even with worst-case hidden terminals, the average throughput gain of contiguous channel bonding was $2.66\times$. In contrast, for the stadium, the gain is reduced to $0.59\times$, i.e., throughput is lost. This indicates that the stadium traffic and channel occupancies yield greater hidden terminal risk than urban environments, even for the same traffic load. Thus, similarly to the hindering case, the results suggest that negligible correlation is detrimental in the presence of hidden terminals.

Finding #13: Even under the stadium's extremely high average load, short durations of lower load can be exploited to yield significant throughput gains. However, the risk is high as hidden terminals could drive the throughput to levels even worse than without any channel bonding.

Chapter summary

In this chapter, we have depicted the key results on the performance of channel bonding and derived the corresponding findings. Through our CTMN model and Komondor simulations, we have identified key characteristics of channel bonding policies, including the fact that different policies per BSS may be required to maximize the metric of interest. Besides, our experimental findings of the on-the-field WACA dataset revealed the underpinning factors controlling throughput gain, from which we highlight the inter-channel correlation. We start depicting potential solutions to the joint spectrum management problem next. In particular, the next chapter treats heuristic-based approaches to reach satisfactory configurations by forsaking optimal performance.

Chapter 5

HEURISTIC-BASED POLICIES

A heuristic – derived from the Greek word meaning "to discover" – is any problem-solving method that employs a practical, flexible, *shortcut* that is not guaranteed to be optimal but is *good enough* for reaching a short-term goal. Heuristics are then used to find quickly satisfactory solutions in complex systems where finding an optimal solution is infeasible or impractical.

5.1 Heuristics to cope with complexity

As for the problem we deal with in this thesis, there are many valuable works in the literature about custom channel allocation and channel bonding solutions in wireless networks (e.g., [104, 134, 139, 144]). Table 5.1 collects some of them. These solutions rely on certain types of statistics (e.g., channel occupancy, packet reception rate, or color conflicts) to operate. We call these prefixed rule-based solutions as *heuristic* solutions in front of machine learning solutions, which we discuss in chapter §6.

While there are multiple works treating channel allocation and channel bonding separately, few of them assess the joint problem altogether in the context of WLANs. We highlight the following ones. A distributed spectrum assignment for home WLANs relying on out-of-band measurements is proposed in [63]. Continuous-time Markov networks are employed in [31] to study a centralized approach for maximizing network fairness. Besides, an algorithm for primary channel selection based on the bonding direction likelihoods was recently presented in [77]. However, such likelihoods are estimated by assuming a known number of users in each channel, which is usually not feasible in real deployments. It is worth noticing that the works mentioned above consider fully-backlogged traffic, thus missing insights on more realistic patterns with different traffic needs. Conversely, an uncertain traffic channel allocation approach was presented in [100]. Still, a centralized controller in the backend is required. Recently, we formulated DyWi (see §5.2), a decentralized, lightweight algorithm, adaptive in the sense that a new primary channel is only adopted when the BSS performance is under a given satisfaction threshold [27]. We depict DyWi in the section below.

Type	Ref.	Description
Protocol	[53]	ARAMIS is a standard-compliant, closed-loop rate adaptation solution that jointly adapts rate and bandwidth on a per-packet basis. Experiments show that ARAMIS accurately adapts to a wide variety of channel conditions with negligible overhead.
	[67]	A channel bonding protocol in which a node is allowed to start a transmission as long as there are some narrow idle channels, and it gradually increases the channel width during transmission whenever new narrow channels become available.
	[76]	The AGILE spectrum adaptation system is able to dynamically tune the channel central frequency and bandwidth of wireless links adapting to the interference and traffic conditions. The developed system can detect under-utilized spectrum fragments and adjust the occupied spectrum.
	[73]	A channel allocation algorithm based on an integer nonlinear programming model is used in a CTMN representation of the system, with the target of maximizing the throughput in channel bonding WLANs.
	[19]	ACORN is an auto-configuration framework for enterprise 802.11n WLANs. ACORN integrates user association and channel allocation functions since they are tightly coupled when wide channels are used.
	[143]	A MAC protocol which supports discontinuous channel bonding is proposed. The protocol modifies the traditional frame structure.
	[114]	A guard-band-aware channel assignment scheme reduces the number of required guard channels for a given transmission by utilizing adjacent channels.
	[67]	A channel bonding protocol with collision detection in which a node gradually increases the transmission bandwidth whenever new narrow channels are found available.

Table 5.1: Channel bonding custom protocols.

5.2 Heuristic-based primary channel selection for dynamic channel bonding

In thesis paper **#4** [27], we formulate dynamic-wise (DyWi), a decentralized, lightweight algorithm that leverages information about the sensed spectrum occupancy in the whole allocated bandwidth of a node (i.e., primary and secondary channels). Based on such occupancy, the primary channel is selected online to maximize the expected throughput. It does so by considering the activity of the target primary channel and the potential bonds that could be established with its adjacent channels. DyWi is adaptive in the sense that a new primary channel is only adopted when the BSS performance is below a given satisfaction threshold. Besides, since DyWi relies just on local information, neither neighbor messaging nor a central controller is required. This property makes DyWi suitable to be implemented in off-the-shelf APs, thus avoiding costly inter-BSS collaboration. Results in IEEE 802.11ax HD deployments show substantial improvements with respect to traditional primary channel selection, even when considering substantial delays due to channel switching.

5.2.1 Online selection of the primary channel

Let an AP belonging to BSS w operate under dynamic channel bonding and have allocated the full available bandwidth according to channel allocation C_w. Is there a way to estimate the best primary channel inside the channelization, $p_w \in C_w$? For the sake of identifying a convenient primary channel, we propose an iterative online primary selection (OPS) approach.

In essence, the AP of w periodically decides the primary channel selection, where each

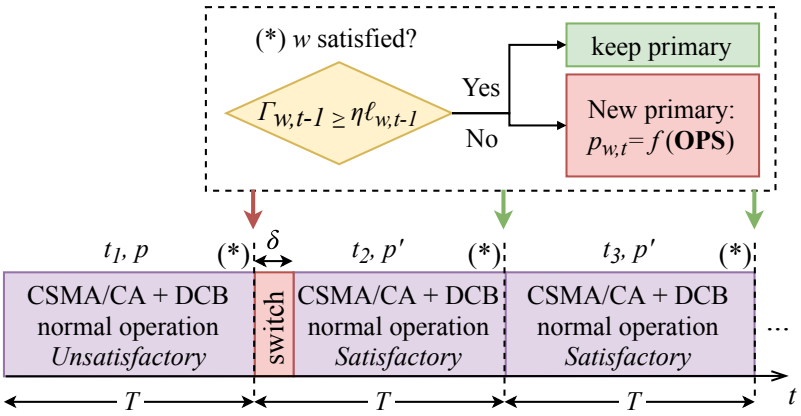


Figure 5.1: Example of the satisfaction-based online primary channel selection. Since w is not satisfied at the end of iteration t_1, the primary channel is switched from $p_{w,1} = p$ to $p_{w,2} = p'$, determined by a given OPS rule. In contrast, with the new primary, w gets satisfied in t_2 and keeps p' in t_3.

decision instant represents the beginning of an iteration. Specifically, at the beginning of a given iteration t, the AP of w measures the throughput $\Gamma_{w,t-1}$ achieved during the last iteration $t-1$ and acts according to a throughput satisfaction condition. In this case, we focus on the successful downlink traffic as the prime performance metric. However, the algorithm can be easily extended to consider other parameters, such as latency. Namely, the primary channel remains the same if w is satisfied because a sufficient share of the generated traffic has been successfully sent during the last iteration,[1] i.e., $\Gamma_{w,t-1} \geq \eta \ell_{w,t-1}$, where η is the satisfaction threshold and $\ell_{w,t-1}$ is the actual generated traffic load in that iteration. Otherwise, w will switch to a new primary channel at the cost of remaining inactive during a period δ, which is required to announce the new primary channel to the associated stations (STAs) and apply the new channel configuration. Note that the way the new primary channel is selected depends on the OPS rule. The temporal evolution of the general procedure is displayed in Fig. 5.1 through an illustrative example.

Algorithm 1 shows the pseudocode of the online primary channel selection for a generic OPS rule. Note that any OPS rule relies on the gathered data about the bandwidth occupancy in the last iteration. Formally, the empirical probability that a subset of n channels in channelization $\mathcal{C}$ was free (or idle) during the last iteration $t-1$ given a primary $p_{w,t}$ is given by

$$\mathcal{F}_{w,t-1}(\mathcal{C}, p_{w,t}, n) = \mathbb{E}_{t-1}\big[P_{\text{rx}}(c) < \text{CCA}, \forall c \in \mathcal{C}(p_{w,t}, n)\big], \tag{5.1}$$

where $P_{\text{rx}}(c)$ is the power received at *basic* channel c and $\mathcal{C}(p_{w,t}, n)$ is the set of *basic* channels used in the transmission, which is mandated by the channelization scheme $\mathcal{C}$. For instance, following the IEEE 802.11ac/ax channelization, for primary $p = 6$ and $n = 2$, the corresponding 40-MHz bonded channel is $\mathcal{C}(6, 2) = \{5, 6\}$ (channels 52 and 56 in the standard). With slight abuse of notation, $\mathbb{E}_{t-1}$ represents the expected value function at iteration $t-1$.

Three OPS rules are evaluated in this work: dynamic-random (DR), dynamic-free (DF), and dynamic-wise (DiWy or DW). In the event of an unsatisfactory iteration, DR selects a new primary channel uniformly at random, DF picks the one found most free during the last

[1] We rely just on data from the last iteration for lowering memory demands and enabling fast adaptability in dynamic environments.

Algorithm 1: Online primary selection. OPS refers to the selected online primary selection rule.

Input: $p_{w,t}$, η, OPS, $\mathcal{C}$;
iteration $t \leftarrow 0$;
while *BSS* w *active* **do**
  while *iteration* t *not finished* **do**
    CSMA/CA normal operation with dynamic channel bonding;
  end
  $t \leftarrow t + 1$;
  $\Gamma_{w,t-1} \leftarrow$ `measure_throughput`$(t-1)$;
  $\mathcal{F}_{w,t-1} \leftarrow$ `get_idleness`$(t-1)$;
  if $\Gamma_{w,t-1} < \eta \ell_{w,t-1}$ **then**
    $p_{w,t} \leftarrow$ `apply_ops`$(p_{w,t-1}, \mathcal{F}_{w,t-1}, \text{OPS}, \mathcal{C})$
    `channel_switch`$(p_{w,t}, \ \delta)$;
  end
end

iteration, and DyWi selects it based on the *forecast* throughput given the probability of bonding in every possible bandwidth. Such probabilities are estimated by periodically measuring the energy in all the secondary channels as done during the PIFS period. Apart from DR and DF, we also consider as baseline the traditional fix primary (FP) allocation, which does not change the primary channel under any circumstances, so it is not referred in Algorithm 1.

5.2.2 Dynamic-wise primary channel selection

A fundamental question arises when considering DyWi regarding the way the primary channel should be selected. Assume a scenario where a BSS w is allocated 80 MHz accounting from channel 1 to 4. At the end of iteration $t-1$, w is unsatisfied and changes its primary from $p_{w,t-1} = 2$ to $p_{w,t}$. Assume also that the probabilities of finding each of its allocated basic channels free in iteration $t-1$ was $\boldsymbol{\pi}_{w,t-1} = [0.93, 0.38, 0.85, 0.85]$, where $\boldsymbol{\pi}_{w,t-1}[c]$ is the probability of finding the *basic* channel c free. Then, two main options may be contemplated as best choice for selecting $p_{w,t}$ to maximize the throughput $\Gamma_{w,t}$ of the upcoming iteration: *i*) to pick the primary with highest probability to be free (i.e., $p_{w,t} = 1$ in this case), or *ii*) to pick the primary providing the highest potential average data rate considering both its probability to be free, as well as the probability of the channels nearby (e.g., $p_{w,t} \in \{3, 4\}$). Notice that there is not an evident choice to provide beforehand since multiple parameters apart from $\boldsymbol{\pi}_{w,t-1}$ will impact the performance. For example, dynamic network activity affects to the SINR, MCS, or packet error rate.

We tackle this point at issue by proposing a lightweight maximization problem for the forecast data rate of BSS w at iteration t. The idea behind this approach is that maximizing the successful data rate should maximize the throughput. Thus, despite being a sub-optimal formulation for maximizing the throughput,[2] selecting the primary channel according to the forecast average data rate is a convenient heuristic, as shown later. In particular, the average data rate is estimated by the probability of transmitting at each possible bandwidth. Hence, we

[2]One should consider different complex parameters such as the buffer status and environment dynamics to define an optimal formulation.

can formulate the problem as

$$\begin{aligned} &\underset{p_{w,t}\neq p_{w,t-1}}{\operatorname{argmax}}\ \hat{r}_{w,t}(p_{w,t}), \text{with} \\ &\hat{r}_{w,t}(p_{w,t}) = \sum_{n\in\mathcal{N}} \mathbb{P}_{w,t}\big(\mathcal{F}_{w,t-1}, p_{w,t}, n\big) r_w\big(n\big), \end{aligned} \tag{5.2}$$

where $\hat{r}_{w,t}(p_{w,t})$ is the forecast average data rate by BSS w at iteration t for new primary $p_{w,t}$, and n is the number of bonded channels, which should be permitted by the channelization scheme. For instance, in the IEEE 802.11ax amendment, $n \in \mathcal{N} = \{1, 2, 4, 8\}$, for 20 to 160-MHz bandwidths. $\mathbb{P}_{w,t}(\mathcal{F}_{w,t-1}, p_{w,t}, n)$ is the probability that w transmits in n contiguous channels in the starting iteration given $p_{w,t}$ is selected, and $r_w(n)$ is the data rate given the bandwidth nb. Note that r also depends on the MCS index, which will vary according to the signal-to-noise ratio (SNR) at the STA.

In order to estimate the probability of transmitting in each possible combination of channels, we rely on the empirical probability $\mathcal{F}_{w,t-1}(\mathcal{C}, p_{w,t}, n)$, which was updated during the CSMA/CA operation in iteration $t-1$. Since dynamic channel bonding is implemented, the largest available bandwidth is always picked in every transmission. Hence, the probability of transmitting in a certain bandwidth is contingent on the probability of transmitting in higher bandwidths. Specifically,

$$\mathbb{P}_{w,t}(\mathcal{F}_{w,t-1}, p_{w,t}, n) = \mathcal{F}_{w,t-1}(\mathcal{C}, p_{w,t}, n) - \sum_{n'\in\{\mathcal{N}|n'>n\}\mathbb{P}_{w,t}(\mathcal{F}_{w,t-1}, p_{w,t}, n')}, \tag{5.3}$$

where the probability of transmitting in wider bandwidths is subtracted due to the constraint $n' \in \{\mathcal{N}|n' > n\}$. For instance, for bands UNII-1 and UNII-2 altogether (i.e., $\mathcal{C}_{\text{20-MHz}} = \{1, ..., 8\}$), we define $\mathbb{P}(\mathcal{F}, p, 8) = \mathcal{F}(\mathcal{C}, p, 8)$ for $n = 8$ (i.e., 160-MHz). Similarly, on the other end, $\mathbb{P}(\mathcal{F}, p, 1) = \mathcal{F}(\mathcal{C}, p, 1) - \mathbb{P}(\mathcal{F}, p, 2) - \mathbb{P}(\mathcal{F}, p, 4) - \mathbb{P}(\mathcal{F}, p, 8)$ for $n = 1$ (i.e., 20-MHz).

As for the complexity of the presented OPS rules, note that they are all computational lightweight, especially DR, since it does not keep track of any data. Despite DF's complexity increases with the number of *basic* channels, $\mathcal{F}_{\text{DF}}\big(|\mathcal{C}_{\text{20-MHz}}|\big)$, its complexity remains also low. DyWi's complexity, however, is bounded by $\mathcal{F}_{\text{DW}}\big(|\mathcal{C}_{\text{20-MHz}}|(\log_2|\mathcal{C}_{\text{20-MHz}}|+1)^2\big)$ for the IEEE 802.11ax channelization. Nonetheless, DyWi is completely tractable in operation time by off-the-shelf network cards since the number of possible bonds in the 5-GHz band is still small.

5.2.3 DyWi performance

Regarding the simulation deployments, we contemplate dense 40x40 m^2 scenarios following the characteristics of the deployments shown in §4.2.1, where BSS A implements dynamic channel bonding and the rest of BSS's implement single-channel or dynamic channel bonding with the same probability 1/2. The simulation time of each scenario is $T_{\text{obs}} = 25$ seconds. As for the configuration of the online algorithms, we set the iteration time $T = 1$ s and threshold for the satisfaction $\eta = 0.9$. Note that we consider a value of η smaller than 1 to provide stability to the algorithm.

Figure 5.2 shows the average throughput $\bar{\Gamma}_{\text{A}}$. We observe that DyWi clearly outperforms FP and DR, even when considering a huge adaptation cost delay $\delta = 100$ ms (i.e., an important

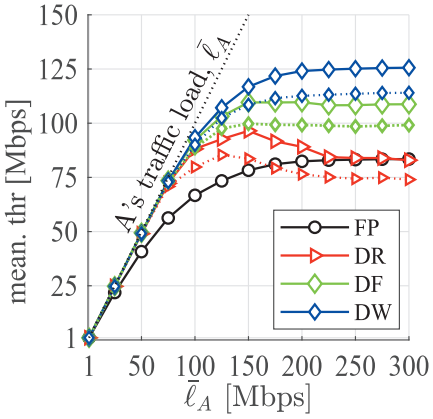


Figure 5.2: Mean throughput of BSS A for different OPS rules: fixed primary (FP), dynamic random (DR), dynamic-free (DF), DyWi (DW). Continuous lines assume no switching cost while dashed lines correspond to 100 ms delay per switch.

10% penalty with respect to the iteration duration $T = 1$ s). While DR may be counterproductive for high loads, both DF and DyWi keep a constant performance after saturating. Such a throughput reduction for DR is caused by the fact that the larger $\bar{\ell}_A$, the harder to remain satisfied, which leads to more frequent channel switching. Then, critically for DR, the random selection of the primary leads to picking each channel with the same probability. Accordingly, the average throughput converges to FP's because, on average (for all the scenarios), the primary channels are equiprobable selected whenever the satisfaction condition is not accomplished. In summary, results show significant improvements with respect to traditional allocation, even under high switching adaptation costs.

To assess the temporal evolution of the different algorithms, Fig. 5.3 plots the cumulative distribution function (CDF) of the number of iterations k required to reach a satisfactory primary channel for moderate, medium, and high traffic loads. We observe that all the OPS rules take very few iterations to maximize performance, meaning that the heuristic can usually rely on its performance estimates. As expected, the lower the load, the higher the value of CDF(k) for any k. Note that there are few unusual scenarios where the CDF varies for FP. Those are the cases where the load is almost adequate from the very beginning since the traffic generation's stochastic nature makes the throughput vary around the satisfaction threshold as the simulation progresses.

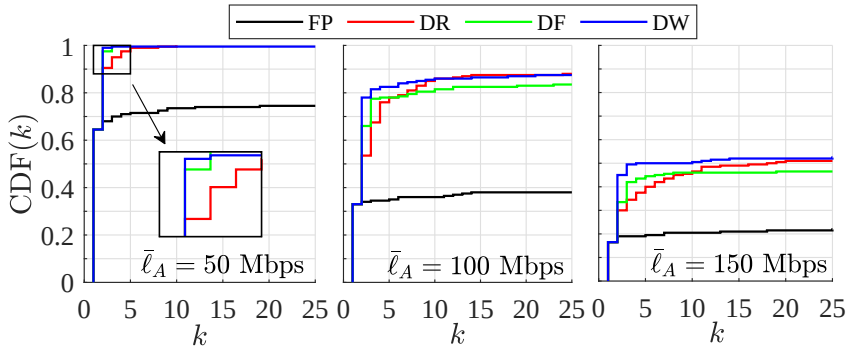


Figure 5.3: CDF of the number of iterations k required to reach satisfaction.

Finding #14: Harnessing heuristics from occupancy of secondary channels in channel bonding BSS's is a useful approach to select the primary that maximizes individual throughput and reaches fast satisfaction under *steady* environments.

Notice that DyWi and the rest of OPS rules have been assessed under steady environments, where only one BSS was able to change its configuration. So, the used heuristics are reliable in the sense that they are not outdated and can provide meaningful hints. For multi-agent setups where multiple BSS's may change its configuration, such heuristics are no longer trustful. So, we leave as future work the design of a CSMA/CA-like mechanism where DyWi can be paused under the observation of performance inconsistencies and resume after a random period. This way, agents would generate a more stable environment at the cost of sacrificing short-term performance maximization.

Chapter summary

This chapter overviewed heuristic-based approaches related to the spectrum management problem and presented the DyWi algorithm for online primary channel selection. Relying upon heuristics allows finding rapid *good enough* configurations by forsaking optimal performance. In the next chapter, we question whether machine learning can aid further in the problem.

Chapter 6

MODEL-FREE REINFORCEMENT LEARNING

This chapter discusses various model-free reinforcement learning approaches to tackle the problem of channel allocation in distributed, dynamic channel bonding WLANs under varying traffic load demands. The focus is on the adaptability or learning speed of the multi-agent-empowered BSS's to achieve adequate performance. We motivate why stateless RL and particularly multi-armed bandits are the most suitable approach for the problem. In contrast to most of the works on RL for spectrum management, we evaluate different MAB exploration techniques by simulating realistic physical and MAC operation in two different settings: a self-contained toy scenario with known optimal configurations and generalized random scenarios to benchmark the MABs under different node densities and traffic load variations.

This chapter has **no thesis paper of reference**. At the time of depositing this dissertation, the contents in this chapter are original and have not been submitted for publication.

6.1 A change of paradigm towards reinforcement learning

As discussed in the chapter before, heuristics are low-complex solutions that work well in steady scenarios. However, in highly dynamic scenarios, its performance is severely undermined since they rely on statistics only from recent observations that tend to be outdated when applied. So, heuristics are limited to the meaningfulness of the gathered observations, and such meaningfulness is hard to keep when the environment changes. Then, in uncoordinated, high-density deployments, where multiple BSS's operating under different traffic regimes adapt their spectrum configurations at their own, estimating and designing accurate hand-crafted spectrum management techniques is unfeasible.

Are there alternatives to tackle this dynamism and high complexity imposed by the joint problem of spectrum allocation and channel bonding? Indeed, ML solutions have been widely studied in wireless networks for a plethora of problems. In fact, we find a clear trend towards ML-aided solutions, which are expected to cope with the ever-increasing complexity of new applications and services like Vehicle to Everything (V2X) communications, Machine Type Communications (mMTC), or Ultra-Reliable Low-Latency Communication (uRLLC). The capability of ML for automatically learning (and adapting) to (un)seen situations can cope with heterogeneous scenarios, including different mobility, connectivity, and performance requirements. That is, a change of paradigm towards solutions aided with artificial intelligence and ML

is getting increasingly common in the context of present and future wireless networks [93,140]. Different surveys on the matter have been published in the last years, from which we highlight [16,37,70,79,124,142] and references therein.

As for the joint problem of channel allocation and channel bonding, multi-channel next-generation WLANs' complex dynamics will make it necessary to rely on some sort of learning to overcome hand-crafted alternatives. That is, relying on prior hard-coded rules and protocols seems not an option in the quest to support the ever-increasing demands of next-generation services and applications in terms of very high throughputs and very low delays. Further, undertaking new protocol designs that could consider all the dynamism and generalize to a complete set of scenarios is merely hopeless.

6.1.1 What is reinforcement learning?

RL refers to sequentially training ML models based on decisions taken towards maximizing a cumulative reward. RL relies on agents that ought to learn how to achieve a goal in an uncertain, potentially complex environment. The agent employs trial and error to come up with a solution to the problem. To get the agent to do what the programmer (or administrator) wants, the agent gets either rewards (e.g., throughput) or penalties (e.g., delay) for the actions it performs (e.g., switch to another primary channel). Its goal is to maximize the total reward (e.g., mean throughput during an observation time). Although the designer sets the reward function and policy, i.e., the rules of the game (e.g., the more throughput, the better), he/she gives the model no hints or suggestions for how to solve the game. It is up to the strategy or exploration algorithm to figure out how to perform the task to maximize the reward, normally starting from random trials and finishing with sophisticated tactics.

More formally, RL models an agent that tries to learn how to behave in an environment by performing actions and observing the collected rewards. At a given iteration t, action a results in a reward observation drawn from a reward distribution $r_t(a) \sim \theta_a$. In the context of WLANs, an agent could be installed into an AP that tries to maximize a certain BSS performance metric by testing different configuration settings and adapting to the collected reward observations.

Model-based vs. model-free

There are two main RL approaches: *i*) **model-based approach** to first estimate the system model from observations and then apply dynamic programming or a computationally efficient heuristic policy such as myopic or Whittle index [90] policies, and *ii*) **model-free approach** to learn the policy directly through interactions with the system without estimating the system model. The model-based approach is less favored for our problem since the user's limited observation capability may result in a bad (and most times outdated) system model estimation.

In particular, since the system interactions in the spectrum management problem are complex and generated by multiple actors (nodes), we must undoubtedly rely on model-free approaches. That is, when getting rid of strong assumptions like Markovian channels (e.g., [48, 49, 99]), it would not make sense to have an agent trying to infer what is the model behind the interactions perceived at the varying power levels detected at each of the channels. Moreover, even in the case where such interference model was stationary (and let us also assume simple enough to be accurately captured through a model), in the very moment that the agent-empowered AP initiated a transmission, the contention and interference generated to sur-

rounding nodes may completely change the learned model consequently. Thus, one can easily derive that having a giant model considering all the possibilities, or a specific models per each configuration, is not viable.

6.1.2 Why RL and not other ML frameworks?

In plain words, RL seems to fit more naturally to the spectrum management problem than the other two principal ML alternatives: supervised learning (SL) and unsupervised learning (UL). We reason why below.

Supervised learning (SL) is the ML task of learning a function that maps an input x to an output y based on example input-output pairs (x_i, y_i). It infers a function from labeled training data consisting of a set of training examples. An SL algorithm then analyzes the training data and produces an inferred function, which can be used for mapping new examples (i.e., generalizing). One way to use SL in our problem would be to try to learn the general and *true* WLAN behavior, function $f(x) = y$, through an estimate $h_\theta(x) = \hat{y}$, where $h_\theta(x)$ is the learned function. Then, once the hypothesis $h_\theta(x)$ is learned – i.e., the error is sufficiently low with respect to $f(x)$ – one could try to infer the input (or configuration) $\hat{x}^*$ that maximizes the performance $\hat{y}^*$, hoping that $\hat{x}^*$ would also result in the actual y^*.

The main three drawbacks of SL for our problem are *i*) designing the model to fit, which should consist of a vast number of attributes (e.g., nodes' locations, configuration parameters, performance metrics, etc.), *ii*) it needs a lot of data to learn accurate functions in complex systems (e.g., multi-parameter non-linear models), so our agents would not have that amount of time to do so,[1] and *iii*) SL is thought to generalize to unknown scenarios, but this is not the focus of our problem because we want the agent to adapt to potentially infinite different worlds from scratch. The only way to overcome such a task with SL would be to measure the performance of numerous settings in a lot of scenarios (primarily offline), then try to infer some general behavior from the true and unknown function $y = f(x)$, where input x is the representation of the action setting and the environment around the agent, and finally predicting the performance $\hat{y} = h_\theta(x')$ for unlabeled (unknown) input x'. The significant issues are that the input domain of x is multi-dimensional (with even categorical dimensions) and tends to infinity. The function f representing WLANs' *behavior* is so complex that learning an accurate estimate h_θ for realistic deployments is simply inconceivable.

Moreover, there is another issue when trying to replicate Wi-Fi through (deep) supervised learning: the problem of estimating a function is different from the problem of maximizing an unknown function. That is, even though we could get a good estimator $h_\theta(x)$ of the *true* function, the optimal of such estimated function may be completely different than the *true* optimal of f. In other words, our oracle may work well for most of the inputs x but fail for the input x^* maximizing $f(x)$. To showcase such an issue, let us introduce the example illustrated in Figure 6.1, where a unidimensional input $x \in [0, 20]$ produces an output y. The *true* and unknown function f is plotted in blue. The pairs (x, y) that have been sampled to later feed the ML model are highlighted in yellow. Finally, the learned function h_θ is plotted in red. We observe that while both functions (h and f) are really similar (leading to a small mean prediction error), we fail at inferring the argmax by just looking at h_θ (around $x^* = 2$, where

[1] We state that generating a dataset rich enough to generalize is unfeasible since data samples should have large multi-dimensional attributes to be meaningful. So, for instance, simply by moving one STA one meter away, a new scenario (or data sample) would be generated.

the *true* argmax is around $x^* = 12$). Again, it is harder to generalize to an unknown maximum than to approximate to a function considering all the domain.

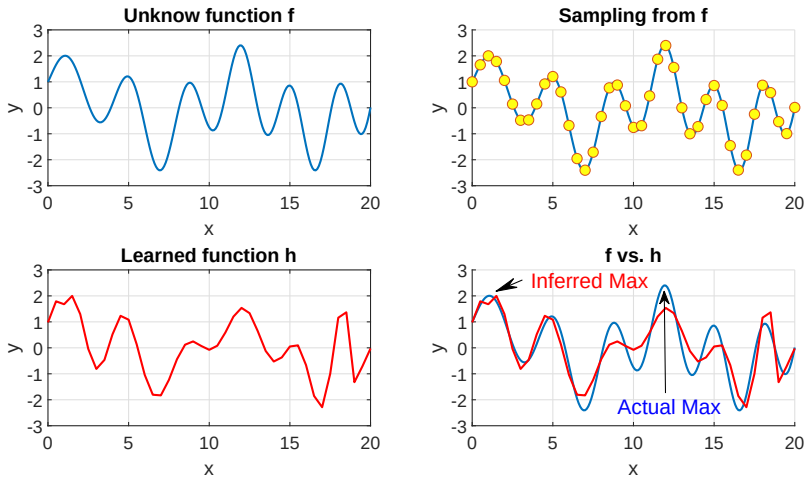


Figure 6.1: Motivation example on the argmax problem.

In **unsupervised learning** (UL), there are only inputs x and no corresponding output variables. That is, there is no y. UL's goal is to model the underlying structure or distribution in the observed data to learn more about it. It is called unsupervised because, unlike SL there are no correct answers since samples are unlabeled. Algorithms are left on their own to discover potentially meaningful structures in the data. UL problems can be further grouped into clustering (to discover the inherent groupings in the data) and association (to discover rules that describe large portions of your data) problems.

Similarly to SL, the underlying structure of the observed data in the problem at issue is expected to be so complex that any attempt to model it through UL will be most likely fruitless. Hence, we believe that following an RL black-box approach is preferable: try to adapt from scratch to whatever the system's observations are, no matter its intrinsic nature.

6.2 Mapping the problem to RL

This section depicts the modifiable attributes, actions, system statuses, and states composing the learning framework for the primary channel and maximum bandwidth allocation in dynamic channel bonding WLANs. Notice that by defining the primary channel and maximum bandwidth, we actually determine the secondary channels since we follow the IEEE 802.11ac/ax channelization $\mathcal{C}$. We then formally formulate the problem to solve. Finally, we discuss the alternatives for learning architectures and RL models.

6.2.1 System model

We consider a static WLAN of W potentially overlapping BSS's $\{w_1, w_2, ..., w_W\}$, each composed by one AP and one or multiple STAs. The TMB path-loss model is assumed [10] and spatially distributed scenarios are captured. Namely, we cover scenarios where the BSS's may

not all be inside the carrier sense range of each other. Therefore, the typical phenomena of home Wi-Fi networks like flow in the middle and hidden/exposed nodes are captured as in [26–28].

As for the channelization, we consider $|C|$ 20-MHz channels (or basic channels), e.g., $|C| = 8$ from channel 36 to 64 following the IEEE 802.11ac/ax standards. Accordingly, a transmitter may bond up to $|C| \times 20$ MHz (e.g., 160 MHz if $|C| = 8$). Note that not any bonding combination is permitted by the IEEE 802.11ac/ax ammendments. For instance, 60 MHz transmission (3 basic channels) are not permitted [5,7]. We assume all the BSSs' have dynamic channel bonding capabilities to bond up to $|C|$ channels. Nevertheless, as explained later in section 6.2.2, the maximum number of channels to be aggregated is limited by the maximum bandwidth attribute. For instance, we can easily restrict a BSS to single-channel transmissions by appropriately setting such attributes. The adaptive RTS/CTS mechanism introduced in the IEEE 802.11ac standard [5] for dynamic bandwidth is considered.

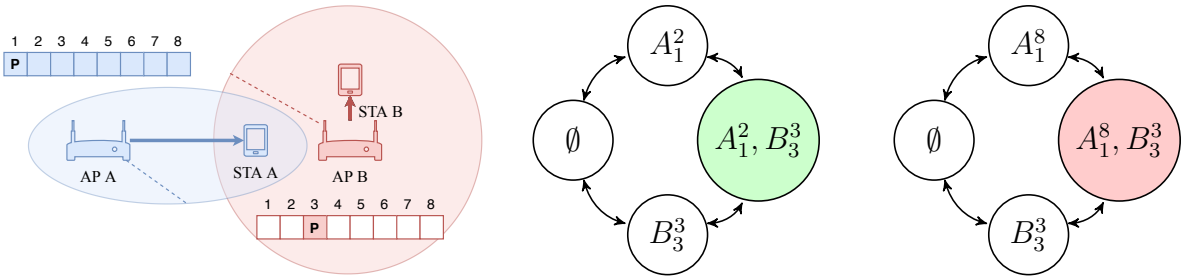
Data packets generated by a BSS w follow a Poisson process with a mean duration between packets given by $1/\ell_w$, where ℓ_w is the mean load in packets per second. Unlike much of the works in the literature, we consider that the mean traffic load of every BSS may vary over time, thus introducing mid/long-term dynamism to the problem.

6.2.2 Attributes and actions

While it is clear that the primary channel is critical to the BSS performance since it is where the backoff procedure runs, the maximum bandwidth is of significant relevance as well. Indeed, once the backoff expires, the channel bonding policy and maximum allowed bandwidth will determine the basic channels to transmit in. For instance, in the example in Figure 2.2c, if the node was allocated 40 MHz (channels 1 and 2) rather than 80 MHz (channels 1 to 4), it would bond the first 2 channels at the end of the second backoff, rather than 4. Notice that, as stated by findings #2 and #3, limiting the bandwidth, or being conservative in general, may be much appropriate to BSS's in order to reduce adverse effects like unfavorable contention or hidden nodes. Next, we motivate why the set of possible maximum bandwidth values b has a profound impact on the BSS performance. So, the more flexible b, i.e., the more values it can get, the higher the potential performance.

The role of the maximum bandwidth

To illustrate why not restricting the maximum bandwidth can be counterproductive, let us consider *toy scenario IV* (Figure 6.2a) consisting of BSS's A and B facing a potential hidden node situation. Notice that STA A is close to AP B, so STA A experiences disturbing interference from AP B. Likewise, AP B is far enough from AP A to avoid contention whenever AP A bonds channel 3 (e.g., when using 80 or 160 MHz). Beyond contention, there is a potential hidden node issue for STA A. If the power of interest (received from AP A) is not sufficiently high with respect to the interference power (received from AP B and the background noise), STA A would not be able to decode any packets from AP A, thus leading to detrimental throughput. In particular, given that we assume a fixed transmission power, STA A would need AP A to transmit in 20 or 40 MHz at maximum to receive enough power of interest to decode the packets. So, in this example, AP A should not transmit in 80 or 160 MHz whenever AP B is transmitting, even when AP A finds its full allocated spectrum free at the backoff termination.



(a) 2-BSS's toy scenario deployment. (b) Restricted: $b_{\mathrm{A}} = 2$. (c) Unrestricted: $b_{\mathrm{A}} = 8$.

Figure 6.2: *Toy scenario IV*: effect of limiting the maximum bandwidth. BSS B (always using single-channel) interferes with STA A, which perceives: (b) negligible interference since the power of interest from AP A is higher enough than the interference because AP A concentrates the transmission power into 40 MHz, and (c) critical interference due to the lower power reception from AP A at 160 MHz.

With this simple example, we corroborate that, while channel bonding is necessary to improve satisfaction under demanding traffic loads, not limiting the maximum bandwidth may also raise drawbacks. Such a limitation should be adapted to each scenario given the heterogeneity and chaotic nature of Wi-Fi deployments.

Action space

Given the relevance of the primary channel and the maximum bandwidth, we consider below two configurable attributes each agent-empowered BSS's can modify during the learning process:

- **Primary channel p_w**: primary channel where the backoff procedure is executed, $p \in C_w$, where C_w is the channel allocation of BSS w.
- **Maximum bandwidth b**: maximum bandwidth (in number of basic channels), $b \in \beta = \{1, 2, 4, ..., |C_w|\}$. Recall that, with dynamic channel bonding, the transmitter can adapt to the sensed spectrum on a per-frame basis. So, the bandwidth limitation just sets an upper bound on the number of basic channels to bond.

We therefore encounter an action space $\mathcal{A}_w$ per BSS w of size

$$O(\mathcal{A}_w) = O(C_w) \times O(\beta_w), \tag{6.1}$$

where $O(\cdot)$ represents the number of elements or cardinality of a set, and every action, or spectrum configuration, $a = (p_w, b_w) \in \mathcal{A}_w$ is a pair of the primary channel $p_w \in C_w$ and maximum bandwidth $b_w \in \beta_w$ attributes. Should we consider a central single agent managing $W' \leq W$ BSS's, the action space increases exponentially to

$$O(\mathcal{A}) = \prod_{w=1}^{W'} O(C_w) \times O(\beta_w). \tag{6.2}$$

Definition 6.2.1 (spectrum configuration)**.** The spectrum configuration of a BSS w is defined as the pair primary channel p_w and maximum bandwidth b_w.

If we assume that BSS's have the same attribute spaces C and $\beta, \forall w$, then we clearly observe the exponential growth of the whole action space $O(\mathcal{A}) = \Big(O(C) \times O(\beta)\Big)^{W'}$.

6.2.3 Statuses, observations and states

Before discussing what is an *state*, which its definition highly depends on the RL framework in use, let us coin a new concept called system *status* to depict how the whole WLAN *looks like* at time t.

Definition 6.2.2 (system status)**.** The system status $\mathcal{H}_t$ is the minimum *information* we need to describe the entire WLAN in a given time instant t.

The system status $\mathcal{H}_t$ of a static WLAN deployment[2] at time t is given by three key parameters of each of the W' agent-empowered BSS in the WLAN like if we were taking a screenshot of the system: the primary channel $p_{w,t}$, the maximum bandwidth $b_{w,t}$, and the quantized mean load $\ell_{w,t}$. Formally, a status is defined by the set

$$\mathcal{H}_t = \{(p_{1,t}, b_{1,t}, \ell_{1,t}), (p_{2,t}, b_{2,t}, \ell_{2,t}), ..., (p_{W',t}, b_{W',t}, \ell_{W',t})\}. \tag{6.3}$$

First, notice that we only consider the attributes of the agent-empowered BSS's since the rest of BSS's are not allowed to modify its configuration; thus, they do not convey any extra information for the status. For instance, let us consider a deployment with just a couple of BSS's, A and B. Assume A is agent-empowered, whereas B is not, so $W' = 1$ and $W = 2$. The system information relies just on the configuration of A since B will never change its own. So, the number of system statuses is just the number of possible configurations A can take.

Second, we rely on a quantization $\mathcal{L}$ of the mean traffic load ℓ to discretize its continuous domain. Further, notice that the agent cannot modify the load ℓ by any means since it is application-dependent. Naturally, the throughput performance depends on the traffic load, the AP configuration, and the rest of BSS's configurations and loads.

Definition 6.2.3 (world)**.** The world (or environment) of a particular BSS w is defined as the combination of configurations and loads of each of the rest BSS's.

From the point of view of a particular BSS w, the combination of current spectrum configurations and loads of the rest BSS's conform its *world* or environment, which is determined as

$$\mathcal{I}_{w,t} = \{p_{w',t}, b_{w',t}, \ell_{w',t} | w' \neq w\}. \tag{6.4}$$

Finally, the amount of possible statuses a certain deployment may have is given by the possible values each of the status parameters can take,

$$O(\mathcal{H}) = \prod_{w=1}^{W'} O(p_w) O(b_w) O(\ell_w), \tag{6.5}$$

which raises again exponentially with the number of agents and size of the attributes' spaces. Assuming all the agent-empowered BSS's have the same capabilities,

$$O(\mathcal{H}) = \Big(O(p) O(b) O(\ell)\Big)^{W'}. \tag{6.6}$$

[2]Static refers to the fixed locations of all the nodes in the WLANs, including APs and STAs.

Naturally, most of the time, the agents cannot know the system's status in a real-time fashion. However, they could rely on partial and delayed observations to then infer the most accurate state through some discretization function. Besides, since observations are architecture-dependent – given they may be limited to local sensing capabilities or, on the contrary, be shared via a central controller – the states' definition is also strictly dependent on the RL model under consideration.

6.2.4 Problem definition

In general terms, our goal is to improve the BSS's performance by maximizing their throughput and minimizing their delay. The key then is to let the BSS's find (learn) the best actions (p, b), where *best* depends on the problem formulation itself. For instance, we could assess a max-min approach to maximize the throughput satisfaction experienced by the less favored BSS. Another common aim in network management is to boost the fairness of some performance metric among contending BSS's in the WLAN. This thesis does not stick into one problem formulation, but we instead evaluate different performance metrics, including individual and aggregated ones.

In any case, we must define the individual performance metrics of each BSS to assess whether such metrics tend to improve (i.e., the learning is working) or not. We focus on the throughput satisfaction ξ and the delay d. First, the throughput satisfaction of a BSS w at iteration t is simply defined as the ratio of throughput $\Gamma_{w,t}$ to generated traffic load $\ell_{w,t}$,

$$\xi_{w,t} = \frac{\Gamma_{w,t}}{\ell_{w,t}}. \tag{6.7}$$

So, $\xi_{w,t} \in [0, 1], \forall w, t$. The throughput satisfaction is a commonly used parameter to indicate the ratio of packets acknowledged to the number of packets generated by a BSS in a given time window. Thus, a satisfaction value $\xi = 1$ indicates that all the traffic has been successfully received at the destination.

Second, the delay is defined as the mean data packet delay, simply computed as the time difference between a data packet being generated and acknowledged. Unfortunately, the delay is not bounded, which is convenient for guaranteeing most RL algorithms' convergence. Thus, we propose a bounded delay ratio defined as

$$\delta_{w,t} = \frac{d^*}{d_{w,t}}, \tag{6.8}$$

where $d_{w,t}$ is the mean delay during iteration t, and d^* is the known minimum achievable delay experienced when transmitting a single data packet at maximum MCS 11. So, $\delta_{w,t} < 1, \forall w, t$ holds because every BSS will always experience a delay higher than d^* simply due to the MAC interframes (SIFS, DIFS) and control packets exchange (RTS, CTS, ACK).

Once the two main performance metrics have been defined, we may formulate the reward function R, the philosopher stone of any RL algorithm. In this work, we assess two definitions of R: only throughput based,

$$R(\xi) = \xi, \tag{6.9}$$

and a throughput-delay mix,

$$R(\xi, \delta) = \xi \cdot \delta, \tag{6.10}$$

respectively. Notice that we propose the latter's product operation given the high correlation between throughput satisfaction and delay. Generally, a high ξ leads to low δ and vice versa. However, for the cases where $\xi = 1$, there may exist configurations providing different delay values. Thus, we anticipate it is worth explicitly considering the delay as well.

Regardless of the reward function R, we aim at maximizing the cumulative reward *as soon as possible*,

$$G = \sum_{t=1}^{T} r_t, \tag{6.11}$$

where T is the number of iterations we have to execute the learning, and r_t is the reward at iteration t. Let us emphasize the complexity of the problem by noting that r_t depends on the reward definition R, on the action a_t selected at t, and also on the world $\mathcal{I}_{w,t}$.

6.3 RL framework for the spectrum management problem

Now that we have depicted the RL problem, we may discuss the pros and cons of different model-free RL approaches in terms of the learning architecture (i.e., where the learning is carried) and RL models (i.e., what algorithms run the learning).

6.3.1 Learning architecture

In broad terms, there are two main types of learning architectures: centralized and distributed. In a **centralized architecture**, all the agents must communicate to a central controller and follow its instructions. In other words, agents become dummy in the sense that they only provide data about their performance and sensing metrics to the controller. It is the central controller responsible for aggregating this data from the dummy agents and process it to generate the new actions each edge agent must undertake. In a way, there would be just a single (super agent) at the central controller conducting the action selection of each BSS w, leading to an aggregated action space $\mathcal{A}$ including all the possible actions of each BSS w, i.e., $\mathcal{A} = \{\mathcal{A}_w\}, \forall w$. So, a super action a would be the combination of all the spectrum configurations of each of the W' controlled BSS's, i.e., $a = \{(p_1, b_1), (p_2, b_2), \ldots, (p_{W'}, b_{W'})\}$.

The centralized architecture has the advantage of counting with a holistic vision of the WLAN, thus having the capability to identify cause-effect relations between the configurations of the BSS's and act explicitly to aid those performing poorly. However, gathering such aggregate data entails non-negligible communication delays from the dummy agents to the central controller, making the data potentially outdated. Besides, relying on a single entity raises the risk of failure in the whole WLAN. Moreover, the aggregated action space grows exponentially with the number of BSS's (6.2), hindering RL algorithms' rapid execution.

In contrast to the centralized architecture, **decentralized** and distributed architectures do not rely on a single central controller. Instead, multiple controllers are used in the distributed design, whereas no controllers are present in decentralized architectures. Since a decentralized architecture can be thought as a composite of different centralized architectures, let us focus on the distributed one to compare the centralized to a radically different architecture. In a distributed architecture, agents operate by themselves without any external controller. That is, agents make decisions on their own, relying on their local data, or via some peer-to-peer communication with neighboring agents in the case of collaborative settings.

Distributed architectures are less likely to fail than centralized ones because the decision-making does not rely on a single entity but all the BSS's in the WLAN. Besides, the processing and communication delay to the central controller is no longer an issue since edge agents count on real-time metrics to conduct their RL algorithms. This is key for efficient and responsive spectrum management in WLANs. As for the drawbacks, decentralized architectures may provide inconsistent performance when not properly organized, leading to sub-optimal configurations. Further, fairness issues become a challenge when there is no direct way to cope with low performing BSS's.

Finally, a hybrid combination of centralized, long-term learning (e.g., big data) and quick, short-term learning in the edge is a topic that deserves to be studied. In [129], we propose a realization of the ITU-T ML-aware architecture for IEEE 802.11 WLANs. Through the definition of ML components and management functions, the ITU-T architecture provides the necessary flexibility for fulfilling different use case requirements, ranging from centralized solutions to decentralized approaches.

6.3.2 RL models

Different ML solutions for the spectrum management problem, especially in RL, have been proposed in recent years. However, certain assumptions these papers have in common hinder their accurate evaluation. For instance, most papers consider synchronous time slots (e.g., [45,47,102,106,122,145]), a strong assumption that does not hold at all in CSMA/CA WLANs. Also, some papers define a binary reward, where actions are simply good or bad (e.g., [47, 49, 145]). This, while easing the RL framework, is not convenient for rewards taking into account continuous-valued performance metrics like throughput or delay. Further, most of the papers consider simple traffic patterns or even fully backlogged regimes (e.g., [74, 101, 133]), thus overlooking the effects of dynamic traffic loads. Also related to the dynamism assessment, most of the papers do not consider packet aggregation, which is a critical feature affecting the throughput and delay. To ease their analysis, some papers even consider conflict graphs rather than actual carrier sense areas (e.g., [101, 106]), thus disregarding spatial distribution effects. Last but not least, a majority of the papers provide custom RL solutions that the reader may find complex to follow. Indeed, in most of cases, the design of states, actions, and rewards is not justified but simply formulated. This leaves short room to extrapolate such solutions to slightly different problems, making it harder to reproduce the presented results.

From the set of model-free RL approaches, we highlight MABs, Temporal Difference (TD) methods like Q-learning, and deep reinforcement learning (DRL).[3] We discuss them below in order of increasing complexity and motivate why we envision MABs as the best choice.

Stateless RL variations

The most common formulations of RL rely on states, which are mapped through a policy π to an action a, i.e., $\pi(s) = a$. However, stateless approaches do also exist. A key benefit of the stateless RL is precisely the absence of states, which eases its design by relying only on action-reward pairs. In this regard, the most representative stateless RL model for non-episodic problems is the MAB formulation. However, stateless variations of TD methods like stateless

[3]We discard Monte Carlo methods since they require episodes to eventually terminate in order to work, which contrasts our vision of WLANs in continuous on-line adaptation in a so-called infinite episode.

Q-learning have also been studied in the literature. The MAB is a problem in which a fixed limited set of actions must be selected between competing (alternative) choices to maximize expected gain. Each choice's properties are only partially known at the time of selection and may become better understood as time passes or by allocating resources to the choice [117]. MABs are a classic RL problem that exemplifies the exploration-exploitation trade-off dilemma.

Various works in the literature on spectrum management use MABs. For instance, the survey in [87] deals with channel allocation for spectrum scheduling based on conflict graphs. A channel selection based on UCB1 is studied in [58], where edge agents distributively select the channel to transmit in a slotted manner. Multi-channel spectrum access through various types of MABs is assessed in [99]. Authors in [49] follow a particular approach to deal with opportunistic spectrum access via restless MABs, where a single user must learn the environment. A recent custom dynamic channel allocation is applied in [119] with chaotic laser theory. Apart from MABs, other works on wireless communications rely on a stateless Q-learning variation. For instance, an spectrum allocation in cognitive radios through distributed stateless Q-learning, where each agent has its own Q-table, is studied in [11]. LTE/Wi-Fi coexistence through time-domain access aided with stateless Q-learning on a single-agent is studied in [45], where actions are defined as the probability of occupying a channel. Finally, authors in [130] rely on decentralized, stateless Q-learning to improve spatial reuse in dense wireless networks.

Temporal Difference

TD learning is a combination of Monte Carlo ideas and dynamic programming (DP) ideas [117]. Like Monte Carlo methods, TD methods can learn directly from raw experience without a model of the environment's dynamics. Like DP, TD methods update estimates based on other learned estimates, without waiting for an outcome (i.e., they *bootstrap*).

There are two main methods for TD: state–action–reward–state–action (SARSA) and Q-learning. SARSA and Q-learning work similarly, assigning values to state-action pairs in a tabular way. The key difference is that SARSA is on-policy, whereas Q-learning is off-policy. This means that SARSA learns action values relative to the behavior policy (or the policy it follows), while Q-learning does so by following a greedy policy. In other words, SARSA uses the value of the action following the same ϵ-greedy policy all the time. In contrast, Q-learning uses the maximum value over all possible actions for the next iteration.

Multiple works using TD have been studied in the literature on wireless communications. Here, we overview some of them and point out the states contemplated. A distributed RL approach for OFDM subcarrier and power allocation is presented in [66], where a global Q-table is shared among agents (base stations). The states are defined as a combination of the user allocation to the subcarriers, interference, and power, and the action is the user allocation to the subcarriers and the power. A backoff stage selection through Q-learning is proposed in [15], where the state is defined as the backoff stage itself, whereas the action is to increase/decrease such stage. Authors in [69] propose centralized Q-learning for spectrum and channel allocation in cognitive radios, where a single agent is installed in the cluster head. The state is defined as a combination of parameters like the nodes' locations, time zones, and band groups, whereas the action is the channel to select. Q-learning is also the algorithm of choice in [14], where a case study is provided in the form of contention window selection in an IEEE 802.11ax WLAN. Similarly to [15], the state is the backoff stage, and the action is to increase/decrease the stage. Finally, the adaptation of different parameters of a wireless link (i.e., channel bandwidth, MCS,

or guard intervals) is proposed in [74]. In this case, the learning is done through a distributed SARSA implementation, where the state is defined the same as the action, consisting of the tuple of values of each link parameter.

Deep Reinforcement Learning

Deep learning (DL) is a branch of ML based on a set of algorithms that attempt to model high-level abstractions in data by using multiple processing layers, with complex structures or otherwise, composed of multiple non-linear transformations. They do so via deep neural networks (DNNs) consisting of multiple hidden layers. So, RL's combination with deep learning results in DRL, where the RL policy or algorithm uses a DNN rather than arrays or tables. The most common form of DRL is a deep learning extension of Q-learning, called deep Q-network (DQN). So, as in Q-learning, the algorithms still work with state-action pairs, conversely to the MAB formulation. In plain, DQN is a much complex version of the TD methods explained before. But why would one make the problem more complicated? It turns out that convergence in tabular Q-learning gets so slow when the number of states and actions increases. That is, it is so costly to keep gigantic tables that we use DNNs to approximate such a table.

DRL has been gaining attention in wireless communications in the last years. For instance, an on-demand channel approach through multi-agent DRL is proposed in [106], where a central controller aids agents at the edge. The state is defined as a tuple of the current and average load, and the action is composed of the primary channel and allocated bandwidth. A DQN is also implemented for contention window optimization in [133], where a single-agent at unique AP runs the learning. States are defined as a history of collision probabilities, and actions are simply the AP's contention window. A joint beamforming, power control, and interference coordination solution based on DQN executed by a single-agent controlling all the BSS's at a central location is studied in [98]. In this work, states are related to positions, power, beamforming, and actions to increase/decrease the power by an amount inside a range of possibilities and beamforming parameters. Similarly, a single-agent DQN is proposed in [122] for channel selection with binary rewards in the context of cognitive radios. States (or observations) are a representation of the channel status (busy or idle), and the action is the selected channel. Finally, the selection of network slices in the form of MCS, CCA, and transmission power is performed in [51] through a single-agent implementing DQN. The state is defined as the combination of different configuration parameters (including channel, required throughput, or RSSI), and the action consists of the tuple MCS, CCA, and transmission power.

For the problem we deal with in this thesis, we propose having a MAB instance in each agent-empowered AP to adapt to the environment without states. The reason lies in the fact that meaningful states are intricate to define and its effectiveness heavily depends on the application and the type of scenario under consideration. In the end, a state is a piece of information that should help the agent by fitting it into its policy. If the state is meaningless, or worst, misleading, it is preferable not to rely on states and go for a stateless approach. As we show later, MABs provide a lightweight and efficient framework to cope with the spectrum management problem. Section §6.4 treats MABs in detail and the evaluation of different MAB algorithms is conducted in §6.5.

Proposed RL approach: First, we envision a **distributed** architecture rather than a centralized one to boost the responsiveness and adaptability in highly dynamic environments. Second, we propose a **MAB** formulation for the RL problem to speed up the learning convergence by disregarding states.

6.4 Multi-Armed Bandits

In this thesis, we tackle the spectrum allocation problem in WLANs as a MAB due to its ready-to-use conception for problems facing the exploration-exploitation trade-off. Namely, we would like the agent to learn the best action quickly, but at the same time, we want the agent to perform as good as possible along its quest to find such action.

6.4.1 The MAB problem

The multi-armed bandit problem is a classic RL problem that exemplifies the exploration vs. exploitation dilemma, where no knowledge on the reward probability distribution of each action is assumed. The typical example to introduce the MAB problem is the gambler who faces multiple slot machines, each with different probability distributions over the money he/she can get after pulling the slots. Naturally, the gambler wants to get as much money as possible, thus achieving the highest reward in the mid/long-term. So, what strategy should he/she follow? That is where the MAB problem arises with its different variations. For instance, a naive approach would be to pull each machine slot a lot of times, so the gambler could guess the "authentic" reward probability of each slot according to the law of large numbers, and then select always the best slot machine. However, such an approach has obvious pitfalls like determining how many times are enough to discover the "true" reward probability or wasting too many trials (or iterations) in a priori lousy machines. Further, in environments where the distribution of such rewards may change due to external actors (as it is the case in WLANs), it would not be viable to get accurate estimates of the reward probability distributions. Simply because from one iteration to the other, the underlying reward distribution may be completely changed.

Formally, a MAB can be described as a tuple of the set of actions and the reward function $< \mathcal{A}, R >$, where:

- There are K arms (slot machines in the previous example or spectrum management configurations in this thesis) with hidden reward probabilities $\{\theta_1, \ldots, \theta_K\}$. $\mathcal{A}$ is the set of actions (or action space), each referring to the interaction with one arm.
- At each time step (or iteration) t, we take an action (or arm) a and observe a reward r_t resulting from performing such action.
- We refer to the *value* of an action a as the expected reward, $Q(a) = \mathbb{E}_{\theta_a}[r|a]$, determined by the reward distribution θ_a.
- Finally, the reward function R defines the reward values obtained at any time step t, $r_t = R(a_t)$. For instance, $R(a_t)$ could be defined in the previous example as the money earned in the last slot pull or as the throughput experienced by a BSS when using action a_t during the last iteration t.

Notice the difference between $R(a_t)$ and $Q(a_t)$. While $r_t = R(a_t)$ is the *actual* reward observed when applying action a_t in iteration t (e.g., throughput observed when playing a_t at iteration t), $Q(a_t)$ is the *expected* value of playing a_t (e.g., mean future throughput when playing a_t regardless t). In other words, $R(a_t)$ is a particular outcome of the reward distribution $\theta(a_t)$, whereas $Q(a_t)$ is a representation (or statistic) of the arm's value (e.g., mean reward).

The goal is to maximize the cumulative reward G over a finite number of iterations T (6.11). The simplest action selection rule (or policy π) is to select one of the actions with the highest estimated value, that is, one of the greedy actions, $\pi_t = a_t = \arg\max_{a \in \mathcal{A}} Q_t(a)$. However, greedy selection always exploits current knowledge to maximize the immediate reward, i.e., it does not sample the rest of a priori inferior actions. So, what happens if the action that we have ranked as the best is not? There are more sophisticated selection rule alternatives like ϵ-greedy methods, which force exploration with some probability ϵ to achieve better knowledge on the rewards of the actions in the whole action space. We depict some of the most relevant ones in §6.4.3.

6.4.2 MAB taxonomies

There are different types of MABs. They can be categorized according to three main concepts: the type of reward, the use of contexts, and the temporal horizon.

Reward taxonomy

According to its reward distribution, there are two main classes of MABs: stochastic and adversarial. The **stochastic** MAB is the most common one, and it is completely determined by the distributions of rewards $\{\theta_1, \ldots, \theta_K\}$ of the actions in the action space. In particular, in iteration t, the distribution of the reward observed by a learner that chose action $a_t \in \mathcal{A}$ is θ_a, regardless of the past rewards and actions. The reward for each action is independent and identically distributed (i.i.d). That is, every time an action is chosen, the reward is sampled independently from this distribution. Both for simplicity and for guaranteeing theoretical convergence in certain MAB algorithms, the reward function is bounded, $r_t = R(a_t) \in [0, 1], \forall a, t$, so normalized performance metrics (e.g., throughput satisfaction) are normally used. The main limitation of stochastic MABs is the difficulty of finding real-world problems that rely on suitable distributions. Indeed, rewards can be non-stationary; thus, the reward distribution probability for a given action may change over time. Next, we present adversarial MABs, a paradigmatic example of non-stationary rewards.

In **adversarial** MABs, learning is still possible in the sense that the regret can be kept sublinear. However, selection rules methods must be completely different from the stochastic ones since adversarial rewards can be arbitrary, as if they are chosen by an "omniscient adversary", so the stochastic assumption on the rewards being generated from a fixed distribution does not longer hold. In plain, there are no probability distributions to learn whatsoever. Then, what can the agent do to face such an adversary? The key idea is to introduce noise in the exploration to choose random actions that the adversary could not expect.

So, what class of MAB suits better the joint spectrum management problem? We anticipate that the stochastic one. Even though the different actions' reward distributions are generally complex to estimate, no omniscient adversary tells what the rewards will be at every iteration. Instead, we have different actors (nodes) that behave according to a certain configuration

each. Such configurations determine what the actors will do, resulting in a stochastic setting. Nonetheless, even though the problem is likely to be stochastic, it is so complex that an adversarial approach when we assume that no distributions should (or could) be learned may also be appropriate to design a fruitful algorithm. Accordingly, we assess in further sections the convenience of both stochastic and adversarial MABs.

Context taxonomy

While the general MAB formulation does not consider states, *contexts* may be used to fine-grain the arm selection based not only on the gathered rewards, but also on further observations about the world $\mathcal{I}$. Essentially, we may categorize MABs in context-less and contextual MABs. Context-less are the general MAB formulation where no contexts are assumed.

Conversely, in **contextual** bandits, reward distributions depend on a context, which is fit to the RL algorithm before making a decision. Now, the observed reward r_t in each iteration t depends both on the context Ω_t and the chosen action a_t. For instance, assume that an AP observes that a given basic channel, say basic channel 2, is busy most of the time. The derived context would then represent that channel 2 is active. Thus, it is not the same to apply the action corresponding to pick primary $p = 2$ when the context indicates channel 2 is busy rather than applying the same action when such channel is idle. Henceforth, it seems that contexts may aid to the spectrum management problem, especially if we leverage knowledge from previous contexts (e.g., similar contexts stochastically repeated every day). However, contextual MABs also pose the challenge of appropriately defining contexts, meaning that a good representation of the world $\mathcal{I}$ should be captured in each context.

There are different approaches within the contextual bandit problem. The simplest one, while the most versatile, considers a separate MAB per context (as if each context was a state). Then, the goal is to find the optimal action $a^*(\Omega_t) = \pi^*(\Omega_t)$ per context Ω_t.

So, should we go for a contextual bandit approach to tackle the joint spectrum management problem? We believe that it is better not to. Essentially, defining contexts may be as complex as defining states. And what would be an appropriate context definition? the current action, $\Omega_t = a_t$? The current reward, $\Omega_t = r_t$? A mix of the action selected and the reward $\Omega_t = f(a_t, r_t)$? A mix including also historical information $\Omega_t = f(\{a_{t-t'}, \ldots, a_t\}, \{r_{t-t'}, \ldots, r_t\})$? Unfortunately, we do not find any definition that suits our multi-agent and non-stationary reward problem. The reason lies in the fact that we want to boost the learning speed to raise the user experience in the short/mid-term. Then, relying on a different MAB per action in a contextual fashion would entail large and potentially unfruitful learning periods.

Temporal taxonomy

The last taxonomy we identify is the temporal one, defined by the duration imposed by the programmer to converge to a good solution, which determines the *horizon*: finite or infinite. **Finite-horizon** refers to when there is a need to perform (explore and exploit) during a given time window (e.g., 5 minutes). So, the number of iterations T the agent counts with is finite. Thus, the RL algorithm must try to get the most accumulated reward (6.11) with that ending in mind.

Infinite-horizon, in contrast, poses no time limit to the agent to converge, i.e., $T \to \infty$. As a matter of fact, in many problems, a finite time horizon cannot be easily specified, so the

infinite horizon formulation fits more naturally. More importantly, stationary problems with infinite time horizon lead to optimal stationary strategies, which offer great simplicity.

For the joint spectrum management problem of this thesis, we will rely on finite horizons to assess the *myopic* nature of the MABs. We call it myopic in the sense that they try to optimize rewards now without any explicit use of forecast information or any direct representation of decisions in the future. Meaning we want BSS's to improve its performance as soon as possible.

Proposed MAB approach: First, we choose to study **stochastic** rather than adversarial MABs because of WLANs' non-adversarial nature. Second, we adopt the standard **context-less** formulation of MABs given the unfeasibility of defining lightweight meaningful context. Third, we study the performance of MABs in **finite-horizon** experiments because we aim at boosting its learning speed.

6.4.3 MAB exploration strategies

Since any stochastic MAB problem can be seen as an instance of an adversarial problem, we may apply all the following algorithms regardless of whether contexts are defined or not. We depict the different action selection strategies of choice below.

Exploration-first. (Algorithm 2 in the Appendix) The most simple MAB algorithm dedicates the first T_{exp} rounds to exploration and the remaining $T - T_{\text{exp}}$ rounds to exploitation. We may mandate the full exploration of the action space, i.e., $T_{\text{exp}} = K$ to explore all the arms, as we do later in the experiments. Notice that in the event of an environment change after the exploration phase, action rewards may drastically change. Exploration-first copes with this issue by simply ranking actions according to their last observed reward. So, if the best action in iteration $t-1$ turns out to be the worst in t, since its reward is updated, the second-best action in $t-1$ is now the best in t and its pick accordingly in $t+1$. Consequently, exploration is implicitly carried when action rewards vary over time.

Epsilon-greedy. (Algorithm 3 in the Appendix) The ϵ-greedy or (e-greedy) algorithm takes the best known action with probability $1-\epsilon$, and explores a random action (previously explored or not) with probability ϵ [125]. The action value Q is estimated, such estimation being represented as $\hat{Q}$, according to the past experience by averaging the rewards associated with the target action a that we have observed so far (up to the current time step t), i.e., $\hat{Q}_t(a) = \frac{1}{N_t(a)} \sum_{\tau=1}^{t} r_\tau \mathbb{1}[a_\tau = a]$, where $\mathbb{1}$ is a binary indicator function telling whether the action a was used in a given iteration, and $N_t(a)$ is the number of times the action a has been selected so far, i.e., $N_t(a) = \sum_{\tau=1}^{t} \mathbb{1}[a_\tau = a]$. In an exploring iteration, any action is picked with same probability, whereas in an exploitation iteration, the best estimated action is picked, i.e., $a = \text{argmax}_{a \in \mathcal{A}} \hat{Q}_t(a)$. As for the experiments later presented, we set the original epsilon value $\epsilon_0 = 1$ to force exploration. Besides, to avoid inefficient exploration after *enough* iterations, we decrease the parameter ϵ to reduce the probability of exploring over time as suggested in [20].

UCB1. (Algorithm 4 in the Appendix) Due to the randomness in e-greedy, we may end up exploring a bad action that we have confirmed in the past. To avoid such inefficient exploration, one approach is to decrease the parameter ϵ in time (as done in ϵ-greedy), and the other is to be *optimistic* about options with high uncertainty and thus to prefer actions for which we have not had a confident value estimation yet. The Upper Confidence Bounds (UCB1) al-

gorithm, introduced first in [82] and further analyzed in [20], measures this potential by an upper confidence bound of the reward value, $\hat{U}_t(a)$, so that the true value is below the bound, $Q(a) \leq \hat{Q}_t(a) + \hat{U}_t(a)$ with high probability. The upper bound $\hat{U}_t(a)$ is a function of $N_t(a)$; a larger number of trials $N_t(a)$ should give us a smaller bound $\hat{U}_t(a)$. In UCB1, we always select the greediest action to maximize the upper confidence bound: $a_t = \text{argmax}_{a \in \mathcal{A}} \hat{Q}_t(a) + \hat{U}_t(a)$. For the cases where no prior knowledge on the distributions is given, the upper confidence bound can be derived as $\hat{U}_t(a) = \sqrt{\frac{2 \log t}{N_t(a)}}$. So, by looking at $\hat{U}_t(a)$, for a given value estimation $\hat{Q}_t(a)$, we see that the more times an action is picked – the larger $N(a)$ – the less probable it becomes since the uncertainty about its reward distribution is reduced.

Thompson sampling. (Algorithms 5 and 6 in the Appendix) We do not assume any prior on the reward distribution θ_a of action a in UCB1. Therefore we have to rely on a generalized estimation of the upper confidence bound. If we know the distribution upfront, we would be able to make better-bound estimations. For example, if we expect every arm's mean reward to be Gaussian, we can set the upper bound as 95% confidence interval by setting $\hat{U}_t(a)$ to be twice the standard deviation. In Thompson sampling, we want to select action a according to the probability that a is optimal. In particular, at each time t, we draw a sample from the prior distribution of every action $a \in \mathcal{A}$, $\tilde{Q}(a) \sim \theta_a$. The best action is then selected among the drawn samples: $a_t = \text{argmax}_{a \in \mathcal{A}} \tilde{Q}(a)$. After the last actual reward $r(a_t)$ of the selected action a_t is observed, we update its reward distribution parameters (e.g., mean and standard deviation for the normal distribution, and α and β parameters for the Beta distribution) accordingly, which is essentially making Bayesian inference to compute the posterior with the known prior and the likelihood of getting the sampled data.

Exp3. (Algorithm 7 in the Appendix) Unfortunately, reward probability distributions are so complex that having optimism is arguably naive, especially in competitive scenarios like WLANs. To conceive rewards that could operate in any manner, the adversarial model was proposed, where the agent must face an omniscient adversary that secretly chooses a sequence of rewards at the start of the game. Auer's illustrative example is to think about a gambler that plays in a rigged casino [21]. The Exponential-weight algorithm for Exploration and Exploitation (Exp3) is is a popular algorithm for adversarial MABs [115]. The trick of the agent is the randomness in his choice of actions. Exp3 works by maintaining a list of weights for each action, using these weights to decide which action to take next randomly, and increasing (decreasing) the relevant weights when a payoff is good (bad).

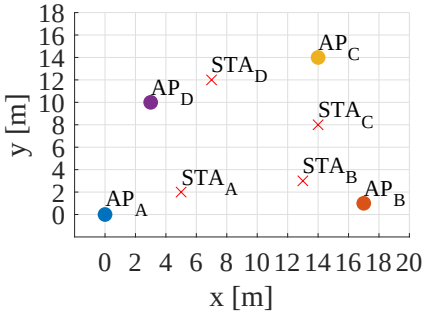
6.5 Evaluation

In this section, we evaluate the performance of the MABs depicted in §6.4.3. We first benchmark them against the optimal global configuration in a toy self-contained deployment, from which we know the optimal configuration for every traffic load. Then, we generalize the results to denser random deployments by comparing the most promising MABs: exploration-first and ϵ-greedy.

6.5.1 Self-contained dataset

We study the deployment illustrated in Figure 6.3a, consisting of $W = 4$ BSS's (with one AP and one STA each) in a system of $n_C = 4$ basic channels. The traffic load is quantized and can

take three values, $\ell_w \in \mathcal{L} = \{20, 50, 150\}$ Mbps $\forall w$.[4] As for the action attributes, the primary channel can take any of the n_C channels in the system, $p \in \{1, 2, 3, 4\}$, and the maximum bandwidth is allowed to be set to $b \in \{1, 2, 4\}$ fulfilling the IEEE 802.11ac/ax channelization restrictions for 20, 40, and 80 MHz bandwidths. So, according to (6.5), the total number of statuses when considering all BSS's to have an agent-empowered AP (i.e., $W' = W = 4$) raises to $O(\mathcal{H}) = (4 \times 3 \times 3)^4 = 1,679,616$. Notice that even a petite deployment like this one leads to a vast number of possible statuses.



(a) Location of nodes.

	A	B	C	D
A	-	40	20	80
B	40	-	80	40
C	20	80	-	80
D	80	40	80	-

(b) Interference matrix.

Figure 6.3: Toy deployment of the self-contained dataset.

The interference matrix elements in Figure 6.3b indicate the minimum bandwidth in MHz that causes two APs to overlap. So, we note that this deployment is complex in the sense that multiple different one-to-one overlaps appear depending on the distance and the transmission bandwidth in use, leading to exposed and hidden node situations hard to prevent beforehand. For instance, AP_A and AP_C can only overlap when using 20 MHz, whereas AP_A and AP_D do always overlap regardless of the bandwidth because their proximity. Instead, AP_A and AP_B overlap whenever 40 MHz (or 20 MHz) bandwidth is used.

To generate the self-contained dataset, we simulated all the possible statuses $\mathcal{H}$ of the deployment in Komondor. So, we first create the deployment, meaning that we deploy the APs and STAs in fixed positions in the map (as in Figure 6.3a). Then, we define the values of the modifiable attributes and the mean traffic load that each BSS experiences. Likewise, we set the configuration capabilities of each BSS, so we empower each agent with the action space $\mathcal{A}_w = \{\{p_w\}, \{b_w\}\}$. For simplicity, we consider all the BSS's to have the same action space $\mathcal{A}_w = \mathcal{A} = p \times b, \forall w$. Finally, for every WLAN's global configuration, we must generate all the input files corresponding to each possible status $\mathcal{H}$ to be later simulated with Komondor. Notice that the input $x = \mathcal{H}$ is mapped to the output y, where y is an array containing multiple performance metrics of each BSS, including throughput and delay. Every status $\mathcal{H}$ is evaluated during $T_\mathcal{H} = 5$ seconds. Given that the WLAN remains steady during any status, we empirically found that 5 seconds is enough to accurately estimate the long-term mean performance. We use Wi-Fi parameters according to 802.11ax [27].

Once we gather the whole dataset of the deployment through Komondor, we use it to benchmark the performance of the MAB algorithms depicted in §6.4.3 under different traffic patterns. For the sake of explanation, assume that the WLAN starts at $t_0 = 0$ with status $\mathcal{H}_0$. At t_1, one agent-empowered BSS picks an action a_1 (e.g., switches the primary channel), leading to status

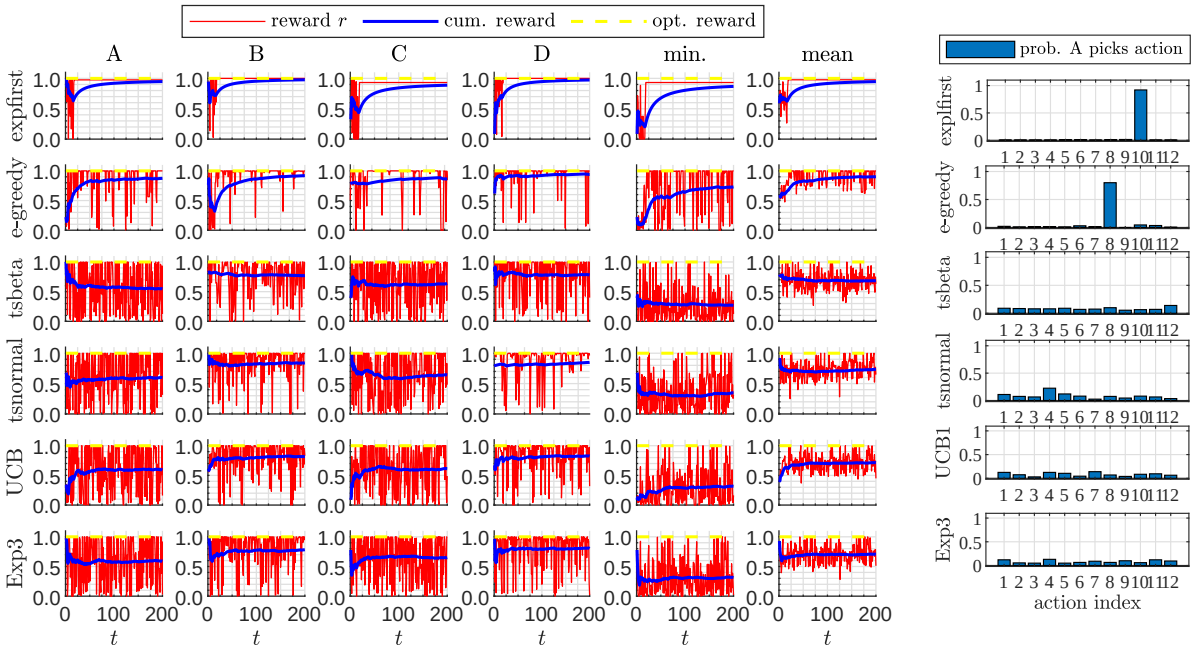
[4]The 4-BSS's self-contained dataset for spectrum management in WLANs can be found at `https://www.upf.edu/web/wnrg/wn-datasets`.

$\mathcal{H}_1$. Notice that, since we already have the dataset generated, it is straightforward (and fast) to check the performance y of all the BSS's in the WLAN at such given status. We essentially use the dataset as a lookup table $x = \mathcal{H} \rightarrow y$, so the computation speed rises.

6.5.2 Benchmarking RL algorithms

Learning under constant traffic demands

We first aim at assessing the learning capacity, including the achievable reward and the corresponding convergence speed of the presented MABs in a WLAN with constant traffic loads. In essence, we ask: do MABs converge to *good* rewards under steady load conditions? How long do they take? We assume each BSS having a high average traffic load, $\ell_w = 50$ Mbps, $\forall w$. So, this experiment's long-term dynamism is due exclusively to the MABs progression in the multi-agent setting. The inner traffic load condition does not vary, but the spectrum management configurations (or actions) do. We run a single simulation for each MAB intending to display a realization of how they behave.



(a) Run chart of the instant, cumulative, and optimal reward of each BSS, minimum and mean of the WLAN under fixed traffic loads.

(b) Prob. BSS A to select each action.

Figure 6.4: Self-contained toy dataset under constant traffic loads.

Figure 6.4a shows the throughput satisfaction evolution of the MABs. For each MAB, and for each BSS w, we plot the instant reward $r_{w,t}$, the normalized cumulative reward $G_{w,t}/t$, the optimal reward of each separate BSS, and the minimum and maximum optimal of the whole WLAN. Notice that some BSS's can only achieve their optimal by forcing the others to underperform, so we refer to the minimum of the WLAN ("min." in the Figure) as the most important metric to assess the MAB fairness. We observe that while exploration-first and e-greedy do converge (learn) to higher satisfaction values close to the optimal, the rest of MABs seem not to learn at all. This relates to the fact that learning based on parameter

estimation is fruitless in dynamic and chaotic multi-agent deployments like this. So, it seems preferable to act quickly with a low-level knowledge of the actions' rewards. Besides, we find that win-win relations are established as shown by the minimum and mean convergence of the whole WLAN. That is, on average, BSS's tend to benefit from others' benefits. As for the learning speed, exploration-first takes approximately a full exploration of the action space with erratic rewards to provide a super-steady performance, whereas e-greedy keeps switching configurations throughout the simulation.

Figure 6.4b shows the probability of BSS A (picked as an example) to select each possible action. We observe that exploration-first and e-greedy tend to use one single action that behaves *sufficiently* well, so they waste less time in exploring than exploiting. In contrast, the rest of MABs *waste* too many iterations exploring the different actions.

Finding #15: In small deployments with constant traffic load, exploration-first and e-greedy (so-called *ligthweight* MABs) achieve near-optimal rewards after *enough* exploration, showcasing their ability to learn in multi-agent Wi-Fi deployments by establishing win-win relationships between BSS's. Further, exploration-first is the only to lead to a super-steady performance at the cost of experiencing vacillating rewards during the early exploration phase.

The usefulness of MAB parameters into the spotlight: why do sophisticated MABs not work?

From the previous experiments, we concluded that UCB1, Thompson Sampling, and Exp3 are not an option for the multi-agent setting of the joint spectrum management problem in WLANs. Here, we delve deep into why, which is tied to the different parameter estimations these algorithms must conduct. First, UCB1 cannot profit from computing an estimation of the upper confidence bound of any action a. In essence, the action selection relies on the sum of such confidence bound and the estimated value $\hat{Q}(a)$. Given the non-stationarity of the multi-agent setting, the estimation $\hat{Q}(a)$ is lousy and may highly vary from one iteration to the other. The algorithm then falls in a loop where most of the actions are selected with similar probability, thus wasting lots of iterations with low performance and generating even more dynamism to the WLAN.

Second, even though Thompson Sampling is usually known to outperform the rest of the presented MABs in stationary environments [91], its performance is well below exploration-first and e-greedy under the multi-agent setting. Similarly to UCB1, the reason lies in trying to estimate the parameters (e.g., α and β in the Beta distribution) defining the probability distribution of the reward of each action while assuming that it is constant, or at least, slowly varying in time, so it would be worth to learn. However, that is not the case at all in the multi-agent setting, where from one iteration to the other, the actual probability distribution of any action may completely change. So, the gathered knowledge from past iterations can turn entirely worthless to estimate the new distribution.

To illustrate the low utility of estimating the probability distribution of an agent's actions in a multi-agent setting, let us consider an elementary example. Assume two single-channel BSS's overlap, and there are two possible primary channels to pick in the system $C_A = C_B = \{1, 2\}$. We call the BSS's A and B, respectively, and assume they have a constant traffic load. From the

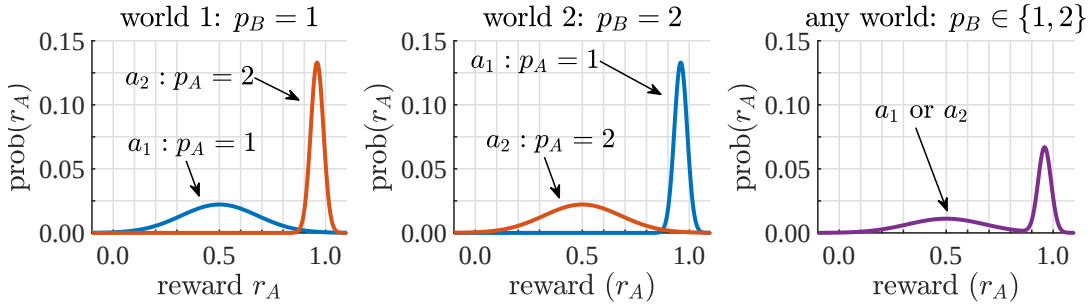


Figure 6.5: Probability distributions of the reward of each action BSS A can take.

point of view of A, its world $\mathcal{I}_A$ is determined just by B's action selection. That is, B selecting primary $p_B = 1$ generates a first world for A where it is better to select $p_A = 2$ to avoid contention and collisions. Likewise, B selecting primary $p_B = 2$ generates a second world for A where it is better to select $p_A = 1$. Figure 6.5 shows possible probability distributions (modeled as Gaussian variables for simplicity) of the reward of A in terms of throughput satisfaction. As anticipated, it is much better for A to select B's opposite action, i.e., the probability of getting a better reward is much higher. So, with this information, would the agent be able to make better decisions? Indeed, it would. However, since MABs do not count on states, the agent does not have a separate reward distribution of each action per world (or state). In contrast, it generates a state-blind distribution like the one shown in the most right plot in Figure 6.5. And the information contained in there, beyond simple statistics like the mean, is of very little use.[5]

Finally, Exp3 tackles adversarial settings, where an omniscient adversary may change the reward distribution of each action at its will. In particular, the *urgency* of an action is the sum of two terms: an exponential term in the success of the action for exploitation, and a constant term for exploration. So, why is Exp3 not working in our multi-player setting? Like UCB1 and Thompson Sampling, Exp3 relies on estimating the reward for the exponential term related to exploitation. And, as in the previous algorithms, such estimations are not reliable at all.

Finding #16: In contrast to the sound performance of exploration-first and e-greedy, the rest of MABs (i.e., Thompson Sampling, UCB1, and EXP3) cannot learn due to their need to over-exploring the action space for estimating algorithm parameters resulting flawed and unfruitful in uncoordinated multi-agent setups.

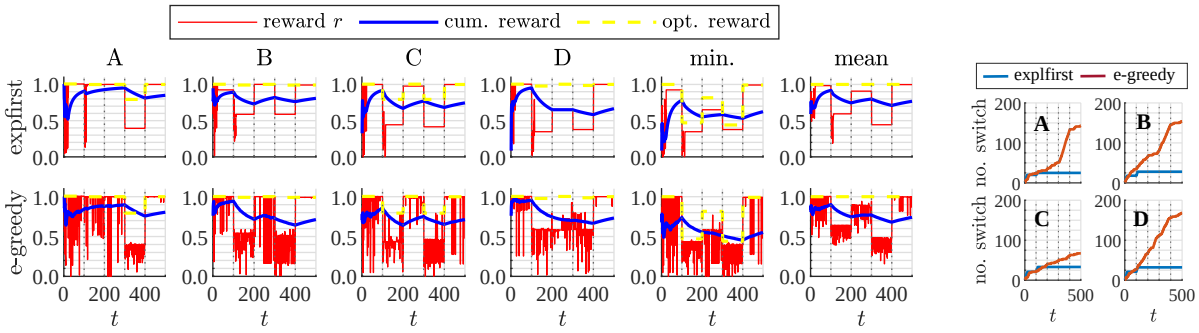
Adapting to varying traffic loads

In the previous experiment, we observed that both exploration-first and e-greedy were able to reach near-optimal rewards with relatively fast convergence in setups with constant traffic loads. In this experiment, we focus on the performance of the MABs mentioned above under varying traffic loads patterns. Are they able to adapt quickly to such changes in the load demand?

We now consider a traffic load pattern where each BSS w changes its mean traffic load to $\ell_w \in \{20, 50, 150\}$ Mbps every 100 iterations. This time, we focus our analysis only on

[5]Moreover, notice that generating the reward distribution with accuracy for every state-action pair is a costly task itself that requires extensive exploration.

exploration-first and e-greedy because they were the only MABs capable of learning under non-varying traffic loads.



(a) Run chart of the instant, cumulative, and optimal reward of each BSS, minimum, and mean of the WLAN in the scenario with varying traffic loads.

(b) Evolution of no. of switches.

Figure 6.6: Self-contained toy dataset under varying traffic loads

Figure 6.6a shows the throughput satisfaction evolution as in Figure 6.4a, this time considering load changes every 100 iterations. We find that both exploration-first and e-greedy can raise their performance after every traffic change, so they learn after every environment alteration. Again, we observe that exploration-first is much more *steady* than e-greedy. In fact, after the initial full exploration phase, exploration-first can quickly adapt to each environment change.

This steadiness is corroborated by looking at the number of action switches shown in Figure 6.6b. Exploration-first hardly changes an action after the full exploration phase in the early stages, whereas e-greedy keeps often changing, although the decreasing update function of the ϵ parameter. This number of switches is an essential metric to the BSS performance. After every change in the spectrum management configuration, the PHY and MAC overheads required to set up the AP and STAs in the BSS entail delays hindering the overall performance. In this regard, exploration-first seems a much more favorable choice. Indeed, one well-known pitfall of e-greedy is its poor asymptotic behavior because it usually continues to explore long after the optimal solution becomes apparent [117].

Finding #17: Both exploration-first and e-greedy perform similarly in terms of cumulative (or long-term) reward, getting close to the optimal minimum in the WLAN after *relatively* few iterations. Nonetheless, exploration-first adapts much more *smoothly* than e-greedy after each load alteration. That is, it requires very few iterations to select its a priori best action once the first full exploration phase is completed. This has substantial benefits given that every time a different action is chosen, the configuration should be broadcast and the STAs in the BSS require some time to adapt to it, leading to service interruptions and detrimental performance.

Can contexts or states aid learning?

Part of the reasoning we conducted on why MABs seemed the most suitable approach for this thesis' problem was related to the fact of avoiding states. This way, there is no need to seek

meaningful state definitions, and the learning can be executed more quickly since only actions and rewards are used to execute the policy.

Nonetheless, in this experiment, we go beyond such qualitative reasoning and provide a quantitative measure on Q-learning performance (relying on states), and contextual MABs (relying on contexts). For the sake of boosting the learning speed, given we consider a relatively short time-horizon, we define the state s and context Ω the same: as merely the binary throughput satisfaction of the BSS,

$$s = \Omega = \begin{cases} \text{true} & \xi > 0.95 \\ \text{false} & \text{otherwise} \end{cases}, \tag{6.12}$$

where ξ is the throughput satisfaction, which we compare against a 0.95 threshold to provide a safety margin. Notice that state space $\mathcal{S} = \{\text{true}, \text{false}\}$ is small and therefore does not provide much information on the environment. However, other more meaningful state definitions considering parameters like the selected action or spectrum occupancy statistics would exponentially increase the size of $\mathcal{S}$ and reduce the learning speed dramatically. For instance, if the state was defined simply as the action, i.e., $s = a$, the state-action table would be 12x12 given the $|\mathcal{A}| = 12$ actions in the action space. Accordingly, 144 state-action elements would exist. So, it would take *at least* 144 iterations to try each of the pairs once. In contrast, the proposed state space generates a more manageable state-action table of 2x12.[6]

Q-learning is a temporal difference algorithm seeking the best action to take given the current state. It is considered off-policy because the Q-learning function learns from actions outside the current policy, like taking random actions. The learning rate α or step size determines to what extent newly acquired information overrides old information. A factor $\alpha = 0$ makes the agent learn nothing (exclusively exploiting prior knowledge). In contrast, a factor $\alpha = 1$ makes the agent consider only the most recent information (ignoring prior knowledge to explore possibilities). We use a high value of $\alpha = 0.7$ because the non-stationary of the deployment. The discount factor γ determines the importance of future rewards. A factor $\gamma = 0$ will make the agent myopic (or short-sighted) by only considering current rewards, while a factor approaching $\gamma \to 1$ will make it strive for a long-term high reward. We use a relatively small $\gamma = 0.3$ to foster exploitation (for rapid convergence) in front of exploration.

Another way in which agents use partial information gathered from the environment is in the form of contexts. In plain, contexts can be formulated as *states for stateless approaches*. That is, a contextual MAB can be instantiated like a particular MAB running separately per context. The learning is separated from one context to the other so that no information is shared between contexts like it is the case for Q-learning and other state-full approaches. In this case, we propose having a separate MAB instance for each of the two contexts defined (true and false).

Figure 6.7 is an extension of Figure 6.4a, this time showing the performance of the proposed Q-learning and contextual MAB. While we observe that both tend to learn, i.e., the individual reward of all BSS's increases over time, we find unsteadiness with peaks of low (even zero) throughput satisfaction throughout all the iterations. Remarkably, such low throughputs may generate service interruptions causing poor user experience. So, we have seen that, by consid-

[6]From this brief discussion on the size of the state-action space we can conclude that more complex approaches like DRL do not suit in the scenarios of this thesis. DRL relies on huge state-action spaces and normally takes long periods to start learning.

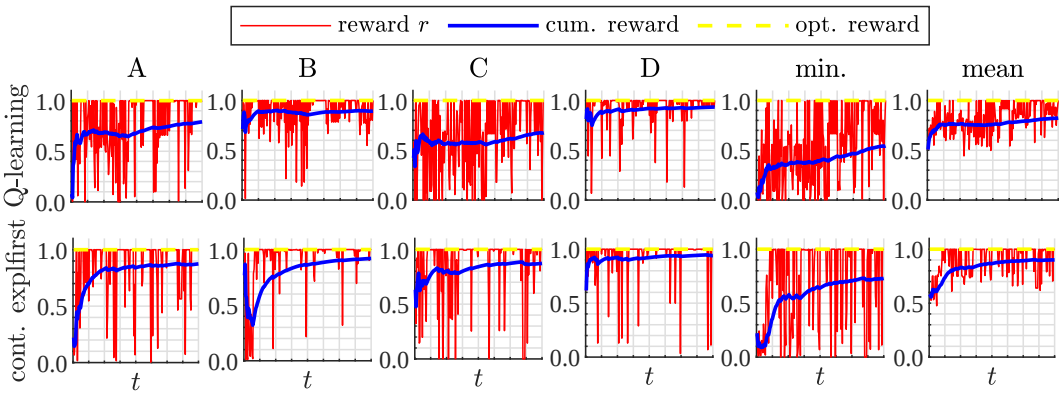


Figure 6.7: Q-learning and contextual MABs under constant traffic loads.

ering states and contexts, agents also tend to learn in the proposed finite horizon, but at a slower pace than the lightweight MABs.

We envision using states and context for larger time horizons when the BSS's may support periods of poor performance until the WLAN reaches steady dynamics providing even higher rewards than MABs. In such cases, more complex and larger state spaces may aid the task. We leave the study of such approaches as future work since this thesis focuses on rapid-adaptation mechanisms for zero-prior knowledge scenarios.

Finding #18: RL algorithms harnessing states or contexts do not boost the learning rate in short-term finite horizons like those studied in this thesis. Therefore, we corroborate that stateless approaches like MABs are a preferable choice for such a task.

6.5.3 Generalization to high-density deployments

One question remains unanswered: what happens if the number of potentially overlapping BSS's and their action spaces increase? That is, what happens in even more complex multi-agent settings? We now leave the holistic toy dataset and assess many random deployments to generalize the MABs behavior.

Are not heuristics enough? DyWi vs. MABs

Before digging deep in the analysis of lightweight MABs in random high-density deployments, let us first corroborate that the MAB formulation for the joint problem of primary channel selection and bandwidth allocation outperforms heuristic-based algorithms. In particular, we compare DyWi [27], the online primary channel selection presented in §5.2, against exploration-first for the very same scenarios. Notice that DyWi assumes specific knowledge of the environment in the form of occupancy statistics per each basic channel. Gathering heuristics usually entails an extra effort in terms of energy consumption, processing, or hardware capabilities. Further, the statistics may be outdated in multi-agent settings. However, we assume a unique agent-empowered BSS as we did to analyze DyWi to compare the case of a single-agent vs. a fixed world.

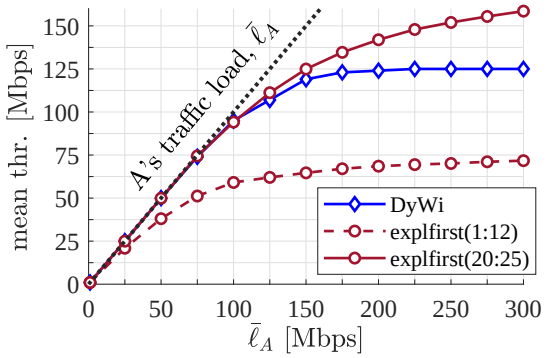


Figure 6.8: Mean throughput achieved with DyWi and exploration-first. The number (X:Y) for exploration-first represent the first and last iteration where the mean is computed.

Figure 6.8 shows the mean throughput achieved with DyWi, as in Figure 5.2, but this time we include the mean throughput of exploration-first in two iteration windows. The first one corresponds to part of the early action space exploration (iterations 1 to 12) and the other window corresponds to the exploitation phase (iterations 20 to 25). We observe two main phenomena: *i*) it is much better to rely on heuristics from scratch since they apply their fixed policy based on statistics no matter the iteration, and *ii*) when the full exploration phase of exploration-first finishes, the mean throughput outperforms DyWi's significantly. So, for steady worlds where only a single agent may change its configuration, heuristics are a good approach for immediate adaptation, but MABs can provide higher mid-term rewards. Nevertheless, let us emphasize that in this single-agent setting, the heuristics gathered by DyWi hold in the sense that the rest of BSS's do never change their configuration. Then, in multi-agent settings like the ones covered in this chapter, heuristics are not a valid option.

Finding #19: Heuristics-based algorithms like DyWi provide immediate adaptation in *steady* environments by using fixed policies relying on observable parameters. In contrast, MABs, by nature, need to learn in a trial and error fashion, which entails unavoidable periods of lower performance at early stages. Nonetheless, MABs provide higher potential short/mid-term rewards.

Dense multi-agent setups

We propose a 20x20 m^2 map where multiple agent-empowered BSS's (from $W' = 6$ to 16) are spread at random over the area. We simply force a 4 m separation between APs, and STAs being separated up to 10 m from their associated AP. As in the previous experiments, each BSS w is assigned a random initial traffic load $\ell_w \in \{20, 50, 150\}$ Mbps, $\forall w$. As for the channelization, we consider now a standard-compliant maximum bandwidth of 160 MHz, thus having 8 possible 20-MHz channels (e.g., from channels 36 to 64 in the IEEE 802.11ac/ax standards). Accordingly, the action space of a BSS w is $\mathcal{A}_w = \{p_w, b_w\} | p_w \in \{1, 2, ..., 8\}, b_w \in \{1, 2, 4, 8\}, \forall w$, thus having 32 (8 x 4) possible configurations. Notice that, in contrast to the self-contained toy dataset, we now lose the optimal baseline reference since it is unfeasible to simulate all the statuses of the different deployments. So, we compare the exploration-first and e-greedy algo-

rithms against an off-the-shelf random configuration we use as a baseline.

In this first experiment, we aim to generalize the assessment on deployments of increasing BSS density. Are the ligthweight MABs still able to learn? We generate 100 WLAN random deployments for three different numbers of BSS's $W' = W \in \{6, 10, 16\}$ and run the same MAB algorithm at every AP in the network. In particular, we simulate exploration-first, e-greedy, and a fixed random configuration, where agents are idle and do never change the initial configuration whatsoever. We run each scenario for two different random seeds to contemplate the randomness of the MAB algorithms.

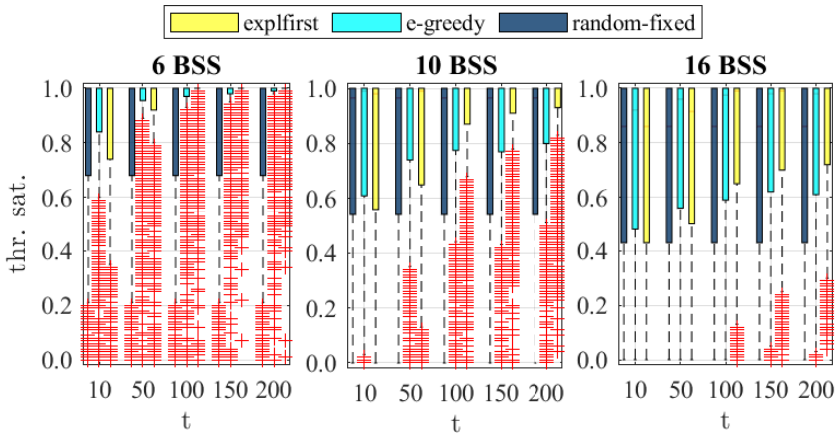


Figure 6.9: Reward distributions at different iterations.

Figure 6.9 shows the distribution of the throughput satisfaction ξ of every BSS for each algorithm and each number of deployed BSS's. Notice that each boxplot inside the subplots is filled with the reward values gathered at a given iteration (10, 50, 100, 150, and 200) in the different simulations, thus covering a vast heterogeneous set of deployments. We observe that, regardless of the node density, exploration-first and e-greedy can learn, thus clearly outperforming the static configurations. In particular, exploration-first outperforms e-greedy as the simulation progresses. However, we also notice that e-greedy is better at earlier stages of the simulation. This is due to the mandatory rule of exploration-first of exploring the full action space, no matter whether the agent is satisfied or not. If agents start from zero knowledge on the reward per action, exploration-first will take more time to *converge*. Still, once the full action space is explored, its performance is significantly higher than e-greedy.

Finding #20: Both exploration-first and e-greedy can learn even in high-density deployments, outperforming by far static configurations. Besides, while exploration-first performs worse than e-greedy at the early stages of the simulation, it reaches much better performance as the simulation progresses.

On the fairness

We observed that both exploration-first and e-greedy were able to learn for every assessed BSS density. However, does the WLAN also reach higher fairness values as the simulation

progresses? Does the greedy behavior of BSS's contribute to a better performance of the whole WLAN? We keep studying the 100 WLAN random deployments from the previous experiment.

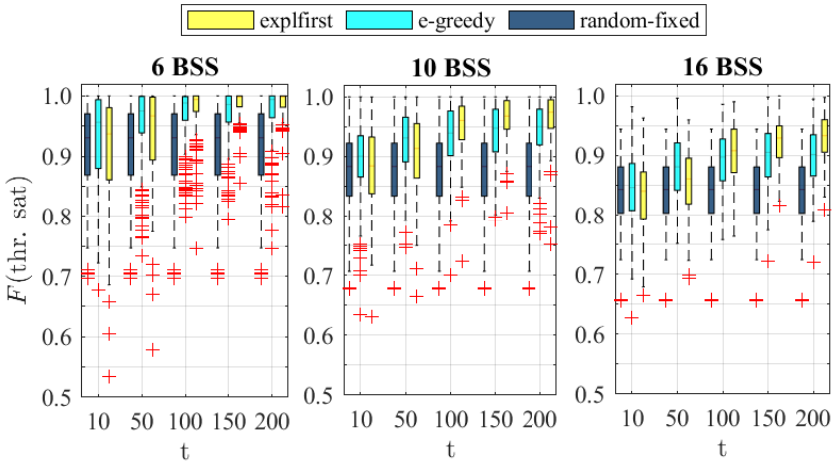


Figure 6.10: Jain's fairness distributions at different iterations.

Figure 6.10 shows the distribution of the Jain's fairness index (4.1) for the throughput satisfaction ξ. Like in Figure 6.9, each boxplot inside the subplots is filled with the reward values gathered at a given iteration in the different simulations. We observe a similar behavior between the evolution of the reward (Figure 6.9) and the evolution of the fairness (Figure 6.10): e-greedy is fairer at the early stages, whereas exploration-first provides significantly higher fairness as the simulation progresses. This is an interesting result showing that multi-agent settings like multi-channel WLANs can result in fair settings without coordination between agents. The key factor is the mutual interests forming a win-win relationship among agents. In plain, always on average terms, any pair of BSS's would mutually benefit if they do not overlap when trying to satisfy their traffic load demands. Unfortunately, we also find unfair outlier scenarios where some BSS's do not perform well due to different reasons like exposed/hidden nodes. In that kind of situation, it seems necessary to coordinate, thus changing the local greediness of the reward definition towards a collaborative one. In turn, that would require communication overheads and compatibility issues (or conflicts of interests between BSS's).

Finding #21: The WLAN's fairness tends to increase, thus reaching global *stable* configurations where most (or all) the BSS's perform similarly. In particular, exploration-first tends to outperform e-greedy, leading to more fair setups as simulations progress.

Adaptation to varying traffic load

In the previous experiments on random deployments, we have observed the learning capabilities of exploration-first and e-greedy under unvarying traffic profiles. In this experiment, we assess how these MABs respond to varying traffic loads, potentially affecting each action's reward. We provide a direct comparison between exploration-first and e-greedy by assessing each agent's reward at every iteration in a 1-vs-1 fashion. In particular, we compute the evolution of

the probability of each MAB outperforming the other. That is, for every agent, we count the number of deployments per iteration where one MAB provided the highest instant reward $r_{w,t}$.

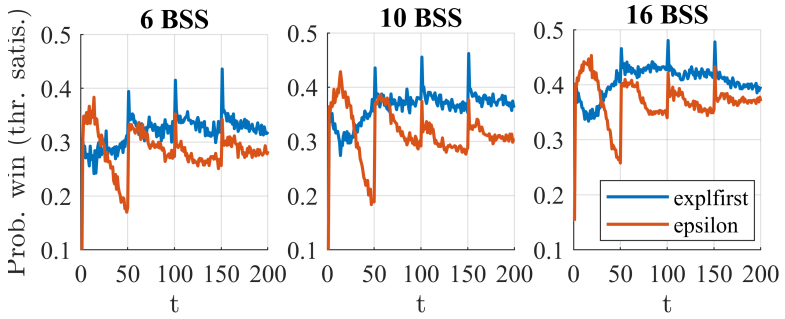


Figure 6.11: Run-chart of the probability of providing the highest instant reward.

Figure 6.11 shows the probability of each MAB outperforming the other in terms of throughput satisfaction. Notice that the curves do not sum 1 in any of the iterations because of the cases where both MABs provided the same reward, most often corresponding to maximum satisfaction, $\xi = 1$. We corroborate that exploration-first is only outperformed by e-greedy in the first full exploration phase. Afterward, it performs better even after the changes in the traffic loads. So, it is preferable for every node density we studied to use exploration-first for mid/long term reward and quicker adaptation to environment changes.

Finding #22: After *paying the price* of performing the first full exploration phase, exploration-first outperforms e-greedy both in terms of mid/long-term reward and fast adaptation in front of environment alternations. We conclude that in situations where there is room to spend a *reasonable* time learning, exploration-first should be the MAB of choice.

Lightweighting the action space

We observed that exploration-first and e-greedy could learn and raise the BSS's performance consequently. We now ask if it is possible to increase the learning convergence speed and reach good performance levels sooner. To that aim, we propose lightweighting the action space of each agent by reducing the possible values the maximum bandwidth attribute may take: from $b \in \{1, 2, 4, 8\}$ to $b \in \{1, 8\}$, i.e., to allow only single-channel ($b = 1$) or to remove any bandwidth restriction ($b = 8$). So, we expect to raise the convergence at the cost of renouncing to potentially better configurations.

We run the MABs in the same deployments of the two experiments before, but we now reduce the maximum bandwidth space b to two values: 1 (20 MHz) and 8 (160 MHz). Notice that the resulting action space has 16 actions rather than 32, i.e., we halve the action space going from $|\mathcal{A}_w| = 32$ to $|\mathcal{A}_w| = 16, \forall w$.

Figure 6.12 shows the mean and standard distribution of the throughput satisfaction when applying exploration-first with the original and the reduced action space. We observe that both action spaces have a similar convergence rate for scenarios with moderate density (6 BSS's). However, the best performance is only reached with the original action space (i.e., keeping all

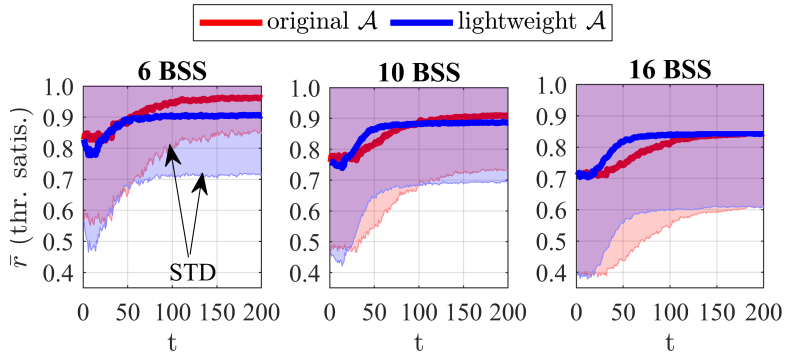


Figure 6.12: Comparison between the original and the lightweight action space when using exploration-first. The curves represent the mean value of the rewards and the shaded areas correspond to the standard deviation (STD), respectively.

the configuration options in the max. bandwidth). However, as the node density increases, it turns out that having a reduced action space allows significantly faster convergence while reaching a similar performance than the original action space in the long-term.

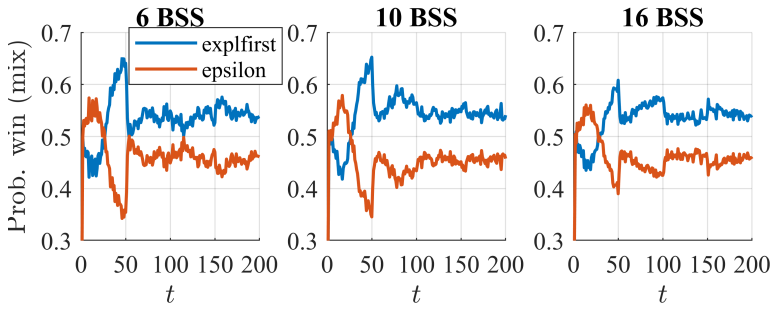
So, is there any rule of thumb for reducing action spaces? We suggest limiting the domain of the less critical attributes, always bearing in mind the trade-off between fast convergence and maximum reachable reward. However, such a task is not straightforward and requires expertise from the RL designer. In our case, we have proposed reducing the maximum bandwidth's possible values $\{b\}$ and keeping all the values for the primary channel $\{p\}$ since we know the primary channel is critical in CSMA/CA networks.

Finding #23: It is preferable to lightweight the action space in challenging (dense and highly loaded) scenarios since it will help increase the learning speed. The downside is to renounce to a potentially better long-term performance. So, we suggest reducing the less critical attributes from the action space whenever possible.

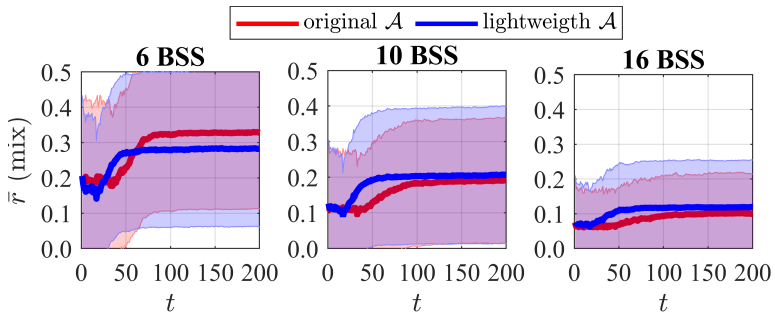
Mixed reward: throughput and delay

Up to now, we have considered the reward to be exclusively dependant on the throughput satisfaction, i.e., on the throughput and the traffic load (6.7). In this experiment, we aim to answer whether another reward definition, including the delay, would vary our MAB analysis in any regard. We repeat the experiments on the action space lightweighting (§6.5.3) and the direct comparison between exploration-first and e-greedy (§6.5.3), this time using the mixed throughput-delay reward definition (6.10).

Figure 6.13a and Figure 6.13b show the probability of each MAB outperforming the other in terms of the mixed reward, and the mean and standard distribution of the throughput satisfaction when applying exploration-first with the original and the reduced action space respectively. That is, such figures are complementary to Figure 6.11 and Figure 6.12, respectively. We observe similar patterns in both pairs of figures, only varying the reward scale, which is expected since the mixed reward is by definition lower than the throughput satisfaction. Therefore, we conclude that exploration-first also outperforms e-greedy when introducing other performance



(a) Run-chart of the probability of providing the highest instant mixed reward (complementary to Figure 6.11).



(b) Original and lightweight action space when using exploration-first for the mixed reward (complementary to Figure 6.12).

Figure 6.13: Mixed throughput-delay reward assessment.

parameters like the delay in the reward definition R.

Finding #24: Exploration-first keeps learning and outperforming e-greedy for the mixed throughput-delay reward. We derive that exploration-first is convenient for the Wi-Fi deployments considered in this thesis regardless of the performance parameters included in the reward definition.

Chapter summary

In this chapter, we envisioned the need for ML to cope with the spectrum management problem in high-density and dynamic settings. We justified why a stateless variation of RL in the form of MABs suits the task of adapting fast in uncoordinated deployments. Results from extensive experiments showed the responsiveness of MABs in front of other RL approaches.

Chapter 7

CONCLUDING REMARKS

7.1 Summary

In this thesis, we shed light on the behavior of channel allocation and channel bonding in WLANs, from its essential characteristics to benchmarking its performance on a variety of dense deployments with different traffic load requirements. To conduct such an in-depth analysis, we have developed different enablers: a CTMN model for channel bonding in spatially distributed WLANs, the Komondor wireless network simulator, and the WACA spectrum analyzer. We used the latter to generate a unique dataset simultaneously capturing all the channels in the 5 GHz Wi-Fi channelization. The dataset contains multiple locations of interest, including a crowded football match in the Camp Nou stadium. We also covered heuristic-based approaches and proposed DyWi, an algorithm for online selection of the primary channel of a dynamic channel bonding BSS. Finally, we explored the use of ML to better cope with the joint spectrum management problem and conclude that RL is the approach to follow due to its trial-and-error black-box conception that disregards the implicit nature or model behind multi-agent WLANs. In this regard, we call into question whether the use of complex RL algorithms really helps the quest and derive that stateless algorithms, especially in the form of lightweight MABs, are an efficient solution for rapid adaptation avoiding extensive or meaningless states.

We believe the research carried in this thesis will contribute to the design of novel spectrum management techniques for next-generation WLANs. We expect that our findings on channel bonding will help overcome the main challenges imposed by spectrum sharing. Besides, we hope that our reasoning on why stateless RL algorithms in the form of lightweight MABs are an efficient solution for rapid deployment will be useful as well for other techniques like scheduling or transmission power control.

7.2 Future work

There are different research topics to continue exploring concerning spectrum management in WLANs. First and foremost, we envision the use of RL as a key enabler for next-generation WLANs. Thus, we will dig deeper into the analysis of multi-agent settings by considering partial and full coordination through other learning architectures that may impose communication overheads but also a potentially holistic knowledge of the environment. In this settings, where we count with relatively large periods for conducting the learning process, we do believe that

the use of more complex state-full RL formulations like DRL can be useful to reach a more profound judgment on the WLAN deployment by identifying complex long-terms patterns. To that aim, the trade-off between meaningful and condensed states remains an essential venue for research.

As for the stateless RL formulation, we believe that the presented lightweight MABs can be improved in several ways. A mechanism to adapt the iteration duration to the level of dynamism in the environment is worth to study. In steady environments (e.g., single-agent), short iterations are enough to estimate action rewards. Instead, under high dynamism, iterations should be much larger or not used at all. In the latter case, the use of heuristic-based approaches could help to identify *good enough* actions and pause the MAB exploration. So, designing a backoff mechanism that triggers heuristics or RL instructions depending on the reliability of the estimated action rewards seems promising. Further, the study of asynchronous agents remains unexplored. While we expect similar behavior to the studied synchronous settings, an in-depth study on the matter would aid detecting the main differences and potential pitfalls.

Finally, we observe that the inter-channel correlation detected in the on-the-field traces played a critical role in channel bonding performance. Consequently, simple models relying just on occupancy severely underestimate the available gains for channel bonding. Thus, we envision the design of a trace-driven model for spectrum activity that takes into account inter-channel correlation. We believe such a model will help in building more efficient new spectrum management algorithms.

Funding

This work has been partially supported by the following grants:

- FI grant from the Generalitat de Catalunya.
- Cisco University Research Program fund, a corporate advised fund of Silicon Valley Community Foundation,Grant number: Project CG No. 890107, Towards Deterministic Channel Access in High-Density WLANs.
- Spanish Ministry of Economy and Competitiveness under the Maria de Maeztu Units of Excellence Programme (MDM-2015-0502).
- WINDMAL PGC2018-099959-B-I00 (MCIU/AEI/FEDER,UE).
- Catalan Government under grant SGR-2017-1188.
- We appreciate the generosity of Futbol Club Barcelona for letting us take measurements in the Camp Nou stadium.

Bibliography

[1] Devices must be professionally installed when operating in the 5470 – 5725 ghz band. Publication Number: 443999 Rule Parts: 15E.

[2] Rice University Wireless Open Access Research Platform. `http://warpproject.org`, last accessed January 30, 2020.

[3] Status of project ieee 802.11ax high efficiency wlan (hew). `http://www.ieee802.org/11/Reports/tgax_update.htm` (accessed July 2017).

[4] IEEE 802.11n enhancements for Higher Throughput, 2009.

[5] IEEE 802.11ac Very High Throughput., 2013.

[6] IEEE 802.11be Extremely High Throughput, 802.11-18/1231r6, 2019.

[7] IEEE P802.11ax/D4.0 High Efficiency WLAN, 2019.

[8] List of WLAN channels. Wikipedia., 2020.

[9] M. Achanta. Method and apparatus for least congested channel scan for wireless access points, Oct. 5 2004. US Patent App. 10/959,446.

[10] T. Adame, M. Carrascosa, and B. Bellalta. The TMB path loss model for 5 GHz indoor WiFi scenarios: On the empirical relationship between RSSI, MCS, and spatial streams. In *2019 Wireless Days (WD)*, pages 1–8. IEEE, 2019.

[11] A. Ahmed, G. Amel, S. Anis, and M. Abdellatif. Resource allocation for multi-user cognitive radio systems using multi-agent q-learning. *Procedia Computer Science*, 10:46–53, 2012.

[12] R. Akl and A. Arepally. Dynamic channel assignment in IEEE 802.11 networks. In *Portable Information Devices, 2007. PORTABLE07. IEEE International Conference on*, pages 1–5. IEEE, 2007.

[13] H. Al-Rizzo, M. Haidar, R. Akl, and Y. Chan. Enhanced channel assignment and load distribution in IEEE 802.11 WLANs. In *Signal Processing and Communications, 2007. ICSPC 2007. IEEE International Conference on*, pages 768–771. IEEE, 2007.

[14] R. Ali, Y. A. Qadri, Y. B. Zikria, T. Umer, B.-S. Kim, and S. W. Kim. Q-learning-enabled channel access in next-generation dense wireless networks for iot-based ehealth systems. *EURASIP Journal on Wireless Communications and Networking*, 2019(1):1–12, 2019.

[15] R. Ali, N. Shahin, Y. B. Zikria, B.-S. Kim, and S. W. Kim. Deep reinforcement learning paradigm for performance optimization of channel observation–based mac protocols in dense wlans. *IEEE Access*, 7:3500–3511, 2018.

[16] M. A. Alsheikh, S. Lin, D. Niyato, and H.-P. Tan. Machine learning in wireless sensor networks: Algorithms, strategies, and applications. *IEEE Communications Surveys & Tutorials*, 16(4):1996–2018, 2014.

[17] N. Anand, E. Aryafar, and E. W. Knightly. WARPlab: a flexible framework for rapid physical layer design. In *Proceedings of the 2010 ACM workshop on Wireless of the students, by the students, for the students*, pages 53–56. ACM, 2010.

[18] M. Y. Arslan, K. Pelechrinis, I. Broustis, S. V. Krishnamurthy, S. Addepalli, and K. Papagiannaki. Auto-configuration of 802.11n WLANs. In *Proceedings of the 6th International COnference*, pages 1–12, 2010.

[19] M. Y. Arslan, K. Pelechrinis, I. Broustis, S. Singh, S. V. Krishnamurthy, S. Addepalli, and K. Papagiannaki. ACORN: An auto-configuration framework for 802.11n WLANs. *IEEE/ACM Transactions on Networking (TON)*, 21(3):896–909, 2013.

[20] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

[21] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Proceedings of IEEE 36th Annual Foundations of Computer Science*, pages 322–331. IEEE, 1995.

[22] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002.

[23] A. Baid and D. Raychaudhuri. Understanding channel selection dynamics in dense Wi-Fi networks. *IEEE Communications Magazine*, 53(1):110–117, 2015.

[24] S. Barrachina-Muñoz, B. Bellalta, and E. Knightly. Wi-Fi all channel analyzer. In *The 14th ACM Workshop on Wireless Network Testbeds, Experimental evaluation & CHaracterization (WiNTECH). Runner-up best-paper award. Runner Up, Best Paper Award*. ACM, 2020.

[25] S. Barrachina-Muñoz, B. Bellalta, and E. Knightly. Wi-Fi channel bonding: an all-channel system and experimental study from urban hotspots to a sold-out stadium. *Submitted to IEEE Transactions on Networking (under major revision)*, 2020.

[26] S. Barrachina-Muñoz, F. Wilhelmi, and B. Bellalta. Dynamic channel bonding in spatially distributed high-density WLANs. *IEEE Transactions on Mobile Computing*, 19(4):821–835, 2019.

[27] S. Barrachina-Muñoz, F. Wilhelmi, and B. Bellalta. Online primary channel selection for dynamic channel bonding in high-density WLANs. *IEEE Wireless Communications Letters*, 9(2):258–262, 2019.

[28] S. Barrachina-Muñoz, F. Wilhelmi, and B. Bellalta. To overlap or not to overlap: enabling channel bonding in high-density WLANs. *Computer Networks*, 152:40–53, 2019.

[29] S. Barrachina-Munoz, F. Wilhelmi, I. Selinis, and B. Bellalta. Komondor: A wireless network simulator for next-generation high-density wlans. In *2019 Wireless Days (WD)*, pages 1–8. IEEE, 2019.

[30] B. Bellalta. Throughput Analysis in High Density WLANs. *IEEE Communications Letters*, 21(3):592–595, 2017.

[31] B. Bellalta, A. Checco, A. Zocca, and J. Barcelo. On the interactions between multiple overlapping wlans using channel bonding. *IEEE Transactions on Vehicular Technology*, 65(2):796–812, 2015.

[32] B. Bellalta, A. Checco, A. Zocca, and J. Barcelo. On the interactions between multiple overlapping WLANs using channel bonding. *IEEE Transactions on Vehicular Technology*, 65(2):796–812, 2016.

[33] B. Bellalta, A. Faridi, J. Barcelo, A. Checco, and P. Chatzimisios. Channel bonding in short-range WLANs. In *European Wireless 2014; 20th European Wireless Conference*, pages 1–7. VDE, 2014.

[34] B. Bellalta, A. Zocca, C. Cano, A. Checco, J. Barcelo, and A. Vinel. Throughput analysis in CSMA/CA networks using continuous time Markov networks: a tutorial. In *Wireless Networking for Moving Objects*, pages 115–133. Springer, 2014.

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

[36] S. Biswas, J. Bicket, E. Wong, R. Musaloiu-e, A. Bhartia, and D. Aguayo. Large-scale measurements of wireless network behavior. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, pages 153–165, 2015.

[37] M. Bkassiny, Y. Li, and S. K. Jayaweera. A survey on machine-learning techniques in cognitive radios. *IEEE Communications Surveys & Tutorials*, 15(3):1136–1159, 2012.

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

[39] D. Brélaz. New methods to color the vertices of a graph. *Communications of the ACM*, 22(4):251–256, 1979.

[40] S. H. R. Bukhari, M. H. Rehmani, and S. Siraj. A survey of channel bonding for wireless networks and guidelines of channel bonding for futuristic cognitive radio sensor networks. *IEEE Communications Surveys & Tutorials*, 18(2):924–948, 2016.

[41] G. Buttazzo, C. Di Franco, and M. Marinoni. Design and analysis of target-sensitive real-time systems. *Software: Practice and Experience*, 46(9):1181–1200, 2016.

[42] X. Chang. Network simulations with opnet. In *WSC'99. 1999 Winter Simulation Conference Proceedings.'Simulation-A Bridge to the Future'(Cat. No. 99CH37038)*, volume 1, pages 307–314. IEEE, 1999.

[43] G. Chen and B. K. Szymanski. Reusing simulation components: cost: a component-oriented discrete event simulator. In *Proceedings of the 34th conference on Winter simulation (WSC): exploring new frontiers*, pages 776–782. Winter Simulation Conference, 2002.

[44] J. K. Chen, G. De Veciana, and T. S. Rappaport. Improved measurement-based frequency allocation algorithms for wireless networks. In *Global Telecommunications Conference, 2007. GLOBECOM'07. IEEE*, pages 4790–4795. IEEE, 2007.

[45] Y. Chen, J. Zhao, Q. Zhu, and Y. Liu. Research on unlicensed spectrum access mechanism based on reinforcement learning in laa/wlan coexisting network. *Wireless Networks*, 26(3):1643–1651, 2020.

[46] S. Chieochan, E. Hossain, and J. Diamond. Channel assignment schemes for infrastructure-based 802.11 WLANs: A survey. *IEEE Communications Surveys and Tutorials*, 12(1), 2010.

[47] L. Cigler and B. Faltings. Decentralized anti-coordination through multi-agent learning. *Journal of Artificial Intelligence Research*, 47:441–473, 2013.

[48] B. C. Csáji and L. Monostori. Value function based reinforcement learning in changing markovian environments. *Journal of Machine Learning Research*, 9(Aug):1679–1709, 2008.

[49] W. Dai, Y. Gai, and B. Krishnamachari. Online learning for multi-channel opportunistic access over unknown markovian channels. In *2014 Eleventh Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pages 64–71. IEEE, 2014.

[50] S. W. Day and A. Wagstaff. Activity in the wifi bands - an objective approach to estimation. In *ARMMS Conference*. Mass, 2014.

[51] S. De Bast, R. Torrea-Duran, A. Chiumento, S. Pollin, and H. Gacanin. Deep reinforcement learning for dynamic network slicing in ieee 802.11 networks. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 264–269. IEEE, 2019.

[52] L. Deek, E. Garcia-Villegas, E. Belding, S.-J. Lee, and K. Almeroth. The impact of channel bonding on 802.11n network management. In *Proceedings of the Seventh COnference on emerging Networking EXperiments and Technologies*, pages 1–12, 2011.

[53] L. Deek, E. Garcia-Villegas, E. Belding, S.-J. Lee, and K. Almeroth. Joint rate and channel width adaptation for 802.11 MIMO wireless networks. In *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2013 10th Annual IEEE Communications Society Conference on*, pages 167–175. IEEE, 2013.

[54] L. Deek, E. Garcia-Villegas, E. Belding, S.-J. Lee, and K. Almeroth. Intelligent channel bonding in 802.11n WLANs. *IEEE Transactions on Mobile Computing*, 13(6):1242–1255, 2014.

[55] A. Eisenblatter, H.-F. Geerdes, and I. Siomina. Integrated access point placement and channel assignment for wireless LANs in an indoor office environment. In *World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. IEEE International Symposium on a*, pages 1–10. IEEE, 2007.

[56] A. Faridi, B. Bellalta, and A. Checco. Analysis of dynamic channel bonding in dense networks of wlans. *IEEE Transactions on Mobile Computing*, 16(8):2118–2131, 2016.

[57] L. Farzinvash and M. Kargahi. A scheduling algorithm for execution-instant sensitive real-time systems. In *2009 15th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, pages 511–518. IEEE, 2009.

[58] Y. Gai and B. Krishnamachari. Distributed stochastic online learning policies for opportunistic spectrum access. *IEEE Transactions on Signal Processing*, 62(23):6184–6193, 2014.

[59] M. X. Gong, B. Hart, L. Xia, and R. Want. Channel bounding and mac protection mechanisms for 802.11ac. In *2011 IEEE Global Telecommunications Conference-GLOBECOM 2011*, pages 1–5. IEEE, 2011.

[60] M. Haidar, R. Akl, H. Al-Rizzo, and Y. Chan. Channel assignment and load distribution in a power-managed wlan. In *Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007. IEEE 18th International Symposium on*, pages 1–5. IEEE, 2007.

[61] S. A. Hanna. A 3-state hypothesis test model for cognitive radio systems. In *2014 IEEE International Symposium on Dynamic Spectrum Access Networks (DYSPAN)*, pages 291–302. IEEE, 2014.

[62] S. A. Hanna and J. Sydor. Distributed sensing of spectrum occupancy and interference in outdoor 2.4 GHz Wi-Fi networks. In *2012 IEEE Global Communications Conference (GLOBECOM)*, pages 1453–1459. IEEE, 2012.

[63] J. Herzen, R. Merz, and P. Thiran. Distributed spectrum assignment for home WLANs. In *2013 Proceedings Ieee Infocom*, pages 1573–1581. IEEE, 2013.

[64] A. Hills. Large-scale wireless LAN design. *IEEE Communications Magazine*, 39(11):98–107, 2001.

[65] M. Höyhtyä, A. Mämmelä, M. Eskola, M. Matinmikko, J. Kalliovaara, J. Ojaniemi, J. Suutala, R. Ekman, R. Bacchus, and D. Roberson. Spectrum occupancy measurements: A survey and use of interference maps. *IEEE Communications Surveys & Tutorials*, 18(4):2386–2414, 2016.

[66] Y. Hu, M. Chen, Z. Yang, M. Chen, and G. Jia. Optimization of resource allocation in multi-cell ofdm systems: A distributed reinforcement learning approach. In *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 1–6. IEEE.

[67] P. Huang, X. Yang, and L. Xiao. Dynamic channel bonding: enabling flexible spectrum aggregation. *IEEE Transactions on Mobile Computing*, 15(12):3042–3056, 2016.

[68] T. Issariyakul and E. Hossain. Introduction to Network Simulator 2 (NS2). In *Introduction to Network Simulator NS2*, pages 21–40. Springer, 2012.

[69] S.-J. Jang, C.-H. Han, K.-E. Lee, and S.-J. Yoo. Reinforcement learning-based dynamic band and channel selection in cognitive radio ad-hoc networks. *EURASIP Journal on Wireless Communications and Networking*, 2019(1):131, 2019.

[70] C. Jiang, H. Zhang, Y. Ren, Z. Han, K.-C. Chen, and L. Hanzo. Machine learning paradigms for next-generation wireless networks. *IEEE Wireless Communications*, 24(2):98–105, 2016.

[71] L. Jiao, V. Pla, and F. Y. Li. Analysis on channel bonding/aggregation for multi-channel cognitive radio networks. In *Wireless Conference (EW), 2010 European*, pages 468–474. IEEE, 2010.

[72] S. Joshi, P. Pawelczak, D. Cabric, and J. Villasenor. When channel bonding is beneficial for opportunistic spectrum access networks. *IEEE Transactions on Wireless Communications*, 11(11):3942–3956, 2012.

[73] C. Kai, Y. Liang, T. Huang, and X. Chen. A channel allocation algorithm to maximize aggregate throughputs in DCB WLANs. *arXiv preprint arXiv:1703.03909*, 2017.

[74] R. Karmakar, S. Chattopadhyay, and S. Chakraborty. Smartla: Reinforcement learning-based link adaptation for high throughput wireless access networks. *Computer Communications*, 110:1–25, 2017.

[75] F. Kelly. *Reversibility and stochastic networks*. Cambridge University Press, 2011.

[76] S. Keranidis, K. Chounos, T. Korakis, I. Koutsopoulos, and L. Tassiulas. Enabling AGILE spectrum adaptation in commercial 802.11 WLAN deployments. In *Proceedings of the 20th annual international conference on Mobile computing and networking*, pages 295–298. ACM, 2014.

[77] S. Khairy, M. Han, L. X. Cai, Y. Cheng, and Z. Han. A renewal theory based analytical model for multi-channel random access in IEEE 802.11ac/ax. *IEEE Transactions on Mobile Computing*, 2018.

[78] M.-S. Kim, T. Ropitault, S. Lee, and N. Golmie. A throughput study for channel bonding in ieee 802.11ac networks. *IEEE Communications Letters*, 21(12):2682–2685, 2017.

[79] P. V. Klaine, M. A. Imran, O. Onireti, and R. D. Souza. A survey of machine learning techniques applied to self-organizing cellular networks. *IEEE Communications Surveys & Tutorials*, 19(4):2392–2431, 2017.

[80] V. Kone, L. Yang, X. Yang, B. Y. Zhao, and H. Zheng. On the feasibility of effective opportunistic spectrum access. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, pages 151–164, 2010.

[81] V. Kone, L. Yang, X. Yang, B. Y. Zhao, and H. Zheng. The effectiveness of opportunistic spectrum access: A measurement study. *IEEE/ACM Transactions on Networking*, 20(6):2005–2016, 2012.

[82] T. L. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

[83] R. Laufer and L. Kleinrock. On the capacity of wireless CSMA/CA multihop networks. In *2013 Proceedings IEEE INFOCOM*, pages 1312–1320. IEEE, 2013.

[84] J. Lee, W. Kim, S.-J. Lee, D. Jo, J. Ryu, T. Kwon, and Y. Choi. An experimental study on the capture effect in 802.11a networks. In *Proceedings of the second ACM international workshop on Wireless network testbeds, experimental evaluation and characterization*, pages 19–26, 2007.

[85] Y. Lee, K. Kim, and Y. Choi. Optimization of AP placement and channel assignment in wireless LANs. In *Local Computer Networks, 2002. Proceedings. LCN 2002. 27th Annual IEEE Conference on*, pages 831–836. IEEE, 2002.

[86] K. K. Leung and B.-J. Kim. Frequency assignment for IEEE 802.11 wireless networks. In *Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th*, volume 3, pages 1422–1426. IEEE, 2003.

[87] F. Li, D. Yu, H. Yang, J. Yu, H. Karl, and X. Cheng. Multi-armed-bandit-based spectrum scheduling algorithms in wireless networks: A survey. *IEEE Wireless Communications*, 27(1):24–30, 2020.

[88] S. C. Liew, C. H. Kai, H. C. Leung, and P. Wong. Back-of-the-envelope computation of throughput distributions in CSMA wireless networks. *IEEE Transactions on Mobile Computing*, 9(9):1319–1331, 2010.

[89] X. Ling and K. L. Yeung. Joint access point placement and channel assignment for 802.11 wireless LANs. *IEEE Transactions on Wireless Communications*, 5(10):2705–2711, 2006.

[90] K. Liu and Q. Zhao. Indexability of restless bandit problems and optimality of whittle index for dynamic multichannel access. *IEEE Transactions on Information Theory*, 56(11):5547–5567, 2010.

[91] L. Liu, R. Downe, and J. Reid. Multi-armed bandit strategies for non-stationary reward distributions and delayed feedback processes. In *CORS conference*, 2019.

[92] M. López-Benítez and F. Casadevall. Modeling and simulation of time-correlation properties of spectrum use in cognitive radio. In *2011 6th International ICST Conference on Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM)*, pages 326–330, 2011.

[93] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, and D. I. Kim. Applications of deep reinforcement learning in communications and networking: A survey. *IEEE Communications Surveys & Tutorials*, 21(4):3133–3174, 2019.

[94] P. Mahonen, J. Riihijarvi, and M. Petrova. Automatic channel allocation for small wireless local area networks using graph colouring algorithm approach. In *Personal, Indoor and Mobile Radio Communications, 2004. PIMRC 2004. 15th IEEE International Symposium on*, volume 1, pages 536–539. IEEE, 2004.

[95] A. Mishra, S. Banerjee, and W. Arbaugh. Weighted coloring based channel assignment for WLANs. *ACM SIGMOBILE Mobile Computing and Communications Review*, 9(3):19–31, 2005.

[96] A. Mishra, V. Brik, S. Banerjee, A. Srinivasan, and W. A. Arbaugh. A client-driven approach for channel management in wireless LANs. In *Infocom*, 2006.

[97] A. Mishra, V. Shrivastava, D. Agrawal, S. Banerjee, and S. Ganguly. Distributed channel management in uncoordinated wireless environments. In *Proceedings of the 12th annual international conference on Mobile computing and networking*, pages 170–181. ACM, 2006.

[98] F. B. Mismar, B. L. Evans, and A. Alkhateeb. Deep reinforcement learning for 5g networks: Joint beamforming, power control, and interference coordination. *IEEE Transactions on Communications*, 68(3):1581–1592, 2019.

[99] T. A. Myrvoll and J. E. Håkegård. Dynamic spectrum access in realistic environments using reinforcement learning. In *2012 International Symposium on Communications and Information Technologies (ISCIT)*, pages 465–470. IEEE, 2012.

[100] A. Nabil, M. J. Abdel-Rahman, and A. B. MacKenzie. Adaptive channel bonding in wireless lans under demand uncertainty. In *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 1–7. IEEE, 2017.

[101] K. Nakashima, S. Kamiya, K. Ohtsu, K. Yamamoto, T. Nishio, and M. Morikura. Deep reinforcement learning-based channel allocation for wireless lans with graph convolutional networks. *IEEE Access*, 8:31823–31834, 2020.

[102] O. Naparstek and K. Cohen. Deep multi-user reinforcement learning for distributed dynamic spectrum access. *IEEE Transactions on Wireless Communications*, 18(1):310–323, 2018.

[103] B. Nardelli and E. W. Knightly. Closed-form throughput expressions for CSMA networks with collisions and hidden terminals. In *INFOCOM, 2012 Proceedings IEEE*, pages 2309–2317. IEEE, 2012.

[104] M. Park. IEEE 802.11ac: Dynamic bandwidth channel access. In *Communications (ICC), 2011 IEEE International Conference on*, pages 1–5. IEEE, 2011.

[105] K. Pelechrinis, T. Salonidis, H. Lundgren, and N. Vaidya. Experimental characterization of 802.11 n link quality at high rates. In *Proceedings of the fifth ACM international workshop on Wireless network testbeds, experimental evaluation and characterization*, pages 39–46. ACM, 2010.

[106] H. Qi, H. Huang, Z. Hu, X. Wen, and Z. Lu. On-demand channel bonding in heterogeneous wlans: A multi-agent deep reinforcement learning approach. *Sensors*, 20(10):2789, 2020.

[107] M. Rademacher, K. Jonas, and M. Kretschmer. Quantifying the spectrum occupancy in an outdoor 5 GHz WiFi network with directional antennas. In *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6. IEEE, 2018.

[108] A. L. Ramaboli, O. E. Falowo, and A. H. Chan. Bandwidth aggregation in heterogeneous wireless networks: A survey of current approaches and issues. *Journal of Network and Computer Applications*, 35(6):1674–1690, 2012.

[109] A. K. Rathi and A. J. Santiago. The new NETSIM simulation program. *Traffic engineering & control*, 31(5), 1990.

[110] J. Riihijarvi, M. Petrova, and P. Mahonen. Frequency allocation for WLANs using graph colouring techniques. In *Wireless On-demand Network Systems and Services, 2005. WONS 2005. Second Annual Conference on*, pages 216–222. IEEE, 2005.

[111] J. Riihijarvi, M. Petrova, P. Mahonen, and J. D. A. Barbosa. Performance evaluation of automatic channel assignment mechanism for IEEE 802.11 based on graph colouring. In *Personal, Indoor and Mobile Radio Communications, 2006 IEEE 17th International Symposium on*, pages 1–5. IEEE, 2006.

[112] G. F. Riley and T. R. Henderson. The ns-3 network simulator. In *Modeling and tools for network simulation*, pages 15–34. Springer, 2010.

[113] F. Salahdine and H. El Ghazi. A real time spectrum scanning technique based on compressive sensing for cognitive radio networks. In *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*, pages 506–511. IEEE, 2017.

[114] H. A. B. Salameh, M. Krunz, and D. Manzi. Spectrum bonding and aggregation with guard-band awareness in cognitive radio networks. *IEEE Transactions on Mobile Computing*, 13(3):569–581, 2014.

[115] Y. Seldin, C. Szepesvári, P. Auer, and Y. Abbasi-Yadkori. Evaluation and analysis of the performance of the exp3 algorithm in stochastic environments. In *European Workshop on Reinforcement Learning*, pages 103–116, 2013.

[116] S. Subramaniam, H. Reyes, and N. Kaabouch. Spectrum occupancy measurement: An autocorrelation based scanning technique using USRP. In *2015 IEEE 16th Annual Wireless and Microwave Technology Conference (WAMICON)*, pages 1–5. IEEE, 2015.

[117] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[118] T. Taher, R. Attard, A. Riaz, D. Roberson, J. Taylor, K. Zdunek, J. Hallio, R. Ekman, J. Paavola, J. Suutala, et al. Global spectrum observatory network setup and initial findings. In *2014 9th International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM)*, pages 79–88. IEEE, 2014.

[119] S. Takeuchi, M. Hasegawa, K. Kanno, A. Uchida, N. Chauvet, and M. Naruse. Dynamic channel selection in wireless communications via a multi-armed bandit algorithm using laser chaos time series. *Scientific reports*, 10(1):1–7, 2020.

[120] A. Varga and R. Hornig. An overview of the OMNeT++ simulation environment. In *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*, page 60. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008.

[121] E. G. Villegas, R. V. Ferré, and J. Paradells. Frequency assignments in IEEE 802.11 WLANs with efficient spectrum sharing. *Wireless Communications and Mobile Computing*, 9(8):1125–1140, 2009.

[122] S. Wang, H. Liu, P. H. Gomes, and B. Krishnamachari. Deep reinforcement learning for dynamic multichannel access in wireless networks. *IEEE Transactions on Cognitive Communications and Networking*, 4(2):257–265, 2018.

[123] W. Wang, J. Gao, and Y. Bai. Non-contiguous channel bonding via NC-OFDM for using TV white spaces. In *Consumer Electronics, Communications and Networks (CECNet), 2013 3rd International Conference on*, pages 427–430. IEEE, 2013.

[124] W. Wang, A. Kwasinski, D. Niyato, and Z. Han. A survey on applications of model-free strategy learning in cognitive wireless networks. *IEEE Communications Surveys & Tutorials*, 18(3):1717–1757, 2016.

[125] C. J. C. H. Watkins. Learning from delayed rewards. 1989.

[126] M. Wellens, J. RiihijäRvi, and P. MäHöNen. Empirical time and frequency domain models of spectrum use. *Physical Communication*, 2(1-2):10–32, 2009.

[127] P. Wertz, M. Sauter, F. Landstorfer, G. Wolfle, and R. Hoppe. Automatic optimization algorithms for the planning of wireless local area networks. In *Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th*, volume 4, pages 3010–3014. IEEE, 2004.

[128] F. Wilhelmi, S. Barrachina-Muñoz, B. Bellalta, C. Cano, A. Jonsson, and G. Neu. Potential and pitfalls of multi-armed bandits for decentralized spatial reuse in wlans. *Journal of Network and Computer Applications*, 127:26–42, 2019.

[129] F. Wilhelmi, S. Barrachina-Muñoz, B. Bellalta, C. Cano, A. Jonsson, and V. Ram. A flexible machine-learning-aware architecture for future wlans. *IEEE Communications Magazine*, 58(3):25–31, 2020.

[130] F. Wilhelmi, B. Bellalta, C. Cano, and A. Jonsson. Implications of decentralized q-learning resource allocation in wireless networks. In *2017 ieee 28th annual international symposium on personal, indoor, and mobile radio communications (pimrc)*, pages 1–5. IEEE, 2017.

[131] F. Wilhelmi, C. Cano, G. Neu, B. Bellalta, A. Jonsson, and S. Barrachina-Muñoz. Collaborative spatial reuse in wireless networks via selfish multi-armed bandits. *Ad Hoc Networks*, 88:129–141, 2019.

[132] F. Wilhelmi, M. Carrascosa, C. Cano, A. Jonsson, V. Ram, and B. Bellalta. Usage of network simulators in machine-learning-assisted 5g/6g networks. *arXiv preprint arXiv:2005.08281*, 2020.

[133] W. Wydmański and S. Szott. Contention window optimization in ieee 802.11 ax networks with deep reinforcement learning. *arXiv preprint arXiv:2003.01492*, 2020.

[134] Y. Xu, J. Wang, Q. Wu, A. Anpalagan, and Y.-D. Yao. Opportunistic spectrum access in unknown dynamic environment: A game-theoretic stochastic learning solution. *IEEE transactions on wireless communications*, 11(4):1380–1391, 2012.

[135] M. Yazid and A. Ksentini. Stochastic modeling of the static and dynamic multichannel access methods enabling 40/80/160 MHz channel bonding in the VHT WLANs. *IEEE Communications Letters*, 23(8):1437–1440, 2019.

[136] S. Yin, D. Chen, Q. Zhang, M. Liu, and S. Li. Mining spectrum usage data: a large-scale spectrum measurement study. *IEEE Transactions on Mobile Computing*, 11(6):1033–1046, 2012.

[137] M. Yu and H. Luo. An adaptive radio resource management technique for APs in WLANs. In *Networks, 2004.(ICON 2004). Proceedings. 12th IEEE International Conference on*, volume 1, pages 85–91. IEEE, 2004.

[138] M. Yu, H. Luo, and K. K. Leung. A dynamic radio resource management technique for multiple APs in WLANs. *IEEE Transactions on Wireless Communications*, 5(7):1910–1919, 2006.

[139] X. Yue, C.-F. Wong, and S.-H. G. Chan. Cacao: Distributed client-assisted channel assignment optimization for uncoordinated wlans. *IEEE Transactions on Parallel and Distributed Systems*, 22(9):1433–1440, 2011.

[140] A. Zappone, M. Di Renzo, and M. Debbah. Wireless networks design in the era of deep learning: Model-based, ai-based, or both? *IEEE Transactions on Communications*, 67(10):7331–7376, 2019.

[141] Y. Zeng, P. H. Pathak, and P. Mohapatra. A first look at 802.11ac in action: Energy efficiency and interference characterization. In *2014 IFIP Networking Conference*, pages 1–9. IEEE, 2014.

[142] C. Zhang, P. Patras, and H. Haddadi. Deep learning in mobile and wireless networking: A survey. *IEEE Communications Surveys & Tutorials*, 21(3):2224–2287, 2019.

[143] W. Zhang, K. S. Kwak, H. Wang, and J. Huh. A practical MAC protocol supporting discontinuous channel bonding. In *Consumer Electronics (ICCE), 2013 IEEE International Conference on*, pages 510–511. IEEE, 2013.

[144] J. Zheng, Y. Cai, N. Lu, Y. Xu, and X. Shen. Stochastic game-theoretic spectrum access in distributed and dynamic environment. *IEEE transactions on vehicular technology*, 64(10):4807–4820, 2014.

[145] C. Zhong, Z. Lu, M. C. Gursoy, and S. Velipasalar. Actor-critic deep reinforcement learning for dynamic multichannel access. In *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 599–603. IEEE, 2018.

Appendix

MAB exploration algorithms

Algorithm 2: Exploration-first.

input: K;
initialization:
  $t = 1$;
  $\hat{Q}(a) = -1, \forall a \in [K]$;
while *active* **do**
    # explore unexplored actions with priority
    $a_t = \text{argmax}_a \hat{Q}(a)$;
    for $(a = 0; a < K; a++)$ **do**
        if $\vec{r}(a) == -1$ **then**
            $a_t = i$;
            break;
        end
    end
    # get performance
    $r_t \leftarrow \texttt{perform}(a_t)$;
    # update reward of action a_t
    $\hat{Q}(a_t) \leftarrow r_t$;
    $t \leftarrow t + 1$;
end

Algorithm 3: Implementation of ϵ-greedy.

input: ϵ_o, K;
initialization:
  $\epsilon_t = \epsilon_o$;
  $t = 1$;
  $N = 0, \forall a \in [K]$;
  $G = 0, \forall a \in [K]$;
  $\hat{Q}(a) = 0, \forall a \in [K]$;
while *active* **do**
    # select action following ϵ-greedy
    $a_t \begin{cases} a \sim U([1, K]), & \text{with prob } \epsilon \\ \text{argmax}_{a \in [K]} \hat{Q}, & \text{otherwise} \end{cases}$;
    $N(a) \leftarrow N(a) + 1$;
    # get performance
    $r_t \leftarrow \texttt{perform}(a_t)$;
    $G(a) \leftarrow G(a) + r_t$;
    # update estimated reward of action a_t
    $\hat{Q}(a_t) \leftarrow G/N(a)$;
    $t \leftarrow t + 1$;
    # update ϵ_t
    $\epsilon_t \leftarrow \epsilon_o / \sqrt{t}$;
end

Algorithm 4: UCB1 algorithm.

input: K;
initialization:
  $t = 0$;
  $N = 0, \forall a \in [K]$;
  $\hat{Q} = 0, \forall a \in [K]$;
while *active* **do**
    # sample according to UCB
    $a_t \leftarrow \text{argmax}_{a \in [K]} \left(\hat{Q}(a) + \sqrt{\frac{2 \log t}{N(a)}} \right)$;
    $N(a) \leftarrow N(a) + 1$;
    # get performance
    $r_t \leftarrow \texttt{perform}(a_t)$;
    $G(a) \leftarrow G(a) + r_t$;
    # update estimated reward of action a_t
    $\hat{Q}(a_t) \leftarrow G/N(a)$;
    $t \leftarrow t + 1$;
end

Algorithm 5: Thompson sampling as a Beta Bernoulli sampler.

input: K;
initialization:
  $t = 0$;
  $\alpha = \vec{1}$;
  $\beta = \vec{1}$;
while *active* **do**
    # draw sample for each arm
    $\tilde{Q}(a) \sim \mathcal{B}(\alpha, \beta), \forall a \in [K]$;
    # pick most probably optimal arm
    $a_t \leftarrow \text{argmax}_{a \in [K]} \tilde{Q}(a)$;
    # get iteration reward ($0 \leq r' \leq 1$)
    $R_t \leftarrow \texttt{perform}(a_t)$;
    # update α and β
    $\alpha = \alpha + R_t$;
    $\beta = \beta + (1 - R_t)$;
    $t \leftarrow t + 1$;
end

Algorithm 6: Thompson sampling with normal distribution.

input: K;
initialization:
  $t = 0$;
  $\hat{Q}(a) = 0, \forall a \in [1, A]$;
  $N(a) = 0, \forall a \in [1, A]$;
while *active* **do**
    # draw sample from each arm's distribution
    $\tilde{Q}(a) \sim \mathcal{N}(\hat{Q}(a), \frac{1}{N(a)+1}), \forall a \in [K]$;
    # pick most probable optimal
    $a_t \leftarrow \text{argmax}_{a \in [K]} \tilde{Q}(a)$;
    $N(a) \leftarrow N(a) + 1$;
    # get iteration reward
    $R_t \leftarrow \texttt{perform}(a_t)$;
    # update parameters of reward distribution of a_t
    $\hat{Q}(a_t) \leftarrow \frac{\hat{Q}(a_t)N(a_t)+R_t}{N(a_t)+2}$;
    $t \leftarrow t + 1$;
end

Algorithm 7: EXP3 algorithm [22].

input: K,γ;
initialization:
  $t = 0$;
  $w(a) = 1, \forall a \in [K]$;
  $\mathcal{P}(a) = 1/K, \forall a \in [K]$;
while *active* **do**
    # sample from posterior
    $a_t \sim \mathcal{P}$;
    # get iteration reward
    $r_t \leftarrow \texttt{perform}(a_t)$;
    # update weights and probabilities
    $\hat{Q} \leftarrow r_t / P(a)$;
    $w(a) \leftarrow w(a) \exp(\frac{\gamma \hat{Q}}{K})$;
    $\mathcal{P}(a) \leftarrow (1-\gamma)\frac{w(a)}{\sum_{i=1}^{K} w(i)} + \frac{\gamma}{K}$;
    $t \leftarrow t + 1$;
end

Chapter 8

PUBLICATIONS

The main publications of this thesis, from **paper #1** to **paper #6**, are appended next in the format of this dissertation.

8.1

Dynamic Channel Bonding in Spatially Distributed High-Density WLANs

Sergio Barrachina-Muñoz, Francesc Wilhelmi, Boris Bellalta

Abstract

In this paper, we discuss the effects on throughput and fairness of dynamic channel bonding (DCB) in spatially distributed high-density wireless local area networks (WLANs). First, we present an analytical framework based on continuous-time Markov networks (CTMNs) for depicting the behavior of different DCB policies in spatially distributed scenarios, where nodes are not required to be within the carrier sense range of each other. Then, we assess the performance of DCB in high-density IEEE 802.11ac/ax WLANs by means of simulations. We show that there may be critical interrelations among nodes in the spatial domain – even if they are located outside the carrier sense range of each other – in a *chain reaction* manner. Results also reveal that, while always selecting the widest available channel normally maximizes the individual long-term throughput, it often generates unfair situations where other WLANs starve. Moreover, we show that there are scenarios where DCB with stochastic channel width selection improves the latter approach both in terms of individual throughput and fairness. It follows that there is not a unique optimal DCB policy for every case. Instead, smarter bandwidth adaptation is required in the challenging scenarios of next-generation WLANs.

1 Introduction

Wireless local area networks (WLANs), with IEEE 802.11 as the most widely used standard, are a cost-efficient solution for wireless Internet access that can satisfy most of the current communication requirements in domestic, public, and business scenarios. However, the scarcity of the frequency spectrum in the industrial, scientific and medical (ISM) radio bands, the increasing throughput demands given by new hungry-bandwidth applications, and the heterogeneity of current wireless network deployments give rise to substantial complexity. Such issues gain importance in dense WLAN deployments, leading to multiple partially overlapping scenarios and coexistence problems.

In this regard, two main approaches to optimizing the scarce resources of the frequency spectrum are being deeply studied in the context of WLANs: channel allocation (CA) and channel bonding (CB). While CA refers to the action of allocating the potential transmission channels (i.e., both the primary and secondary channels) for a WLAN or group of WLANs, CB is the technique whereby nodes are allowed to use a contiguous set of idle channels for transmitting in larger bandwidths, thus potentially achieving a higher throughput.

This paper focuses on CB, which was first introduced in the IEEE 802.11n (11n) amendment by allowing two 20 MHz basic channels to be aggregated into a 40 MHz channel. Newer amendments like IEEE 802.11ac (11ac) extend the number of basic channels that can be aggregated up to 160 MHz channel widths. It is expected that IEEE 802.11ax (11ax) will boost the use of wider channels [1]. Nonetheless, due to the fact that using wider channels increases the contention and interference among nodes, undesirable lower performances may be experienced when applying static channel bonding (SCB), especially in high-density WLAN scenarios. To mitigate such a negative effect, dynamic channel bonding (DCB) policies are used to select the bandwidth in a more flexible way based on the instantaneous spectrum occupancy. A well-known example of DCB policy is *always-max* (AM)[1] [2, 3], where transmitters select the widest channel found idle when the backoff counter terminates. To the best of our knowledge, the works in the literature assessing the performance of DCB only study the SCB and AM policies, while they also assume fully overlapping scenarios where all the WLANs are within the carrier sense range of the others [2, 4–6]. Therefore, there is an important lack of insights on the performance of CB in more realistic WLAN scenarios, where such a condition usually does not hold.

With this work, we aim to extend the state of the art by providing new insights on the performance of DCB under saturation regimes in WLAN scenarios that are not required to be fully

[1] Some papers in the literature indistinctively use the terms DCB and AM. In this paper, we notate AM as a special case of DCB.

overlapping; where the effect of carrier sense and communication ranges play a crucial role due to spatial distribution interdependencies. Namely, the operation of a node has a direct impact on the nodes inside its carrier sense range, which in turn may affect nodes located outside such range in complex and hard to prevent ways. Besides, we assess different DCB policies, including a stochastic approach that selects the channel width randomly.

In order to evaluate different DCB policies, we first introduce the Spatial-Flexible Continuous Time Markov Network (SFCTMN), an analytical framework based on continuous-time Markov networks (CTMNs). This framework is useful for describing the different phenomena that occur in WLAN deployments when considering DCB in spatially distributed scenarios. In this regard, we analytically depict such complex phenomena by means of illustration through several toy scenarios. Finally, we evaluate the performance of the proposed policies in large high-density 11ax WLAN scenarios by means of simulations using `Komondor`, a particular release (v1.0.1) of the Komondor [7] wireless networks simulator.[2] We find that, while AM is normally the best DCB policy for maximizing the individual long-term throughput of a WLAN, it may generate unfair situations where some other WLANs starve. In fact, there are cases where less aggressive policies like stochastic channel width selection improve AM both in terms of individual throughput and fairness. This leads to the need of boosting wider channels through medium adaptation policies. The contributions of this paper are as follows:

1. Novel insights on the effects of DCB in high-density scenarios. We depict the complex interactions given in spatially distributed deployments – i.e., considering path loss, signal-to-interference-plus-noise ratio (SINR) and clear channel assessment (CCA) thresholds, co-channel and adjacent channel interference, etc. – and discuss the influence that nodes have among them.

2. Generalization of DCB policies including *only-primary* (i.e., selecting just the primary channel for transmitting), SCB, AM, and *probabilistic uniform* (PU) (i.e., selecting the channel width stochastically).

3. Algorithm for modeling WLAN scenarios with CTMNs that extends the one presented in [6]. Such an extension allows us capturing non-fully overlapping scenarios, taking into consideration spatial distribution implications. Moreover, this algorithm allows us to model any combination of DCB policies in a network.

4. Performance evaluation of the presented DCB policies in high-density WLAN scenarios by means of simulations. The selected physical (PHY) and medium access control (MAC) parameters are representative of single user (SU) transmissions in 11ax WLANs.

2 Related work

Several works in the literature assess the performance of CB by means of analytical models, simulations or testbeds. Authors in [8, 9] experimentally analyze SCB in IEEE 802.11n WLANs and show that the reduction of Watt/Hertz when transmitting in larger channel widths causes lower SINR at the receivers. This lessens the coverage area consequently and increases the probability of packet losses due to the accentuated vulnerability to interference. Nonetheless, they also show that DCB can provide significant throughput gains when such issues are palliated by properly adjusting the transmission power and data rates.

The advantages and drawbacks of CB are accentuated with the 11ac and 11ax amendments since larger channel widths are allowed (up to 160 MHz). Nevertheless, it is important to emphasize that in the dense and short-range WLAN scenarios expected in the coming years [1], the issues concerning low SINR values may be palliated. The main reason lies in the shorter distances between transceiver and receiver, and the usage of techniques like spatial diversity multiple-input multiple-output (MIMO)[10]. An empirical study on CB in 11ac is followed in [11], where authors show that throughput increases by bonding channels. By means of simulations, authors in [3, 12] assess the performance of DCB in 11ac WLANs, resulting in significant throughput gains. Still, they also corroborate that these gains are severely compromised by the activity of overlapping wireless networks.

[2] All of the source code of SFCTMN and `Komondor` is open, encouraging sharing of algorithms between contributors and providing the ability for people to improve on the work of others under the GNU General Public License v3.0. The repositories can be found at `https://github.com/sergiobarra/SFCTMN` and `https://github.com/wn-upf/Komondor`, respectively.

There are other works in the literature that follow an analytical approach for assessing the performance of CB. For instance, authors in [4] analytically model and evaluate the performance of CB in short-range 11ac WLANs, proving significant performance boost in presence of low to moderate external interference. In [2], authors show that CB can provide significant performance gains even in high-density scenarios, though it may also cause unfairness. Non-saturation regimes are considered in [13, 14], where authors propose an analytical model for the throughput performance of CB in 11ac/11ax WLANs under both saturated and non-saturated traffic loads. An analytical framework to study the performance of opportunistic CB where 11ac users coexist with legacy users is presented in [15]. Recently, an analytical model based on renewal theory showed that 11ac/11ax DCB can improve throughput even in the presence of legacy users [16]. Literature on CTMN models for DCB WLANs is further discussed in Section 4.

As for particular CB solutions or algorithms, an intelligent scheme for jointly adopting the rate and bandwidth in MIMO 11n WLANs is presented in [10]. Testbed experiments show that such scheme (ARAMIS) accurately adapts to a wide variety of channel conditions with negligible overhead and achieving important performance gains. A DCB protocol with collision detection is presented in [17], in which a node gradually increases the transmission bandwidth whenever new narrow channels are found available. A stochastic spectrum distribution framework accounting for WLANs demand uncertainty is presented in [18], showing better performance compared to the naive allocation approach. Authors in [19] show that the maximal throughput performance can be achieved with DCB under the CA scheme with the least overlapped channels among WLANs. A dynamic bandwidth selection protocol for 11ac WLANs is proposed in [20] to prevent the carrier sensing decreasing and outside warning range problems. In [16] authors propose a heuristic primary channel selection for maximizing the throughput of multi-channel users. Finally, [21] proposes a prototype implementation for commercial 11ac devices showing up to 1.85x higher throughput when canceling time-domain interference.

We believe that this is the first work providing insights into the performance of DCB in spatially distributed scenarios, where the effect of partially overlapping nodes plays a crucial role due to the spatial interdependencies. We also provide an algorithm for generating the CTMNs to model such kind of scenarios. Besides, we assess the performance of different DCB policies, including a novel stochastic approach, and show that always selecting the widest available channel may be sub-optimal in some scenarios.

3 System model under consideration

In this section, we first depict the notation regarding channelization that is used throughout this article. We also define the DCB policies that are studied and provide a general description of the carrier sense multiple access with collision avoidance (CSMA/CA) operation in IEEE 802.11 WLANs. Finally, we expose the main assumptions considered in the presented scenarios.

3.1 Channelization

Let us discuss the example shown in Figure 1 for introducing the channel access terminology and facilitating further explanation. In this example, the system channel C_{sys} counts with $N_{\text{sys}} = 8$ basic channels and WLAN X is allocated with the channel $C_{\text{X}} = \{1, 2, 3, 4\}$ and primary channel $p_{\text{X}} = 3$. Note that in this particular example the AP of WLAN X does not select the entire allocated channel, but a smaller one, i.e., $C_{\text{X}}^{\text{tx}} = \{3, 4\} \subseteq C_{\text{X}}$. Two reasons may be the cause: *i*) basic channels 1 and/or 2 are detected busy at the end of the backoff, or *ii*) the DCB policy determines not to pick them. More formally, the definitions of the channelization terms used throughout the paper are as follows:

- **Basic channel c**: the frequency spectrum is split into basic channels of width $|c| = 20$ MHz.
- **Primary channel p_w**: a basic channel with different roles depending on the node state. All the nodes belonging to the same WLAN w must share the same primary channel p_w. Essentially, this channel is used to *i*) sense the medium for decrementing the backoff when the primary channel's frequency band is found free, and *ii*) listening to control and data packets.
- **Channel C**: a channel $C = \{c_1, c_2, ..., c_N\}$ consists of a contiguous set of N basic channels. The width (or bandwidth) of a channel is $N|c|$.
- **Channelization scheme $\mathcal{C}$**: the set of channels that can be used for transmitting is determined by the channel access specification and the system channel (C_{sys}), whose bandwidth is

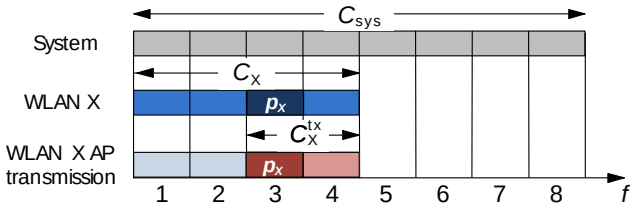


Figure 1: Channel access notation.

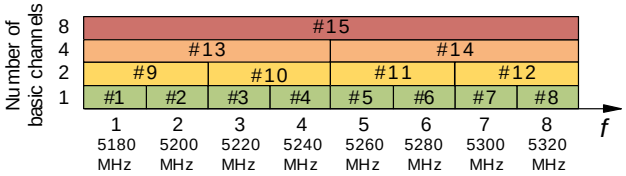


Figure 2: Simplified channelization of 11ac and 11ax.

given by $N_{\text{sys}}|c|$. Namely, all the nodes in the system must transmit in some channel included in $\mathcal{C}$. A simplified version of the channelization considered in the 11ac and 11ax standards is shown in Figure 2, $\mathcal{C} = \{\{1\}, \{2\}, ..., \{1,2\}, \{3,4\}, ..., \{1,2,3,4\}, ..., \{1,2,...,8\}\}$.

- **WLAN's allocated channel C_w**: nodes in a WLAN w must transmit in a channel contained in $C_w \in \mathcal{C}$. Different WLANs may be allocated with different primary channels and different available channel widths.
- **Transmission channel C_n^{tx}**: a node n belonging to a WLAN w has to transmit in a channel $C_n^{\text{tx}} \subseteq C_w \in \mathcal{C}$, which will be given by i) the set of basic channels in C_w found idle by node n at the end of the backoff (C_n^{free}),[3] and ii) the implemented DCB policy.

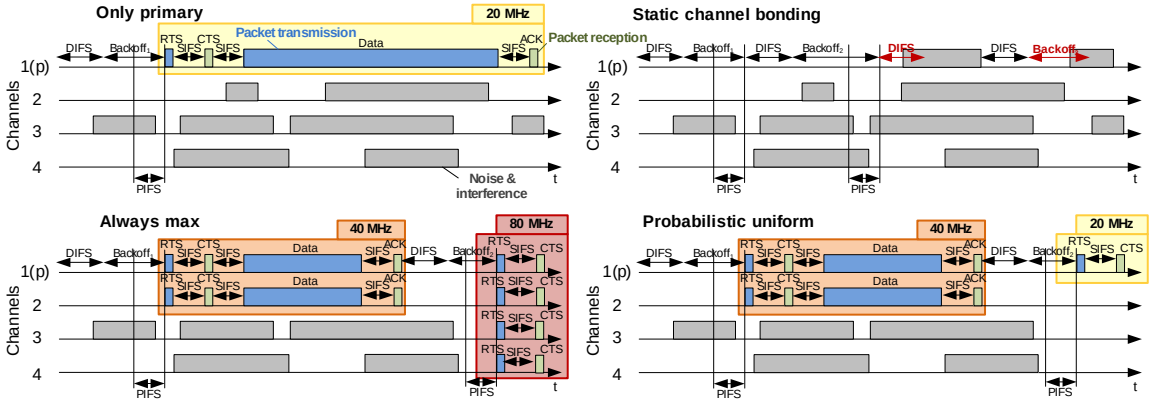


Figure 3: CSMA/CA temporal evolution of a node operating under different DCB policies in an 11ax channelization scheme. The DIFS and backoff in red represent that the sensed interference in the primary channel forces resetting the backoff procedure. While the legacy packets (RTS, CTS, and ACK) duration is the same no matter the bandwidth, the data duration is clearly reduced when transmitted in 40 MHz.

3.2 CSMA/CA operation in IEEE 802.11 WLANs

According to the CSMA/CA operation, when a node n belonging to a WLAN w has a packet ready for transmission, it measures the power sensed in the frequency band of p_w. Once the primary channel has been detected free, i.e., the power sensed by n at p_w is smaller than its CCA threshold,

[3]Note that, in order to include secondary channels for transmitting, a WLAN must listen them free during at least a point coordination function interframe space (PIFS) period before the backoff counter terminates as shown in Figure 3. While such PIFS condition is not considered in the SFCTMN framework for the sake of analysis simplicity, the `Komondor` simulator does.

the node starts the backoff procedure by selecting a random initial value of BO $\in [0, \text{CW} - 1]$ time slots of duration T_e. The contention window is defined as $\text{CW} = 2^b \text{CW}_{\text{min}}$, where $b \in \{0, 1, 2, ..., m\}$ is the backoff stage with maximum value m, and CW_{min} is the minimum contention window. When a packet transmission fails, b is increased by one unit, and reset to 0 when the packet is acknowledged.

After computing BO, the node starts decreasing the backoff counter while sensing the primary channel. Whenever the power sensed by n at p_w is higher than its CCA, the backoff is paused and set to the nearest higher time slot until p_w is detected free again, at which point the countdown is resumed. When the backoff timer reaches zero, the node selects the transmission channel C_n^{tx} based on the set of idle basic channels C_n^{free} and on the DCB policy. The selected transmission channel is then used throughout the whole packet exchanges involved in a data packet transmission between the transceiver and receiver. Namely, request to send (RTS) – used for notifying the selected transmission channel – clear to send (CTS), and acknowledgment (ACK) packets are also transmitted in C_n^{tx}. Likewise, any other node that receives an RTS in its primary channel with enough power to be decoded will enter in network allocation vector (NAV) state, which is used for deferring channel access and avoiding packet collisions (especially those caused by hidden node situations).

3.3 DCB policies

The DCB policy determines the transmission channel a node must pick from the set of available ones. When the backoff terminates, any node belonging to a WLAN w operates according to the DCB policy as follows:

- **Only-primary (OP) or single-channel**: pick just the primary channel for transmitting if it is found idle.
- **Static channel bonding (SCB)**: exclusively pick the full channel allocated in its WLAN when found free. Namely, nodes operating under SCB cannot transmit in channels different than C_w.
- **Always-max (AM)**: pick the widest possible channel found free in C_w for transmitting.
- **Probabilistic uniform (PU)**: pick with same probability any of the possible channels found free inside the allocated channel C_w.

For the sake of illustration, let us consider the example shown in Figure 3, which shows the evolution of a node implementing different DCB policies. In this example, a node is allowed to transmit in the set of basic channels $C_w = \{1(p), 2, 3, 4\}$, where $p_w = 1$ is the primary channel. While OP picks just the primary channel, the rest of policies try to bond channels in different ways. In this regard, SCB is highly inefficient in scenarios with partial interference. In fact, no packets can be transmitted with SCB in this example since the basic channel $\{3\} \in C_w$ is busy during the PIFS durations previous to the backoff terminations. However, more flexible approaches like AM and PU are able to transmit more than one frame in the same period of time. On the one hand, AM adapts in an aggressive way to the channel state. Specifically, it is able to transmit in 40 and 80 MHz channels at the end of the first and second backoff, respectively. On the other hand, the stochastic nature of PU makes it more conservative than AM. In the example, the node could transmit in 1 or 2 basic channels with the same probability (1/2) at the end of the first backoff. Likewise, after the second backoff, a channel composed of 1, 2 or 4 basic channels could be selected with probability (1/3).

3.4 Main assumptions

In this paper, we present results gathered via the SFCTMN framework based on CTMNs, and also via simulations through the `Komondor` wireless network simulator. While in the latter case we are able to introduce more realistic implementations of the 11ax amendment, in the analytical model we use relaxed assumptions for facilitating subsequent analysis. This subsection depicts the general assumptions considered in both cases.

1. **Channel model**: signal propagation is isotropic. Also, the propagation delay between any pair of nodes is considered negligible because of the small carrier sense in the above 1GHz ISM bands where WLANs operate. Besides, the transmission power is divided uniformly among the basic channels in the selected transmission channel. We also consider an adjacent channel interference model that replicates P_ν of the power transmitted per Hertz into the two basic channels that are contiguous to the actual transmission channel.

2. **Packet errors**: a packet is lost if *i*) the power of interest received at the receiver is less than its CCA, *ii*) the SINR (γ) perceived at the receiver does not accomplish the capture effect (CE), i.e., $\gamma <$ CE, or *iii*) the receiver was already receiving a packet. In the latter case, the decoding of the first packet is ruptured only if the CE is no longer accomplished because of the interfering transmission. We assume an infinite maximum number of retransmissions per packet, whose effect is negligible in most of the cases because of the small probability of retransmitting a data packet more than few times [22].

3. **Modulation and coding scheme (MCS)**: the MCS index used by each WLAN is the highest possible according to the SINR perceived by the receiver, and it is kept constant throughout all the simulation. We assume that the MCS selection is designed to keep the packet error rate constant and equal to $\eta = 0.1$ given that static deployments are considered.[4] Note that the η value is only considered if none of the three possible causes of packet error explained in the item above are given.

4. **Traffic**: downlink traffic is considered. In addition, we assume a full-buffer mode where APs always have backlogged data pending for transmission.

4 The CTMN model for WLANs

The analysis of CSMA/CA networks through CTMN models was firstly introduced in [23]. Such models were later applied to IEEE 802.11 networks in [2, 4, 6, 13, 14, 19, 24, 25], among others. Experimental results in [26, 27] demonstrate that CTMN models, while idealized, provide remarkably accurate throughput estimates for actual IEEE 802.11 systems. A comprehensible example-based tutorial of CTMN models applied to different wireless networking scenarios can be found in [28]. Nevertheless, to the best of our knowledge, works that model DCB through CTMNs study just the SCB and AM policies, while assuming fully overlapping scenarios. Therefore, there is an important lack of insights on more general deployments, where such conditions usually do not hold and interdependencies among nodes may have a critical impact on their performance. For instance, an optimal channel allocation algorithm to achieve maximal throughput with DCB was recently presented in [29]. However, this work does not consider the implications of either spatial distribution nor CE.

In this section, we depict our extended version of the algorithm introduced in [6] for generating the CTMNs corresponding to spatially distributed WLAN scenarios, which is implemented in the SFCTMN framework. With this extension, as the condition of having fully overlapping networks is no longer required for constructing the corresponding CTMNs, more factual observations can be made.

4.1 Implications

Modeling WLAN scenarios with CTMNs requires the backoff and transmission times to be exponentially distributed. It follows that, because of the negligible propagation delay, the probability of packet collisions between two or more nodes within the carrier sense range of the others is zero. The reason is that two WLANs will never end their backoff at the same time, and therefore they will never start a transmission at the same time either. Besides, in overlapping single-channel CSMA/CA networks, it is shown that the state probabilities are insensitive to the backoff and transmission time distributions [27, 30]. However, even though authors in [6] prove that the insensitivity property does not hold for DCB networks, the sensitivity to the backoff and transmission time distributions is very small. Therefore, the analytical results obtained using the exponential assumption offer a good approximation for deterministic distributions of the backoff, data rate, and packet length.

4.2 Constructing the CTMN

In order to depict how CTMNs are generated, let us consider the toy scenario (*Scenario I*) shown in Figure 4, which is composed of two fully overlapping WLANs implementing AM.[5] The channel allocation employed in such a scenario can be defined as $C_{\mathrm{A}} = \{1, 2, 3, 4\}$ with $p_{\mathrm{A}} = 2$, and $C_{\mathrm{B}} = \{3, 4\}$ with

[4] In 802.11 devices, given a minimum receiver sensitivity and SINR table, a maximum decoding packet error rate of 10% is usually tried to be guaranteed when selecting the MCS index.

[5] *Scenario I* is selected for conveniently depicting the algorithm. CTMNs corresponding to non-fully overlapping scenarios (e.g., *Scenario III* in Section 5) can be also generated with the very same algorithm.

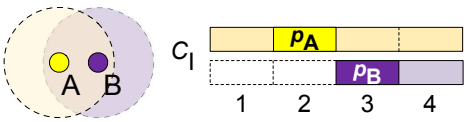


Figure 4: *Scenario I.* WLANs A and B are inside the carrier sense range of each other with potentially overlapping basic channels 3 and 4.

$p_B = 3$. That is, there are four basic channels in the system, and the set of valid transmission channels according to the 11ax channel access scheme is $\mathcal{C}_I = \{\{1\},\{2\},\{3\},\{4\},\{1,2\},\{3,4\},\{1,2,3,4\}\}$ (see Figure 2). Due to the fact that both WLANs are inside the carrier sense range of each other, their APs could transmit simultaneously at any time t only if their transmission channels do not overlap, i.e., $C_A^{tx}(t) \cap C_B^{tx}(t) = \emptyset$. Notice that slotted backoff collisions cannot occur because their counters decrease continuously in time, and therefore two transmissions can be neither started nor finished at the very same time.

4.2.1 States

A state in the CTMN is defined by the set of WLANs active and the basic channels on which they are transmitting. Essentially, we say that a WLAN is active if it is transmitting in some channel, and inactive otherwise. We define two types of state spaces: the global state space (Ψ) and the feasible state space ($\mathcal{S}$).

- **Global state space**: a global state $\psi \in \Psi$ is a state that accomplishes two conditions: *i*) the channels in which the active WLANs are transmitting comply with the channelization scheme $\mathcal{C}$, and *ii*) all active WLANs transmit inside their allocated channels. That is, Ψ only depends on the particular channelization scheme $\mathcal{C}$ in use and on the channel allocation of the WLANs in the system. In this paper, we assume that every transmission should be made in channels inside C_{sys} that are composed of $a = 2^k$ contiguous basic channels, for some integer $k \leq \log_2 N_{sys}$, and that their rightmost basic channels fall on multiples of a, as stated in the 11ac and 11ax amendments.
- **Feasible state space**: a feasible state $s \in \mathcal{S} \subseteq \Psi$ exists only if each of the active WLANs in such state started their transmissions by accomplishing the CCA requirement derived from the assigned DCB policy. Namely, given a global state space, $\mathcal{S}$ depends only on the spatial distribution and on the DCB policies assigned to each WLAN.

The CTMN corresponding to the toy *Scenario I* is shown in Figure 5. Regarding the notation, we represent the states by the most left and most right basic channels used in the transmission channels of each of their active WLANs. For instance, state $s_4 = A_1^2 B_3^4$ refers to the state where A and B are transmitting in channels $C_A^{tx} = \{1,2\}$ and $C_B^{tx} = \{3,4\}$, respectively. Concerning the state spaces, states ψ_6, ψ_7, ψ_8, ψ_9, ψ_{10}, ψ_{11}, $\psi_{12} \notin \mathcal{S}$ are not reachable (i.e., they are global but not feasible) for two different reasons. First, states ψ_{11} and ψ_{12} are not feasible because of the overlapping channels involved. Secondly, the rest of unfeasible states are so due to the fact that AM is applied, thus at any time t that WLAN A(B) finishes its backoff and B(A) is not active, A(B) picks the widest available channel, i.e., $C_A^{tx}(t) = \{1,2,3,4\}$ or $C_B^{tx}(t) = \{3,4\}$, respectively. Likewise, any time A(B) finishes its backoff and B(A) is active, A(B) picks again the widest available channel, which in this case would be $C_A^{tx}(t) = \{1,2\}$ for A and $C_B^{tx}(t) = \{3,4\}$ for B if A is not transmitting in its full allocated channel, respectively. Some states such as $s_5 = A_1^2$ are reachable only via backward transitions. In this case, when A finishes its backoff and B is transmitting in $C_B^{tx}(t) = \{3,4\}$ (i.e., s_3), A picks just $C_A^{tx}(t) = \{1,2\}$ because the power sensed in channels 3 and 4 exceeds the CCA as a consequence of B's transmission. That is, s_5 is only reachable through a backward transition from s_4, given when B finishes its transmission in state s_4.

4.2.2 CTMN algorithm: finding states and transitions

The first step for constructing the CTMN is to identify the global state space Ψ, which is simply composed by all the possible combinations given by the system channelization scheme and the channel allocations of the WLANs. The feasible states in $\mathcal{S}$ are later identified by exploring the

Algorithm 1: CTMN generation of spatially distributed DCB WLAN scenarios.

$i \coloneqq 1$; # Index of the last state found
$k \coloneqq 1$; # Index of the state currently being explored
$s_k \coloneqq \emptyset$; # State currently being explored
$\mathcal{S} \coloneqq \{s_k\}$; # Set of feasible states
$Q \coloneqq [\,]$; # Transition rate matrix
Generate the global state space Ψ
$\Psi \coloneqq$ `generate_psi_space()` ;
while $s_k \in \mathcal{S}$ **do**
  foreach *WLAN* X **do**
    # If WLAN is active in s_k
    if $\exists a, b$ s.t. $X_a^b \in s_k$ **then**
      foreach $\psi \in \Psi$ **do**
        # If there exists a *backward* transition
        if $s_k - X_a^b == \psi$ **then**
          if $\psi \notin \mathcal{S}$ **then**
            $i \coloneqq i$+1;
            $k' \coloneqq i$;
            $S \coloneqq S \cup \psi$;
          else
            # Get index of state ψ in $\mathcal{S}$
            $k' \coloneqq$ `get_index`(φ^*);
          # New *backward* transition $s_k \rightarrow s_{k'}$
          $Q_{k,k'} \coloneqq \mu_{\mathrm{X}}(s_k)$
    # If WLAN is NOT active in s_k
    else
      $\Phi \coloneqq \emptyset$; # Set of global states reachable from s_k
      $\Phi^* \coloneqq \emptyset$; # Set of feasible states reachable from s_k
      # Find possible forward states
      foreach $\psi \in \Psi$ **do**
        if $\exists a, b$ s.t. $s_k + X_a^b == \psi$ **then**
          $\Phi \coloneqq \Phi \cup \psi$;
      # Function f finds feasible states and corresponding transition probabilities according to the DCB policy $\mathcal{D}$ and the basic channels found free
      $\mathcal{C}_{\mathrm{X}}^{\mathrm{free}}(s_k) \coloneqq \{P_{\mathrm{X}}^{\mathrm{rx}}(s_k, C_{\mathrm{X}}) < \mathrm{CCA_X} : 0, 1\}$;
      $\{\Phi^*, \vec{\alpha}\} \coloneqq f(\mathcal{D}, \mathcal{C}_{\mathrm{X}}^{\mathrm{free}}, \Phi)$;
      foreach $\varphi^* \in \Phi^*$ **do**
        if $\varphi \notin S$ **then**
          $i \coloneqq i$+1;
          $k' \coloneqq i$;
          $S \coloneqq S \cup \varphi^*$;
        else
          # Get index of state φ^* in $\mathcal{S}$
          $k' \coloneqq$ `get_index`(φ^*);
        # New *forward* transition $s_k \rightarrow s_{k'}$
        $Q_{k,k'} \coloneqq \vec{\alpha}(\varphi^*)\lambda_{\mathrm{X}}$;
  $k \coloneqq k$+1;

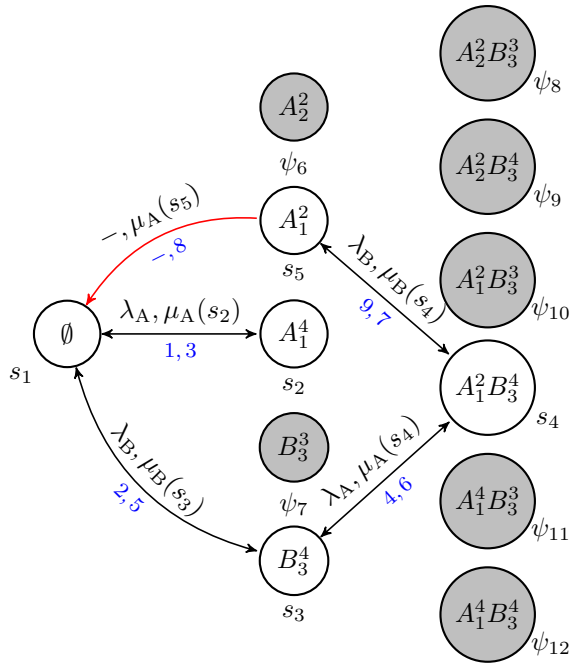


Figure 5: CTMN of *Scenario I* when applying AM. Circles represent states. All the states are global. Specifically, feasible states are displayed in white, while non-feasible states are gray colored. Two-way transitions are noted with forward and backward rates λ, μ, respectively, to avoid cluttering in the figure. The only backward transition is colored in red. The blue pair of numbers beside the transition edges represent the algorithm's discovery order of the forward and backward transitions, respectively.

states in Ψ. Algorithm 1 shows the pseudocode for identifying both $\mathcal{S}$ and the transitions among such states, which are represented by the transition rate matrix Q.[6]

Essentially, while there are discovered states in $\mathcal{S}$ that have not been explored yet, for any state $s_k \in \mathcal{S}$ not explored, and for each WLAN X in the system, we determine if X is active or not. If X is active, we then set possible *backward* transitions to already known and not known states. To do so, it is required to fully explore Ψ looking for states where: *i)* other active WLANs in the state remain transmitting in the same transmission channel, and *ii)* WLAN X is not active.

On the other hand, if WLAN X is inactive in state s_k, we try to find *forward* transitions to other states. To that aim, the algorithm fully explores Ψ looking for states where *i*) other active WLANs in the state remain transmitting in the same transmission channel, and *ii*) X is active in the new state as a result of applying the implemented DCB policy ($\mathcal{D}$) as shown in line 33. It is important to remark that in order to apply such policy, the set of idle basic channels in state s_k, i.e., $\mathcal{C}_{\mathrm{X}}^{\mathrm{free}}(s_k)$, must be identified according to the power sensed in each of the basic channels allocated to X, i.e., $P_{\mathrm{X}}^{\mathrm{rx}}(s_k, C_{\mathrm{X}})$, and on its CCA level. Thereafter, the transmission channel is selected through the f function, which applies $\mathcal{D}$.

Each transition between two states s_i and s_j has a corresponding transition rate $Q_{i,j}$. For *forward* transitions, the packet transmission attempt rate (or simply backoff rate) has an average duration $\lambda = 1/(\mathrm{E}[B] \cdot T_{\mathrm{slot}})$, where $\mathrm{E}[B]$ is the expected backoff duration in time slots, determined by the minimum contention window, i.e., $\mathrm{E}[B] = \frac{\mathrm{CW}_{\mathrm{min}}-1}{2}$. Furthermore, for *backward* transitions, the departure rate (μ) depends on the duration of a successful transmission, i.e., $\mu = 1/T_{\mathrm{suc}} = (T_{\mathrm{RTS}}+T_{\mathrm{SIFS}}+T_{\mathrm{CTS}}+T_{\mathrm{SIFS}}+T_{\mathrm{DATA}}+T_{\mathrm{SIFS}}+T_{\mathrm{BACK}}+T_{\mathrm{DIFS}}+T_{\mathrm{e}})^{-1}$, which in turn depends on both the data rate (r) given by the selected MCS and transmission channel width, and on the packet length $\mathrm{E}[L_{\mathrm{data}}]$. In the algorithm, we simply consider that the data rate of a WLAN X depends on the state of the system, which collects such information, i.e., $\mu_{\mathrm{X}}(s)$.

Depending on the DCB policy, different feasible *forward* transitions may exist from the very same state, which are represented by the set Φ^*. As shown in line 43, every feasible *forward* transition rate is weighted by a transition probability vector ($\vec{\alpha}$) whose elements determine the probability of

[6]Notice that we use $X_a^b \in s$ to say that a WLAN X transmits in a range of contiguous basic channels $[a, b]$ when the CTMN is in state s. With slight abuse of notation, $s - X_a^b$ represents the state where all WLANs that were active in s remain active except for X, which becomes inactive after finishing its packet transmission. Similarly, $s{+}X_a^b$, represents the state where all the active WLANs in s remain active and X is transmitting in the range of basic channels $[a, b]$.

transiting to each of the possible global states in Φ. Namely, with a slight abuse of mathematical notation, the probability to transit to any given feasible state $\varphi^* \in \Phi$ is $\vec{\alpha}(\varphi^*)$. As a consequence, $\vec{\alpha}$ must follow the normalization condition $\sum \vec{\alpha}(\varphi^*) = 1$.

For the sake of illustration, in the CTMNs of Figures 5, 7 and 9, states are numbered according to the order in which they are discovered. Transitions between states are also shown below the edges. Note that with SFCTMN, since non-fully overlapping networks are allowed, transitions to states where one or more WLANs may suffer from packet losses due to interference are also reachable (see Section 5).

4.3 Performance metrics

Since there are a limited number of possible channels to transmit in, the constructed CTMN will always be finite. Furthermore, it will be irreducible due to the fact that backward transitions between neighboring states are always feasible. Therefore, a steady-state solution to the CTMN always exists. However, due to the possible existence of one-way transitions between states, the CTMN is not always time-reversible and the local balance may not hold [31]. Accordingly, it prevents to find simple product-from solutions to compute the equilibrium distribution of the CTMNs. The equilibrium distribution vector $\vec{\pi}$ represents the fraction of time the system spends in each feasible state. Hence, we define π_s as the probability of finding the system at state s. In order to obtain $\vec{\pi}$ we can use the transition rate matrix Q given the system of equations $\vec{\pi}Q = 0$.

As an example, for *Scenario I*, considering that its elements are sorted by the discovery order of the states, $\vec{\pi} = (\pi_\emptyset, \pi_{\mathrm{A}_1^4}, \pi_{\mathrm{B}_3^4}, \pi_{\mathrm{A}_1^2\mathrm{B}_3^4}, \pi_{\mathrm{A}_1^2})$. Besides, the corresponding transition rate matrix is

$$Q = \begin{pmatrix} * & \lambda_\mathrm{A} & \lambda_\mathrm{B} & 0 & 0 \\ \mu_\mathrm{A}(s_2) & * & 0 & 0 & 0 \\ \mu_\mathrm{B}(s_3) & 0 & * & \lambda_\mathrm{A} & 0 \\ 0 & 0 & \mu_\mathrm{A}(s_4) & * & \mu_\mathrm{B}(s_4) \\ \mu_\mathrm{A}(s_5) & 0 & 0 & \lambda_\mathrm{B} & * \end{pmatrix},$$

where λ_A, λ_B and $\mu_\mathrm{A}(s)$, $\mu_\mathrm{B}(s)$ are the packet generation and departure rates in state s of WLANs A and B, respectively. The diagonal elements represented by '*' in the matrix should be replaced by the negative sum of the rest of items of their row, e.g., $Q_{4,4} = -\big(\mu_\mathrm{A}(s_4)+\mu_\mathrm{B}(s_4)\big)$, but for the sake of illustration we do not include them in the matrix. Once $\vec{\pi}$ is computed, estimating the average throughput experienced by each WLAN is straightforward. Specifically, the average throughput of a WLAN w is

$$\Gamma_w := \mathrm{E}[L]\left(\sum_{s\in\mathcal{S}}\{\gamma_w(s) > \mathrm{CE} : 0,1\}\mu_w(s)\pi_s\big(1 - \eta\big)\right),$$

where $\mathrm{E}[L]$ is the expected data packet length, $\gamma_w(s)$ is the SINR perceived by the STA in WLAN w in state s, CE is the capture effect, and η is the constant packet error probability. The system aggregate throughput is therefore the sum of the throughputs of all the WLANs, i.e., $\Gamma := \sum_{w=1}^{M} \Gamma_w$. Besides, in order to evaluate the fairness of a given scenario, we can use $\mathcal{P} := \sum_{w=1}^{M} \log_{10} \Gamma_w$ and/or $\mathcal{F} := \big(\sum_w^M \Gamma_w\big)^2 / \big(M \sum_w^M \Gamma_w^2\big)$, for proportional fairness and the Jain's Fairness Index, respectively.

4.4 Captured phenomena

Even though most of the well-known wireless phenomena are captured by SFCTMN, there are some important features that cannot be implemented due to its mathematical modeling nature. Essentially, the main limitations are the inability to capture backoff collisions and the constraints in terms of execution time for medium size networks. Besides, only the overhead of the RTS/CTS packets are considered in SFCTMN, i.e., the time of a successful transmission takes also the transmission time of such packets into account. However, the main purpose of the RTS/CTS mechanism of avoiding hidden nodes is not captured by the generated CTMNs since no packets are actually transmitted. That is, only average performance is captured through states modeled without differentiating from the type of packet being transmitted.

To cope with the abovementioned limitations, we make use of a simulator in Section 6. Basically, when we lose the benefits of analytically modeling the networks, with `Komondor` we are allowed to simulate large networks and get more realistic insights. A comparison of the features implemented in each tool is shown in Table 1.

5 Interactions in frequency and space

In this section, we draw some relevant conclusions about applying different DCB policies in CSMA/CA WLANs by analyzing four representative toy scenarios with different channel allocations and spatial distributions. To that aim, we use the SFCTMN analytical framework and validate the gathered results by means of the `Komondor` wireless network simulator.[7]

In summary, the main outcomes derived from the analytical analyses performed below in this section are: *i*) the feasible system states depend on the DCB policies followed by each of the WLANs, *ii*) maximizing the instantaneous throughput may not be the optimal strategy to maximize the long-term throughput, *iii*) in non-fully overlapping scenarios, cumulative interference and flow starvation may appear and cause poor performance to some WLANs, *iv*) there is not a unique optimal DCB policy. Note that, otherwise stated, in this paper the optimal DCB policy $\mathcal{D}_w^*$ for WLAN w is the one that maximizes its throughput, i.e., $\mathcal{D}_w^* = \text{argmax}_{\mathcal{D}}\, \Gamma_w$.

5.1 Feasible states dependence on the DCB policy

In Table 2 we show the effect of applying different DCB policies on the average throughput experienced by WLANs A and B (Γ_A and Γ_B, respectively), and by the whole network (Γ) in scenarios *I* and *II* (presented in Figures 4 and 6, respectively). Let us first consider *Scenario I*. As explained in Section 4 and shown in Figure 5, the CTMN reaches 5 feasible states when WLANs implement AM. Instead, due to the fact that both WLANs overlap in channels 3 and 4 when transmitting in their whole allocated channels – i.e., $C_\text{A}^\text{tx} = C_\text{A} = \{1,2,3,4\}$ and $C_\text{B}^\text{tx} = C_\text{B} = \{3,4\}$, respectively – the SCB policy reaches just three feasible states. Such states correspond to those with a single WLAN transmitting, i.e., $\mathcal{S} = \{\emptyset,\ \text{A}_1^4,\ \text{B}_3^4\}$. In the case of OP, both WLANs are forced to pick just their primary channel for transmitting and, therefore, $\mathcal{S} = \{\emptyset, \text{A}_2^2, \text{B}_3^3, \text{A}_2^2\text{B}_3^3\}$. Notice that state A_2^2B_3^3 is feasible because A and B have different primary channels and do not overlap when transmitting in them.

Table 1: Tool features comparison.

Tool	SFCTMN	Komondor
Type	Analytical model	Simulator
DCB policies	✓	✓
Hidden & exposed nodes	✓	✓
RTS/CTS	✗[(a)]	✓
Flow-in-the-middle	✓	✓
Information asymmetry	✓	✓
Backoff collision	✗	✓
Scalable[(b)]	✗	✓

[(a)] RTS/CTS overhead considered in throughput measurements.
[(b)] Assumable execution time for up to 300 nodes.

Table 2: DCB policy effect on the average throughput [Mbps] in *Scenario I* and *Scenario II*. The values obtained through `Komondor` are displayed in parentheses, while the other correspond to the SFCTMN framework.

Policy	***Scenario I***				***Scenario II***			
$\mathcal{D}$	$\lvert\mathcal{S}\rvert$	Γ_A	Γ_B	$\overline{\Gamma}$	$\lvert\mathcal{S}\rvert$	Γ_A	Γ_B	$\overline{\Gamma}$
OP	4	109.36 (109.36)	109.36 (109.36)	109.36 (109.36)	4	109.36 (109.36)	109.36 (109.36)	109.36 (109.36)
SCB	3	132.75 (123.21)	132.75 (137.09)	132.75 (130.15)	3	102.65 (102.24)	102.65 (102.24)	102.65 (102.24)
AM	5	206.68 (204.70)	199.67 (201.91)	203.17 (203.31)	3	102.65 (102.24)	102.65 (102.24)	102.65 (102.24)
PU	10	142.70 (142.69)	142.00 (142.01)	142.35 (142.35)	6	109.30 (109.29)	109.30 (109.27)	109.30 (109.28)

The last policy studied is PU, which is characterized by providing further exploration of the

[7] For the sake of saving space, the evaluation setups and corresponding results of the scenarios considered through the paper are detailed in https://github.com/sergiobarra/data_repos/tree/master/barrachina2018performance

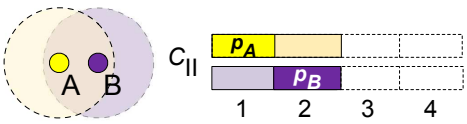


Figure 6: *Scenario II*. WLANs A and B are inside the carrier sense range of each other with potentially overlapping basic channels 1 and 2.

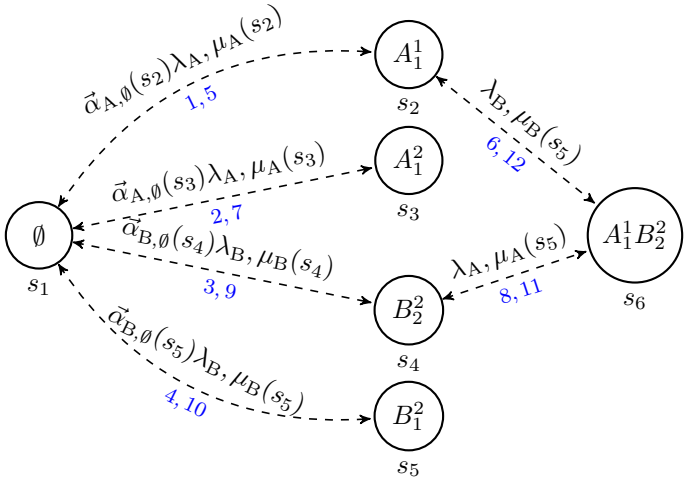


Figure 7: CTMN corresponding to *Scenario II*. Transitions edges are dashed for referring to those that may be given or not depending on the DCB policy. For instance, state s_6 is only reachable for the OP and PU policies. The discovery order of the states and transitions (displayed in blue) corresponds to the PU policy.

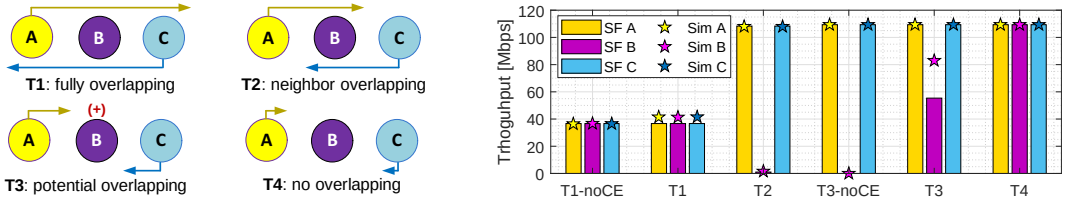
global state space Ψ. It usually allows expanding the feasible state space $\mathcal{S}$ accordingly because more transitions are permitted. In *Scenario I*, whenever the CTMN is in state $\emptyset$ and the backoff of A or B expires, the WLANs pick each of the possible available channels with the same probability. Namely, the CTMN will transit to A_2^2, A_1^2 or A_1^4 with probability 1/3 when A's backoff counter terminates, and to B_3^3 or B_3^4 with probability 1/2 whenever B's backoff counter terminates.

Likewise, if the system is in state B_3^3 and A terminates its backoff counter, the CTMN will transit to the feasible states $\mathrm{A}_2^2\mathrm{B}_3^3$ or $\mathrm{A}_1^2\mathrm{B}_3^3$ with probability 1/2. Similarly, whenever the system is in state A_2^2 or A_1^2, and B finishes its backoff, B will pick the transmission channels $\{3\}$ or $\{3,4\}$ with same probability 1/2 making the CTMN to transit to the corresponding state where both WLANs transmit concurrently. These probabilities are called transition probabilities and are represented by the vector $\vec{\alpha}_{\mathrm{X},s}(s')$. For instance, in the latter case, the probability to transit from $s = \mathrm{A}_2^2$ to $s' = \mathrm{A}_2^2\mathrm{B}_3^3$ when B terminates its backoff is $\vec{\alpha}_{\mathrm{B},\mathrm{A}_2^2}(\mathrm{A}_2^2\mathrm{B}_3^3) = 1/2$.

5.2 Instantaneous vs. long-term throughput

Intuitively, one could think that, as it occurs in *Scenario I*, always picking the widest channel found free by means of AM, i.e., maximizing the throughput of the immediate packet transmission (or instantaneous throughput), may be the best strategy for maximizing the long-term throughput as well. However, the *Scenario II* depicted in Figure 6 is a counterexample that illustrates such lack of applicable intuition. It consists of two overlapping WLANs as in *Scenario I*, but with different channel allocation: $C_\mathrm{A} = C_\mathrm{B} = \{1,2\}$ with $p_\mathrm{A} = 1$ and $p_\mathrm{B} = 2$, respectively. The CTMNs that are generated according to the different DCB policies – generalized to any value the corresponding transition probabilities $\vec{\alpha}_{\mathrm{A},\emptyset}, \vec{\alpha}_{\mathrm{B},\emptyset}$ may have – are shown in Figure 7.

Regarding the transition probabilities, Table 3 shows the vectors $\vec{\alpha}_{\mathrm{A},\emptyset}, \vec{\alpha}_{\mathrm{B},\emptyset}$ that are given for each of the studied DCB policies in *Scenario II*. Firstly, with OP, due to the fact that WLANs are only allowed to transmit in their primary channel, the CTMN can only transit from state $\emptyset$ to states A_1^1 or B_2^2, i.e., $\vec{\alpha}_{\mathrm{A},\emptyset}(s_2) = \vec{\alpha}_{\mathrm{B},\emptyset}(s_4) = 1$. Similarly, with SCB, WLANs can only transmit in their complete allocated channel, thus, when being in state $\emptyset$ the CTMN transits only to A_1^2 or B_1^2, i.e., $\vec{\alpha}_{\mathrm{A},\emptyset}(s_3) = \vec{\alpha}_{\mathrm{B},\emptyset}(s_5) = 1$. Notice that AM generates the same transition probabilities (and respective average throughput) than SCB because whenever the WLANs have the possibility to transmit – which only happens when the CTMN is in state $\emptyset$ – both A and B pick the widest channel available,



(a) Topologies. (b) Average throughput.

Figure 8: *Scenario III*. Yellow and blue arrows indicate the carrier sense range of WLANs A and C, respectively. *T1-noCE* and *T3-noCE* refer to topologies *T1* and *T3* when B does not accomplish the capture effect condition. *SF* and *Sim* refer to the values obtained through SFCTMN and `Komondor`, respectively.

i.e., $C_{\mathrm{A}}^{\mathrm{tx}} = C_{\mathrm{B}}^{\mathrm{tx}} = \{1, 2\}$. Finally, PU picks uniformly at random any of the possible transitions that A and B provoke when terminating their backoff in state $\emptyset$, i.e., $\vec{\alpha}_{\mathrm{A},\emptyset}(s_2) = \vec{\alpha}_{\mathrm{A},\emptyset}(s_3) = 1/2$ and $\vec{\alpha}_{\mathrm{B},\emptyset}(s_4) = \vec{\alpha}_{\mathrm{B},\emptyset}(s_5) = 1/2$, respectively.

Note that state $\mathrm{A}_1^1\mathrm{B}_2^2$, when both WLANs are transmitting at the same time, is reachable from states A_1^1 and B_2^2 for both OP and PU. In such states, when either A or B terminates its backoff and the other is still transmitting in its primary channel, only a transition to state $\mathrm{A}_1^1\mathrm{B}_2^2$ is possible, i.e., $\vec{\alpha}_{\mathrm{A},s_2}(s_6) = \vec{\alpha}_{\mathrm{B},s_4}(s_6) = 1$.

Table 3: Transition probabilities from state $\emptyset$ of WLANs A and B in *Scenario II* for different DCB policies.

$\mathcal{D}$	$\lvert\mathcal{S}\rvert$	$\vec{\alpha}_{\mathrm{A},\emptyset}(s_2)$	$\vec{\alpha}_{\mathrm{A},\emptyset}(s_3)$	$\vec{\alpha}_{\mathrm{B},\emptyset}(s_4)$	$\vec{\alpha}_{\mathrm{B},\emptyset}(s_5)$
OP	4	1.0	0.0	1.0	0.0
SCB	3	0.0	1.0	0.0	1.0
AM	3	0.0	1.0	0.0	1.0
PU	6	0.5	0.5	0.5	0.5

Interestingly, as shown in Table 2, applying OP in *Scenario II*, i.e., being conservative and unselfish, is the best policy to increase both the individual average throughput of A and B (Γ_{A}, Γ_{B}, respectively) and the system's aggregated one (Γ). Instead, being aggressive and selfish, i.e., applying SCB or AM, provides the worst results both in terms of individual and system's aggregate throughput. In addition, PU provides similar results than OP in average because most of the times that A and B terminate their backoff counter, they can only transmit in their primary channel since the secondary channel is most likely occupied by the other WLAN. In fact, state $\mathrm{A}_1^1\mathrm{B}_2^2$ is the dominant state for both OP and PU. Specifically, the probability of finding the CTMN in state $\mathrm{A}_1^1\mathrm{B}_2^2$, i.e., $\vec{\pi}_{s_6}$, is 0.9802 for OP and 0.9702 for PU, respectively. Therefore, the slight differences on throughput experienced with OP and PU are given because of the possible transition from $\emptyset$ to the states A_1^2 and B_1^2 in PU, where WLANs entirely occupy the allocated channel, thus preventing the other for decreasing its backoff.

Despite being a very simple scenario, we have shown that it is not straightforward to determine the optimal DCB policy that the AP in each WLAN must follow. Evidently, in a non-overlapping scenario, AM would be the optimal policy for both WLANs because of the non-existence of inter-WLAN contention. However, this is not a typical case in dense scenarios and, consequently, AM may not be adopted as de facto DCB policy, even though it provides more flexibility than SCB. This simple toy scenario also serves to prove that some intelligence should be implemented in the APs in order to harness the information gathered from the environment.

Concerning the throughput differences in the values obtained by SFCTMN and `Komondor`,[8] we note that the main disparities correspond to the AM and SCB policies. It is important to remark that while SFCTMN considers neither backoff collisions nor NAV periods, `Komondor` actually does so in a more realistic way. Therefore, in `Komondor`, whenever there is a slotted backoff collision, the RTS packets can be decoded by the STAs in both WLANs if the CE is accomplished. That is why the average throughput is consequently increased.

Regarding the NAV periods, an interesting phenomenon occurs in *Scenario I* when implementing SCB, AM or PU. While the RTS packets sent by B cannot be decoded by A because its primary

[8] In `Komondor` the throughput is simply computed as the number of useful bits (corresponding to data packets) that are successfully transmitted divided by the observation time of the simulation.

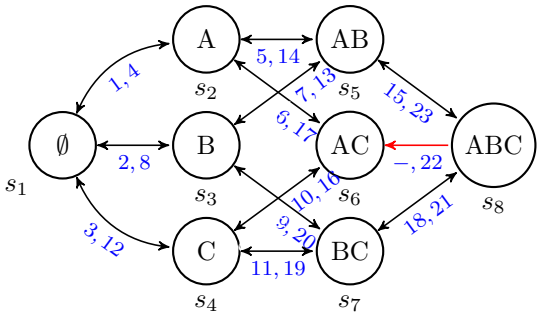


Figure 9: CTMN corresponding to *Scenario III-T3*. For the sake of visualization, neither the transition rates nor the transmission channels are included in the figure. The discovery order of the transitions is represented by the pairs in blue.

channel is always outside the possible transmission channels of B (i.e., $p_{\mathrm{A}} = 2 \notin C_{\mathrm{B}}^{\mathrm{tx}} = \{3, 3\}$ or $\{3, 4\}$), the opposite occurs when A transmits them. Due to the fact that the RTS is duplicated in each of the basic channels used for transmitting, whenever A transmits in its whole allocated channel, B is able to decode the RTS (i.e., $p_{\mathrm{B}} = 3 \in C_{\mathrm{A}}^{\mathrm{tx}} = \{1, 2, 3, 4\}$) and enters in NAV consequently.

5.3 Cumulative interference and flow starvation

When considering non-fully overlapping scenarios, i.e., where some of the WLANs are not inside the carrier sense range of the others, complex and hard to prevent phenomena may occur. As an illustrative example, let us consider the case shown in Figure 8a, where 3 WLANs sharing a single channel (i.e., $C_{\mathrm{A}} = C_{\mathrm{B}} = C_{\mathrm{C}} = \{1\}$) are deployed composing a line network. As the carrier sense range is fixed and is the same for each AP, by locating the APs at different distances we obtain different topologies that are worth to be analyzed. We name these topologies from *T1* to *T4* depending on the distance between consecutive APs, which increases according to the topology index. Notice that all the DCB policies discussed in this work behave exactly the same way in single-channel scenarios. Therefore, in this subsection, we do not make distinctions among them.

The average throughput experienced by each WLAN in each of the regions is shown in Figure 8b. Regarding topology *T1*, when APs are close enough to be inside the carrier sense range of each other in a fully overlapping manner, the medium access is shared fairly because of the CSMA/CA mechanism. For that reason, the throughput is decreased to approximately 1/3 with respect to topology *T4*. Specifically, the system spends almost the same amount of time in the states where just one WLAN is transmitting, i.e., $\pi(\mathrm{A}_1^1) = \pi(\mathrm{B}_1^1) = \pi(\mathrm{C}_1^1) \approx 1/3$. The neighbor overlapping case in topology *T2* is a clear case of flow-in-the-middle (FIM) starvation. Note that A and C can transmit at the same time whenever B is not active, but B can only do so when neither A nor C are active. Namely, B has very few transmission opportunities because A and C are transmitting almost permanently and B must continuously pause its backoff consequently.

An interesting and hard to prevent phenomenon occurs in the potential central node overlapping case at topology *T3*. Figure 9 shows the corresponding CTMN. In this case, the cumulated interference perceived by B from both A and C, prevents the former to decrease its backoff, thus generating a new FIM-like scenario such as in topology *T2*. However, in this case, B is able to decrement the backoff any time A or C are not transmitting. This leads to two possible outcomes regarding packet collisions. On the one hand, if the capture effect condition is accomplished by B (i.e., $\gamma_{\mathrm{B}} > \mathrm{CE}$) no matter whether A and C are transmitting, B will be able to successfully exchange packets and the throughput will increase accordingly. On the other hand, if the capture effect condition is not accomplished, B will suffer a huge packet error rate because most of the initiated transmissions will be lost due the hidden node effect caused by the concurrent transmissions of A and C (i.e., $\gamma_{\mathrm{B}} < \mathrm{CE}$ when A and C transmit). This phenomena may be recurrent and have considerable impact on high-density networks where multiples WLANs interact with each other. Therefore, it should be foreseen in order to design efficient DCB policies.

Finally, in topology *T4*, WLANs achieve the maximum throughput, as expected. The fact is that they are isolated (i.e., outside the carrier sense range of each other), which allows holding successful transmissions without having to pause their backoff.

Concerning the differences on the average throughput values estimated by SFCTMN and `Komondor`, we observe two phenomena with respect to backoff collisions in topologies *T1* and *T3*. In *T1*, due to the fact that simultaneous transmissions (or backoff collisions) are permitted and captured in

`Komondor`, the throughput is slightly smaller or higher depending on whether the capture effect condition is accomplished (T1) or not (T1-noCE), respectively. Note that backoff collisions have a negligible effect in T2 since B suffers from heavy FIM and it hardly ever transmits.

The most notable difference is given in *T3*. In this topology, SFCTMN estimates that B is transmitting just the 50.15% of the time. That is, since A and C operate like in isolation, most of the time they transmit concurrently, causing backoff freezing at B. However, `Komondor` estimates that B transmits about the 75% of the time, capturing a more realistic behavior. Such a difference is caused by the fact that the insensitivity property does not hold in this setup, since the Markov chain is not reversible. For instance, whenever the system is in state $s_6 = \mathrm{A}_1^1\mathrm{C}_1^1$ and A finishes its transmission (transiting to $s_4 = C_1^1$), B decreases its backoff accordingly while C is still active. Therefore, it is more probable to transit from s_4 to $s_7 = \mathrm{B}_1^1\mathrm{C}_1^1$ than to $s_6 = \mathrm{A}_1^1\mathrm{C}_1^1$ again because, in average, the remaining backoff counter of B will be smaller than the generated by A when finishing its transmission. This is in fact not considered by the CTMN, which assumes the same probability to transit from s_4 to s_6 than to s_7 because of the exponential distribution and the memoryless property.

5.4 Variability of optimal policies

Table 4: Policy combinations effect on throughput and fairness in the WLANs of *Scenario IV*. The complete table can be found at Appendix B of the supplementary material.

Policy			**States**	**Throughput [Mbps]**				**Fair.**
$\mathcal{D}_A$	$\mathcal{D}_B$	$\mathcal{D}_C$	$\lvert S\rvert$	Γ_A	Γ_B	Γ_C	Γ	$\mathcal{J}$
AM	AM	AM	5	199.96	3.58	199.96	**403.49**	0.67853
AM	PU	AM	10	149.41	62.45	149.41	361.27	0.89679
PU	AM	PU	25	109.84	108.44	109.84	328.12	**0.9999**
AM	AM	PU	9	111.31	106.91	110.33	328.55	**0.9997**

Most often, the best DCB policy for increasing the own throughput, no matter what policies the rest of WLANs may implement, is AM. Nonetheless, there are exceptions like the one presented in *Scenario II*. Besides, if achieving throughput fairness between all WLANs is the objective, other policies may be required. Therefore, there is not always an optimal common policy to be implemented by all the WLANs. In fact, there are cases where different policies must be assigned to different WLANs in order to increase both the fairness and individual throughput. For instance, let us consider another toy scenario (*Scenario IV*) using the topology *T2* of *Scenario III*, where three WLANs are located in a line in such a way that they are in the carrier sense range of the immediate neighbor. In this case, however, let us assume a different channel allocation: $C_\mathrm{A} = C_\mathrm{B} = C_\mathrm{C} = \{1, 2\}$ and $p_\mathrm{A} = p_\mathrm{C} = 1$, $p_\mathrm{B} = 2$.

Table 6 shows the individual and aggregated throughputs, and the Jain's fairness index for different combinations of DCB policies. We note that, while implementing AM in all the WLANs the system's aggregated throughput is the highest (i.e., $\Gamma = 403.49$ Mbps), the throughput experienced by B is the lowest (i.e., $\Gamma_\mathrm{B} = 3.58$ Mbps), leading to a very unfair FIM situation as indicated by $\mathcal{J} \approx 0.69$. We also find a case where implementing AM does not maximize the individual throughput of B. Namely, when A and C implement AM (i.e., $\mathcal{D}_A = \mathcal{D}_C = \mathrm{AM}$), it is preferable for B to implement PU and force states in which A and C transmit only in their primary channels. This increases considerably both the throughput of B and the fairness accordingly.

Looking at the fairest combinations, we notice that A, C or both must implement PU in order to let B transmit with a similar amount of opportunities. This is achieved by the stochastic nature of PU, which lets the CTMN to explore more states. Accordingly, B experiences the highest throughput and the system achieves complete fairness (i.e., $\mathcal{J} \approx 1$). Nonetheless, the price to pay is to significantly decrease the throughput of A and C. In this regard, other fairness metrics like $\mathcal{P}$ would determine the optimality of a certain combination of policies in a different way. For instance, in the scenario under evaluation, the combination providing the highest proportional fairness is AM-PU-AM (i.e., $\mathcal{P} \approx 6.16$) followed closely by the rest of scenarios with some WLAN implementing PU (i.e., $\mathcal{P} \approx 6.11$).

In essence, this toy scenario is a paradigmatic example showing that with less aggressive policies like PU (of probabilistic nature), not only more states in the CTMN can be potentially explored with respect to AM, but also the probability of staying in states providing higher throughput (or fairness) may increase. Therefore, since most of the times it does not exist a global policy that satisfies all

the WLANs in the system, different policies should be adopted depending on the parameter to be optimized.

6 Evaluation of DCB in dense WLANs

In this section, we study the effects of the presented DCB policies by simulating WLAN deployments of different node densities in `Komondor`. We first draw some general conclusions from analyzing the throughput and fairness when increasing the number of WLANs per area unit. Then, we discuss what is the optimal policy that a particular WLAN should locally pick in order to maximize its own throughput. The evaluation setup (11ax parameters, transmission power, path loss model, etc.) is extensively detailed in Appendix C of the supplementary material.

6.1 Network density vs. throughput

Figure 12 shows the general scenario considered for conducting the experiments presented in this section. For the sake of speeding up `Komondor` simulations, we assume that each WLAN is composed just of 1 AP and 1 STA. Due to the fact that APs and STAs are located randomly in the map, the number of STAs should not have a significant impact on the results because only downlink single-user traffic is assumed. Essentially, we consider a rectangular area $A_{\text{map}} = 100 \times 100$ m^2, where M WLANs are placed uniformly at random with the single condition that any pair of APs must be separated at least $d^{\text{min}}_{\text{AP-AP}} = 10$ m. The STA of each WLAN is located also uniformly at random at a distance $d_{\text{AP-STA}} \in [d^{\text{min}}_{\text{AP-STA}}, d^{\text{max}}_{\text{AP-STA}}] = [1, 5]$ m from the AP. The channelization $\mathcal{C}$ counts with $N_{\text{sys}} = 8$ basic channels (160 MHz) and follows the 11ax proposal (see Figure 2). Channel allocation is also set uniformly at random, i.e., every WLAN w is assigned a primary channel $p_{\text{w}} \sim U[1, 8]$ and allocated channel C_{X} containing $N_{\text{w}} \sim U\{1, 2, 4, 8\}$ basic channels.

For each number of WLANs studied (i.e., $M = 2, 5, 10, 20, 30, 40, 50$), we generate $N_{\text{D}} = 50$ deployments with different random node locations and channel allocations. Then, for each of the deployments, we assign to all the WLANs the same DCB policy. Namely, we simulate $N_{\text{M}} \times N_{\text{D}} \times N_{\text{P}} = 7 \times 50 \times 4 = 1400$ scenarios, where N_{M} is the number of different M values studied and N_{P} is the number of DCB policies considered in this paper. Besides, all the simulations have a duration of $T_{\text{obs}} = 20$ seconds.

In Figure 10, we show by means of boxplots the average throughput per WLAN for each of the presented DCB policies. As expected, when there are few WLANs in the area, the most aggressive policies (i.e., SCB and AM) provide the highest throughput. In contrast, PU, and especially OP, perform the worst, as they do not extensively exploit the free bandwidth. However, when the scenario gets denser, the average throughput obtained by all the policies except SCB tends to be similar. This occurs because WLANs implementing AM or PU tend to carry out single-channel transmissions since the PIFS condition for multiple channels are most likely not accomplished. In the case of SCB, part of the bandwidth in the WLAN's allocated channel will most likely be occupied by other WLANs, and therefore its backoff counter will get repeatedly paused. Thus, its average throughput in dense scenarios is considerably low with respect to the other policies.

In order to asses the use of the spectrum, we first define the average bandwidth usage of a WLAN w as $\text{E}[\text{BW}_w] = \frac{1}{T_{\text{obs}}} \sum_{c=1}^{N_{\text{sys}}} t_w^{\text{tx}}(c) \cdot |c|$, where $t_w^{\text{tx}}(c)$ is the time that WLAN w is transmitting in a channel containing at least the basic channel c. The average spectrum used by all the WLANs, i.e., $\text{BW} = \sum_{w=1}^{M} \text{E}[\text{BW}_w]$ is shown in Figure 11 for the different DCB policies. Similarly to the throughput, while OP and PU do not leverage the free spectrum in low-density scenarios, SCB and AM do so by exploiting the most bandwidth. Instead, when the number of nodes per area increases, SCB suffers from heavy contention periods, which reiterates the need for flexibility to adapt to the channel state. In this regard, we note that AM is clearly the policy exploiting the most bandwidth in average for any number of WLANs. Nonetheless, neither the average throughput per WLAN nor the spectrum utilization may be a proper metric when assessing the performance of the whole system. Namely, having some WLANs experiencing high throughput when some others starve is often a situation preferable to be avoided. In that sense, we focus on the fairness, which is both indicated by the *boxes* and outliers in Figure 10, and more clearly represented by the expected Jain's fairness index shown in Figure 11.

As expected, the policy providing the highest fairness is OP. In fact, no matter the channel allocation, WLANs only pick their primary channel for transmitting when implementing OP; hence the fairness is always maximized at the cost of probably wasting part of the frequency spectrum, especially when the node density is low. In this regard, PU also provides high fairness while exploiting

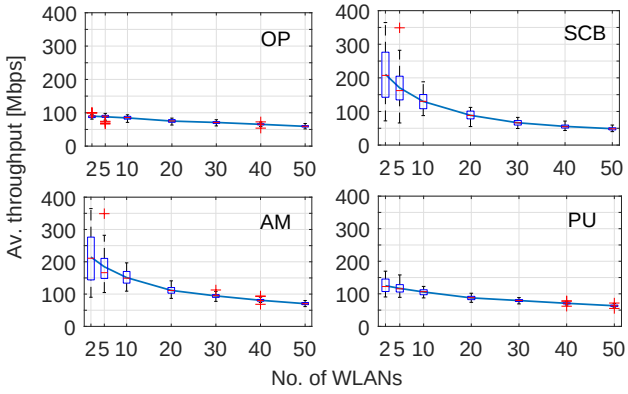


Figure 10: Node density effect on the average WLAN throughput. On each box, the central mark indicates the median, and the bottom and top edges of the box indicate the 25th and 75th percentiles, respectively. The whiskers extend to the most extreme data points not considered outliers, and the outliers are plotted individually using the '+' symbol.

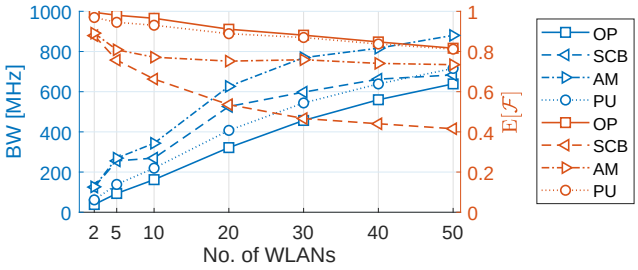


Figure 11: Node density effect on total bandwidth and fairness.

the spectrum more, which increases the average throughput per WLAN accordingly. Regarding the aggressive policies, SCB is clearly the most unfair policy due to its *all or nothing* strategy. Therefore, it seems preferable to prevent WLANs from applying SCB in dense scenarios because of the number of WLANs that may starve or experience really low throughput. However, even though being aggressive, AM is able to adapt its transmission channel to the state of the medium, thus providing both higher throughput and fairness. Still, as indicated by the *boxes* and *outliers* of Figure 10, AM is not *per se* the optimal policy. In fact, there are scenarios where PU performs better in terms of both fairness and throughput. Consequently, there is room to improve the presented policies with some smarter adaptation or learning approaches (e.g., tunning properly the transition probabilities $\vec{\alpha}$ when implementing stochastic DCB).

There are also some phenomena that we have observed during the simulations that are worth to be mentioned. Regarding backoff decreasing slowness, it can be the case that a WLAN w is forced to decrease its backoff counter very slowly due to the fact that neighboring WLANs operate in a channel including the primary channel of w. That is why more fairness is achieved with PU in dense networks as such neighboring WLANs do not always pick the whole allocated channel. Thus, they let w to decrease their backoff more often, and to proceed to transmit accordingly. Finally, concerning the transmission power and channel width, we have observed that transmitting just in the primary channel can also be harmful to other WLANs because of the higher transmission power used per 20 MHz channel. While this may allow using higher MCS and respective data rates, it may also cause packet losses in neighboring WLANs operating with the same primary channel due to heavy interference, especially for OP.

6.2 Local optimal policy

With the following experiment, we aim to identify what would be the optimal policy that a particular WLAN should adopt in order to increase its own throughput. In this case, we consider three rectangular maps (sparse, semi-dense and high-dense) with one WLAN (A) located at the center, and $M - 1 = 24$ WLANs spread uniformly at random in the area (see Figure 12). Besides, we now consider that WLAN A has $N_A \sim U[1, 20]$ STAs.[9] Channel allocation (including the primary

[9]Note that the average results considering just one STA per WLAN are really similar to the ones presented in this work.

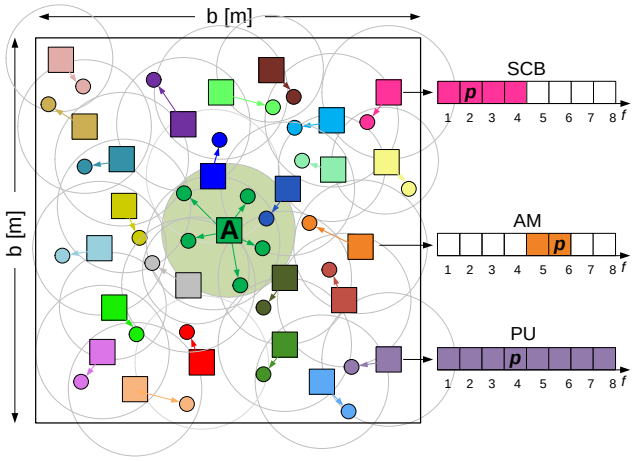


Figure 12: Central WLAN deployment with A placed in the middle and 24 WLANs spread uniformly at random.

channel) is set uniformly at random to all the WLANs, except A. While the central WLAN is also set with a random primary channel, it is allocated the widest channel (i.e., $C_{\mathrm{A}} = \{1, ..., 8\}$) in order to provide more flexibility and capture complex effects. While the DCB policies of the rest $M - 1$ WLANs are picked uniformly at random (i.e., they will implement OP, SCB, AM or PU with same probability 1/4), A's policy is set deterministically. Specifically, for each $N_b = 3$ map sizes (i.e., 75, 100 and 150 m^2), we generate $N_{\mathrm{D}} = 400$ deployments following the aforementioned conditions for each of the DCB policies that A can implement. That is, we simulate $N_{\mathrm{m}} \times N_{\mathrm{D}} \times N_{\mathrm{P}} = 4800$ scenarios. The simulation time of each scenario is also $T_{\mathrm{obs}} = 20$ seconds.

Figure 13 shows the average throughput experienced by A in the considered maps. The first noticeable result is that, in dense scenarios, SCB is non-viable for WLANs with wide allocated channels because they are most likely prevented to initiate transmissions. In fact, A is not able to successfully transmit any data packets in 61%, 84% and 99% of the scenarios simulated for SCB in the sparse, semi-dense and high-dense maps, respectively. Regarding the rest of policies, on average, A's throughput is higher when implementing AM in all the maps. Especially, AM (and SCB in some cases) stands out in sparse deployment. Nevertheless, for dense deployments, there is a clear trend to pick just one channel when implementing AM or PU. That is why OP provides an average throughput relatively close to the ones achieved by these policies. Nonetheless, as the high standard deviation of the throughput indicates, there are important differences regarding Γ_{A} among the evaluated scenarios. Table 5 compares the share of scenarios where AM or PU provide the highest individual throughput for A. We say that AM is better than PU if $\mathrm{E}[\Gamma_{\mathrm{A}}^{\mathrm{AM}}] - \mathrm{E}[\Gamma_{\mathrm{A}}^{\mathrm{PU}}] > \delta_{\Gamma}$, and vice versa. We use the margin $\delta_{\Gamma} = 1$ Mbps for capturing the cases where AM and PU perform similarly.

We see that in most of the cases AM performs better than PU. However, in some scenarios, PU outperforms AM. This improvement is accentuated for the high-density map, where a significant 57% (19% + 38%) of the scenarios achieve the same or highest throughput with PU. Also, there are a scenarios where the throughput experienced by PU with respect to AM is significantly higher (up to 61.5, 127.4 and 41.9 Mbps of improvement for the sparse, semi-dense and high-dense scenarios, respectively). This mainly occurs when the neighboring nodes occupy A's primary channel through complex interactions caused by information asymmetries, keeping its backoff counter frozen for long periods of time. These are clear cases where adaptive policies could importantly improve the performance.

Therefore, as a rule of thumb for dense networks, we can state that, while AM reaches higher throughput on average, stochastic DCB is less risky and performs relatively well. Nonetheless, even though PU is fairer than AM on average, it does not guarantee the absence of starving WLANs either. It follows that WLANs must be provided with some kind of adaptability to improve both the individual throughput and fairness with acceptable certainty.

7 Conclusions

In this work, we show the effects on spatially distributed WLANs of different DCB policies, including a new approach that stochastically selects the transmission channel width. By means of modeling

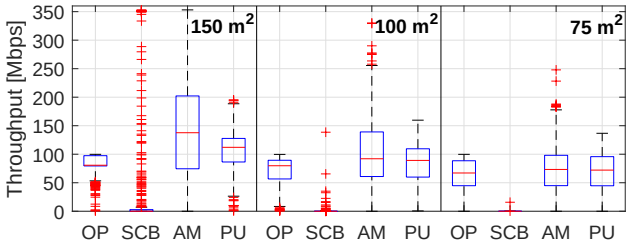


Figure 13: DCB policy effect on central WLAN A.

Table 5: Share of scenarios where AM or PU provide the highest individual throughput for WLAN A.

Deployment	PU best	AM best	Draw
Sparse 150 m^2	90/400 (23%)	260/400 (65%)	50/400 (13%)
Semi 100 m^2	61/400 (15%)	246/400 (62%)	93/400 (23%)
High 75 m^2	77/400 (19%)	172/400 (43%)	151/400 (38%)

WLAN scenarios through CTMNs, we provide relevant insights such as the instantaneous vs. long-term throughput dilemma, i.e., always selecting the widest available channel found free does not always maximize the individual throughput. Besides, we show that often there is not an optimal global policy to be applied to each WLAN, but different policies are required, specially in non-fully overlapping scenarios where *chain reaction* actions are complex to foresee.

Simulations corroborate that, while AM is normally the optimal policy to maximize the individual long-term throughput, there are cases, particularly in high-density scenarios, where stochastic DCB performs better both in terms of individual throughput and fairness among WLANs. We conclude that the performance of DCB can be significantly improved through adaptive policies capable of leveraging gathered knowledge from the medium and/or via information distribution. In this regard, our next works will focus on studying machine learning based policies to enhance WLANs spectrum utilization in high-density scenarios.

Acknowledgment

This work has been partially supported by the Spanish Ministry of Economy and Competitiveness under the Maria de Maeztu Units of Excellence Programme (MDM-2015-0502), and by a Gift from the Cisco University Research Program (CG#890107, Towards Deterministic Channel Access in High-Density WLANs) Fund, a corporate advised fund of Silicon Valley Community Foundation. The work done by S. Barrachina-Muñoz is supported by a FI grant from the Generalitat de Catalunya.

Appendices

A Dynamic channel bonding flowchart

In Figure 14, a simple flowchart of the transmission channel selection is shown.

B Optimal combination of DCB policies

Table 6 shows the individual and aggregated throughputs, and the Jain's fairness index for all the combinations of DCB policies in *Secnario IV*.

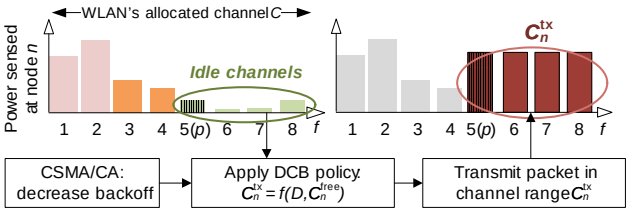


Figure 14: Flowchart of the transmission channel selection. In this example channel 5 is the primary channel and a DCB policy $\mathcal{D} = \text{AM}$ is applied.

Table 6: Effect of different DCB policy combinations on throughput and fairness in the WLANs of *Scenario IV*. The values obtained through `Komondor` are displayed in parentheses, while the other correspond to the SFCTMN framework.

Policy			**States**	**Throughput [Mbps]**				**Fairness**
$\mathcal{D}_A$	$\mathcal{D}_B$	$\mathcal{D}_C$	$\lvert S\rvert$	Γ_A	Γ_B	Γ_C	Γ	$\mathcal{J}$
AM	AM	AM	5	199.96 (199.35)	3.58 (4.76)	199.96 (199.37)	**403.49** (403.48)	0.67853 (0.68247)
AM	PU	AM	10	149.41 (128.72)	62.45 (86.88)	149.41 (128.72)	361.27 (344.31)	0.89679 (0.97131)
PU	AM	PU	14	109.84 (109.49)	108.44 (109.13)	109.84 (109.51)	328.12 (328.13)	**0.99996** (1.00000)
AM	AM	PU	9	111.31 (109.64)	106.91 (109.06)	110.33 (109.49)	328.55 (328.19)	**0.99970** (1.00000)
AM	PU	PU	12	111.29 (109.63)	106.94 (109.07)	110.33 (109.49)	328.56 (328.18)	**0.99971** (1.00000)
PU	PU	PU	14	109.85 (109.52)	108.44 (109.10)	109.85 (109.51)	328.13 (328.13)	**0.99996** (1.00000)

C Evaluation setup

The values of the parameters considered in the simulations are shown in Table 7. Regarding the path loss, we use the dual-slope log-distance model for 5.25 GHz indoor environments in room-corridor condition [32]. Specifically, the path loss in dB experienced at a distance d is defined by

$$\text{PL}(d) = \begin{cases} 53.2\text{+}25.8\log_{10}(d) & \text{if } d \leq d_1 \text{ m} \\ 56.4\text{+}29.1\log_{10}(d) & \text{otherwise} \end{cases}, \tag{1}$$

where $d_1 = 9$ m is the break point distance.

The MCS index used for each possible channel bandwidth (i.e., 20, 40, 80 or 160 MHz) was the highest allowed according to *i*) the power power budget established between the WLANs and their corresponding STA/s, and *ii*) the minimum sensitivity required by the MCSs. As stated by the 11ax amendment, the number of transmitted bits per OFDM symbol used in data transmissions is given by the channel bandwidth and the MCS parameters, i.e., $r = Y_\text{sc} Y_\text{m} Y_\text{c} V_\text{s}$, where Y_sc is the number of data sub-carriers, Y_m is the number of bits in a modulation symbol, Y_c is the coding rate, and $V_\text{s} = 1$ is the number of single user spatial streams (note that we only consider one stream per transmission).

The number of data sub-carriers depends on the transmission channel bandwidth. Specifically, Y_sc can be 234, 468, 980 or 1960 for 20, 40, 80, and 160 MHz, respectively. For instance, the data rate provided by MCS 11 in a 20 MHz transmission is $s = (234 \times 10 \times 5/6 \times 1)\sigma^{-1} = 121.9$ Mbps. However, control frames are transmitted in legacy mode using the basic rate $r_\text{leg} = 24$ bits per OFDM symbol of MCS 0, corresponding to $s_\text{leg} = 6$ Mbps since the legacy OFDM symbol duration σ_leg must be considered. With such parameters we can define the duration of the different packets

Table 7: Parameters considered in the presented scenarios.

Parameter	**Description**	**Value**
f_c	Central frequency	5 GHz
$\|c\|$	Basic channel bandwidth	20 MHz
L_D	Frame size	12000 bits
N_a	No. of frames in an A-MPDU	64
CW_{min}	Min. contention window	16
m	No. of backoff stages	5
MCS	11ax MCS index	0 - 11
η	MCS's packet error rate	0.1
CCA	CCA threshold	-82 dBm
P_{tx}	Transmission power	15 dBm
G_{tx}	Transmitting gain	0 dB
G_{rx}	Reception gain	0 dB
PL(d)	Path loss	see (1)
P_ν	Adjacent power leakage factor	-20 dB
CE	Capture effect threshold	20 dB
N	Background noise level	-95 dBm
T_e	Empty backoff slot duration	9 µs
T_{SIFS}	SIFS duration	16 µs
T_{DIFS}	DIFS duration	34 µs
T_{PIFS}	PIFS duration	25 µs
$T_{PHY\text{-}leg}$	Legacy preamble	20 µs
$T_{PHY\text{-}HE\text{-}SU}$	HE single-user preamble	164 µs
σ_{leg}	Legacy OFDM symbol duration	4 µs
σ	11ax OFDM symbol duration	16 µs
L_{BACK}	Length of a block ACK	432 bits
L_{RTS}	Length of an RTS packet	160 bits
L_{CTS}	Length of a CTS packet	112 bits
L_{SF}	Length of service field	16 bits
L_{MD}	Length of MPDU delimiter	32 bits
L_{MH}	Length of MAC header	320 bits
L_{TB}	Length of tail bits	18 bits

transmissions, and the duration of a successful and collision transmission accordingly:

$$
\begin{aligned}
T_{RTS} &= T_{PHY\text{-}leg} + \left\lceil \frac{L_{SF}+L_{RTS}+L_{TB}}{r_{leg}} \right\rceil \sigma_{leg}, \\
T_{CTS} &= T_{PHY\text{-}leg} + \left\lceil \frac{L_{SF}+L_{CTS}+L_{TB}}{r_{leg}} \right\rceil \sigma_{leg}, \\
T_{DATA} &= T_{PHY\text{-}HE\text{-}SU} + \\
&\quad + \left\lceil \frac{L_{SF}+N_a(L_{MD}+L_{MH}+L_D)+L_{TB}}{r} \right\rceil \sigma, \\
T_{BACK} &= T_{PHY\text{-}leg} + \left\lceil \frac{L_{SF}+L_{BACK}+L_{TB}}{r_{leg}} \right\rceil \sigma_{leg}.
\end{aligned}
$$

References

[1] B. Bellalta. IEEE 802.11ax: High-efficiency WLANs. *IEEE Wireless Communications*, 23(1):38–46, 2016.

[2] B. Bellalta, A. Checco, A. Zocca, and J. Barcelo. On the interactions between multiple overlapping WLANs using channel bonding. *IEEE Transactions on Vehicular Technology*, 65(2):796–812, 2016.

[3] M. Park. IEEE 802.11ac: Dynamic bandwidth channel access. In *Communications (ICC), 2011 IEEE International Conference on*, pages 1–5. IEEE, 2011.

[4] B. Bellalta, A. Faridi, J. Barcelo, A. Checco, and P. Chatzimisios. Channel bonding in short-range wlans. In *European Wireless 2014; 20th European Wireless Conference; Proceedings of*, pages 1–7. VDE, 2014.

[5] S. Vasthav, S. Srikanth, and V. Ramaiyan. Performance analysis of an IEEE 802.11ac WLAN with dynamic bandwidth channel access. In *Communication (NCC), 2016 Twenty Second National Conference on*, pages 1–6. IEEE, 2016.

[6] A. Faridi, B. Bellalta, and A. Checco. Analysis of Dynamic Channel Bonding in Dense Networks of WLANs. *IEEE Transactions on Mobile Computing*, 2016.

[7] Sergio Barrachina-Muñoz, Francesc Wilhelmi, Ioannis Selinis, and Boris Bellalta. Komondor: a wireless network simulator for next-generation high-density wlans. In *2019 Wireless Days (WD)*, pages 1–8. IEEE, 2019.

[8] L. Deek, E. Garcia-Villegas, E. Belding, S. Lee, and K. Almeroth. The impact of channel bonding on 802.11n network management. In *Proceedings of the Seventh Conference on emerging Networking EXperiments and Technologies*, page 11. ACM, 2011.

[9] M. Y. Arslan, K. Pelechrinis, I. Broustis, S. V. Krishnamurthy, S. Addepalli, and K. Papagiannaki. Auto-configuration of 802.11n WLANs. In *Proceedings of the 6th International Conference*, page 27. ACM, 2010.

[10] L. Deek, E. Garcia-Villegas, E. Belding, S. Lee, and K. Almeroth. Joint rate and channel width adaptation for 802.11 MIMO wireless networks. In *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2013 10th Annual IEEE Communications Society Conference on*, pages 167–175. IEEE, 2013.

[11] Y. Zeng, P. Pathak, and P. Mohapatra. A first look at 802.11 ac in action: Energy efficiency and interference characterization. In *Networking Conference, 2014 IFIP*, pages 1–9. IEEE, 2014.

[12] M. X. Gong, B. Hart, L. Xia, and R. Want. Channel bounding and MAC protection mechanisms for 802.11ac. In *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, pages 1–5. IEEE, 2011.

[13] M. Kim, T. Ropitault, S. Lee, and N. Golmie. A Throughput Study for Channel Bonding in IEEE 802.11ac Networks. *IEEE Communications Letters*, 2017.

[14] S. Barrachina-Muñoz, F. Wilhelmi, and B. Bellalta. To overlap or not to overlap: Enabling channel bonding in high-density WLANs. *Computer Networks*, 152:40 – 53, 2019.

[15] M. Han, Sami K., L. X. Cai, and Y. Cheng. Performance Analysis of Opportunistic Channel Bonding in Multi-Channel WLANs. In *Global Communications Conference (GLOBECOM), 2016 IEEE*, pages 1–6. IEEE, 2016.

[16] S. Khairy, M. Han, L. Cai, Y. Cheng, and Z. Han. A renewal theory based analytical model for multi-channel random access in ieee 802.11 ac/ax. *IEEE Transactions on Mobile Computing*, 2018.

[17] P. Huang, X Yang, and L. Xiao. Dynamic channel bonding: enabling flexible spectrum aggregation. *IEEE Transactions on Mobile Computing*, 15(12):3042–3056, 2016.

[18] A. Nabil, M. J. Abdel-Rahman, and A. B. MacKenzie. Adaptive Channel Bonding in Wireless LANs Under Demand Uncertainty. To appear in the Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Montreal, QC, Canada, October 2017.

[19] C. Kai, Y. Liang, T. Huang, and X. Chen. A Channel Allocation Algorithm to Maximize Aggregate Throughputs in DCB WLANs. *arXiv preprint arXiv:1703.03909*, 2017.

[20] Y. Chen, D. Wu, T. Sung, and K. Shih. Dbs: A dynamic bandwidth selection mac protocol for channel bonding in ieee 802.11 ac wlans. In *Wireless Communications and Networking Conference (WCNC), 2018 IEEE*, pages 1–6. IEEE, 2018.

[21] S. Byeon, H. Kwon, Y. Son, C. Yang, and S. Choi. Reconn: Receiver-driven operating channel width adaptation in ieee 802.11ac wlans. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pages 1655–1663. IEEE, 2018.

[22] P. Chatzimisios, A. Boucouvalas, and V. Vitsas. Performance analysis of IEEE 802.11 DCF in presence of transmission errors. In *Communications, 2004 IEEE International Conference on*, volume 7, pages 3854–3858. IEEE, 2004.

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

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

[25] B. Bellalta. Throughput Analysis in High Density WLANs. *IEEE Communications Letters*, 21(3):592–595, 2017.

[26] B. Nardelli and E. Knightly. Closed-form throughput expressions for CSMA networks with collisions and hidden terminals. In *INFOCOM, 2012 Proceedings IEEE*, pages 2309–2317. IEEE, 2012.

[27] S. C. Liew, C. H. Kai, H. C. Leung, and P. Wong. Back-of-the-envelope computation of throughput distributions in CSMA wireless networks. *IEEE Transactions on Mobile Computing*, 9(9):1319–1331, 2010.

[28] B. Bellalta, A. Zocca, C. Cano, A. Checco, J. Barcelo, and A. Vinel. Throughput analysis in CSMA/CA networks using continuous time Markov networks: a tutorial. In *Wireless Networking for Moving Objects*, pages 115–133. Springer, 2014.

[29] C. Kai, Y. Liang, T. Huang, and X. Chen. To bond or not to bond: An optimal channel allocation algorithm for flexible dynamic channel bonding in wlans. In *Vehicular Technology Conference (VTC-Fall), 2017 IEEE 86th*, pages 1–6. IEEE, 2017.

[30] H. Salameh, M. Krunz, and D. Manzi. Spectrum bonding and aggregation with guard-band awareness in cognitive radio networks. *IEEE Transactions on Mobile Computing*, 13(3):569–581, 2014.

[31] F Kelly. *Reversibility and stochastic networks*. Cambridge University Press, 2011.

[32] D. Xu, J. Zhang, X. Gao, P. Zhang, and Y. Wu. Indoor office propagation measurements and path loss models at 5.25 GHz. In *Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th*, pages 844–848. IEEE, 2007.

Sergio Barrachina-Muñoz is a Ph.D. candidate at Universitat Pompeu Fabra (UPF). He received his B.Sc. Degree in Telematics Engineering and M.Sc. in Intelligent Interactive Systems in 2015 and 2016, respectively, both from UPF. He joined the Wireless Networking Research Group in 2015, where he works under the supervision of Dr. Boris Bellalta. His main research interests are focused on machine learning for next-generation WLANs. Sergio is a recipient of a FI grant from the Generalitat de Catalunya.

Francesc Wilhelmi holds a B.Sc. degree in Telematics Engineering from Universitat Pompeu Fabra (2015). With the aim of applying new techniques for solving many well-known problems in communications, Francesc obtained his M.Sc. in Intelligent and Interactive Systems also from UPF in 2016. He is now a Ph.D. candidate in the Wireless Networking Group at UPF. The main topics of his PhD Thesis are related to applying reinforcement learning techniques to spatial reuse in high-density WLANs.

Boris Bellalta is an Associate Professor in the Department of Information and Communication Technologies (DTIC) at Universitat Pompeu Fabra (UPF). He obtained his degree in Telecommunications Engineering from Universitat Politècnica de Catalunya (UPC) in 2002 and the Ph.D. in Information and Communication Technologies from UPF in 2007. His research interests are in the area of wireless networks, with emphasis on the design and performance evaluation of new architectures and protocols. The results from his research have been published in more than 100 international journal and conference papers. He is currently involved in several international and national research projects, including the coordination of the ENTOMATIC FP7 collaborative project. At UPF he is giving several courses on networking, queuing theory and wireless networks. He is co-designer and coordinator of the interuniversity (UPF and UPC) master's degree in Wireless Communications.

8.2

To Overlap or not to Overlap: Enabling Channel Bonding in High-Density WLANs

Sergio Barrachina-Muñoz, Francesc Wilhelmi, Boris Bellalta

Abstract

Wireless local area networks (WLANs) are the most popular kind of wireless Internet connection because of their simplicity of deployment and operation. As a result, the number of devices accessing the Internet through WLANs such as laptops, smartphones, or wearables, is increasing drastically at the same time that applications' throughput requirements do. To cope with these challenges, channel bonding (CB) techniques are used for enabling higher data rates by transmitting in wider channels, thus increasing spectrum efficiency. However, important issues like higher potential co-channel and adjacent channel interference arise when bonding channels. This may harm the performance of the carrier sense multiple access (CSMA) protocol because of recurrent backoff freezing, while making nodes more sensitive to hidden node effects. In this paper, we address the following point at issue: is it convenient for high-density (HD) WLANs to use wider channels and potentially overlap in the spectrum? First, we highlight key aspects of DCB in toy scenarios through a continuous time Markov network (CTMN) model. Then, by means of extensive simulations covering a wide range of traffic loads and access point (AP) densities, we show that dynamic channel bonding (DCB) – which adapts the channel bandwidth on a per-packet transmission – significantly outperforms traditional single-channel on average. Nevertheless, results also corroborate that DCB is more prone to generate unfair situations where WLANs may starve. Contrary to most of the current thoughts pushing towards non-overlapping channels in HD deployments, we highlight the benefits of allocating channels as wider as possible to WLANs altogether with implementing adaptive access policies to cope with the unfairness situations that may appear.

1 Introduction

Although remarkable technological improvements have been achieved in the last decades, wireless local area networks (WLANs), with IEEE 802.11's Wi-Fi as the most widely used standard, still face important challenges that degrade their performance. Particularly, the frequency spectrum is becoming scarce and inefficient because of the rising number of wireless devices, the characteristically heterogeneous and random WLAN deployments, and the raising throughput demands (e.g., some virtual reality applications require more than 1 Gbps to operate properly [1]). All these circumstances lead to dense or high-dense (HD) scenarios with coexistence issues since WLANs try to selfishly serve their users in non-collaborative deployments.

As a result, there is a clear need of exploiting the spectrum in a more efficient way by maximizing transmissions' bandwidth. One of the most promising techniques to overcome such a challenge is channel bonding (CB). The main idea behind CB is to allow using wider bandwidths in order to transmit at higher transmission rates, increasing the potential throughput accordingly.[1] CB for WLANs was firstly introduced in the IEEE 802.11n-2009 amendment [2] by letting two separated 20 MHz channels (or basic channels) get combined into a 40 MHz channel. Later, IEEE 802.11ac-2013 [3] introduced the capability of transmitting also in 80 and 160 MHz channels. Future amendments like the IEEE 802.11ax-2019 [4] or EXtreme throughput (XT) [5], which is expected to support up to 320 MHz transmissions, will boost the use of wider channels. A survey of CB schemes for different types of wireless networks is provided in [6].

There are important drawbacks, however, when it comes to transmitting in wider channels: essentially, the larger the bandwidth used for transmitting, the wider the spectrum suffering from co-channel and

[1] According to the well-known Shannon-Hartley capacity theorem, the capacity (or raw throughput) of a channel increases with the bandwidth.

adjacent channel interference at neighboring nodes. That is, CB may be counterproductive since WLANs nearby are more likely to partially overlap, causing severe performance degradation due to the listen-before-talk nature of the carrier sense multiple access (CSMA) protocol. This effect is further exacerbated when following static channel bonding (SCB) [7]. Besides, the signal-to-interference-plus-noise ratio (SINR) at the receiver decreases for wider channels since the transmission power is spread through the whole transmission bandwidth (or subcarriers). Accordingly, situations like the hidden node problem are more prone to occur when implementing CB [8].

In this regard, dynamic channel bonding (DCB) allows adapting the selected transmission bandwidth to the channel status right before transmitting. This provides a higher degree of flexibility that improves the instantaneous throughput in a simple and efficient way. Then, we can differentiate two approaches with respect to spectrum management in WLANs: *i*) fostering transmissions in non-overlapping basic channels, or *ii*) enabling faster transmissions in wider channels that may potentially overlap in the spectrum. Alas, in high density (HD) spatially distributed scenarios,[2] the complex interrelations among nodes (located inside or outside the carrier sense range of each other) complicate the task of *a priori* estimating the optimal spectrum management approach on a per-WLAN basis.

Significant research has been conducted on the impact of DCB on spatially distributed WLANs' performance under saturation regimes. However, to the best of our knowledge, the effects of unsaturated traffic patters, which fit better to real world problems, are still unknown. While saturated regimes offer valuable insights on worst-case scenarios, WLANs are characteristically unsaturated with load patterns that deeply depend on the application/s being supported. In such scenarios, overlapping approaches seem to be even more convenient since the sensed channels usually remain free during larger periods of time.

In this paper, we compare the performance of traditional single-channel with channel bonding (including a stochastic variant) in networks under different traffic load regimes. To do so, we first introduce an analytical model to depict the behavior of the aforementioned CB approaches (or policies) in spatially distributed WLAN networks. The model is based on continuous time Markov networks (CTMNs) and captures both saturated and unsaturated regimes. Then, by means of simulations, we evaluate the performance of the CB policies in terms of throughput and delay in toy scenarios and HD WLAN deployments. We find that for low individual and neighboring traffic loads single-channel can improve CB in terms of delay since the time to access the channel is reduced. However, in general, DCB significantly outperforms traditional single-channel in most of the evaluated scenarios. Nonetheless, DCB is more prone to cause starvation and hidden nodes, which may lead to unfair scenarios with highly unbalanced performance among WLANs. Results suggest that future WLANs should be allocated all the available bandwidth and dynamically adapt to the spectrum.

The remainder of this article is organized as follows. In Section 2, we introduce CB for IEEE 802.11 WLANs, present the related work, and define the policies considered in this work. Then, in Section 3 we analytically model the interactions of DCB in spatially distributed deployments and point out key aspects of its performance through toy scenarios. DCB in HD WLANs is assessed in Section 4 by means of simulations. We conclude with some final remarks at Section 5.

2 Channel bonding

2.1 Channel bonding in IEEE 802.11 WLANs

CB is a technique whereby nodes, i.e., access points (APs) and stations (STAs), are allowed to use contiguous sets of available basic channels for their transmissions, thus potentially achieving higher throughput. Namely, by doubling the channel bandwidth, approximately the double data capacity can be achieved if the modulation coding scheme (MCS) is kept. CB for WLANs was firstly introduced in the IEEE 802.11n-2009 amendment [2], where high throughput (HT) STAs are allowed to transmit in more than one 20 MHz channel (or basic channel). Specifically, this amendment allowed bonding up to two basic channels composing a 40 MHz channel in the 2.4/5 GHz bands. Works in the literature [8–10] show important improvements achieved with CB in IEEE 802.11ac networks when properly adjusting the transmission power and data rates in WLANs operating at 5 GHz. Note that in the traditional 2.4

[2] In spatially distributed scenarios, nodes are not necessarily within the carrier sense range of each other. Thus, different groups of potentially overlapping WLANs may be given.

GHz band, CB has been found to be counterproductive since only three non-overlapping basic channels are allowed [11].

Later, the IEEE 802.11ac-2013 amendment [3] increased the maximum number of bonded 20 MHz channels to 8, allowing very high throughput (VHT) STAs to transmit in up to 160 MHz in the 5 GHz band. Currently, the IEEE Task Group 11ax (TGax) is working on the IEEE 802.11ax amendment [4], which is expected to be published by 2019. As in IEEE 802.11ac, high efficiency (HE) STAs are also allowed to bond up to 8 basic channels. Moreover, the frequency spectrum efficiency is expected to be boosted in IEEE 802.11ax by combining orthogonal frequency-division multiple access (OFDMA) [12] with preamble puncturing, an optional capability for enabling non-contiguous CB. Recently, the EXtreme throughput (XT) study group [5] has been created with the objective of increasing the peak throughput and capacity of WLANs. The motivation behind XT are the expectations that more than 1 GHz of additional unlicensed spectrum may be available around 2020. Thus, it will allow exploiting further spectrum at the 6 GHz band by transmitting in bandwidths up to 320 MHz. Note that in this manuscript we consider only contiguous bandwidths. That is, we use representative values of IEEE 802.11ax (e.g., 78.125 kHz subcarrier spacing or up to 1024-QAM MCS) to provide results in an appropriate scale when assessing its potential on contiguous DCB. Non-contiguous CB is left for future works, that can take the results presented in this paper as a baseline.

Notwithstanding, implementing CB in ever-increasingly complex WLAN deployments requires a careful balance of trade-offs. First, regarding channelization,[3] the density of neighboring nodes and the number of independent basic channels (which are regulated by governmental institutions) determine the feasibility of deploying interference-free networks. Essentially, as transmission channels get wider, frequency spectrum reuse becomes arduous, and the probability of packet collisions due to co-channel and adjacent channel interference increases. Secondly, the higher the bandwidth, the smaller the transmitted power per Hz and corresponding coverage range. This, on the one hand, reduces the interference with other WLANs operating in a (partially) overlapping spectrum. On the other hand, it reduces the SINR at the destination STAs, resulting in lower transmission rates if the receiver is not close enough to the transmitter. In addition, higher packet loss rates can arise because of the hidden node problem resulting from the lower SINR at higher bandwidths. In this regard, authors in [13] show that parameters like the strength of neighboring links and interferer loads strongly affect the performance of CB.

In short, the multiple spatial distribution factors such as transmission powers, clear channel assessment (CCA) levels, allocated channels, or environment's path loss, make it really difficult to generalize to an optimal set of rules for transmission channel selection. It follows that bandwidth adaptation is required in order to cope with the challenging scenarios of next-generation WLANs.

2.2 Related work

Since its emergence in the IEEE 802.11n amendment, CB has shown a great potential in WLANs. In [8–10], authors provides insight into the factors affecting CB performance in IEEE 802.11ac WLANs. These works show that increasing the bandwidth incurs into a lower SINR at the receivers, thus compromising its effectiveness. The fact is that transmitting in larger channel widths entails a reduction of Watt/Hertz, which accentuates the vulnerability to interference. In order to palliate the inherent limitations, several solutions have been proposed for 802.11 WLANs. Authors in [14] propose ARAMIS, a CB solution for simultaneously adapting the rate and channel bandwidth, boosted by spatial diversity in multiple-input multiple-output (MIMO) IEEE 802.11ac WLANs. Alternatively, [15] formulates a distributed CB scheme based on adaptive channel clear assessment (CCA).

Newer amendments like IEEE 802.11ac or IEEE 802.11ax broaden the capabilities of CB by providing larger channel widths (up to 160 MHz), thus accentuating the advantages and drawbacks of potential spectrum access solutions. Nonetheless, it is worth noting that next-generation deployments will be characterized by short-range WLAN scenarios [12]. This reduction of the AP-STA distance, altogether with the usage of techniques like spatial diversity MIMO[14], contributes to palliate issues regarding low SINR values. There are several works on CB for IEEE 802.11ac relying on simulations. For instance, [7, 16, 17] show significant throughput gains compared to single-channel. Authors in [18] conduct an empirical study corroborating the performance gains of CB in IEEE 802.11ac WLANs. However, these

[3]Channelization is the process of setting independent channels on neighboring APs in order to avoid interference among their WLANs.

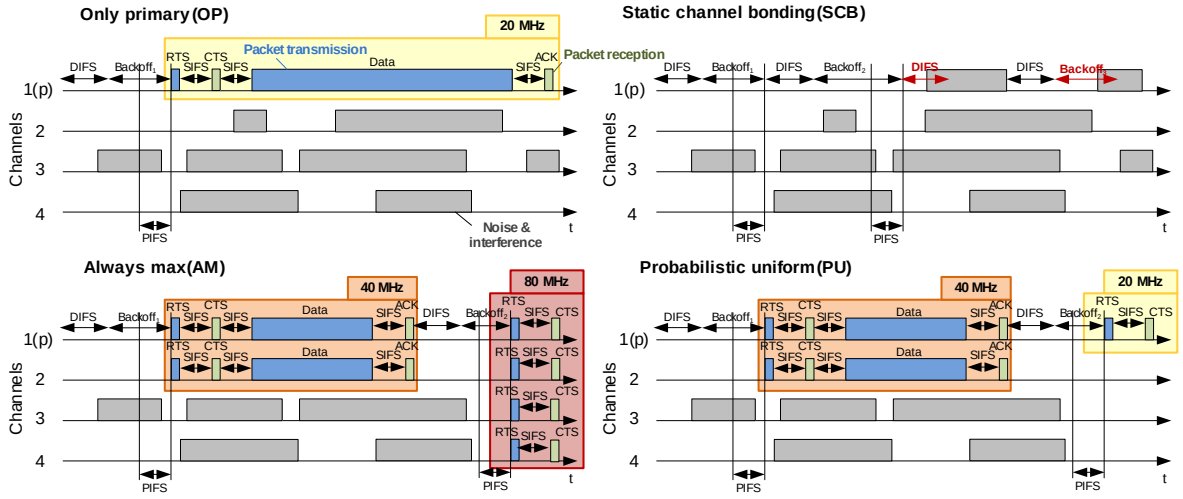


Figure 1: CSMA/CA temporal evolution of a node operating under different CB policies in a IEEE 802.11ax channelization scheme (from [17]). The DIFS and backoff durations in red represent that the sensed interference in the primary channel forces reseting the backoff procedure. While the duration of the legacy packets (RTS, CTS and ACK/BACK) is the same no matter the bandwidth, the data duration is clearly reduced when transmitted in 40 and 80 MHz.

works also highlight that such gains are importantly affected by the operation of neighboring networks. To the best of our knowledge, there are not experimental works yet on CB for preliminary implementations of IEEE 802.11ax.

One of the first fine-grained spectrum access design to dynamically change both the channel width and center frequency was formulated in [19]. This approach, compatible with IEEE 802.11a WLANs, was proven to significantly outperform static allocations. More recently, a frame-level wideband spectrum adaptation prototype supported by specially-constructed preambles and spectrum detection is presented in [20]. In this regard, a collision detection protocol for DCB is presented in [21]. Authors in [22] designed a dynamic bandwidth selection protocol to diminish the *carrier sensing decreasing* and *outside warning range problems*. Other works proposed a heuristic primary channel selection for CB users [23], a probabilistic spectrum distribution framework considering uncertain traffic load [24], or a prototype implementation for IEEE 802.11ac based on time-domain interference cancellation [25].

Analytical models have been also widely used in the literature on CB in WLANs[23, 26–29]. CB in short-range IEEE 802.11ac WLANs is assessed in [26], where authors show significant gains under moderate neighboring activity. High-density deployments are evaluated in [27], showing the exposure to unfairness situations. Opportunistic CB under the presence of legacy users is assessed in [28]. The authors in [30] propose an optimal channel allocation algorithm for DCB WLANs, showing by means of a CTMN model that the scheme with the least overlapped channels provides the highest throughput. An analytical throughput model under unsaturated traffic loads is formulated in [29] for CB in IEEE 802.11ac and IEEE 802.11ax. Also extendable to IEEE 802.11ax WLANs, authors in [23] propose a model based on renewal theory for studying the performance of CB with coexisting legacy users.

While the literature on CB and general spectrum access has extensively covered a wide variety of scenarios under different assumptions, this work focuses on spatially distributed HD WLANs under non-fully-backlogged buffers. The presented results allow us to assess whether (and under what circumstances) it is convenient or not to apply overlapping approaches in front of traditional non-overlapping schemes. To the best of our knowledge, this is the first work on DCB policies considering both spatial distribution and non-saturated traffic regimes in HD WLANs.

2.3 CB policies and CSMA/CA operation

All the aforementioned 802.11 WLAN standards operate essentially the same way when the well-known CSMA with collision avoidance (CSMA/CA) protocol is enabled. CSMA/CA works as follows: when a node n belonging to a WLAN w has a packet ready for transmission, it measures the power sensed in

its primary channel p_w, and determines whether it is idle or occupied according to the CCA level. Once p_w has been detected idle during a DCF interframe space (DIFS), n starts the backoff procedure by selecting a random initial value $b \in [0, \mathrm{CW} - 1]$, where CW is the contention window. After computing b, the node starts decreasing its counter while sensing the primary channel. Whenever the power sensed by n at p_w is higher than its CCA, the backoff is paused until p_w is detected free again, at which point the countdown is resumed. When the backoff timer expires, the node selects the transmission channel C_n^{tx} based on the set of idle basic channels[4] and on the implemented spectrum management rules. In this paper we refer to such rules as CB policies. Namely, when the backoff terminates, the node operates according to the implemented policy $\mathcal{D}$ as follows:

- **Only-primary (OP)**: if CB is not considered, we simply refer to the traditional single-channel (or only-primary operation), i.e., a node can only pick its primary channel for transmitting.
- **Static channel bonding (SCB)**: exclusively picks the whole allocated channel if found entirely free (i.e., all the basic channels inside the allocated channels are free).
- **DCB - Always-max (AM)**: picks the widest possible combination of basic channels found free.
- **DCB - Probabilistic uniform (PU)**: picks with same probability any of the possible combinations of basic channels found free.

Note that the computational complexity of the presented policies is very low and can be easily implemented in off-the-shelf STAs. In fact, the most complex one is PU, which does only require to compute the outcome of a uniform random variable to determine the number of 20 MHz-channels to bond given 4 possible outcomes at the most (i.e., 1, 2, 4 or 8).

The selected transmission channel is then used throughout the packet exchanges involved in a data packet transmission (i.e., RTS, CTS, data, and ACK). The duration of a successful transmission is then given by

$$T_{\mathrm{suc}} = T_{\mathrm{RTS}} + 3\,T_{\mathrm{SIFS}} + T_{\mathrm{CTS}} + T_{\mathrm{DATA}} + T_{\mathrm{BACK}} + T_{\mathrm{DIFS}} + T_{\mathrm{e}}, \tag{1}$$

where T_{SIFS} and T_{DIFS} are the Short Interframe Space (SIFS) and DIFS duration, respectively, and T_{e} is the duration of an empty backoff slot. T_{RTS}, T_{CTS}, T_{DATA} and T_{BACK} are the transmission duration of the RTS, CTS, data, and block acknowledgment (BACK) packets, respectively. Likewise, any other node that receives an RTS in its primary channel with enough power to be decoded will enter in network allocation vector (NAV) state, which is used for deferring channel access and avoiding packet collisions (especially those caused by hidden node situations).

In Figure 1, the temporal evolution of a node operating under the different CB policies is shown. In this example, the node is allowed to transmit in the set of basic channels $C_w = \{1(p), 2, 3, 4\}$, where $p_w = 1$ is the primary channel. While OP picks just the primary channel, the rest of policies try to bond channels in different ways. In this regard, SCB is highly inefficient in scenarios with partial interference. In fact, no packets can be transmitted with SCB in this example since the basic channel $\{3\} \in C_w$ is busy when both backoffs terminate. Instead, more flexible approaches like AM and PU are able to transmit more than one frame in the same period of time. On the one hand, AM adapts in an aggressive way to the channel state. In this example, it is able to transmit in 40 and 80 MHz channels at the end of the first and second backoff, respectively. On the other hand, the stochastic nature of PU makes it more conservative than AM. In the example, the node could transmit in 1 or 2 basic channels with the same probability (1/2) when the first backoff terminates. Likewise, after the termination of the second backoff, a channel composed of 1, 2 or 4 basic channels could be selected with equal probability too (1/3).

3 Understanding the interactions between spatially distributed WLANs

In this Section, we first analytically model the interactions given in spatially distributed WLANs under different traffic loads. Essentially, we show that the probabilities of transiting from one state to another

[4]Note that, in order to include secondary channels for transmitting, a WLAN must listen to them free during at least a Point coordination function (PCF) Interframe Space (PIFS) period before the backoff counter terminates, as shown in Figure 1.

in the generated CTMNs are determined by the CB policies of the WLANs in the network. Later, we present two toy scenarios and simulate them by means of CTMNs and `11axHDWLANsSim`,[5] release v1.2.1b of the Komondor wireless simulator [31].

3.1 The CTMN model for WLANs

CTMNs have been widely used in the literature to model the behavior of WLAN networks. An approach which accurately models the behavior of unsaturated CSMA/CA networks operating in single-channel was introduced in [32]. Such a model is extended in [27] to capture the coupled dynamics of a group of overlapping WLANs using CB. Later, authors in [17] introduced a framework (SFCTMN) which extended the CTMN algorithm presented in [33] for characterizing CB policies in spatially distributed scenarios where all WLANs are saturated. However, to the best of our knowledge, spatial distribution effects like WLAN starvation are not considered in works studying DCB under unsaturated regimes.

Below we model such scenarios through CTMNs too. To do so, we extend the model presented in [17] by considering unsaturated traffic loads as proposed in [32]. For simplicity, we consider only downlink traffic and that each WLAN is composed by one access point (AP) and one station (STA). Hence, we simply refer to the WLAN activity as a single entity.

3.1.1 Assumptions and implications

Modeling WLAN scenarios with CTMNs requires the backoff and transmission times to be exponentially distributed. We also assume that the propagation delay between any pair of nodes is negligible. This has a main implication: the probability of slotted backoff collisions between two or more nodes within carrier sense range is zero. Nonetheless, packet collisions resulting from the cumulated interference of simultaneous transmissions of nodes outside the carrier sense range are possible. Besides, an infinite maximum number of retransmissions per packet is assumed. Note that effect of assuming an infinite maximum number of retransmissions is almost negligible in most of the cases because of the small probability of retransmitting a data packet more than a few times [34].

3.1.2 States in the CTMN

A state s in the CTMN is defined by the set of active WLANs (i.e., that are transmitting) and the basic channels selected for the transmission. The set of feasible states is represented by $\mathcal{S}$. Essentially, with slight abuse of notation, we say that a WLAN w is active in state s, i.e., $w \in s$ if it is transmitting, and inactive otherwise. States are represented by the most left and most right basic channels used in the transmission channels of each of the active WLANs. For instance, in state $s = \mathrm{A}_2^2\mathrm{B}_1^4$, there are two active WLANs: A and B. While A is transmitting in the basic channel $C_{\mathrm{A}}^{\mathrm{tx}} = \{2\}$ (20 MHz), B is doing so in a bonded channel $C_{\mathrm{B}}^{\mathrm{tx}} = \{1, 2, 3, 4\}$ (80 MHz). The state in which there is no active WLAN is represented by $\emptyset$.

A transition between two states s and s' in the CTMN has a corresponding transition rate $Q_{s,s'}$. For *forward* transitions, the average packet transmission attempt rate is $\rho_w \lambda_w$, where $\lambda = 1/(\mathrm{E}[B] \cdot T_{\mathrm{slot}})$, being $\mathrm{E}[B]$ the expected backoff duration in time slots. Parameter ρ_w is the long-run stationary probability that WLAN w has packets ready for transmission when the primary channel is sensed idle and so the backoff counter is active. Consequently, ρ_w depends on the traffic load ℓ_w of WLAN w. Note that a WLAN becomes saturated ($\rho_w = 1$) whenever it is not able to carry its traffic load, i.e., whenever it generates more packets than the ones it transmits. For *backward* transitions, the departure rate (μ) depends on the duration of a successful transmission (T_{suc}), which in turn depends on both the data rate (r) given by the selected MCS and transmission channel width, and on the average data packet length ($\mathrm{E}[L]$). Thus, we simply say that the data rate of a WLAN w depends on the state of the system, which contains such information, i.e., $\mu_w(s)$.

[5] All of the source code of Komondor is open, encouraging sharing of algorithms between contributors and providing the ability for people to improve on the work of others under the GNU General Public License v3.0. The repository can be found at `https://github.com/wn-upf/Komondor`.

3.1.3 Analytical performance metrics

The equilibrium distribution vector $\vec{\pi}$ represents the fraction of time the system spends in each state. We define $\vec{\pi}_s$ as the probability of finding the system at state s. Hereof, in continuous-time Markov processes with stationary distribution, $\vec{\pi}$ is given by solving the system of equations $Q\vec{\pi} = 0$, where the matrix item $Q_{s,s'}$ is the transition rate from state s to s'. Once $\vec{\pi}$ is computed, estimating the average throughput experienced by each WLAN is straightforward. Specifically, the average throughput of WLAN w is

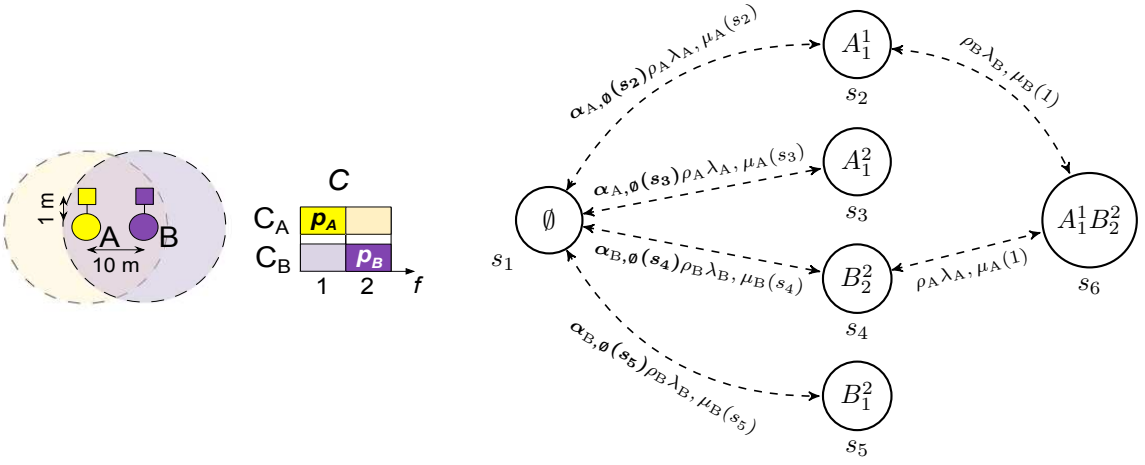
$$\Gamma_w := \mathrm{E}[L]\Big(\sum_{s\in\mathcal{S}}\{\gamma_w(s) > \mathrm{CE} : 0, 1\}\mu_w(s)\pi_s\big(1-\eta\big)\Big), \tag{2}$$

where $\mathrm{E}[L]$ is the expected data packet length, $\gamma_w(s)$ is the SINR perceived by the receiving STA in WLAN w in state s, CE is the capture effect threshold, and η is the MCS packet error probability.[6]

Note that the unknown ρ parameters must be obtained by solving a non-linear system of equations, which in general does not have a closed-form. As done in [27], in this work we use an iterative fixed-point approach for updating the ρ values until the throughput of all the WLANs converges to their corresponding traffic load, or they become saturated.

3.2 Constructing CTMNs for CSMA/CA WLANs

Let us consider the toy *Scenario I* shown in Figure 2(a), which is composed of two potentially overlapping WLANs, to depict a small example of how CTMNs are constructed. The channel allocation of this scenario can be defined as $\mathcal{C}$: $C_\mathrm{A} = \{1(p), 2\}$ with $p_\mathrm{A} = 1$, and $C_\mathrm{B} = \{1, 2(p)\}$ with $p_\mathrm{B} = 2$. That is, there are two basic channels in the system, and the set of valid transmission channels according to the IEEE 802.11ax channel access scheme is $\{\{1\}, \{2\}, \{1, 2\}\}$. We say that both WLANs are potentially overlapping because they are inside the carrier sense range of each other and thus their signals will overlap when transmitting in the same channel at the same time t, i.e., when $C_\mathrm{A}^\mathrm{tx}(t) \cap C_\mathrm{B}^\mathrm{tx}(t) \neq \emptyset$. In this case, due to the primary channel allocation, A and B will only overlap when both transmit in their whole allocated channel $\{1, 2\}$.



(a) Scheme of *toy scenario I*.

(b) CTMN corresponding to *toy scenario I*.

Figure 2: *Toy scenario I.* a) WLANs A and B are inside the carrier sense range of each other with potentially overlapping basic channels 1 and 2. b) Note that certain states and transition edges are not given in the CTMN depending on the applied combination of CB policies. For instance, state s_6 is only reachable for the OP and PU policies, while the transitions from s_1 to s_3 and s_5 are not possible when OP is implemented.

[6] A maximum decoding packet error rate of 10% is usually tried to be guaranteed when selecting the MCS index in 802.11 devices.

Different feasible states and *forward* transitions may exist in the CTMN depending on the implemented CB policies. Every feasible transition is weighted by a transition probability vector $\alpha_{w,s}(s')$ whose elements determine the probability of WLAN w to transit from state s to s'. Table 1 collects the number of feasible states ($|\mathcal{S}|$) and transition probabilities that are given for each of the studied CB policies in *Scenario I*. The corresponding CTMNs are shown in Figure 2(b).

Table 1: Transition probabilities from state $\emptyset$ of WLANs A and B in *toy scenario I* for different CB policies.

$\mathcal{D}$	$\lvert\mathcal{S}\rvert$	$\vec{\alpha}_{\mathrm{A},\emptyset}(s_2)$	$\vec{\alpha}_{\mathrm{A},\emptyset}(s_3)$	$\vec{\alpha}_{\mathrm{B},\emptyset}(s_4)$	$\vec{\alpha}_{\mathrm{B},\emptyset}(s_5)$
OP	4	1.0	0.0	1.0	0.0
SCB	3	0.0	1.0	0.0	1.0
AM	3	0.0	1.0	0.0	1.0
PU	6	0.5	0.5	0.5	0.5

For instance, with OP, since WLANs are only allowed to transmit in their primary channel, the CTMN can only transit from state $\emptyset$ to states A_1^1 or B_2^2, i.e., $\alpha_{\mathrm{A},\emptyset}(s_2) = \alpha_{\mathrm{B},\emptyset}(s_4) = 1$. Instead, with SCB, WLANs can only transmit in their complete allocated channel, thus, when being in state $\emptyset$ the CTMN transits to the *all or nothing* states A_1^2 or B_1^2, i.e., $\alpha_{\mathrm{A},\emptyset}(s_3) = \alpha_{\mathrm{B},\emptyset}(s_5) = 1$. Notice that in this particular case AM generates the same transition probabilities (and respective average throughput) than SCB because whenever the WLANs have the possibility to transmit – which only happens when the CTMN is in state $\emptyset$ – they pick the widest channel available, i.e., $\{1, 2\}$. Finally, PU picks uniformly at random any of the possible transitions when the backoff terminates in $\emptyset$, i.e., $\alpha_{\mathrm{A},\emptyset}(s_2) = \alpha_{\mathrm{A},\emptyset}(s_3) = 1/2$ and $\alpha_{\mathrm{B},\emptyset}(s_4) = \alpha_{\mathrm{B},\emptyset}(s_5) = 1/2$, respectively.

3.3 Empirical performance metrics and toy evaluation setup

Note that, even though the analytical expression of the throughput by CTMN (2) is pretty accurate [17], there are other performance metrics hard to capture with enough accuracy because of the required assumptions like the nonexistence of backoff collisions. That is why in this work we rely on the event-based wireless network simulator `11axHDWLANsSim` [31]. The performance metrics considered in this work are defined as follows:

- **Throughput $\boldsymbol{\Gamma}$**: total number of data bits successfully sent (i.e., acknowledged) during the observation time. That is, only the useful data (i.e., no headers) of each of the transmitted frames is considered for computing the throughput.
- **Access delay $\boldsymbol{\delta}$**: average duration between two consecutive channel accesses whenever there is backlogged data.
- **Packet delay $\boldsymbol{d}$**: average delay between a packet arrival (insertion in the buffer queue) and its corresponding acknowledgment after being transmitted.
- **Drop ratio $\boldsymbol{\varphi}$**: ratio of packets that are dropped by the buffer. A packet is dropped if it is generated when the queue of the buffer is full (i.e., when the buffer already has N_b packets at the queue).
- **No. of aggregated packets per frame $\boldsymbol{n_a}$**: average number of aggregated packets per frame. A frame can contain up to N_a packets.

The parameters of the simulation setups evaluated in this work[7] are collected in Table 2 of the Appendix, which correspond to the IEEE IEEE 802.11ax simulation setup presented in [17]. However, for the sake of simplicity, in these toy scenarios we consider for the moment no MCS error rate ($\eta = 0$) and highest MCS corresponding to 1024-QAM 5/6. Regarding the traffic load, note that we consider that a WLAN w generates a data packet every $t_w \sim \text{Exponential}(1/\ell_w)$, following a Poisson process.

[7]For the sake of saving space, the full details of the evaluation setups (e.g., nodes positions) and corresponding results of the scenarios considered through the paper are detailed in `https://github.com/sergiobarra/data_repos/tree/master/barrachina2018tooverlap`.

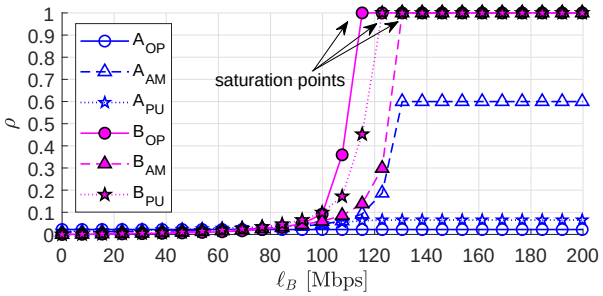


Figure 3: Saturation in *toy scenario I*. The long-run stationary probability ρ is estimated through SFCTMN. Traffic load of WLAN A is fixed to $\ell_A = 76.8$ Mbps. WLAN B gets saturated at different traffic loads ℓ_B depending on the CB policy.

3.4 Toy scenario I: to overlap or not to overlap?

In Figure 3 there is plotted the long-run stationary probability ρ of both WLANs when operating under different policies and traffic loads. Likewise, in Figure 4, we plot the average throughput, access delay, packet delay, drop ratio and number of aggregated data packets per frame. While we keep the traffic load of A constant to $\ell_{\mathrm{A}} = 76.8$ Mbps, the load of B is the x-axis independent variable $\ell_{\mathrm{B}} \in [0, 240]$ Mbps. We assume that both WLANs implement exactly the same policy in each case.

Given the duration of a successful slot in a CSMA/CA IEEE 802.11ax network (1), the maximum capacity for a successful transmission of a frame containing N_a packets, i.e., $r = N_a L_D / T_{\mathrm{suc}}(N_a)$, using MCS 11 is $r_{20} = 109.71$ Mbps for single-channel (20 MHz) transmissions, and $r_{40} = 207.18$ Mbps for two bonded channels (40 MHz) transmissions. Thus, even in isolation, whenever the traffic load of a WLAN surpasses that rates, it gets saturated. Note also the effect of the overhead introduced by the PHY and MAC layers since the raw transmission data rates provided by 1024-QAM 5/6 are $r^*_{20} = 121.9$ and $r^*_{40} = 243.8$ Mbps, for 20 and 40 MHz, respectively.

The saturation points of WLAN B are shown in Figure 3, where for OP it gets saturated at approximately $\ell_B \approx r_{20}$. Instead, regarding A's saturation point, we note that, as single-channel capacity already copes with ℓ_{A} (i.e., $\ell_{\mathrm{A}} < r_{20}$), it never gets saturated (i.e., $\rho_{\mathrm{A}} < 1$) no matter neither the policy selected nor ℓ_{B}.[8] As expected, with AM, B gets saturated for higher ℓ_{B} since more frames can be transmitted per unit of time. Note that in isolation, B would saturate for a ℓ_{B} close to r_{40}. In this case, however, the whole channel is shared with A when both implement AM and saturation is reached at a lower value $\ell_{\mathrm{B}} \approx 130$ Mbps.

In terms of throughput, the higher the traffic load required to saturate a WLAN, the higher its potential value. That is, AM provides the highest Γ_{B} for high ℓ_{B}, while any policy combination copes with ℓ_{A} (i.e., $\ell_{\mathrm{A}} = \Gamma_{\mathrm{A}}$). Regarding the CTMN model, all the throughput estimations completely match the simulator results with exception of the slight difference given in the $\mathrm{B_{AM}}$ and $\mathrm{B_{PU}}$ curves. On the one hand, the main reason lies in the fact that the CTMN model assumes that all the frames contain exactly N_a packets, while the simulator has not such a restriction. Thus, frames containing less than N_a packets are completely possible in the simulations conducted. This effect is specially noticeable at curve $\mathrm{B_{PU}}$. On the other hand, while simultaneous slotted backoff terminations are not captured by the CTMN model, the simulator does so. Hence, since in this particular scenario concurrent transmissions are decodable due to the proximity AP-STA, $\mathrm{B_{AM}}$ is slightly smaller than the simulated one.

As it occurs with the throughput, for high ℓ_B, it is more convenient for B to use the aggressive CB approach provided by AM in order to decrease the delay. However, interestingly, we note that for low ℓ_B, OP is the best policy since the delay to access the channels is significantly reduced compared to AM. The reason is that with AM the two WLANs must share the channel as they transmit using the full 40 MHz spectrum. Consequently, backoff counters get frozen during larger periods of time and the delay between consecutive channel accesses increases accordingly. This effect can also be seen in the average number of packets aggregated per frame, where, for low traffic loads, AM aggregates more packets on average since the buffer of one WLAN is able to be filled with more packets during the transmission of

[8] Note that $\ell < r$ is a mandatory condition in order to ensure unsaturated regimes. The reason lies in the overheads caused by the headers, control packets and inter frame spaces of the MAC layer.

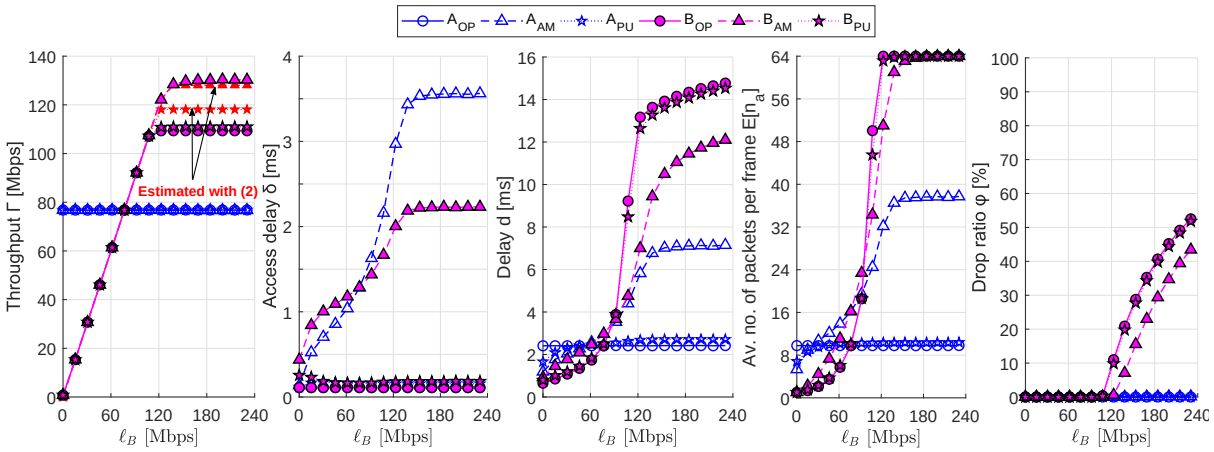


Figure 4: Performance metrics of *toy scenario I*. Results correspond to `11axHDWLANsSim` simulations of 1000 seconds each. Traffic load of WLAN A is fixed to $\ell_A = 76.8$ Mbps. The throughput curves in red correspond to the only values obtained by the CTMN model (2) that do not completely match the simulator results.

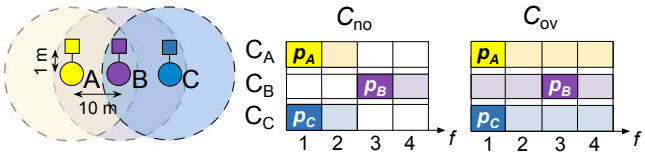


Figure 5: *Toy scenario II*. A *neighbor overlapping* network where two channel allocations are considered.

the other. In turn, when the backoff expires, larger frames are sent.

In this particular scenario, we see that an overlapping approach is the best both in terms of delay and throughput when the traffic load is moderate to high. Instead, for low traffic loads, the delay is reduced with OP, since it avoids overlaps making channel access independent on the other WLAN's activity.

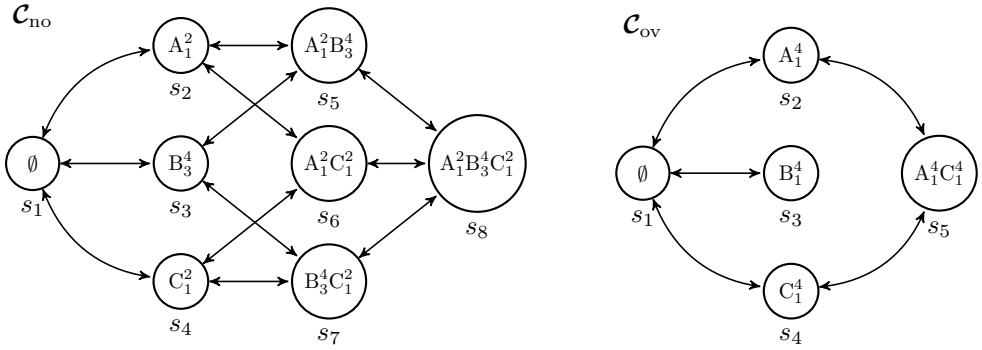
3.5 Toy Scenario II: drawbacks of overlapping

Toy scenario II shown in Figure 5 comprises a network of three WLANs where the central one (B) is in the carrier sense range of the other two (A and C). Instead, A and C are outside the carrier sense of each other (i.e., the edge WLANs never overlap in any basic channel). All the WLANs implement AM. We consider two different channel allocations for comparing the non-overlapping vs. overlapping approaches, respectively:

- $\mathcal{C}_{\text{no}}$: $C_{\text{A}} = C_{\text{C}} = \{1(p), 2\}$ and $C_{\text{B}} = \{3(p), 4\}$.
- $\mathcal{C}_{\text{ov}}$: $C_{\text{A}} = C_{\text{C}} = \{1(p), 2, 3, 4\}$ and $C_{\text{B}} = \{1, 2, 3(p), 4\}$.

Note that, as shown in Figure 6(a) and Figure 6(b), different states are reached depending on the channel allocation of the WLANs. On the one hand, $\mathcal{C}_{\text{no}}$ allows any combination of concurrent transmissions by sacrificing potential allocated bandwidth. On the other hand, WLANs must contend for the channel when $\mathcal{C}_{\text{ov}}$ is allocated. In turn, their data transmission rate is approximately doubled with respect to $\mathcal{C}_{\text{no}}$ (i.e., $r_{80} \approx 2\, r_{40}$). In Figure 7, the packet delay, throughput, and drop ratio experienced by the WLANs under different traffic loads is shown. Note that A and C behave exactly the same way since they are symmetrically deployed and have same channel allocation. We evaluate the aforementioned performance metrics for three different values of ℓ_{B} (i.e., 76.8, 192.0 and 307.2 Mbps) and a several values of $\ell_A = \ell_C = \ell_e$ in the range $[0, 600]$ Mbps.

As expected, for the non-overlapping channelization ($\mathcal{C}_{\text{no}}$), there is no dependence among WLANs. Essentially, performance is just a consequence of the WLANs' own traffic load. Thus, saturation is



(a) Non-overlapping channel allocation $\mathcal{C}_{\text{no}}$. (b) Overlapping channel allocation $\mathcal{C}_{\text{ov}}$.

Figure 6: CTMNs of *Toy scenario II*.

reached when ℓ approximates r_{40} since each WLAN is allocated two basic channels. This saturation effect can be seen in the throughput and drop ratio curves in Figure 7. Note that packets start to be dropped in saturation, i.e., when $\rho \approx 1 \rightarrow \varphi > 0$. Besides, due to the fact that channels do not overlap, the maximum throughput reachable by any WLAN is slightly less than r_{40}.

Regarding the overlapping channel allocation ($\mathcal{C}_{\text{ov}}$), results show that B's performance is really deteriorated when the traffic load of A and C increases. Essentially, while A and C can transmit at the same time whenever B is not active, B can only do so when neither A nor C are active. This is a clear case of unfair WLAN starvation. Namely, the larger ℓ_e, the fewer the transmission opportunities for B, like A and C transmit during the majority of the time. As a consequence of such flow-in-the-middle (FIM) starvation, B suffers from high delay, low throughput, and high drop ratio. Interestingly, ℓ_{B} does not practically affect to d_{A} or d_{C}, since B starves even for $l_{\text{B}} = 76.8$ Mbps when ℓ_e is high. Instead, for low ℓ_e, the overlapping setup is more convenient for B when its traffic load is high (see performance for $l_{\text{B}} = 307.2$ Mbps).

As for the throughput estimation of the CTMN model, we note that only WLAN B at the overlapping allocation shows different numerical results, even though following the same trend as the simulator. The main reason lies in two aspects. First, a fixed number of N_a data payloads aggregated per frame is assumed in the CTMN model for every WLAN. Instead, the simulator realistically aggregates *up to* N_a according to the buffer status of each WLAN. Thus, in the model, A and C access the channel less frequently than in the simulator for low-moderate loads but do so during larger periods to transmit larger frames. Instead, since A and C are not saturated in the simulator for such traffic load, shorter but more frequent frames are transmitted. Second, the simulator's implementation of the NAV state – resulting from properly decoding an RTS or CTS of a WLAN within the carrier sense – is not captured by the CTMN. These make WLAN B at the simulator to have fewer chances to access the channel, thus experiencing a significantly reduced (and more representative) throughput. While analytical models such as the presented provide important insights into the system states and behavior of WLANs, inaccuracies when capturing complex scenarios are hard to prevent. Solving those inaccuracies in the model may be not possible or require complex extensions. For instance, in our case, we should accurately represent the buffer occupancy for each node, which would add many more dimensions and states to the CTMN.

In conclusion, we see that, in terms of delay, the non-overlapping channel allocation is the most convenient for keeping a fair deployment where all WLANs are capable to cope with low to moderate traffic loads. Besides, while an overlapping approach is really convenient for the edge WLANs in terms of throughput and delay, it is not the case for the central WLAN B, which actually starves for high edge traffic load. Nonetheless, when neighboring activity is low, B's performance is improved for high ℓ_B since it is able to use the full frequency spectrum like in isolation.

The presented toy scenarios suggest that there is not a unique spectrum allocation approach that suits all the cases. In fact, WLANs' performance depends on multiple factors like spatial distribution and traffic loads, but also on the metric objective to be optimized, which may be designed to foster individual or collaborative behaviors. Nonetheless, we have seen that, as a rule of thumb, AM and overlapping channel allocations are the most convenient for improving the average performance. In

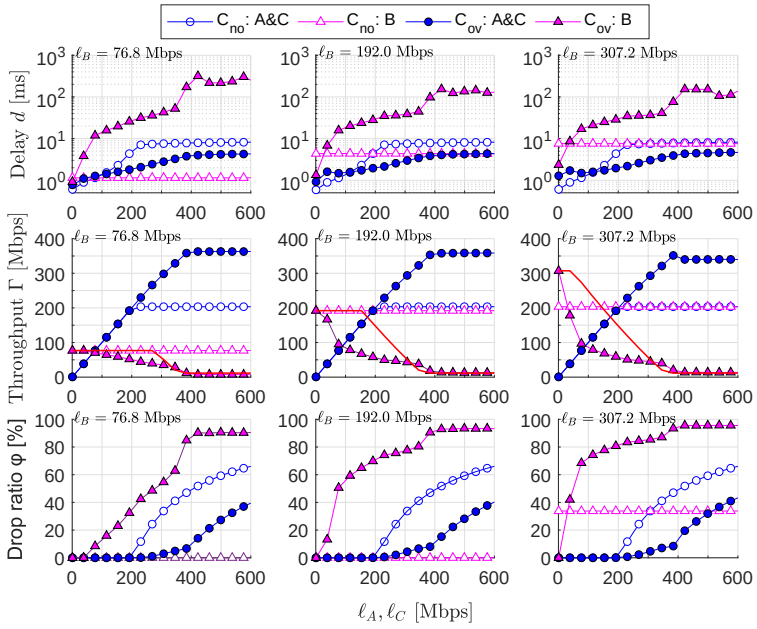


Figure 7: Performance metrics of *toy scenario II*. Results correspond to simulations of 1000 seconds each. The throughput curves in red correspond to the unique case (C_{ov}:B) where the CTMN model does not match the simulator.

turn, with AM we run the risk of jeopardizing WLANs that may fall in FIM starvation or hidden node situations with higher probability, resulting in less fair scenarios.

4 Performance evaluation in HD scenarios

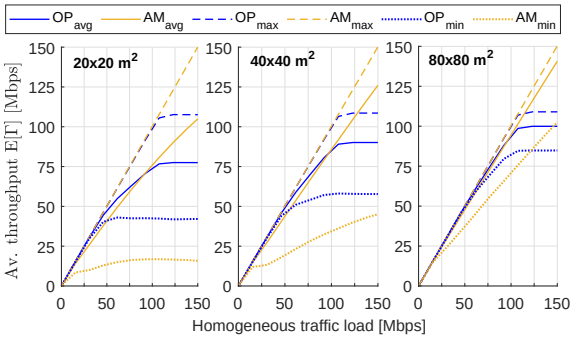
In this section, we analyze the performance of DCB in two different types of IEEE 802.11ax WLAN deployments. Namely, we first assess the impact of the node density in networks with homogeneous CB policies and traffic load. We then discuss what is the optimal CB policy that a WLAN should pick in a completely random HD deployment. The IEEE 802.11ax configuration and other setup parameters used in the following simulations are detailed in Table 2 in the Appendix A.

4.1 Node density effect on CB policies

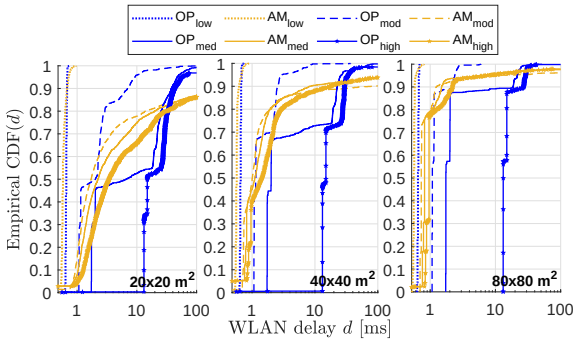
In order to get insights into the node density effect on the efficiency of CB, we now assess the performance of single-channel (non-overlapping approach) and DCB (overlapping approach) in a network consisting of 6 WLANs randomly located in a square map of different sizes: 20x20, 40x40 and 80x80 m^2. All the WLANs are set with the same policy in each case (OP or AM). In addition, the same traffic load is assumed for all the WLANs. The minimum distance between any two APs is set to $d^{\min}_{\text{AP-AP}} = 8$ m and each WLAN is located uniformly at random at a distance $d_{\text{AP-STA}} \in [d^{\min}_{\text{AP-STA}}, d^{\max}_{\text{AP-STA}}] = [1, 4]$ m from the AP. Regarding the channel allocation, all the WLANs are set with random primary channel in the eight basic channels considered in the system (i.e., $p_w \sim U[1, 8], \forall w$). The set of allocated basic channels is set to the maximum allowed $C_w = \{1, ..., 8\} \forall w$ for contiguous spectrum transmissions.[9] Specifically, we generate $N_{\text{D}} = 200$ deployments following the aforementioned conditions for each of the $N_{\text{P}} = 2$ policies considered. Besides, we evaluate each policy for $N_\ell = 11$ values of the homogeneous traffic load ranging from 0.768 to 150 Mbps in the $N_{\text{M}} = 3$ maps of different sizes. The simulated time of each of the $N_{\text{D}} \times N_{\text{P}} \times N_\ell \times N_{\text{M}} = 13200$ scenarios is 20 seconds.

In Figure 8(a) it is shown the average, maximum and minimum throughput experienced by the network. We note that OP has a clear limitation regarding the maximum achievable value, which leads

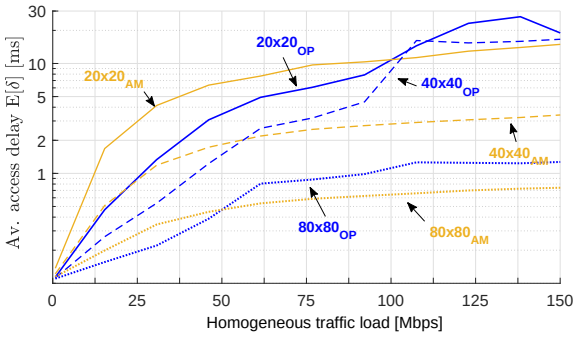
[9] Note that allocating the whole bandwidth to all the WLANs is an interesting extreme case when precalculated channel allocation is overlooked.



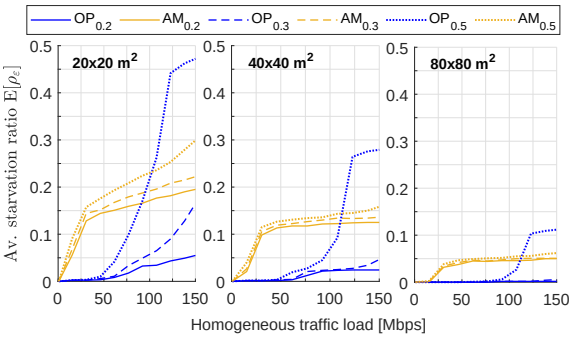
(a) Average, minimum, and maximum throughput. The legend's subscripts *avg*, *max* and *min* refer to the average, maximum, and minimum of the performance metrics.



(b) Empirical CDF of the individual delay. The legend's subscripts *low*, *mod*, *med*, and *high* refer to $\ell = 0.768, 30.72, 61.44, 122.88$ Mbps, respectively.



(c) Average access delay for OP and AM in the three map deployments considered.



(d) Average starvation ration according to three starvation thresholds. Subscripts in the legend represent the value of ε.

Figure 8: Node density effect on network performance. 6 WLANs with same traffic load and CB policies are deployed in squared maps of different areas.

to saturation even in the least dense map (80x80 m^2). In fact, such maximum is quasi-independent of the node density and corresponds to the maximum effective data rate provided by 20 MHz transmissions $r_{20} = 109.71$ Mbps. Instead, for mid to low dense scenarios, AM is able to cope with the traffic load in the majority of the cases as reflected by the average and maximum throughput curves of maps 40x40 and 80x80 m^2. As indicated by the minimum throughput, we find more situations with precarious performance in AM, which corroborates the *risky* nature of aggressive DCB policies. That is, while AM is convenient on average, it is more prone to generate unfair scenarios where at least one WLAN experiences poor performance.

The cumulative distribution function (CDF) of the packet delay experienced by the WLANs is plotted in Figure 8(b). In contrast to the throughput, studying the CDF is convenient for the packet delay since its average value may explode even when just one of the WLAN starves. As expected, the higher the traffic load, the less the probability of achieving small delays regardless of the selected policy. As suggested in previous sections, we confirm that OP leads to a smaller delay than AM for low loads when WLANs are likely to overlap (see 20x20 m^2 map). Instead, for higher loads, since OP is by default constrained to the maximum data rate provided by a 20 MHz channel, it is more probable to achieve acceptable delays with AM. For less dense scenarios, the risk of generating unfair situations is importantly reduced in AM (see 80x80 m^2 map). We note that the packet delay matches completely with the average access delay shown in Figure 8(c). A key aspect to consider in this regard is the effect on the average backoff duration in presence of multiple hidden node collisions, leading to a significant increase of the contention window.

To conclude our observations on the fairness and intrinsic risk of DCB, we study the average starvation ratio. We say that a WLAN w starves if it is not able to successfully transmit a certain fraction ε of its

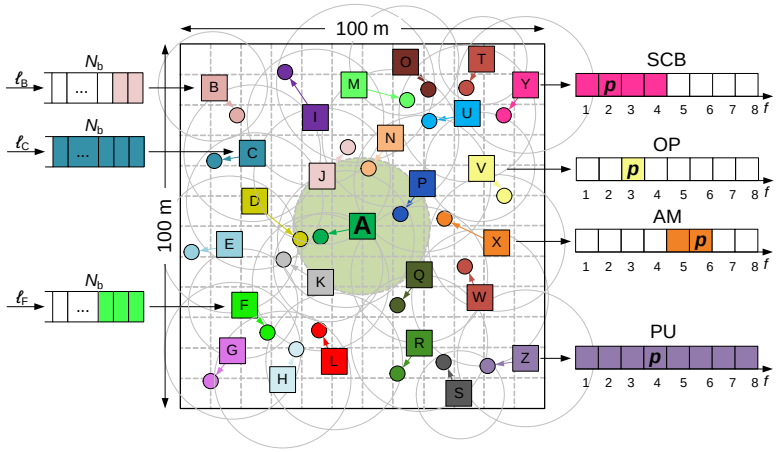


Figure 9: Network deployment with WLAN A placed in the middle and the rest 24 WLANs spread uniformly at random in a 100x100 m2 area.

traffic load ℓ_w. Specifically, w starves if its average throughput Γ_w is less than the selected starvation threshold, i.e., when $\Gamma_w < \varepsilon \ell_w$. The starvation ratio ρ_ε of a particular scenario is computed as the fraction of starving WLANs. For instance, if 2 of 6 WLANs are found to be starving, the corresponding starvation ratio would be 2/6. The average value of this ratio is plotted in Figure 8(d) for the different maps. As expected, the higher the traffic load and/or map density, the higher the starvation ratio for both policies. Looking at the low-mid traffic loads we confirm that OP outperforms AM when it comes to avoid unfair scenarios in uncontrolled deployments. Indeed, OP completely avoids starvation for each of the thresholds when the traffic load is low. This contrasts with AM, which outperforms OP for scenarios under very high loads, but is not able to properly cope with the FIM and hidden node situations in low density deployments.

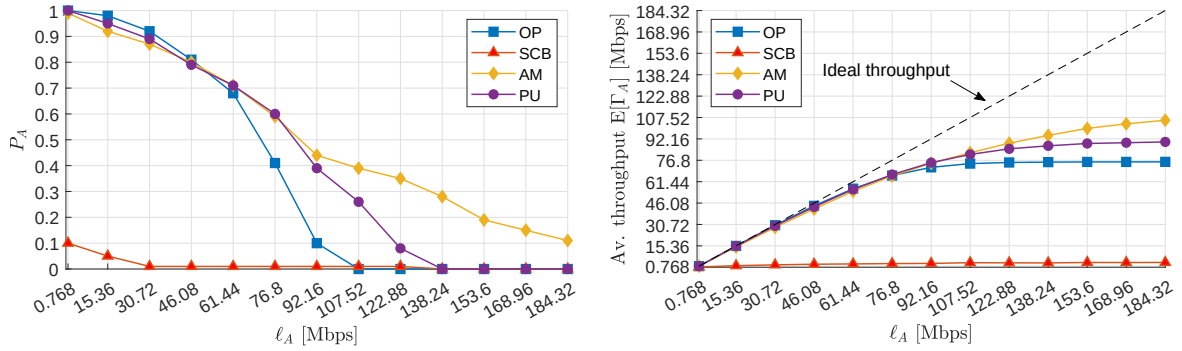
4.2 Optimal individual policy in uncontrolled HD WLANs

In this subsection we discuss what is the optimal CB policy that a particular WLAN (A) should locally implement for improving either its own throughput or delay, i.e., $\mathcal{D}^*_{\Gamma,\mathrm{A}} = \mathrm{argmax}_{\mathcal{D}}\, \Gamma_\mathrm{A}$ or $\mathcal{D}^*_{d,\mathrm{A}} = \mathrm{argmin}_{\mathcal{D}}\, d_\mathrm{A}$, respectively. As shown in Figure 9, we consider a 100 x 100 m^2 area with WLAN A located at the center, and 24 WLANs spread uniformly at random in the area with the single condition that any pair of APs must be separated at least $d^{\min}_{\text{AP-AP}} = 10$ m. The STA[10] of each WLAN is located also uniformly at random at a distance $d_{\text{AP-STA}} \in [d^{\min}_{\text{AP-STA}}, d^{\max}_{\text{AP-STA}}] = [1, 5]$ m from the AP.

Regarding the channel allocation, all the WLANs are set with random primary channel in the eight basic channels considered in the system (i.e., $p_\mathrm{w} \sim U[1, 8], \forall w$). The set of allocated basic channels is assigned uniformly at random as well. That is, the number of allowed basic channels for transmitting is $|C_w| \sim U\{1, 2, 4, 8\}, \forall w \neq \mathrm{A}$, with the exception of WLAN A, which is allocated the widest channel (i.e., $C_\mathrm{A} = \{1, ..., 8\}$). Besides, we consider now bursty traffic dependent on the average traffic load (ℓ), where a burst of $n_b = 10$ packets is generated each $t_b \sim \text{Exponential}(n_b/\ell)$ in order to provide more realistic traffic patterns.

While the CB policies of the rest of WLANs are also set uniformly at random (i.e., they implement OP, SCB, AM or PU with same probability 1/4), A is fixed to a desired policy. Specifically, we generate $N_\mathrm{D} = 100$ deployments following the aforementioned conditions for each of the $N_\mathrm{P} = 4$ policies that A can implement. Besides, we evaluate each policy for $N_\ell = 13$ values of A's traffic load ranging from 0.768 to 184.32 Mbps (i.e., from 64 to 15360 packets/s). The rest of WLANs are set with random average traffic load inside such a range, i.e., $\ell_w \sim U[0.768, 184.32], \forall w \neq \mathrm{A}$. Hence, we simulate $N_\mathrm{D} \times N_\mathrm{P} \times N_\ell = 5200$ scenarios. The simulation time of each scenario is 10 seconds.

[10]Note that in the considered scenarios, having one or multiple STAs per AP does not significantly affect the obtained average results. The main reason is that STAs are randomly placed near the AP and the destination is selected at random for each frame transmission. Therefore, the unique effect of considering more STAs is a probable slight decrease (increase) on the average throughput because of the performance anomaly resulting from the lower (higher) MCS picked by the STA placed farthest from (closest to) the AP.



(a) Probability that A is able to successfully transmit all its traffic load.

(b) Traffic load effect on the average throughput experienced by WLAN A.

Figure 10: Throughput analysis of the *central WLAN scenario*.

Figure 10(a) shows the probability of WLAN A to successfully transmit its traffic load, i.e., $P_{\mathrm{A}} = \mathcal{P}\big(\Gamma_{\mathrm{A}} \geq (1-\epsilon_\Gamma)\ell_{\mathrm{A}}\big)$. Note that we use a margin of error $\epsilon_\Gamma = 0.05$ to cope with the stochastic packet generation of the performed simulations. The average throughput of A for each of the policies is plotted in Figure 10(b). As expected, SCB is viable only for few scenarios when the traffic load is extremely small. This is because the rest of WLANs most likely prevent A to initiate any transmission by occupying part of its allocated channel C_{A}. Instead, the other policies perform much better – specially, AM – since they avoid saturation with high probability even for high traffic loads. While A avoids saturation in some scenarios for $\ell_{\mathrm{A}} < 92.16$ Mbps with OP and $\ell_{\mathrm{A}} < 122.88$ Mbps with PU, respectively, the aggressive adaptability nature of AM allows avoiding saturation in scenarios even where $\ell_{\mathrm{A}} = 184.32$ Mbps. Nevertheless, we note that there are scenarios where AM heavily suffers from the hidden node problem since the SINR at the receiver is importantly reduced. Accordingly, even though the AP may find the whole spectrum free, the STA is not able to decode most of the RTS packets due to interference. This effect can be clearly seen in the P_{A} improvement of OP or PU for low traffic loads (i.e., < 46.08 Mbps).

The average packet delay experienced by A under different traffic loads is shown in Figure 11(a). For the sake of representation, we consider only those scenarios free of outliers.[11]. Note that we do not plot the delay of SCB since its performance is clearly deficient, as shown by Figures 10(a) and 10(b). As a significant result, we note that the smallest average delay is provided by AM for all the studied loads, except for $\ell_{\mathrm{A}} = 0.768$ Mbps. This proves that the delay reduction observed by OP at low traffic loads does also hold for uncontrolled scenarios where neighboring WLANs may have different CB policies and higher traffic loads.

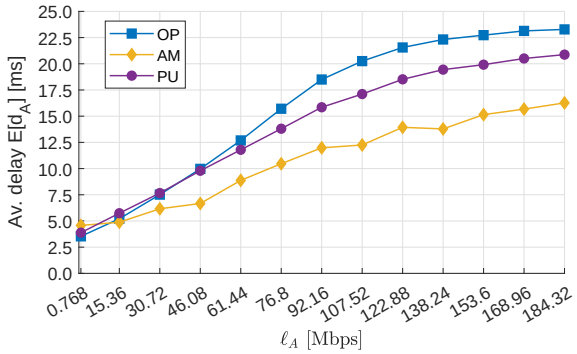
Despite the average superior performance of AM, there are few scenarios where a less aggressive approach like PU or even OP outperforms it. In this regard, we assess below the share of scenarios where each policy provides the smallest average packet delay for WLAN A. In particular, Figure 11(b) compares AM against OP and Figure 11(c) does so for AM and PU. Finally, Figure 11(d) compares the number of scenarios where AM is better than the best combination of OP and PU, i.e., AM is compared against $\mathcal{D} = \mathrm{argmin}_{\mathrm{OP,PU}}\, d_A$ for each of the simulated scenario. We say that for any given pair of policies $\mathcal{D}_1$, $\mathcal{D}_2$, three types of outcomes are categorized according to a predefined delay margin $\delta_d = 1$ ms:

$$\texttt{if}\ \ \mathrm{E}\big[d_{\mathrm{A}}^{\mathcal{D}_1}\big] - \mathrm{E}\big[d_{\mathrm{A}}^{\mathcal{D}_2}\big] \begin{cases} < -\delta_d, & \mathcal{D}_1 \text{ better than } \mathcal{D}_2 \\ > \delta_d, & \mathcal{D}_2 \text{ better than } \mathcal{D}_1 \\ \text{otherwise}, & \text{draw} \end{cases}.$$

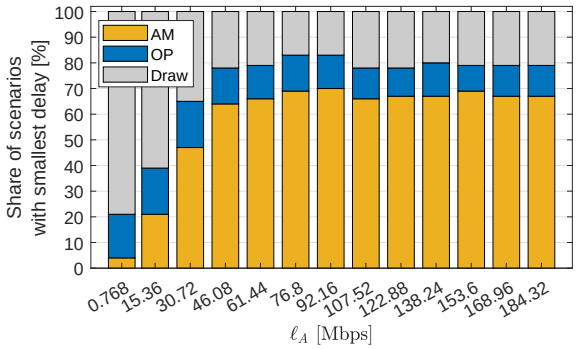
The delay margin allows us capturing those cases where $\mathcal{D}_1$ and $\mathcal{D}_2$ perform similarly.

We see that in most of the cases AM outperforms OP and PU, specially for scenarios with mid-high traffic loads. Nonetheless, for low loads, we note that there is always a better choice than AM for reducing the delay, corroborating the outcomes from previous sections. In addition, there is a significant share of scenarios where OP and, especially, PU provide similar or even smaller delays than AM for all

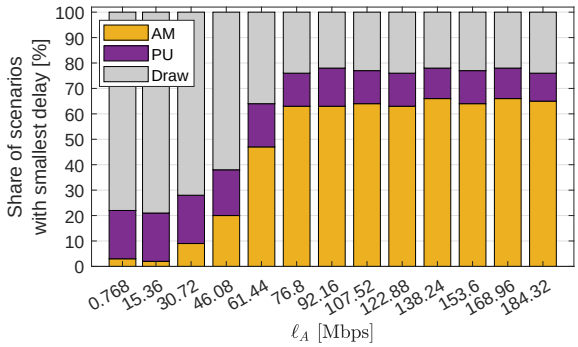
[11] A scenario is labeled as outlier if $d_{\mathrm{A}} \geq 100$ ms. Only 3% of the scenarios where $\mathcal{D}_{\mathrm{A}} = \mathrm{OP}, \mathrm{AM}, \mathrm{PU}$ are outliers.



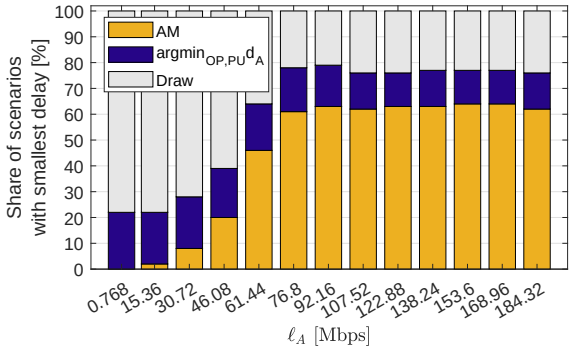
(a) Traffic load effect on the average delay (without outliers) experienced by WLAN A.



(b) Share of scenarios where AM or OP provides the smallest delay for A.



(c) Share of scenarios where AM or PU provides the smallest delay for A.



(d) Share of scenarios where AM or the best policy among OP and PU provides the smallest delay for A.

Figure 11: Delay analysis of the *central WLAN scenario*.

traffic loads. This mainly occurs when A and its neighboring nodes are able to concurrently transmit in different channels through interactions that are not given when implementing AM. Essentially, when A transmits in its whole available bandwidth, neighboring WLANs with primary channels overlapping with A's transmission must wait until it is finished. Afterwards, such WLANs are able to terminate their backoffs and could select a transmission channel including A's primary in turn. This generates *all or nothing* states like the one shown in *Scenario I* that keep A's backoff frozen for longer periods of time. Instead, if A transmits in narrower channels by implementing OP or PU, such WLANs could transmit at the same time in non-overlapping channels and enable more successful parallel transmissions.

In summary, we see that overlapping approaches can significantly enhance traditional single-channel performance in terms of delay and throughput in uncontrolled and realistic HD deployments. Still, there are cases when an overlapping approach that always selects the maximum available bandwidth can be counterproductive in the mid/long-term. Despite the intrinsic uncertainty of spatially distributed WLAN deployments, we can state as a rule of thumb that DCB is convenient when applied through spectrum-adapting policies. Nonetheless, as indicated by the scenarios where OP and/or PU outperformed AM, there is room for further improvement through smarter adaptation by adopting policies on a per-WLAN basis. Hence, we envision that the most effective way of using DCB is to allocate all the nodes with the whole available frequency spectrum, and to smartly assign the primary channel. Moreover, deeper improvements could be achieved by endowing the nodes with the capability to recover from lousy situations like FIM, which are more likely to happen when neighboring WLANs implement aggressive DCB.

5 Conclusions

In this work, we assess the performance of CB in WLANs under different traffic loads. By modeling and simulating CB policies in spatially distributed scenarios we shed light on the question: is it convenient to share wider channels and potentially overlap in the spectrum or not? We show that, while

the performance of SCB is clearly poor for moderate-high traffic loads, spectrum-adapting DCB can significantly outperform the traditional single-channel approach in terms of throughput and delay, even in high-density deployments. However, we also remark two main outcomes concerning DCB: *i*) the intrinsic risk when it comes to generating unfair scenarios as a consequence of the hidden node and FIM situations, and *ii*) the exposure to doing more harm than good in terms of delay for low traffic regimes.

In this regard, the intricate nature of uncontrolled WLAN deployments leaves room for further improvements in spectrum efficiency, while prevents designing effective predefined rules. Therefore, our next work will focus on two aspects: studying machine learning based DCB policies to efficiently adapt to traffic load needs, and jointly combining DCB with adequate primary channel allocation.

Acknowledgment

This work has been partially supported by a Gift from CISCO University Research Program (CG#890107) & Silicon Valley Community Foundation, by the Spanish Ministry of Economy and Competitiveness under the Maria de Maeztu Units of Excellence Programme (MDM-2015-0502), and by the Catalan Government under grant SGR-2017-1188. The work by S. Barrachina-Muñoz is supported by an FI grant from the Generalitat de Catalunya.

References

[1] M. Elbamby, C. Perfecto, M. Bennis, and K. Doppler. Toward low-latency and ultra-reliable virtual reality. *IEEE Network*, 32(2):78–84, 2018.

[2] IEEE 802.11n. Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY): Enhancements for High Throughput. *IEEE*, 2009.

[3] I. P802.11ac. Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Enhancements for Very High Throughput for Operation in Bands below 6 GHz. *IEEE*, 2014.

[4] IEEE 802.11 Task Group AX. Status of Project IEEE 802.11ax High Efficiency WLAN (HEW). `http://www.ieee802.org/11/Reports/tgax_update.htm`. Accessed: 2018-09-21.

[5] EXtreme Throughput (XT) 802.11. *Doc.: IEEE 802.11-18/0789r10*, 2018.

[6] S. Bukhari, M. Rehmani, and S. Siraj. A survey of channel bonding for wireless networks and guidelines of channel bonding for futuristic cognitive radio sensor networks. *IEEE Communications Surveys & Tutorials*, 18(2):924–948, 2016.

[7] M. Park. IEEE 802.11 ac: Dynamic bandwidth channel access. In *Communications (ICC), 2011 IEEE International Conference on*, pages 1–5. IEEE, 2011.

[8] L. Deek, E. Garcia-Villegas, E. Belding, S. Lee, and K. Almeroth. The impact of channel bonding on 802.11 n network management. In *Proceedings of the Seventh Conference on emerging Networking Experiments and Technologies*, page 11. ACM, 2011.

[9] M. Arslan, K. Pelechrinis, I. Broustis, S. Krishnamurthy, S. Addepalli, and K. Papagiannaki. Auto-configuration of 802.11n wlans. In *Proceedings of the 6th International Conference*, page 27. ACM, 2010.

[10] M. Arslan, K. Pelechrinis, I. Broustis, S. Krishnamurthy, S. Addepalli, and K. Papagiannaki. ACORN: An auto-configuration framework for 802.11 n WLANs. *IEEE/ACM Transactions on Networking (TON)*, 21(3):896–909, 2013.

[11] V. Shrivastava, S. Rayanchu, J. Yoonj, and S. Banerjee. 802.11n under the microscope. In *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*, pages 105–110. ACM, 2008.

[12] B. Bellalta. IEEE 802.11ax: High-efficiency WLANs. *IEEE Wireless Communications*, 23(1):38–46, 2016.

[13] L. Deek, E. Garcia-Villegas, E. Belding, S. Lee, and K. Almeroth. Intelligent channel bonding in 802.11 n wlans. *IEEE Transactions on Mobile Computing*, 13(6):1242–1255, 2014.

[14] L. Deek, E. Garcia-Villegas, E. Belding, S. Lee, and K. Almeroth. Joint rate and channel width adaptation for 802.11 MIMO wireless networks. In *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2013 10th Annual IEEE Communications Society Conference on*, pages 167–175. IEEE, 2013.

[15] W. Wang, F. Zhang, and Q. Zhang. Managing channel bonding with clear channel assessment in 802.11 networks. In *Communications (ICC), 2016 IEEE International Conference on*, pages 1–6. IEEE, 2016.

[16] M. Gong, B. Hart, L. Xia, and R. Want. Channel bounding and MAC protection mechanisms for 802.11ac. In *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, pages 1–5. IEEE, 2011.

[17] S. Barrachina-Muñoz, F. Wilhelmi, and B. Bellalta. Dynamic Channel Bonding in Spatially Distributed High-Density WLANs. *arXiv preprint arXiv:1801.00594*, 2018.

[18] Y. Zeng, P. Pathak, and P. Mohapatra. A first look at 802.11ac in action: Energy efficiency and interference characterization. In *Networking Conference, 2014 IFIP*, pages 1–9. IEEE, 2014.

[19] S. Yun, Daehyeok Kim, and L. Qiu. Fine-grained spectrum adaptation in wifi networks. In *Proceedings of the 19th annual international conference on Mobile computing & networking*, pages 327–338. ACM, 2013.

[20] W. Wang, Y. Chen, Z. Wang, J. Zhang, K. Wu, and Q. Zhang. Wideband spectrum adaptation without coordination. *IEEE Transactions on Mobile Computing*, 16(1):243–256, 2017.

[21] P. Huang, X. Yang, and L. Xiao. Dynamic channel bonding: enabling flexible spectrum aggregation. *IEEE Transactions on Mobile Computing*, 15(12):3042–3056, 2016.

[22] Y. Chen, D. Wu, T. Sung, and K. Shih. DBS: A dynamic bandwidth selection MAC protocol for channel bonding in IEEE 802.11 ac WLANs. In *Wireless Communications and Networking Conference (WCNC), 2018 IEEE*, pages 1–6. IEEE, 2018.

[23] S. Khairy, M. Han, L. Cai, Y. Cheng, and Z. Han. A Renewal Theory based Analytical Model for Multi-channel Random Access in IEEE 802.11ac/ax. *IEEE Transactions on Mobile Computing*, 2018.

[24] A. Nabil, M. Abdel-Rahman, and A. MacKenzie. Adaptive Channel Bonding in Wireless LANs Under Demand Uncertainty. To appear in the Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Montreal, QC, Canada, October 2017.

[25] S. Byeon, H. Kwon, Y. Son, C. Yang, and S. Choi. RECONN: Receiver-Driven Operating Channel Width Adaptation in IEEE 802.11ac WLANs. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pages 1655–1663. IEEE, 2018.

[26] B. Bellalta, A. Faridi, J. Barcelo, A. Checco, and P. Chatzimisios. Channel bonding in short-range WLANs. In *European Wireless 2014; 20th European Wireless Conference; Proceedings of*, pages 1–7. VDE, 2014.

[27] B. Bellalta, A. Checco, A. Zocca, and J. Barcelo. On the interactions between multiple overlapping WLANs using channel bonding. *IEEE Transactions on Vehicular Technology*, 65(2):796–812, 2016.

[28] M. Han, Sami K., L. X. Cai, and Y. Cheng. Performance Analysis of Opportunistic Channel Bonding in Multi-Channel WLANs. In *Global Communications Conference (GLOBECOM), 2016 IEEE*, pages 1–6. IEEE, 2016.

[29] M. Kim, T. Ropitault, S. Lee, and N. Golmie. A Throughput Study for Channel Bonding in IEEE 802.11ac Networks. *IEEE Communications Letters*, 2017.

[30] C. Kai, Y. Liang, T. Huang, and X. Chen. A Channel Allocation Algorithm to Maximize Aggregate Throughputs in DCB WLANs. *arXiv preprint arXiv:1703.03909*, 2017.

[31] S. Barrachina-Muñoz, F. Wilhelmi, I. Selinis, and B. Bellalta. Komondor: a Wireless Network Simulator for Next-Generation High-Density WLANs. *arXiv preprint arXiv:1811.12397*, 2018.

[32] R. Laufer and L. Kleinrock. On the capacity of wireless CSMA/CA multihop networks. In *INFOCOM, 2013 Proceedings IEEE*, pages 1312–1320. IEEE, 2013.

[33] A. Faridi, B. Bellalta, and A. Checco. Analysis of Dynamic Channel Bonding in Dense Networks of WLANs. *IEEE Transactions on Mobile Computing*, 2016.

[34] P. Chatzimisios, A. Boucouvalas, and V. Vitsas. Performance analysis of IEEE 802.11 DCF in presence of transmission errors. In *Communications, 2004 IEEE International Conference on*, volume 7, pages 3854–3858. IEEE, 2004.

[35] D. Xu, J. Zhang, X. Gao, P. Zhang, and Y. Wu. Indoor office propagation measurements and path loss models at 5.25 GHz. In *Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th*, pages 844–848. IEEE, 2007.

A Appendix - Evaluation setup

The values of the parameters considered in the simulations presented throughout this paper are shown in Table 2. Regarding the path loss, we use the dual-slope log-distance model for 5.25 GHz indoor environments in room-corridor condition [35]. Specifically, the path loss in dB experienced at a distance d is defined by

$$\mathrm{PL}(d) = \begin{cases} 53.2 + 25.8\log_{10}(d) & \text{if } d \leq d_1 \text{ m} \\ 56.4 + 29.1\log_{10}(d) & \text{otherwise} \end{cases}, \tag{3}$$

where $d_1 = 9$ m is the break point distance.

The MCS index used for each possible channel bandwidth (i.e., 20, 40, 80 or 160 MHz) was the highest allowed according to the power power budget established between the WLANs and their corresponding STA/s and the minimum sensitivity required by the MCSs. As stated by the IEEE 802.11ax amendment, the number of transmitted bits per OFDM symbol used in the data transmissions is given by the channel bandwidth and the MCS parameters, i.e., $r = Y_{\mathrm{sc}}Y_{\mathrm{m}}Y_{\mathrm{c}}V_{\mathrm{s}}$, where Y_{sc} is the number of data sub-carriers, Y_{m} is the number of bits in a modulation symbol, Y_{c} is the coding rate, and $V_{\mathrm{s}} = 1$ is the number of single user spatial streams (note that we only consider one stream per transmission).

The number of data sub-carriers depends on the transmission channel bandwidth. Specifically, Y_{sc} can be 234, 468, 980 or 1960 for 20, 40, 80, and 160 MHz, respectively. For instance, the data rate provided by MCS 11 in a 20 MHz transmission is $s = (234 \times 10 \times 5/6 \times 1)\sigma^{-1} = 121.9$ Mbps. However, control frames are transmitted in legacy mode using the basic rate $r_{\mathrm{leg}} = 24$ bits per OFDM symbol of MCS 0, corresponding to $s_{\mathrm{leg}} = 6$ Mbps since the legacy OFDM symbol duration σ_{leg} must be considered. With such parameters we can compute the duration of each packet transmission:

$$T_{\mathrm{RTS}} = T_{\text{PHY-leg}} + \left\lceil \frac{L_{\mathrm{SF}} + L_{\mathrm{RTS}} + L_{\mathrm{TB}}}{r_{\mathrm{leg}}} \right\rceil \sigma_{\mathrm{leg}},$$

$$T_{\mathrm{CTS}} = T_{\text{PHY-leg}} + \left\lceil \frac{L_{\mathrm{SF}} + L_{\mathrm{CTS}} + L_{\mathrm{TB}}}{r_{\mathrm{leg}}} \right\rceil \sigma_{\mathrm{leg}},$$

$$T_{\mathrm{DATA}}(N_a) = T_{\text{PHY-HE-SU}} + \left\lceil \frac{L_{\mathrm{SF}} + N_{\mathrm{a}}(L_{\mathrm{MD}} + L_{\mathrm{MH}} + L_{\mathrm{D}}) + L_{\mathrm{TB}}}{r} \right\rceil \sigma,$$

$$T_{\mathrm{BACK}} = T_{\text{PHY-leg}} + \left\lceil \frac{L_{\mathrm{SF}} + L_{\mathrm{BACK}} + L_{\mathrm{TB}}}{r_{\mathrm{leg}}} \right\rceil \sigma_{\mathrm{leg}}.$$

Table 2: Evaluation setup (from simulation setup in [17]).

Parameter	**Description**	**Value**
f_{c}	Central frequency	5.25 GHz
$\lvert c \rvert$	Basic channel bandwidth	20 MHz
L_{D}	Data packet size	12000 bits
N_{b}	Buffer capacity	150 packets
N_{a}	Max. no. of packets in a frame	64
$\mathrm{CW}_{\mathrm{min}}$	Min. contention window	16
m	No. of backoff stages	5
MCS	IEEE 802.11ax MCS index	0 - 11
η	MCS's packet error rate	0.1
CCA	CCA threshold	-82 dBm
P_{tx}	Transmission power	15 dBm
G_{tx}	Transmitting gain	0 dB
G_{rx}	Reception gain	0 dB
$\mathrm{PL}(d)$	Path loss	see (3)
P_{ν}	Adjacent power leakage factor	-20 dB
CE	Capture effect threshold	20 dB
N	Background noise level	-95 dBm
T_{e}	Empty backoff slot duration	9 μs
T_{SIFS}	SIFS duration	16 μs
T_{DIFS}	DIFS duration	34 μs
T_{PIFS}	PIFS duration	25 μs
$T_{\mathrm{PHY\text{-}leg}}$	Legacy preamble	20 μs
$T_{\mathrm{PHY\text{-}HE\text{-}SU}}$	HE single-user preamble	164 μs
σ_{leg}	Legacy OFDM symbol duration	4 μs
σ	OFDM symbol duration	16 μs
L_{BACK}	Length of a block ACK	432 bits
L_{RTS}	Length of an RTS packet	160 bits
L_{CTS}	Length of a CTS packet	112 bits
L_{SF}	Length of service field	16 bits
L_{MD}	Length of MPDU delimiter	32 bits
L_{MH}	Length of MAC header	320 bits
L_{TB}	Length of tail bits	18 bits

8.3

Komondor: a Wireless Network Simulator for Next-Generation High-Density WLANs

Sergio Barrachina-Muñoz, Francesc Wilhelmi*,
Ioannis Selinis, and Boris Bellalta

Abstract

Komondor is a wireless network simulator for next-generation wireless local area networks (WLANs). The simulator has been conceived as an accessible (ready-to-use) open source tool for research on wireless networks and academia. An important advantage of Komondor over other well-known wireless simulators lies in its high event processing rate, which is furnished by the simplification of the core operation. This allows outperforming the execution time of other simulators like ns-3, thus supporting large-scale scenarios with a huge number of nodes. In this paper, we provide insights into the Komondor simulator and overview its main features, development stages and use cases. The operation of Komondor is validated in a variety of scenarios against different tools: the ns-3 simulator and two analytical tools based on Continuous Time Markov Networks (CTMNs) and the Bianchi's DCF model. Results show that Komondor captures the IEEE 802.11 operation very similarly to ns-3. Finally, we discuss the potential of Komondor for simulating complex environments – even with machine learning support – in next-generation WLANs by easily developing new user-defined modules of code.

1 Introduction

The Institute of Electrical and Electronics Engineers (IEEE) 802.11 Wireless Local Area Networks (WLANs) are evolving fast to satisfy the new strict requirements in terms of data rate and user density. In particular, various IEEE 802.11 amendments have been introduced in the past few years or are under active development to accommodate the need for higher capacity, exponential growth in number of devices, and novel use-cases. [1]. An example of next-generation high-density deployment is depicted in Fig. 1 where multiple WLANs are allocated with different channels and dynamic channel bonding (DCB) policies.

Of particular interest is the IEEE 802.11ax (11ax) amendment [2], that is under active development and which was introduced to address the demands and challenges that WLANs will face in the congested 2.4/5 GHz bands [3]. Other important amendments for next-generation wireless networks are the IEEE 802.11ay [4] and EXtreme Throughput (XT) 802.11 [5], which aim to exploit the 60 GHz and ≤ 6 GHz frequency bands, respectively. Amendments like the aforementioned ones lay the foundation of next-generation WLANs by including new features such as multiple-antenna techniques like Downlink/Uplink Multi-User Multiple-Input-Multiple-Output (DL/UL MU-MIMO), spatial reuse techniques like BSS coloring, and efficient use of channel resources like DL/UL Orthogonal Frequency Division Multiple Access (OFDMA). Therefore, it becomes necessary to provide reliable simulation tools able to assess the performance and behavior of next-generation WLANs in multiple scenarios/cases, especially in high-density deployments.

In this paper, we present Komondor,[1] an open source, event-driven simulator based on the CompC++ COST library [6]. Komondor is focused on fulfilling the need for assessing the novel features introduced in recent and future amendments, which may be endowed with applications driven by machine learning (ML). The motivation for developing and building the presented wireless network simulator is motivated by:

i) The lack of analytical models for capturing next-generation techniques in spatially distributed and/or high-density deployments.

ii) The lack of next-generation WLAN-oriented simulators.

*The contribution of the first two authors is the same.

[1] All of the source code of Komondor, under the GNU General Public License v3.0., is open, and potential contributors are encouraged to participate. The repository can be found at `https://github.com/wn-upf/Komondor`

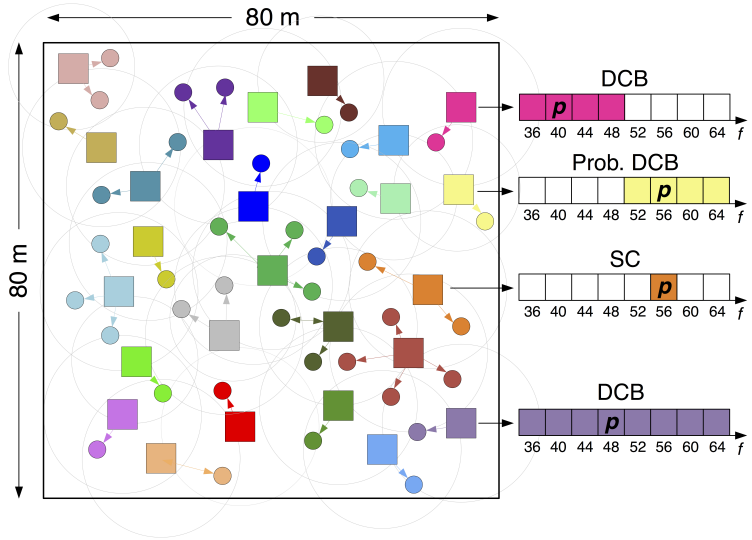


Figure 1: Dense scenario composed of 25 WLANs. Note that each WLAN has its own channel allocation and DCB policy.

iii) The complexity of extending simulators comprising an exhaustive implementation of the physical (PHY) layer.

iv) The large (or intractable) execution time required by other simulators to simulate high-density deployments.

v) The need for conveniently incorporating ML-based agents in the simulation tool.

In short, Komondor is designed to efficiently implement new functionalities by relying on flexible and simplified PHY layer dependencies, to be faster than most off-the-shelf simulators, and to provide reliable simulations and a gentle learning curve to new users.

2 Wireless network simulators

Wireless network simulators can be categorized into continuous-time and discrete-event. On the one hand, continuous-time simulators continuously keep track of the system dynamics by dividing the simulation time into very small periods of time. On the other hand, in discrete-event simulators, events are used to characterize changes in the system. Accordingly, for the latter, events are ordered in time and normally allow running faster simulations than continuous-time simulators. In addition, discrete-event simulators allow tracing events with higher precision.

From the family of discrete-event driven network simulators, only a few ones are publicly available. OMNET++ [7] is a component-based C++ simulation library that is not open-source and is used for modeling communication networks and distributed multiprocessor systems. OPNET is another commercial simulator that allows the integration of external components. NetSim [8] was conceived to provide an accurate simulation model oriented to the world wide web. To that purpose, the simulator was written in Java, which compromises simulation time with programming flexibility. When it comes to open source network simulators, a MATLAB-based link-level simulator was presented in [9] for supporting the IEEE 802.11g/n/ac/ah/af technologies. The ns-2 simulator [10] is another network simulator known for its accuracy and the integration with the network animator. Finally, the ns-3, which was introduced in 2006 to replace the ns-2, presents significant advantages over the ns-2 due to its detailed simulation features, becoming very popular among the research community [11]. Table 1 highlights in a nutshell the most important characteristics of the overviewed network simulators and Komondor.

Among the family of overviewed discrete-event simulators, we highlight the ns-3 open-source simulator due to its popularity and use it as a baseline for comparing against Komondor. Despite

[2] Although ns-2 and ns-3 do not provide a default graphical animation tool, there are tools supporting live animation, e.g., PyViz or NetAnim for ns-3 and NAM for ns-2.

[3] An integration with OpenAI Gym has been recently provided to ns-3 [12], but the ML-based operation is not part of the simulator.

Table 1: Comparison of wireless network simulators.

Simulator	Open-source	Source lang.	Complexity	GUI	11ax features	ML/based module
ns-3	Yes	C++	High	No[1]	Partial	No[2]
ns-2	Yes	C++/OTcl	Low	No[1]	No	No
OMNET++	No	C/C++	Medium	Yes	No	No
OPNET	No	C++	Medium	Yes	No	No
NetSim	No	Java	Low	Yes	No	No
Komondor	**Yes**	**C/C++**	**Low**	**No**	**Partial**	**Yes**

the plethora of features that are supported in ns-3, it has some inherent limitations, such as the high complexity for developing new features/models as an extension of the simulator core. In particular, compatibility with the already existing/supported models is required and must be carefully ensured. For example, beamforming for previous mature amendments (i.e. IEEE 802.11n/ac) is not available yet, owing to the effort required to integrate it. Moreover, the integration of new features strongly depends on the willingness of the community to contribute to the development.

With respect to the IEEE 802.11ax operation – rates and support for information elements are being developed – the implementation is mostly based on the Draft 1.0 [13]. Such a draft dates from 2016 and does not include most of the core IEEE 802.11ax functionalities. At the time of submitting this paper, only the Single-User Protocol Data Unit (SU PPDU) and MIMO with up to four antennas are supported in ns-3, whereas OFDMA and MU-MIMO are not supported in the official distribution [14].

Apart from the official resources, we find few ns-3 works publicly available that support IEEE 802.11ax features, which may (or may not) be integrated into future releases. For example, we highlight the works with regard to the OFDMA that have been carried out by Getachew Redieteab et al. (based on the IEEE 802.11ax specification framework document [15]) and Cisco [16]. However, none of these works completely follow the latest developments in the IEEE 802.11ax standard and have not been validated through extensive simulations and testbed results, as had previously occurred with the OFDM [17]. In addition to OFDMA, the spatial reuse operation (i.e., BSS Color [18]) is under active development, whereas extensions of the capture effect have been applied to ns-3 to follow the IEEE 802.11ax guidelines and studied in [19] and in a testbed [20].

3 Komondor Design Principles

3.1 Architecture

Komondor aims to realistically capture the operation of WLANs. Henceforth, it reproduces actual transmissions on a per-packet basis. To that purpose, Komondor is based on the COST library, which allows building interactions between components (e.g., wireless nodes, buffers, packets) through synchronous and asynchronous events. While the former are messages explicitly exchanged between components through input/output ports, the latter are based on timers. In practice, components perform a set of operations until a significant event occurs. For instance, a node that is decreasing its backoff may freeze it when an overlapping node occupies the channel. The beginning and end of such a transmission are examples of significant events, whereas decreasing the backoff counter is not. Nevertheless, events may be triggered by different timers. In the previous example, a node's transmission begins once the backoff timer terminates (i.e., the backoff timer triggers the beginning of the transmission), while the end of the transmission is triggered by the packet transmission timer. Fig. 2 shows the schematic of a COST component, which is composed of inports, outports, and a set of timers.

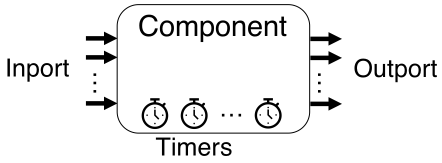


Figure 2: COST component. While inports and outports allow to directly communicate with other components, timers trigger events specific to the component.

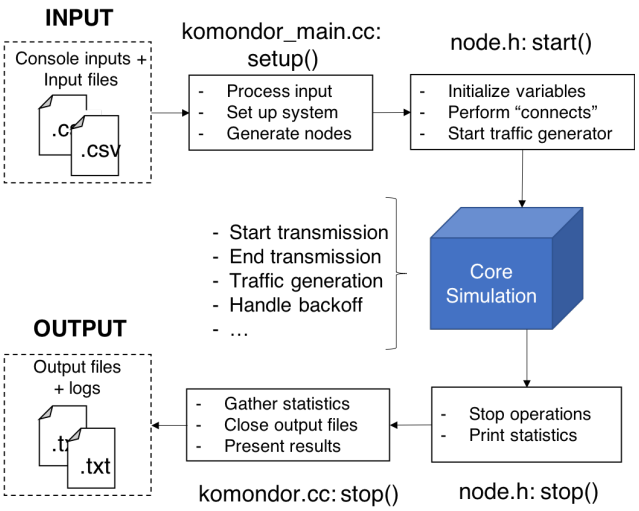


Figure 3: Komondor execution flowchart.

3.2 IEEE 802.11 Features

Komondor entails a long-term project in which several contributors are involved. That is, the simulator is continuously evolving to include novel techniques and generally improve performance. The current version of Komondor (v2.0) includes the following fully tested IEEE 802.11ax features:

- **Distributed coordination function (DCF)**: the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) captures the basic Wi-Fi operation for accessing the channel. Moreover, Contention Window (CW) adaptation is considered.
- **Buffering and packet aggregation**: several traffic generator models are implemented in Komondor such as deterministic, Poisson or full-buffer. Besides, multiple media access control protocol data unit (MPDU) can be aggregated into the same PLCP protocol data unit (PPDU) in order to reduce the generated communication overheads.
- **Dynamic channel bonding (DCB)**: multiple channel widths can be selected during transmissions by implementing DCB policies in order to maximize the spectrum efficiency. Some of these policies were already evaluated in [21], [22].
- **Modulation coding scheme (MCS) selection**: the MCS is negotiated between any transmitter-receiver pair according to the Signal-to-Interference-and-Noise Ratio (SINR), thus supporting multiple transmission rates.
- **Ready-to-send/Clear-to-send (RTS/CTS) and Network Allocation Vector (NAV)**: virtual carrier sensing is implemented in order to minimize the number of collisions by hidden-nodes.

Future development stages are under progress including other features such as OFDMA, MU-MIMO transmissions, beamforming, spatial reuse, and ML-based configuration.

3.3 Execution Flowchart

Komondor is composed of several modules that allow performing simulations with a high degree of freedom. Fig. 3 summarizes the operational mode of Komondor from a user's point of view. A more detailed user's guide providing a quick-start and guided execution examples is available in the Komondor's Github repository.

3.3.1 Input and Setup/Start

as for the execution console command for starting Komondor simulations, arguments are designed in a simple and efficient way. Examples of console arguments are the file names of the inputs, the activation flags of the logs, the simulation time and the random seed. In addition, input files (in CSV format) are used to define the environment and have been conceived in a way that the user can

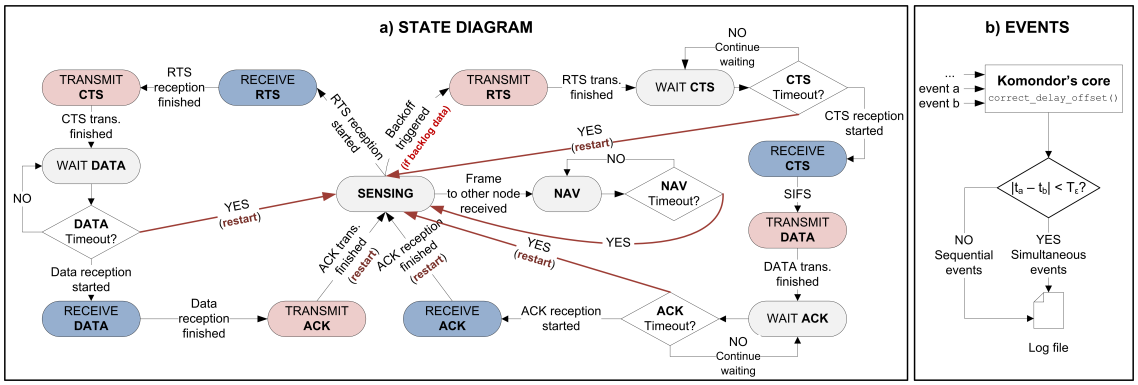


Figure 4: Komondor's state diagram and events. a) States are reachable by different transitions. b) Simultaneous events are properly processed through delay offset correction.

easily modify important simulation parameters such as the traffic load, the path-loss model, or the data packet size. Once the environment is generated and nodes are initialized, traffic is exchanged between nodes until the simulation time runs out.

3.3.2 Stop and Output

when the simulation finishes, the closing is handled and statistics are gathered. Then, extensive and detailed performance statistics are per default provided by Komondor (e.g., throughput, delay, spectrum utilization, or collisions). Moreover, the user can efficiently include as much as metrics as desired.

3.4 States and events

The Komondor's core operation is based on states, which represent the status (or situation) in which a node can be involved. A state diagram summarizing both states and transitions is shown in Fig. 4. Roughly, a given node starts in the SENSING state, where multiple events can occur (e.g., a new packet is buffered or a new transmission is detected). Then, according to the noticed event, the node transits to the corresponding state.

3.4.1 States

we depict below each state and how a node must behave in front of new events.

- **SENSING**: a node senses the channel with two main purposes. First, to follow the CSMA/CA operation to gain access to the channel (in case there is backlogged data in the buffer/s). Second, to wait for incoming transmissions, so that either carrier sensing or receiving procedures are held. In case of being immersed in a backoff procedure, a node detecting a "new transmission" event would sense the power received in its primary channel, and assess whether to freeze the backoff countdown. Similarly, whenever an "end transmission" event occurs, the channel is sensed in order to determine whether the backoff counter can be resumed or remain paused.

- **TRANSMIT**: a transmitter node is currently transmitting a packet. No matter what events may occur, during the packet transmission, the node blocks its receiver capabilities and remains in the same state until the transmission is finished.

- **RECEIVE**: when a node is receiving and decoding an incoming packet, it will behave in front of a new event according to its implication in the channel of interest. Of especial importance are those new transmission events triggered by other nodes that have gained access to the medium. Specifically, if a new transmission generates enough interference, the ongoing reception will be discarded, thus leading to a packet loss.

- **WAIT states**: these states allow modeling the situations where a node that transmitted a packet is expecting for the corresponding response. Namely, after transmitting RTS, CTS or DATA packets, the transmitter will wait for the corresponding CTS, DATA or ACK/BACK packets, respectively. If the response packet is not received before the corresponding timeout

is triggered, the transmitter assumes that either the transmitted packet or the response packet is lost and resets to SENSING state. Wait states are particularly useful to detect packet losses when anomalies in the network (e.g., hidden terminal problem) occur.

- **NAV**: when a node enters in NAV state due to the successful reception of a frame addressed to a different destination, it sets a NAV timer and keeps listening to its primary channel. If a new frame is successfully received during the NAV, the timer is updated, provided that the new NAV time is larger than the current remaining time.

3.4.2 Events

each time a node performs an action that can affect the system (e.g., it starts transmitting a frame), an event is announced. Events in Komondor are lined up on the time axis and handled by the core entity. Events management is similar in ns-3. However, the latter exhibits a significant limitation, since events that are scheduled at the exact same time can be executed in any order. Such a development feature may lead to unpredictable results and is incompatible with real-world situations in which events can occur simultaneously. Some inconsistencies may occur in case that the execution order affects multiple simultaneous events (e.g., two packets arriving at the exact same time). To solve this, Komondor, which is also a discrete-event simulator, employs temporal variables to compare the exact timestamps at which two or more events were generated. As a result, Komondor is able to successfully simulate the behavior of simultaneous events while keeping the logic of the states.

3.5 Developing new modules

Komondor has been conceived to be easily modified and extended. In particular, several modules have been provided to represent different simulation capabilities (e.g., propagation, channel access or traffic generation). Accordingly, Komondor can be potentially extended to support the operation of other IEEE 802.11 amendments such as 11n, 11ac, 11ad or 11ay. In addition, ML-based modules can also be introduced. A complete manual can be found at the Komondor's repository.

4 Validation

In this Section, we validate the operation of Komondor and show its potential for dealing with high-density scenarios. In particular, we show the reliability of the simulator, despite its reduced complexity of the PHY.[4] The validation of the Komondor's operation is done through a set of illustrative scenarios, and our results are compared with the ones obtained with ns-3.[5] In addition to ns-3, a mutual validation is performed with the Continuous Time Markov Networks (CTMNs) modeling introduced in [24], and which is extended for spatially distributed networks in the Spatial-Flexible Continuous Time Markov Network (SFCTMN) framework [21]. As for high-density scenarios, we make use of the Bianchi's DCF analytical model [25] to validate the results in fully-overlapping deployments, where all the nodes are within the carrier sense of the others. The results shown in the following subsections were obtained according to the parameters defined in Table 2. The duration of the RTS, CTS and data frame is computed as follows:

$$T_{\text{RTS}} = T_{\text{PHY-leg}} + \left\lceil \frac{L_{SF} + L_{\text{RTS}}}{L_{s,l}} \right\rceil \sigma_{\text{leg}},$$

$$T_{\text{CTS}} = T_{\text{PHY-leg}} + \left\lceil \frac{L_{SF} + L_{\text{CTS}}}{L_{s,l}} \right\rceil \sigma_{\text{leg}},$$

$$T_{\text{D}} = T_{\text{HE-SU}} + \left\lceil \frac{L_{\text{SF}} + L_{\text{MH}} + N_{\text{agg}} L_{\text{D}}}{L_{s,l}} \right\rceil \sigma.$$

Note that full-buffer traffic is assumed in all the scenarios throughout this work for comparative purposes. Moreover, we have considered the residential path-loss model recommended in the IEEE 802.11ax [23], which inflicts high losses due to its large number of obstacles (e.g., walls).

[4] For instance, channel effects are assumed to remain static during the whole transmission of a given frame, and the propagation delay is considered to be negligible.

[5] Details on the ns-3 implementation used in the simulations presented throughout this paper can be found at `https://github.com/wn-upf/Komondor/tree/master/Documentation/Validation/ns-3`. For instance, this implementation includes the 11ax residential scenario propagation loss [23] and has a PLCP training duration updated according to the 11ax amendment [2].

Table 2: Parameters considered in the presented scenarios.

Parameter	Description	Value
f_{c}	Central frequency	5 GHz
$\|c\|$	Basic channel bandwidth	20 MHz
MCS	11ax MCS index	0-11
G_{tx}	Transmitting gain	0 dB
G_{rx}	Reception gain	0 dB
PL(d)	Path loss (Residential scenario)	see [23]
N	Background noise level	-95 dBm
σ_{leg}	Legacy OFDM symbol duration	4 µs
σ	OFDM symbol duration (GI-32)	16 µs
N_{sc}	Number of subcarriers (20 MHz)	234
N_{ss}	Number of spatial streams	1
T_{e}	Empty slot duration	9 µs
T_{SIFS}	SIFS duration	16 µs
T_{DIFS}	DIFS duration	34 µs
T_{PIFS}	PIFS duration	25 µs
$T_{\mathrm{PHY\text{-}leg}}$	Legacy preamble duration	20 µs
$T_{\mathrm{HE\text{-}SU}}$	HE single-user field duration	100 µs
T_{ACK}	ACK duration	28 µs
T_{BACK}	Block ACK duration	32 µs
$T_{\mathrm{PPDU}}^{\mathrm{max}}$	Max. PPDU duration	5484 µs
$L_{s,l}$	Size OFDM symbol (legacy)	24 bits
L_{D}	Data packet size	11728 bits
N_{agg}	No. of frames in an A-MPDU	1, 64
L_{RTS}	Length of an RTS packet	160 bits
L_{CTS}	Length of a CTS packet	112 bits
L_{SF}	Length of service field	16 bits
L_{MH}	Length of MAC header	320 bits
CW	Contention window (fixed)	15

4.1 Analyzing toy Scenarios

Komondor has been conceived as a friendly and ready-to-use wireless network simulator that can be used by researches and teachers to study fundamental networking issues. In particular, scenarios and environment configurations can be conveniently modified through structured input files. The scenarios proposed in this Section are a clear example of toy scenarios where different networking concepts such as *flow starvation* or *additive interference* take place. Furthermore, a given user can easily analyze WLAN scenarios through the implemented logs generation system and statistics reporting. Accordingly, particular phenomena in the PHY and medium access control (MAC) layers can be tracked (e.g., channel contention, packet collisions, physical carrier sensing, energy detection, or buffer dynamics).

4.2 Basic Operation

We first aim to validate the basic IEEE 802.11 operation of the DCF implemented in Komondor when RTS/CTS is applied. For that, we consider a single Access Point (AP) scenario (we name it *Scenario 1*) with one and two stations (STAs), where full-buffer downlink traffic is held. The two-STAs case allows us to assess the proper behavior of Komondor in presence of multiple STAs. To validate this scenario, we compare the Komondor results with the ones provided by ns-3 and the SFCTMN framework. Fig. 5 shows the simulation results obtained from each tool, for packet aggregation ($N_{\mathrm{agg}} = 64$) and no-aggregation ($N_{\mathrm{agg}} = 1$). We note that the average throughput obtained by each simulation tool is almost identical, either for packet aggregation or not. In addition, having multiple STAs leads to the same result as for a single one since the destination STA is picked at random in every transmission.

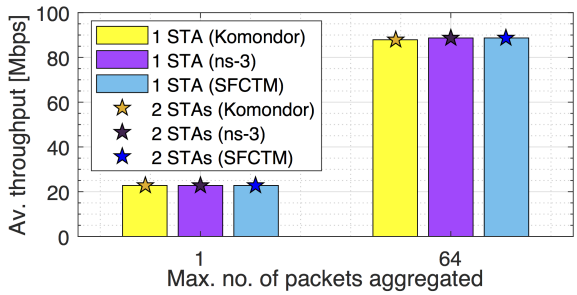
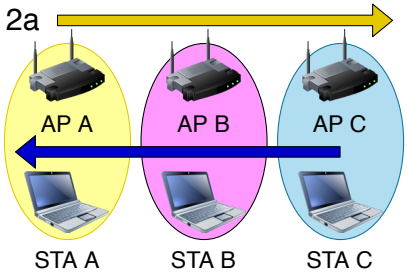
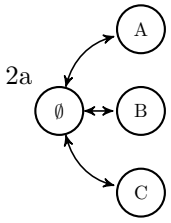


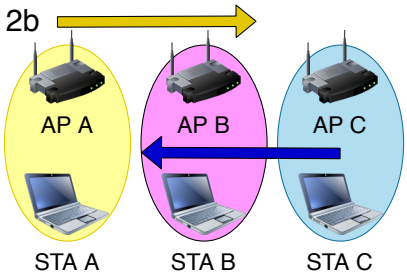
Figure 5: Average throughput experienced by the WLAN of *Scenario 1*, for $N_{\text{agg}} = 1$ and $N_{\text{agg}} = 64$. Results obtained from each simulation tool are shown.



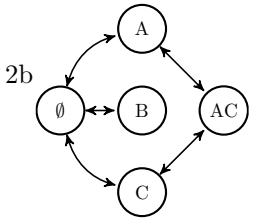
(a) Topology of *Sce2a*.



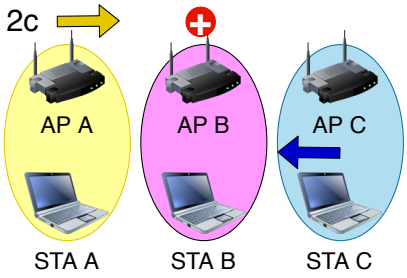
(b) CTMN of *Sce2a*.



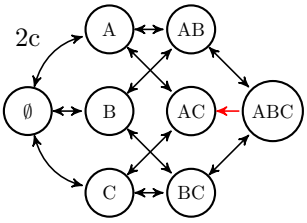
(c) Topology of *Sce2b*.



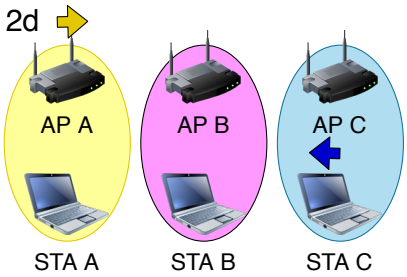
(d) CTMN of *Sce2b*.



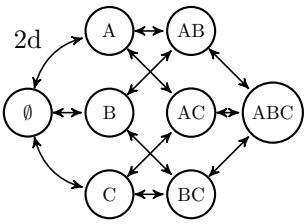
(e) Topology of *Sce2c*.



(f) CTMN of *Sce2c*.



(g) Topology of *Sce2d*.



(h) CTMN of *Sce2d*.

Figure 6: Topologies and corresponding CTMNs of scenarios 2a-2d. The yellow and blue arrows represent the area of interference from transmitters in WLANs A and C, respectively, whereby medium contention is forced.

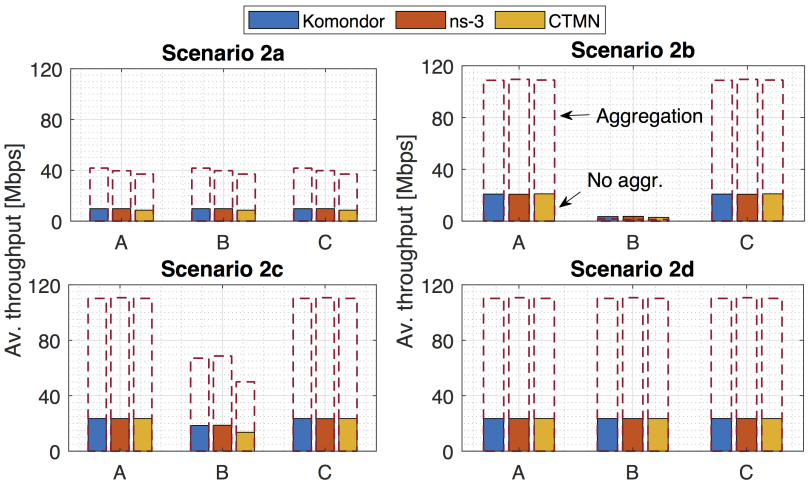


Figure 7: Average throughput experienced by each WLAN in scenarios 2a-2d. $N_{\text{agg}} = 1$ and $N_{\text{agg}} = 64$ are represented through solid bars and dashed lines, respectively.

4.3 Complex inter-WLAN interactions

In order to validate the behavior of Komondor in front of more complex inter-WLAN interactions, we now focus on the three-WLANs scenarios shown in Fig. 6. We name them *Scenario 2a-2d*. The interactions occurring in such scenarios are illustrated through CTMNs, where states[6] represent the WLANs that are currently transmitting. Note that each of these scenarios reflects different situations that are of particular interest since they generalize different well-known phenomena in wireless networks:

- **Fully overlapping (Fig. 6a)**: all the nodes cause contention to all the others when transmitting. For that, the distance between consecutive APs and between AP and STA of the same WLAN is set to $d_{\text{AP,AP}} = d_{\text{AP,STA}} = 2$ m, respectively.
- **Flow starvation (Fig. 6c)**: contention is triggered in a pair-wise manner, so that WLAN_A and WLAN_C do not interfere each other. For that, the distance is set to $d_{\text{AP,AP}} = 4$ m and $d_{\text{AP,STA}} = 2$ m. Note that this case could be also extended to show a hidden node effect if AP_A or AP_C were intended to transmit to a STA located at the location of AP_B.
- **Potential overlap (Fig. 6e)**: contention only occurs at WLAN_B when both WLAN_A and WLAN_C transmit concurrently. Otherwise, the channel is sensed as free. Note that, in this case, packets are successfully transmitted in WLAN_B whenever it access the channel. The distances are $d_{\text{AP,AP}} = 5$ m and $d_{\text{AP,STA}} = 2$ m for WLAN_A and WLAN_C, and $d_{\text{AP,STA}} = 3$ m for WLAN_B.
- **No overlapping (Fig. 6g)**: none of the nodes causes contention to any other when transmitting. That is, every WLAN operates like in isolation. The distances in this case are $d_{\text{AP,AP}} = 10$ m and $d_{\text{AP,STA}} = 2$ m.

The average throughput experienced by each WLAN in each scenario is shown in Fig. 7. As previously done, we compare the performance of Komondor with ns-3 and SFCTMN. Note that results gathered by both Komondor and ns-3 are very similar in all the cases. Concerning the differences in the average throughput values estimated by both simulators and SFCTMN, we observe two phenomena with respect to backoff collisions in topologies of *Scenario 2a* and *2c*. First, in *2a*, the throughput is slightly higher when the capture effect condition is ensured. This is due to the fact that concurrent transmissions (or backoff collisions) are permitted and captured in the simulators. Second, the most notable difference is given in *2c*, which is caused by the assumption of continuous time backoffs in the CTMN. These are clear examples of the limitations of the analytical tool.

4.4 High-density and simulator performance

Finally, we assess the performance of Komondor when dealing with high-density scenarios. Notice that being able to simulate scenarios with a large number of nodes is a key feature due to the ever-increasing trend towards short-range and dense deployments. In this situation, we show the results

[6]Note that CTMN states are not related by any means to Komondor states.

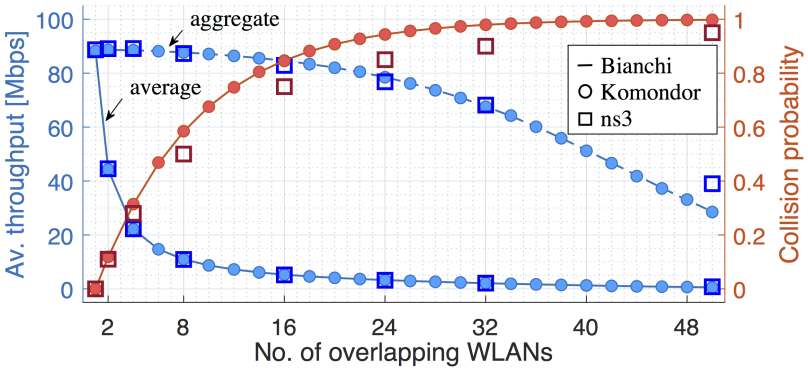


Figure 8: Throughput (average and aggregate) and collision probability vs. number of overlapping WLANs. Only some ns-3 points are plotted for the sake of visualization.

of different fully-overlapping scenarios, ranging from 1 to 50 WLANs, each consisting in of one AP and one STA. The validation is performed against the Bianchi's analytical model and ns-3. The MCS for all the WLANs is set to 256-QAM. Fig. 8 shows the results in terms of throughput (average and aggregate) and collision probability obtained for fully overlapping networks of different sizes. For comparison purposes, the simulation time used in each scenario has been set to 100 seconds, for both Komondor and ns-3. Notice that such a fully overlapping setting frames a worst-case situation regarding packet collisions. This impacts on the number of events and the simulation time as the network density increases. Nevertheless, much more positive results are expected to be achieved in more realistic non-fully overlapping dense scenarios.

As shown, Komondor maintains its accuracy with respect to Bianchi's model, even when dealing with a lot of nodes. Regarding ns-3, slight differences are noticed in the collisions probability due to the error rate model, where collisions are based on the dropped RTS frames and the use of the Extended Interframe Space (EIFS). Moreover, differences in the throughput increase with the number of nodes, as previously addressed in [26].

To conclude this section, we provide insights into the execution complexity of Komondor. Fig. 9 shows the execution time and the number of generated events in Komondor and ns-3 for each number of WLANs.[7] As shown, the execution complexity of ns-3 is significantly higher than in Komondor. We identify the cause of this difference to be the complex PHY implementation in ns-3, which leads to a larger number of generated events.

5 Komondor and potential use cases

Apart from small deployments consisting of few WLANs under single-channel operation [27], more complex scenarios capturing DCB or high-density scenarios have been already validated and analyzed by using Komondor. In this section, we briefly discuss further potential uses such as the implementation of next-generation WLAN techniques or the inclusion of learning agents to perform efficient spectrum access and spatial reuse.

5.1 Potential usage

Complex wireless environments can be already extensively simulated by Komondor as a result of its reduced computational complexity in comparison to other well-known simulators such as ns-3. A prominent example of a complex scenario mixing both high-density deployments and DCB is discussed in [21], where authors assessed the performance of different DCB policies versus node density (see Fig. 1). In [22], a similar deployment is analyzed while considering different traffic loads. A set of scenarios including DCB is shown in Fig. 10, which were validated in Komondor's validation report v0.1.[8] New features like spatial reuse, MIMO, beamforming and MU communications through OFDMA and/or MU-MIMO are currently under development.

[7]Note that the execution time is strongly dependent of the computer used and its status at the moment of performing the simulation. In our case, we used an Intel Core i5-4300U CPU @ 1.9 GHz x 4 and 7.7 GiB memory.

[8]Komondor's validation report v0.1: `https://github.com/wn-upf/Komondor/blob/master/Documentation/Other/validation\_report\_v01.pdf`.

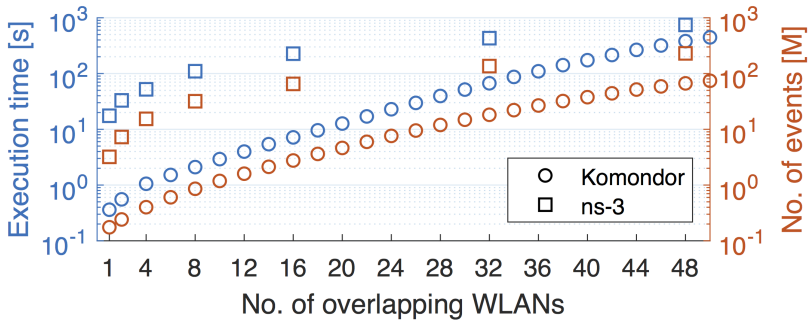


Figure 9: Execution time and number of generated events vs. number of overlapping WLANs.

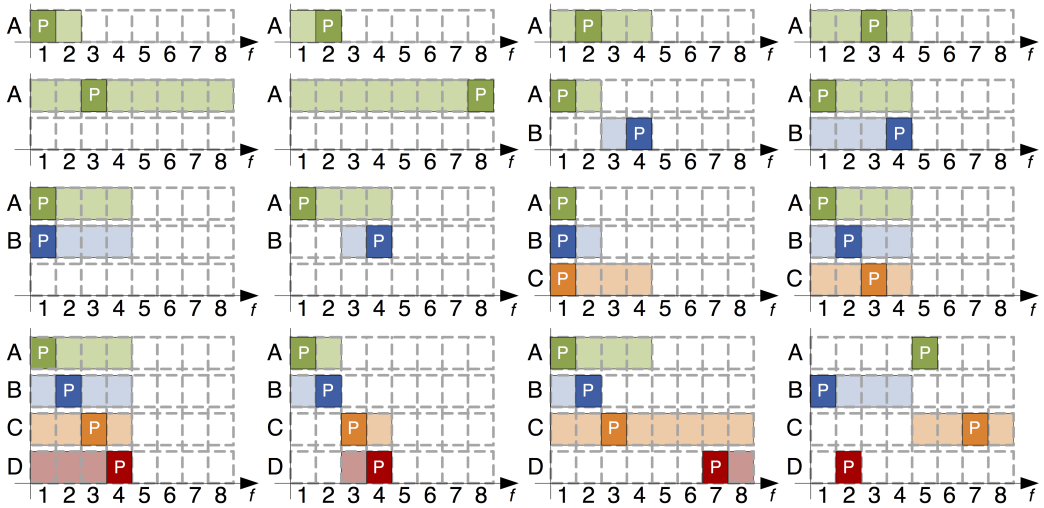
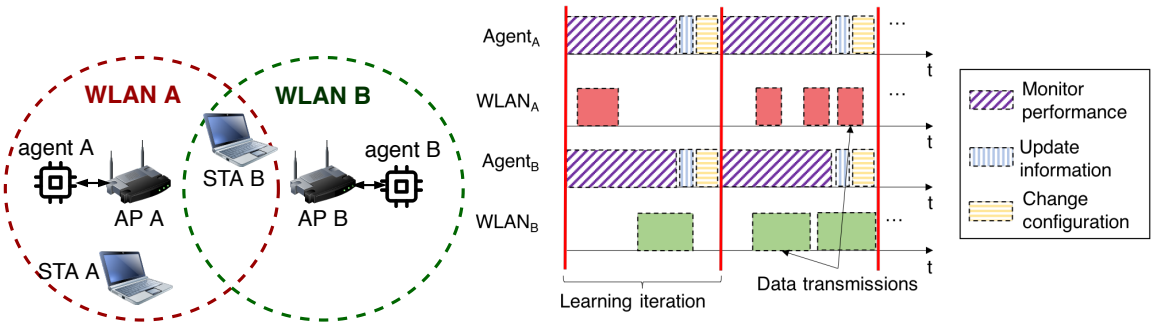


Figure 10: Scenarios with different DCB capabilities.



(a) Agents embedded to APs

(b) Learning operation followed by agents

Figure 11: ML-based operation implemented in Komondor.

5.2 Machine learning agents

In addition to simulating advanced techniques proposed by the latest IEEE 802.11 amendments, Komondor permits including intelligent agents. In particular, agents are embedded to APs (see Fig. 11a) to perform the following operations (see Fig. 11b): *i)* monitor WLAN's performance, *ii)* run an implemented learning method, and *iii)* suggest new configurations to be applied by the WLAN, according to generated knowledge.

The application of intelligent agents has been previously studied in [27, 28], where decentralized learning is employed to both Transmit Power Control (TPC) and Carrier Sense Threshold (CST) adjustment.

6 Conclusions

In this work, we presented Komondor, a wireless network simulator that stems from the need of providing a reliable and low-complexity simulation tool able to capture the operation of novel WLAN mechanisms like DCB or spatial reuse. The operation of Komondor has been validated against the ns-3 simulator and analytical tools such as CTMNs and Bianchi's DCF model. In this regard, we have shown its effectiveness when dealing with high-density scenarios, thereby outperforming ns-3 with respect to the simulation time. The provided validation is fundamental for the next development stages, which contemplate the inclusion of novel techniques in WLANs that have not been fully implemented in other well-known simulators. Some future implementations contemplate OFDMA, MU-MIMO, and the spatial reuse operation, naming a few among others. Finally, we have discussed the potential of Komondor regarding complex scenarios and ML integration. In particular, a preliminary ML-based architecture is already implemented, so that intelligent agents can rule self-configuring operations at different communication levels.

Acknowledgments

This work has been partially supported by a Gift from CISCO University Research Program (CG#890107) & Silicon Valley Community Foundation, by the Spanish Ministry of Economy and Competitiveness under the Maria de Maeztu Units of Excellence Programme (MDM-2015-0502), and by the Catalan Government under grant SGR-2017-1188. The work by S. Barrachina-Muñoz is supported by an FI grant from the Generalitat de Catalunya.

References

[1] I. Selinis, K. Katsaros, M. Allayioti, S. Vahid, and R. Tafazolli, "The Race to 5G Era; LTE and Wi-Fi," *IEEE Access*, vol. 6, no. 1, pp. 56 598–56 636, December 2018.

[2] "IEEE p802.11ax/d2.0," pp. 1–596, November 2017.

[3] B. Bellalta, "IEEE 802.11ax: High-efficiency WLANs," *IEEE Wireless Communications*, vol. 23, no. 1, pp. 38–46, February 2016.

[4] Y. Ghasempour, C. R. da Silva, C. Cordeiro, and E. W. Knightly, "IEEE 802.11 ay: Next-generation 60 GHz communication for 100 Gb/s Wi-Fi," *IEEE Communications Magazine*, vol. 55, no. 12, pp. 186–192, 2017.

[5] L. C. et al., "EXtreme Throughput (XT) 802.11," 2018.

[6] G. Chen and B. K. Szymanski, "Reusing simulation components: cost: a component-oriented discrete event simulator," in *Proceedings of the 34th conference on Winter simulation (WSC): exploring new frontiers*. Winter Simulation Conference, 2002, pp. 776–782.

[7] A. Varga and R. Hornig, "An overview of the OMNeT++ simulation environment," in *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, p. 60.

[8] A. K. Rathi and A. J. Santiago, "The new NETSIM simulation program," *Traffic engineering & control*, vol. 31, no. 5, 1990.

[9] J. Milos, L. Polak, M. Slanina, and T. Kratochvil, "Link-level simulator for WLAN networks," in *Link-and System Level Simulations (IWSLS), International Workshop on*. IEEE, 2016, pp. 1–4.

[10] T. Issariyakul and E. Hossain, "Introduction to Network Simulator 2 (NS2)," in *Introduction to Network Simulator NS2*. Springer, 2012, pp. 21–40.

[11] G. F. Riley and T. R. Henderson, "The ns-3 network simulator," in *Modeling and tools for network simulation*. Springer, 2010, pp. 15–34.

[12] P. Gawłowicz and A. Zubow, "ns3-gym: Extending openai gym for networking research," *arXiv preprint arXiv:1810.03943*, 2018.

[13] R. Stacey *et al.*, "Proposed TGax draft specification," *IEEE 802.11-16/0024rl*, 2016.

[14] "Wi-Fi design," https://www.nsnam.org/docs/models/html/wifi-design.html, 2018.

[15] R. Stacey, "Specification framework for tgax," https://mentor.ieee.org/802.11/dcn/15/11-15-0132-17-00ax-spec-framework.docx, 2016.

[16] Cisco, "NS3 Simulator of 802.11ax (ns3-802.11ax-simulator)," {https://github.com/cisco/ns3-802.11ax-simulator}, year=2017.

[17] G. Pei and T. R. Henderson, "Validation of OFDM error rate model in ns-3," *Boeing Research Technology*, pp. 1–15, 2010.

[18] I. Selinis, M. Filo, S. Vahid, J. Rodriguez, and R. Tafazolli, "Evaluation of the DSC Algorithm and the BSS Color Scheme in Dense Cellular-like IEEE 802.11ax Deployments," in *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Sept 2016, pp. 1–7.

[19] I. Selinis, K. Katsaros, S. Vahid, and R. Tafazolli, "Exploiting the Capture Effect on DSC and BSS Color in Dense IEEE 802.11 ax Deployments," in *Proceedings of the Workshop on ns-3*. ACM, 2017, pp. 47–54.

[20] E. Khorov, A. Kureev, I. Levitsky, and A. Lyakhov, "Testbed to Study the Capture Effect: Can We Rely on this Effect in Modern Wi-Fi Networks," in *2018 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, June 2018, pp. 1–5.

[21] S. Barrachina-Munoz, F. Wilhelmi, and B. Bellalta, "Dynamic Channel Bonding in Spatially Distributed High-Density WLANs," *(forthcoming) IEEE Transactions on Mobile Computing*, 2019.

[22] S. Barrachina-Muñoz, F. Wilhelmi, and B. Bellalta, "To Overlap or not to Overlap: Enabling Channel Bonding in High-Density WLANs," *Computer Networks*, vol. 152, pp. 40 – 53, 2019.

[23] S. Merlin, et al., "TGax Simulation Scenarios," doc. IEEE 802.11-14/0980r16, 2016.

[24] B. Bellalta, A. Zocca, C. Cano, A. Checco, J. Barcelo, and A. Vinel, "Throughput analysis in csma/ca networks using continuous time markov networks: a tutorial," in *Wireless Networking for Moving Objects*. Springer, 2014, pp. 115–133.

[25] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE Journal on selected areas in communications*, vol. 18, no. 3, pp. 535–547, 2000.

[26] R. Patidar, S. Roy, T. R. Henderson, and M. Mehrnoush, "Validation of wi-fi network simulation on ns-3," 2017.

[27] F. Wilhelmi, S. Barrachina-Muñoz, B. Bellalta, C. Cano, A. Jonsson, and G. Neu, "Potential and pitfalls of multi-armed bandits for decentralized spatial reuse in wlans," *Journal of Network and Computer Applications*, vol. 127, pp. 26–42, 2019.

[28] F. Wilhelmi, C. Cano, G. Neu, B. Bellalta, A. Jonsson, and S. Barrachina-Muñoz, "Collaborative spatial reuse in wireless networks via selfish multi-armed bandits," *Ad Hoc Networks*, 2019.

8.4

Online Primary Channel Selection for Dynamic Channel Bonding in High-Density WLANs

Sergio Barrachina-Muñoz, Francesc Wilhelmi, and Boris Bellalta*

Abstract

In order to dynamically adapt the transmission bandwidth in wireless local area networks (WLANs), dynamic channel bonding (DCB) was introduced in IEEE 802.11n. It has been extended since then, and it is expected to be a key element in IEEE 802.11ax and future amendments such as IEEE 802.11be. While DCB is proven to be a compelling mechanism by itself, its performance is deeply tied to the primary channel selection, especially in high-density (HD) deployments, where multiple nodes contend for the spectrum. Traditionally, this primary channel selection relied on picking the most free one without any further consideration. In this paper, in contrast, we propose dynamic-wise (DyWi), a light-weight, decentralized, online primary channel selection algorithm for DCB that improves the expected WLAN throughput by considering not only the occupancy of the target primary channel but also the activity of the secondary channels. Even when assuming important delays due to primary channel switching, simulation results show a significant improvement both in terms of average delay and throughput.

1 Introduction

Modern applications like augmented reality, virtual reality, or real-time 8K video are pushing next-generation (nextGen) wireless local area networks (WLANs) to support ever-increasing demands on performance. In addition, the complexity of nextGen high-density (HD) deployments, where numerous wireless devices will contend for accessing the medium, hinders even more the task of achieving high throughput and low latency.

In order to improve the performance of nextGen WLANs, we focus in this work on spectrum management techniques. In particular, two well-known techniques have been widely studied in this regard: channel allocation (CA) and dynamic channel bonding (DCB). CA is the method to assign portions of the spectrum (or channels) to one or multiple WLANs. In contrast, DCB is a technique whereby two or more channels are bonded according to their instant occupancy, enabling wider bandwidths per transmission, and thus potentially reaching higher data rates.

Although much has been understood from the works on CA and DCB in the literature, little has been assessed with respect to combining CA with DCB altogether in WLANs, particularly for high-density (HD) deployments. Nevertheless, while DCB has shown a tremendous potential to outperform traditional single-channel (SC) [1–3], its performance is severely tied to the CA. Especially, the primary channel selection is critical since it runs the backoff procedure of the carrier sense multiple access with collision avoidance (CSMA/CA) protocol. Still, always selecting the least occupied channel as the primary is no longer appropriate for DCB since potential bonds with adjacent channels should be also considered.

As for works combining CA and DCB for WLANs, we find a distributed algorithm for jointly allocating channel center frequencies and bandwidths [4], or a centralized approach for maximizing the network fairness [5]. In addition, a heuristic-based algorithm for primary channel selection based on the bonding direction likelihoods was recently presented in [6]. However, such likelihoods are estimated by assuming a known number of users in each channel, which is normally not feasible in real deployments. It is worth noticing that the aforementioned works consider fully-backlogged

*All the authors are with the Wireless Networking research group at Universitat Pompeu Fabra, Barcelona, Spain (e-mail: {sergio.barrachina, francisco.wilhelmi, boris.bellalta}@upf.edu). This work has been partially supported by the Spanish Ministry of Economy and Competitiveness under the Maria de Maeztu Units of Excellence Programme (MDM-2015-0502), by a Gift from the Cisco University Research Program (CG#890107, Towards Deterministic Channel Access in High-Density WLANs) Fund, a corporate advised fund of Silicon Valley Community Foundation, and by PGC2018-099959-B-I00 (MCIU/AEI/FEDER,UE). The work done by S. Barrachina-Muñoz is supported by a FI grant from the Generalitat de Catalunya.

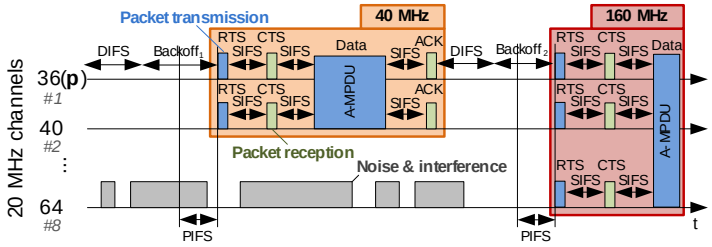


Figure 1: CSMA/CA operation of a node operating under DCB in the IEEE 802.11ac/ax channelization scheme for the UNII-1 and UNII-2 bands. The numbers preceded by a pound symbol (#) represent a simpler channel indexation.

traffic, thus missing insights on more realistic patterns with different traffic needs. Finally, an uncertain traffic CA approach was presented in [7]. Still, a centralized controller in the backend is required.

In this paper, we formulate dynamic-wise (DyWi), a decentralized, lightweight algorithm that leverages information about the sensed spectrum occupancy in the whole allocated bandwidth of a node (i.e., primary and secondary channels). Based on such occupancy, the primary channel is selected in an online manner with the aim of maximizing the expected throughput. It does so by considering not only the activity of the target primary channel but also the potential bonds that could be established with its adjacent channels. DyWi is adaptive in the sense that a new primary channel is only adopted when the WLAN performance is below a given satisfaction threshold. Besides, since DyWi relies just on local information, neither neighbor messaging nor a central controller is required. This property makes DyWi suitable to be implemented in off-the-shelf access points (APs), thus avoiding costly inter-WLAN collaboration.

Simulations of DyWi in IEEE 802.11ax HD deployments show important improvements with respect to traditional primary channel selection, even when considering substantial delays due to channel switching.

2 Primary channel selection for DCB

2.1 Dynamic channel bonding

DCB was first introduced in IEEE 802.11n (2009), where two contiguous 20-MHz channels could be bonded to form a single 40-MHz channel. Then, IEEE 802.11ac (2013) extended the DCB capability to bond up to 8 20-MHz channels, reaching a maximum of 160-MHz transmission bandwidth. While IEEE 802.11ax (2019) keeps such a limit, future amendments like EXtreme throughput (i.e., IEEE 802.11be) aim to support up to 320-MHz transmissions. Figure 1 shows the operation timeline of a node implementing DCB with primary channel $p = 1$. Note that the bandwidth selection is decided according to the occupancy of the secondary channels during the PCF Interframe Space (PIFS) previous to the backoff termination. Accordingly, transmissions of 40 and 160 MHz are performed after the expiration of the first and second backoff in the example, respectively.

2.2 Online selection of the primary channel

Let an AP belonging to WLAN w operate under DCB and have allocated the full available bandwidth according to a channelization scheme $\mathcal{C}$. As an example, the channelization (or allowed transmission channels) in Fig. 1 is $\mathcal{C} = \{\{1\},\{2\},...,\{8\},\{1,2\},...,\{7,8\},\{1,2,3,4\},\{5,6,7,8\},\{1,...,8\}\}$, where $\mathcal{C}_{\text{20-MHz}} = \{\{1\},\{2\},...,\{8\}\} \subset \mathcal{C}$ is the set of 20-MHz (*basic* or non-bonded) channels in $\mathcal{C}$. Then, since the primary channel must be 20-MHz, $p \in \mathcal{C}_{\text{20-MHz}}$. Note that throughout the rest of the paper, we use $b = 20$ MHz to denote the bandwidth of a *basic* channel.

For the sake of identifying a convenient primary channel in an online manner, we rely on an iterative algorithm. In essence, the AP of w periodically makes a decision about the primary channel selection, where each decision instant represents the beginning of an iteration. Specifically, at the beginning of a given iteration t, the AP of w measures the throughput $s_{w,t-1}$ achieved during the last iteration $t-1$ and acts according to a satisfaction condition.[1] Namely, the primary channel remains

[1] In this work, we focus on the successful downlink traffic as the main performance metric. However, the algorithm can be easily extended to consider other parameters such as latency.

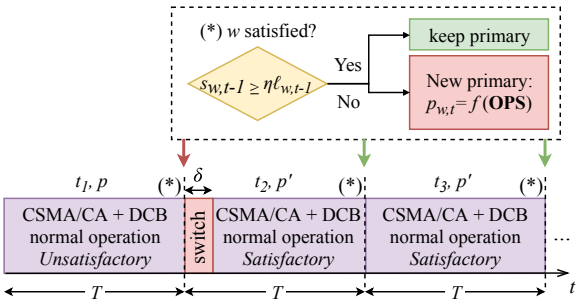


Figure 2: Example of the satisfaction-based online primary channel selection. Since w is not satisfied at the end of iteration t_1, the primary channel is switched from $p_{w,1} = p$ to $p_{w,2} = p'$, determined by a given OPS rule. In contrast, with the new primary, w gets satisfied in t_2 and keeps p' in t_3.

Algorithm 1: Online primary selection. OPS refers to the selected online primary selection rule.

Input: $p_{w,t}$, η, OPS, $\mathcal{C}$;
iteration $t \leftarrow 0$;
while *WLAN w active* **do**
  while *iteration t not finished* **do**
    CSMA/CA normal operation with DCB;
  $t \leftarrow t$+1;
  $s_{w,t\text{-}1} \leftarrow$ `measure_throughput(`t-1`)`;
  $\mathcal{F}_{w,t\text{-}1} \leftarrow$ `get_idleness(`t-1`)`;
  if $s_{w,t\text{-}1} < \eta \ell_{w,t\text{-}1}$ **then**
    $p_{w,t} \leftarrow$ `apply_ops(`$p_{w,t\text{-}1}$`,`$\mathcal{F}_{w,t\text{-}1}$`,OPS,`$\mathcal{C}$`)`
    `channel_switch(`$p_{w,t}$`,` δ`)`;

the same if w is satisfied because a sufficient share of the generated traffic has been successfully sent during the last iteration,[2] i.e., $s_{w,t\text{-}1} \geq \eta \ell_{w,t\text{-}1}$, where η is the satisfaction ratio and $\ell_{w,t\text{-}1}$ is the actual generated traffic load in that iteration. Otherwise, w will switch to a new primary channel at the cost of remaining inactive during a period δ, required to announce the new primary channel to the associated stations (STAs), and to apply the new channel configuration. Note that the way the new primary channel is selected depends on the online primary selection (OPS) rule. The temporal evolution of the general procedure is displayed in Fig. 2 through an illustrative example.

Algorithm 1 shows the pseudocode of the online primary channel selection for a generic OPS rule. Note that any OPS rule relies on the gathered data about the bandwidth occupancy in the last iteration. Formally, the empirical probability that a subset of n channels in channelization $\mathcal{C}$ was free (or idle) during the last iteration $t-1$ given a primary $p_{w,t}$ is given by

$$\mathcal{F}_{w,t-1}(\mathcal{C}, p_{w,t}, n) = \mathbb{E}_{t-1}\big[P_{\text{rx}}(c) < \text{CCA}, \forall c \in \mathcal{C}(p_{w,t}, n)\big], \tag{1}$$

where $P_{\text{rx}}(c)$ is the power received at *basic* channel c, CCA is the clear channel assessment (CCA) threshold, and $\mathcal{C}(p_{w,t}, n)$ is the set of *basic* channels used in the transmission, which is mandated by the channelization scheme $\mathcal{C}$. For instance, following the IEEE 802.11ac/ax channelization, for primary $p = 6$ and $n = 2$, the corresponding 40-MHz bonded channel is $\mathcal{C}(6, 2) = \{5, 6\}$ (channels #52 and #56 in the standard). With slight abuse of notation, $\mathbb{E}_{t-1}$ represents the expected value function at iteration $t-1$.

Three OPS rules are evaluated in this work: dynamic-random (DR), dynamic-free (DF), and dynamic-wise (DiWy or DW). In the event of an unsatisfactory iteration, DR selects a new primary channel uniformly at random, DF picks the one found most free during the last iteration, and DyWi selects it based on the forecast throughput given the probability of bonding in every possible bandwidth. Such probabilities are estimated by periodically measuring the energy in all the secondary channels as done during the PIFS period. Apart from DR and DF, we also consider as baseline the traditional fix primary (FP) allocation, which does not change the primary channel under any circumstances, so it is not referred in Algorithm 1.

[2]We rely just on data from the last iteration for lowering memory demands and enabling fast adaptability in dynamic environments.

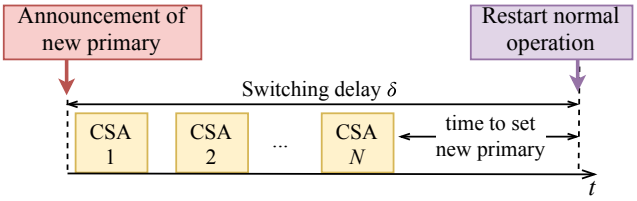


Figure 3: Channel switch procedure.

Regarding the channel switch operation, the 802.11h standard (later included in amendments like 802.11ax) states that the AP may notify the associated STAs through Channel Switch Announcement (CSA) frames [8]. As depicted in Fig. 3, the channel switch is scheduled so that all the STAs in the WLAN have the opportunity to receive at least one CSA element before the actual switch. Specifically, the number of remaining frames to be broadcast until switching to the new primary channel is known as CSA count value. Accordingly, the switching delay δ is the sum of the time it takes to broadcast the CSA frames and the time left to STAs to adapt to the new primary channel.

2.3 Dynamic-wise (DyWi) primary channel selection

A main question arises when considering DyWi regarding the way the primary channel should be selected. Assume a scenario where a WLAN w is allocated 80 MHz accounting from channel 1 to 4. At the end of iteration t-1, w is unsatisfied and changes its primary from $p_{w,t\text{-}1} = 2$ to $p_{w,t}$. Assume also that the probabilities of finding free each of its allocated 20-MHz channels in iteration t-1 was $\boldsymbol{\pi}_{w,t\text{-}1} = [0.93, 0.38, 0.85, 0.85]$, where $\boldsymbol{\pi}_{w,t\text{-}1}[c]$ is the probability of finding the *basic* channel c free. Then, two main options may be contemplated as best choice for selecting $p_{w,t}$ to maximize the throughput $s_{w,t}$ of the upcoming iteration: *i*) to pick the primary with highest probability to be free (i.e., $p_{w,t} = 1$ in this case), or *ii*) to pick the primary providing the highest potential average data rate considering both its probability to be free, as well as the probability of the channels nearby (e.g., $p_{w,t} \in \{3, 4\}$). Notice that there is not an accurate choice to provide beforehand since multiple parameters apart from $\boldsymbol{\pi}_{w,t\text{-}1}$ will impact the performance. For example, dynamic network activity affects to the signal-to-interference-plus-noise ratio (SINR), modulation coding scheme (MCS), or packet error rate.

We tackle this point at issue by proposing a lightweight maximization problem for the forecast data rate of WLAN w at iteration t. The idea behind this approach is that maximizing the successful data rate should lead to maximizing the throughput as well. Thus, in spite of being a sub-optimal formulation for maximizing the throughput,[3] selecting the primary channel according to the forecast average data rate is a convenient heuristic, as shown later. In particular, the average data rate is estimated by the probability of transmitting at each possible bandwidth. Then, we can formulate the problem as

$$\begin{aligned} &\underset{p_{w,t} \neq p_{w,t\text{-}1}}{\operatorname{argmax}} \ \hat{r}_{w,t}(p_{w,t}), \text{with} \\ &\hat{r}_{w,t}(p_{w,t}) = \sum_{n \in \mathcal{N}} \mathbb{P}_{w,t}\big(\mathcal{F}_{w,t-1}, p_{w,t}, n\big) r_w\big(n\big), \end{aligned} \tag{2}$$

where $\hat{r}_{w,t}(p_{w,t})$ is the forecast average data rate by WLAN w at iteration t for new primary $p_{w,t}$, and n is the number of bonded channels, which should be permitted by the channelization scheme. For instance, in the IEEE 802.11ax amendment, $n \in \mathcal{N} = \{1, 2, 4, 8\}$, for 20 to 160-MHz bandwidths. $\mathbb{P}_{w,t}(\mathcal{F}_{w,t-1}, p_{w,t}, n)$ is the probability that w transmits in n contiguous channels in the starting iteration given $p_{w,t}$ is selected, and $r_w(n)$ is the data rate given the bandwidth nb. Note that r also depends on the MCS index, which will vary according to the signal-to-noise ratio (SNR) at the STA.

In order to estimate the probability of transmitting in each possible combination of channels, we rely on the empirical probability $\mathcal{F}_{w,t-1}(\mathcal{C}, p_{w,t}, n)$, which was updated during the CSMA/CA operation in iteration t-1. Since DCB is implemented, the largest available bandwidth is always picked per transmission. Hence, the probability of transmitting in a certain bandwidth is contingent on the probability of transmitting in higher bandwidths. Specifically,

[3] One should consider different complex parameters such as the buffer status and environment dynamics to define an optimal formulation.

Table 1: Evaluation setup.

Param.	Description	Value
f_c	Central frequency	5.25 GHz
b	*Basic* channel bandwidth	20 MHz
L_d	Data packet size	12000 bits
N_b	Buffer capacity	150 packets
N_a	Max. no. of aggregated packets in a frame	64
CW_{min}	Min. contention window	16
m	No. of backoff stages	5
MCS	IEEE 802.11ax MCS index	0 - 11
η	MCS's packet error rate	0.1
CCA	CCA threshold	-82 dBm
P_{tx}	Transmission power	15 dBm
G_{tx}	Transmitting gain	0 dB
G_{rx}	Reception gain	0 dB
PL(d)	TMB indoor path loss for 11ax	see [11]
CE	Capture effect threshold	20 dB
N	Background noise level	-95 dBm
$\mathcal{C}$	Channelization for UNII-1 & UNII-2	36(1) - 64(8)
$\lvert\mathcal{C}_{20\text{-MHz}}\rvert$	No. of 20-MHz channels in the system	8
$\mathcal{N}$	Allowed no. of bonded channels	$\{1, 2, 4, 8\}$
OPS	Online primary selection rule	DR, DF, DW
T	Iteration duration	1 s
T_{obs}	Simulation duration	25 s
η	Satisfaction ratio	0.9
δ	Switching delay	0, 100 ms

$$\mathbb{P}_{w,t}(\mathcal{F}_{w,t-1}, p_{w,t}, n) = \mathcal{F}_{w,t-1}(\mathcal{C}, p_{w,t}, n) - \sum_{n' \in \{\mathcal{N} | n' > n\}} \mathbb{P}_{w,t}(\mathcal{F}_{w,t-1}, p_{w,t}, n'), \tag{3}$$

where the probability of transmitting in wider bandwidths is subtracted due to the constraint $n' \in \{\mathcal{N}|n' > n\}$. For instance, for bands UNII-1 and UNII-2 altogether (i.e., $\mathcal{C}_{20\text{-MHz}} = \{1, ..., 8\}$), we define $\mathbb{P}(\mathcal{F}, p, 8) = \mathcal{F}(\mathcal{C}, p, 8)$ for $n = 8$ (i.e., 160-MHz). Similarly, on the other end, $\mathbb{P}(\mathcal{F}, p, 1) = \mathcal{F}(\mathcal{C}, p, 1) - \mathbb{P}(\mathcal{F}, p, 2) - \mathbb{P}(\mathcal{F}, p, 4) - \mathbb{P}(\mathcal{F}, p, 8)$ for $n = 1$ (i.e., 20-MHz).

As for the complexity of the presented OPS rules, note that they are all computational lightweight; especially DR since it does not keep track of any data. Despite DF's complexity increases with the number of *basic* channels, $\mathcal{F}_{DF}\big(|\mathcal{C}_{20\text{-MHz}}|\big)$, its complexity remains also low. DyWi's complexity, however, is bounded by $\mathcal{F}_{DW}\big(|\mathcal{C}_{20\text{-MHz}}|(\log_2 |\mathcal{C}_{20\text{-MHz}}| + 1)^2\big)$ for the IEEE 802.11ax channelization. Nonetheless, DyWi is completely tractable in operation time by off-the-shelf network cards since the number of possible bonds in the 5-GHz band is still small.

3 System model

For evaluating the performance of the OPS rules presented in Section 2, we simulate IEEE 802.11ax HD deployments using the Komondor [9] wireless network simulator v1.2.1c. For simplicity, we consider negligible propagation delay, downlink traffic, and WLANs composed by one AP and one STA. The packet arrival process at each AP follows a Poisson process, generating packets every $t_n \sim \text{Exponential}\big(L_d/\bar{\ell}\big)$, where $L_d/\bar{\ell}$ is the average packet arrival rate given a fixed packet length L_d and average arrival bit rate $\bar{\ell}$ [10]. As for the packet reception model, we consider that a packet is lost if the SINR perceived at the receiver is below the capture effect (CE) threshold. Note that transmitted power is spread over the channels used in the transmission bandwidth. We also consider the same CCA (-82 dBm) both in primary and secondary channels to make channel access more restrictive, as proposed in [1].

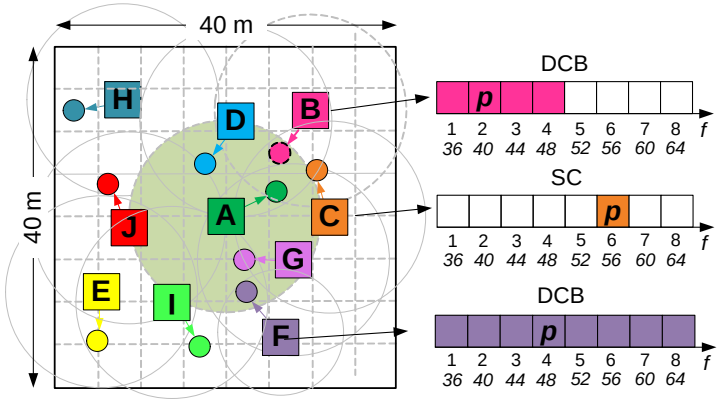
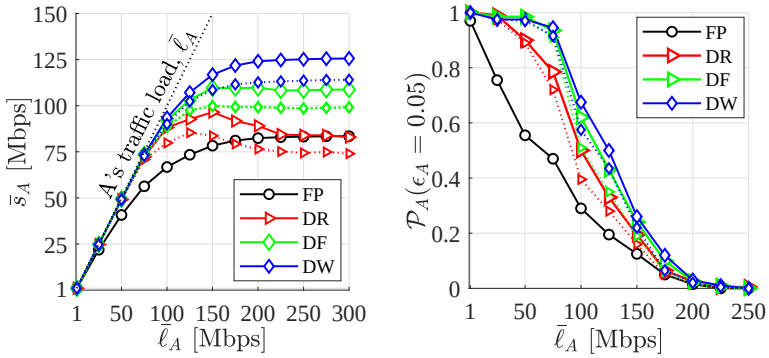


Figure 4: Deployment with WLAN A in the center. The IEEE 802.11ac/ax 20-MHz channel indexation is displayed in italic. WLANs may implement SC or DCB in different allocated bandwidths.



(a) Average throughput. (b) Prob. of *desired throughput*.

Figure 5: Performance metrics of WLAN A for the different primary selection rules. Continuous lines assume no switching cost while dashed lines correspond to $\delta = 100$ ms.

Regarding the simulation deployments, we contemplate dense 40x40 m^2 scenarios like the one shown in Fig. 4. One WLAN (A) remains always located at the center in every scenario, and other 9 WLANs are spread uniformly at random in the area. The only condition is that any pair of APs must be separated at least $d^{\min}_{\text{AP-AP}} = 10$ m. The STA of each WLAN is located also uniformly at random at a distance $d_{\text{AP-STA}} \in [d^{\min}_{\text{AP-STA}}, d^{\max}_{\text{AP-STA}}] = [1, 5]$ m from the AP. Note that WLANs are not required to be within the carrier sense range of the others, i.e., the simulations capture spatial distribution effects.

Regarding the CA, all the WLANs are set with a random primary channel in the eight *basic* channels of the UNII-1&2 sub-bands (i.e., $p_{\text{w}} \sim U[1, 8], \forall w$). The set of allocated basic channels is assigned uniformly at random as well. That is, the number of allowed basic channels for transmitting in w is $|\mathcal{C}_w| \sim U\{1, 2, 4, 8\}, \forall w$, except for WLAN A, which is allocated the whole bandwidth in all the scenarios (i.e., $\mathcal{C}_{\text{A}} = \{1, ..., 8\}$). While A implements DCB, the rest of WLANs implement SC or DCB with same probability 1/2).

We generate $N_{\text{D}} = 200$ random deployments following the aforementioned conditions and evaluate $N_\ell = 17$ values of A's average traffic load $\bar{\ell}_A$ ranging from 1 to 400 Mbps. The rest of WLANs are set with random average traffic load inside this range, i.e., $\bar{\ell}_w \sim U[1, 400]$ Mbps. In addition to traditional FP, we consider the $N_{\text{P}} = 3$ OPS schemes proposed in Section 2 (i.e., DR, DF, DW). For the latter ones, two switching delay costs $\delta \in \{0, 100\}$ ms are assessed.[4] Consequently, we simulate $N_{\text{D}} \times N_\ell \times (1 + 2N_{\text{P}}) = 23,800$ scenarios. The simulation time of each scenario is $T_{\text{obs}} = 25$ seconds. As for the configuration of the online algorithms, we set the iteration time $T = 1$ s and satisfaction ratio $\eta = 0.9$. Note that we consider a value of η smaller than 1 to provide stability to the algorithm.

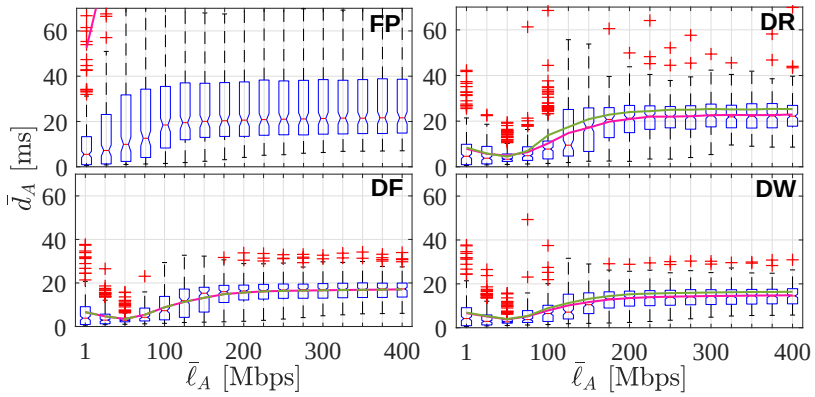


Figure 6: Distribution of the average delay of A for $\delta = 0$ ms. The mean value for the different loads is represented through the purple and green lines for $\delta = 0$ and $\delta = 100$ ms, respectively.

4 Performance Evaluation

Figs. 5 and 6 collect key performance metrics of WLAN A in the proposed scenario for different traffic loads. Namely, Fig. 5a shows the average throughput $\bar{s}_\mathrm{A}$ computed as the number of bits successfully transmitted (acknowledged) divided by the simulation time T_obs. The probability $\mathcal{P}_\mathrm{A}$ of successfully transmitting *sufficient* traffic is plotted at Fig. 5b. Such a probability is computed as the portion of scenarios accomplishing $\bar{s}_\mathrm{A} \geq (1-\epsilon_s)\bar{\ell}_\mathrm{A}$, where $\epsilon_s = 0.05$ is set to deal with the non-deterministic traffic generation. Finally, Fig. 6 shows the average packet delay $\bar{d}_\mathrm{A}$.

As shown by the different metrics, DyWi clearly outperforms FP and DR both in terms of throughput and delay. In fact, even when considering a huge adaptation cost delay $\delta = 100$ ms (i.e., an important 10% penalty with respect to the iteration duration T), $\mathcal{P}_A$ is prominently increased for moderate and medium loads. While DR may be counterproductive for high loads (see Fig. 5a), both DF and DyWi keep a constant performance after saturating. Such a throughput reduction for DR is caused by the fact that the larger $\bar{\ell}_A$, the harder to remain satisfied, which leads to more frequent channel switching. Then, critically for DR, the random selection of the primary leads to picking each channel with the same probability. Accordingly, the average throughput converges to FP's because, on average (for all the scenarios), the primary channels are equiprobable selected whenever the satisfaction condition is not accomplished.

As for the average delay, even though we can see by the outliers in Fig. 6 that adopting (2) does not guarantee optimal performance for every scenario, DyWi outperforms the rest of selection rules in most of the cases. A phenomenon worth noticing is the reduction of the average delay from low to moderate traffic loads for online selection algorithms. The reason lies in the buffer dynamics. Specifically, during the normal operation of an unsatisfactory iteration, while for low traffic loads, the number of aggregated packets per frame is pretty low, it is much greater for higher loads. Accordingly, since the buffer tends to remain unsaturated for low loads, during the new configuration setup after an unsatisfactory iteration, the buffer is normally filled up. As a result, the first frame of the next iteration – usually containing many aggregated data packets– affects much more to the average delay for low traffic loads than for high loads.

In order to assess the temporal evolution of the different algorithms, Fig. 7 plots the cumulative distribution function (CDF) of the number of iterations k required to reach a satisfactory primary channel for moderate, medium and high traffic loads. As expected, the lower the load, the higher the value of CDF(k) for any k. Note that there are few unusual scenarios where the CDF varies for FP. Those are the cases where the load is almost adequate from the very beginning and the stochastic nature of the traffic generation makes the throughput to vary around $\eta\bar{\ell}_A$ as the simulation progresses.

It is also observed that the highest value of the CDF is normally provided by DyWi, which wisely adapts to the medium. However, DR may be even better for intricate scenarios (high k) and medium/high loads. This suggests that for such difficult scenarios, relying on (2) may not be optimal due to the unexpected and harming interactions generated at the moment of changing the primary. Nevertheless, even though these unusual cases leave room for further improvement, DyWi is an effective solution that may be adopted in off-the-self WLANs due to its light-weight complexity

[4]The number and time between CSA frames are configurable (and dependant on the AP manufacturer). In this work, we consider a practical value $\delta = 100$ ms that allows stressing the performance of the evaluated algorithms.

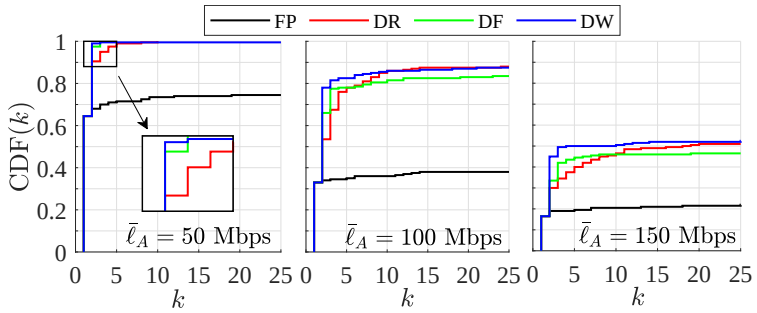


Figure 7: Cumulative distribution function of the number of iterations k required to reach satisfaction.

and direct improvement.

5 Conclusions

In this work, we have formulated DyWi, a lightweight and decentralized online primary channel selection algorithm for WLANs. DyWi aims at maximizing the throughput by iteratively estimating the occupancy of the primary and secondary channels, thus boosting potential bonds. Results show significant improvements with respect to traditional allocation, even under high switching adaptation costs.

References

[1] M. Park, "IEEE 802.11 ac: Dynamic bandwidth channel access," in *Communications (ICC)*. IEEE, 2011, pp. 1–5.

[2] L. Deek, E. Garcia-Villegas, E. Belding, S.-J. Lee, and K. Almeroth, "Intelligent channel bonding in 802.11n WLANs," *IEEE Transactions on Mobile Computing*, vol. 13, no. 6, pp. 1242–1255, 2013.

[3] S. Barrachina-Muñoz, F. Wilhelmi, and B. Bellalta, "Dynamic channel bonding in spatially distributed high-density WLANs," *(In press) IEEE Transactions on Mobile Computing*, 2019.

[4] J. Herzen, R. Merz, and P. Thiran, "Distributed spectrum assignment for home WLANs," in *INFOCOM*. IEEE, 2013, pp. 1573–1581.

[5] B. Bellalta, A. Checco, A. Zocca, and J. Barcelo, "On the interactions between multiple overlapping WLANs using channel bonding," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 2, pp. 796–812, 2016.

[6] S. Khairy, M. Han, L. X. Cai, Y. Cheng, and Z. 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] A. Nabil, M. J. Abdel-Rahman, and A. B. MacKenzie, "Adaptive channel bonding in wireless LANs under demand uncertainty," in *Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE, 2017, pp. 1–7.

[8] IEEE, "Spectrum and transmit power management extensions in the 5 ghz band in europe," *IEEE Std 802.11h-2003*, pp. 1–75, Oct 2003.

[9] S. Barrachina-Muñoz, F. Wilhelmi, I. Selinis, and B. Bellalta, "Komondor: a wireless network simulator for next-generation high-density WLANs," *Wireless Days (WD)*, 2019.

[10] S. Barrachina-Muñoz, F. Wilhelmi, and B. Bellalta, "To overlap or not to overlap: enabling channel bonding in high-density WLANs," *Computer Networks*, vol. 152, pp. 40 – 53, 2019.

[11] T. Adame, M. Carrascosa, and B. Bellalta, "The TMB path loss model for 5 GHz indoor WiFi scenarios: On the empirical relationship between RSSI, MCS, and spatial streams," *Wireless Days (WD)*, 2019.

8.5

Wi-Fi All-Channel Analyzer

Sergio Barrachina-Muñoz, Boris Bellalta, Edward Knightly

Abstract

In this paper, we present WACA, the first system to simultaneously measure the energy in all 24 Wi-Fi channels that allow channel bonding at 5 GHz with microsecond scale granularity. With WACA, we perform a first-of-its-kind measurement campaign in areas including urban hotspots, residential neighborhoods, universities, and a sold-out stadium with 98,000 fans and 12,000 simultaneous Wi-Fi connections. The gathered dataset is a unique asset to find insights otherwise not possible in the context of multi-channel technologies like Wi-Fi. To show its potential, we compare the performance of contiguous and non-contiguous channel bonding using a trace-driven framework. We show that while non-contiguous outperforms contiguous channel bonding's throughput, occasionally bigger by a factor of 5, their average throughputs are similar.

1 Introduction

Channel bonding is a key mechanism for increasing Wi-Fi data rates as the maximum data rate increases in proportion to the total channel bandwidth. In Wi-Fi, while the *basic* channel width remains 20 MHz, the maximum *bonded* channel width has increased from 40 MHz in 802.11n [1] to 160 MHz in 802.11ac/ax [2, 3], and 320 MHz in 802.11be [4]. During this time, the standard has evolved to not only support wider bandwidths, but also to enable more sophisticated channel bonding policies: in 802.11n, only "static" channel bonding was allowed in which a fixed group of pre-configured channels must always be bonded. Today, the standard enables a far richer set of capabilities including dynamic selection of channel width as well as bonding both contiguous and non-contiguous channels.

In this paper, we make the following three contributions. First, we develop WACA as the first system to simultaneously measure all 24 Wi-Fi channels at 5 GHz with a 10 μsec sampling rate. WACA employs multiple synchronized WARP software defined radios (SDRs) and has 24 antennas and 24 radio frequency (RF) chains to match the number of Wi-Fi channels that allow channel bonding (from channel 36 to 161) in the IEEE 802.11ac/ax standards. Our approach contrasts with prior work that uses one or several RF chains, thereby encountering *second* scale channel switching delays. Thus, no prior methods capture all channels at the channel-access μsec timescale.

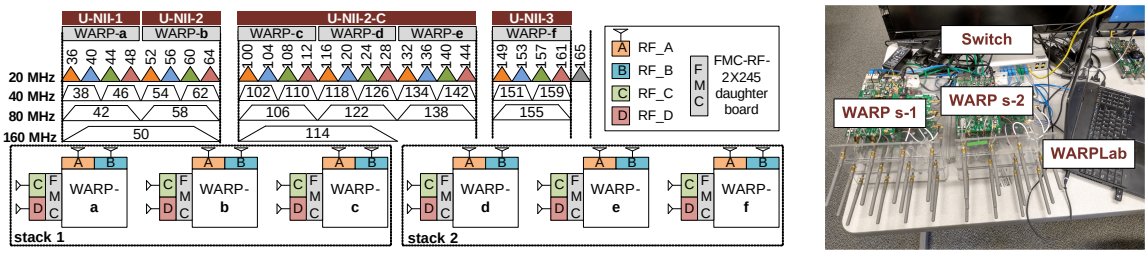
Second, using WACA, we conduct extensive measurement campaigns covering two continents, dense urban areas, and places of interest such as university campuses, apartment buildings, shopping malls, and hotels. We also perform measurements in the Futbol Club Barcelona's Camp Nou, one of the largest sports stadiums in the world. The shortest campaign took 20 minutes and the longest covered more than 1 week, and the total number of samples exceeds 10^{11}. In all cases, we record signal strength on all channels at 10 μsec sample rate, from which we infer the epochs for which each channel is occupied.[1][2] Notice that the traces compose an unprecedented dataset to find insights otherwise not possible. We highlight Wi-Fi as the main research application of the dataset but others can also be taken into account, e.g., cognitive radios, coexistence between Wi-Fi and LTE, and coexistence between Wi-Fi and Internet of Things (IoT) technologies, to list a few examples.

Third, we introduce a trace-driven framework to evaluate the performance of channel bonding policies using the aforementioned high-resolution traces of channel activity. As the channel occupancies are highly dynamic, we group epochs according to their average utilization. While the stadium measurements have band occupancies as high as 99%, even dense urban scenarios with many competing basic service sets (BSS's) yielded maximum occupancies near 45%.

Finally, we use the trace-driven framework to compare the performance of contiguous vs. non-contiguous channel bonding and find that the increased flexibility of the latter can yield a throughput improvement of up to 5×. Unfortunately, the scenarios for these gains occur quite rarely yielding

[1] The WACA dataset v1 can be found at `https://www.upf.edu/web/wnrg/wn-datasets`.

[2] All of the source code of WACA is open, encouraging sharing of algorithms between contributors and providing the ability for people to improve on the work of others under the GNU General Public License v3.0. The repository can be found at `https://github.com/sergiobarra/WACA_WiFiAnalyzer`.



(a) IEEE 802.11ac/ax channelization and assignation per RF.

(b) Deployment schematic.

Figure 1: The WACA all WiFi channels spectrum analyzer.

modest average gains of less than 10%, which may ultimately favor contiguous channel bonding, since it is simpler to implement.

Due to space constrains, we leave the study of key factors such as spectral correlation, bandwidth prevention, or standard compliant channel bonding policies as future work.

2 Related work

Prior work performed spectrum measurements for Wi-Fi traffic, e.g., [5–15]. Example objectives include creating interference maps [15], assessing interference behavior [16], surveying Wi-Fi usage [10], quantifying spectrum occupancy in outdoor testbeds [5, 6], designing efficient scanning methods [7, 11], modeling spectrum use [8], opportunistic spectrum access [9, 14], dynamic channel selection [17], and assessing real-world network behavior by examining data from thousands APs [12]. Unfortunately, no such prior work provided simultaneous measurement across the entire 5 GHz band, which we require for our channel bonding study. While some papers do provide multi-channel measurements, e.g., [5, 9, 14, 18], they do so via sequential scanning, thus taking on the order of seconds to change from one channel to the next, orders of magnitude beyond the transmission time scale for channel bonding. Namely, WACA measures all channels simultaneously using SDRs having a sampling rate of 10μsec. Moreover, prior work does not consider (for example) stadiums.

Throughput gains of channel bonding have been demonstrated previously in testbeds. In particular, IEEE 802.11n static channel bonding has been shown to be affected not only by link signal quality, but also by the power and rates of neighboring links [19]. Likewise, intelligent channel bonding management was shown to benefit from identifying the signal strength of neighboring links and interference patterns [20]. High bandwidths were shown to be vulnerable to increased thermal noise or the power per Hertz reduction [21, 22]. Nonetheless, existing experimental results have targeted only on one or few controlled links at most. In contrast, we develop WACA as a monitoring system in order to measure multiple BSS's in various operational settings, which allows tackling how a channel bonding BSS interacts with surrounding BSS's in a broad set of scenarios.

Simulation studies and analytical models have also been employed to study channel bonding, e.g., early simulation studies demonstrated throughput gains of channel bonding compared to single-channel transmission [23, 24]. Likewise, analytical models have been proposed to study channel bonding, especially through Markov chains [25, 26]. Analytical models for unsaturated traffic have been also proposed [27, 28]. As well, different channel bonding policies were introduced and modeled for spatially distributed scenarios [29]. However, such work does not have an experimental validation as presented here. In this regard, WACA datasets can be used for validation of the aforementioned methods under real-world channel occupancies in future work.

3 Wi-Fi All-Channel Analyzer

3.1 Overview

Our objective is to simultaneously capture activity on all Wi-Fi channels, i.e., all 24 basic (non aggregated) 20 MHz channels in the 5 GHz band that permit channel bonding. In principle, this could be achieved with an off-the-shelf spectrum analyzer. However, most spectrum analyzers are not capable of dealing with the required bandwidth of this objective, i.e., they cannot simultaneously measure the entire Wi-Fi 5-GHz band: 645 MHz ranging from channel 36 to 161 (i.e., from 5170 to 5815 MHz). Moreover, wide-band spectrum analyzers that can cover this bandwidth lack resolution to analyze basic channels within the band.

Parameter	Value
Active RFs (channels 36 to 161), $\mathcal{R}$	$\{1, 2, ..., 24\}$
Iteration's measurement duration, T_{m}	1 s
Iteration's processing duration, T_{proc}	~ 9 s
Duration of a complete iteration, T_{it}	~ 10 s
Original no. of power samples per channel per second, n_{s}^*	10^7
Downsampled no. of power samples per channel per second, n_{s}	10^5
Time between consecutive samples, T_{s}	10 µs
Number of iterations, N_{it}	125-59500

Table 1: WARPLab setup.

Likewise, one could envision a system comprising 24 off-the shelf Wi-Fi cards as sniffers, one per basic channel. Unfortunately, such a system would be quite unwieldy and would introduce a challenge of ensuring synchronicity among wireless cards: restricting the delay between channel measurements to the order of nano/microseconds is unfeasible due to the hardware interrupt latency and jitter from the different peripherals [30, 31].

Thus, we design WACA to simultaneously measure power (and I/Q signals if required) on all 5-GHz basic channels. Key benefits of WACA include the simplicity of experimental procedures (from deployment to post-processing), a dedicated RF chain per channel (covering the whole band and easing hardware failure detection), and the ease of adaptation/configuration empowered by the WARPLAb framework [32].

3.2 Building Blocks

The key building blocks of WACA are *i*) six WARP v3 programmable wireless SDRs [33], *ii*) six FMC-RF-2X245 dual-radio FMC daughterboards,[3] *iii*) 24 5-GHz antennas (one per RF chain), and *iv*) one Ethernet switch to enable communication between the WARPLab host (e.g., PC) and the WARP boards. The preeminent building block is WARP, a scalable and extensible programmable wireless platform to prototype advanced wireless networks. The FMC-RF-2X245 module dual-radio FPGA Mezzanine Card (FMC) daughterboard extends the capability of WARP v3 boards from 2 to 4 RF chains. Therefore, by combining 2 stacks of 3 WARP boards each with their corresponding FMC-RF-2X245 daughterboards, we realize 24 RF chains (with one 5-GHz antenna each) enabling us to assign a single RF chain per basic channel. Finally, the Ethernet switch enables the communication from the WARP nodes to the WARPLab host. Figure 1a shows the assignment of the RF chains to each basic channel allowed for bonding and Figure 1b depicts the physical realization of WACA.

3.3 Measurement Methodology

An iterative procedure is followed for collecting power samples. Namely, in each iteration, WACA first simultaneously measures the power at each basic channel during T_{m} and then takes T_{proc} to process and forward the measurements to the WARPLab host. Both tasks are sequentially performed until the end of the measurement campaign. Table 1 shows the main WARPLab parameters used for measurements. Notice that all parameters are fixed except N_{it}, used for determining the duration of the measurement campaign.

WARP boards install the MAX2829 transceiver, which has a fixed 10 Msps received signal strength sampling rate. Accordingly, since the measurement duration in an iteration is $T_{\text{m}} = 1$ second, the number of consecutive samples captured per basic channel per iteration is $n_{\text{s}}^* = 10^7$. Then, in each iteration, we store $|\mathcal{R}| \times n_{\text{s}}^* = 24 \times 10^7$ samples. Nonetheless, to decrease the amount of required memory, we downsample the gathered samples in each iteration by a $100\times$ factor, thus reducing the data size per iteration from 60 MB to 600 KB. Essentially, while the transceiver measures 1 power sample every 100 ns by default, we keep just 1 sample every $T_s = 10$ µs. Notice that the resulting time scale is also suitable given Wi-Fi timings. Indeed, the short interframe space (SIFS) is the smallest interframe space and takes 16 µs ($> T_s$).

[3]FMC-RF-2X245 datasheet: `https://mangocomm.com/products/modules/fmc-rf-2x245`, retrieved January 30, 2020.

As for the duration of processing and forwarding (period in which no data is collected), T_{proc} entails a significant yet unavoidable delay overhead with respect to the total duration of an iteration $T_{\text{it}} = T_{\text{m}} + T_{\text{proc}}$. Specifically, for the host PC used in all the scenarios in the dataset (Intel Core i5-4300U CPU @ 1.9 GHz x 4 and 7.7 GiB memory) and the parameters listed in Table 1, $T_{\text{proc}} \approx 9$ s for $T_{\text{m}} = 1$ s. Once initiated, WACA operates by itself, and no human intervention is required.

3.4 Validation

Before deploying WACA for the measurement campaigns, we perform an extensive set of in-lab controlled experiments for validation, comprising over 6×10^8 power samples collected in WACA cross validated with controlled and known transmissions from commercial APs. We conduct the validation of all the boards by first measuring the power perceived against distance and transmission bandwidth. Then, we explore the spectrum behavior when setting up different off-the-shelf channel bonding configurations. The measurements were gathered in an empty and large event room (about 300 m^2). Figure 2 shows the complete deployment. In particular, we deploy WACA including the PC hosting the WARPLab framework (lower part of the figure), 3 PCs receiving `iperf` traffic (right part of the figure), and 3 PCs sending `iperf` traffic with the corresponding 3 APs enabling the `iperf` connections (left part of the figure). The AP models we use are Asus RT-AC87U (AS), TP-Link AC1750 (TP), and Linksys WRT 3200 ACM (LS). Remarkably, all of these APs are only capable of performing static channel bonding.

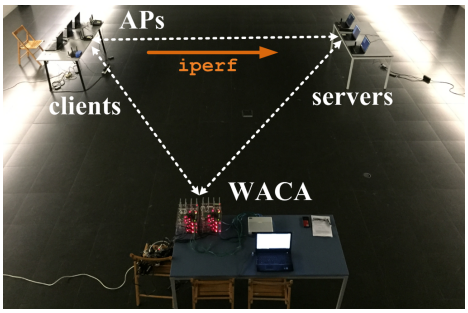


Figure 2: Validation testbed at the main events room of Universitat Pompeu Fabra's Communication campus.

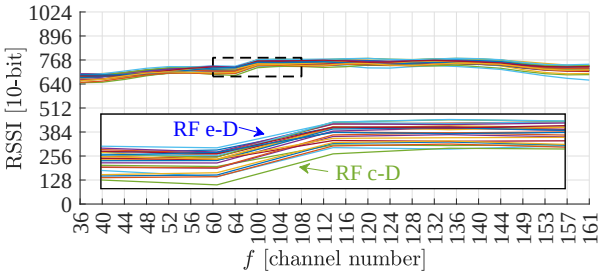
3.4.1 RSSI vs. central frequency

Ideally, we would like each of the 24 RFs to perceive exactly the same RSSI value when receiving the same signal modulated at a central frequency f. Besides, we would also like each of the RFs to perceive exactly the same RSSI value at every central frequency f when the same signal is modulated at each f. In other words, the RSSI vs. f should be the same flat curve for all the RFs. In order to validate such premises, we use a WARP node acting as a transmitter and sequentially connect the transmitter RF (which keeps generating the same 20-MHz signal) to each of the 24 RFs of WACA for every channel from 36 to 161 via an RF-SMA connector. As shown in Figure 3, a similar RSSI is perceived at every RF. Besides, a similar flat frequency response is achieved for every RF.

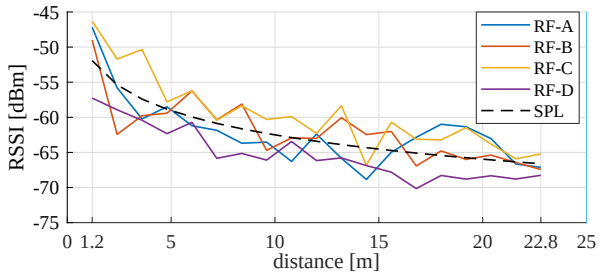
3.4.2 RSSI vs. distance

We measure the RSSI in dBm[4] at each basic channel at different distances d from the AP to WACA ranging from 1.2 to 22.8 meters. We took 10^5 samples per point. For redundancy, we use 4 RFs to measure the same channel $p = 124$.[5] As expected, the RSSI tends to decrease with the distance due to the path loss effect as shown in Figure 3b. For the sake of representation, we also plot the simplified path loss (SPL) of the mean power received at each point: $\text{RSSI}(d) = \text{RSSI}(d_0) - \alpha 10 \log_{10}(d)$, with $d = 1$ m. Despite the experiments were conducted in a large, empty room without furniture, we can see the multipath effects in the sparse RSSI values perceived by each RF due to the reflection of the walls. Such effect is anticipated since the minimum separation between RFs is 6 cm.

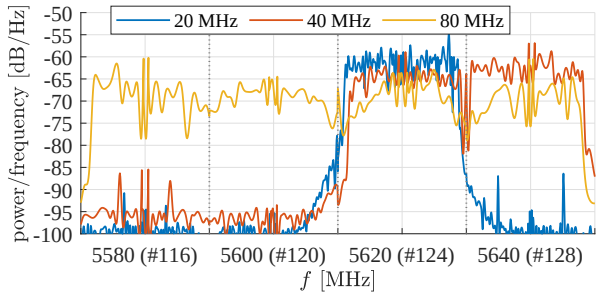
[4]We follow the MAX2829 transceiver data sheet to convert from 10-bit RSSI values to dBm values as done in [34].
[5]We selected this channel since negligible activity was detected.



(a) RSSI vs. central frequency. Each curve represents an RF.



(b) RSSI vs. distance. The curve named SPL stands for the simplified path loss model with $\alpha = 1.15$.



(c) Transmission bandwidth effect on power.

Figure 3: Validation metrics.

3.4.3 RSSI vs. bandwidth

This experiment aims at measuring the RSSI perceived at a fixed distance (3.6 m) in four basic channels at the same time for 20, 40, and 80 MHz transmissions. The general procedure followed in the experiment can be summarized in 4 steps: *i)* the AS AP is set up with bandwidth $b \in \{20, 40, 80\}$ MHz and primary channel $p = 124$; *ii)* an `iperf` client is placed close to the AP and associates to it; *iii*) an *iperf* server is located at distance $d = 3.6$ m from the AP and also associates to it; finally, *iv*) the `iperf` communication is triggered and WACA captures the resulting RSSI at channels 116, 120, 124, and 128, thus covering 80 MHz. Notice that it is enough to use just one WARP board with the corresponding daughterboard to cover the 4 basic channels of interest. Theoretically, when doubling the bandwidth, the transmission power gets reduced by a half (3 dB). We corroborate this fact by looking at the similar reduction factor in the power detected per Hz in Figure 3c.

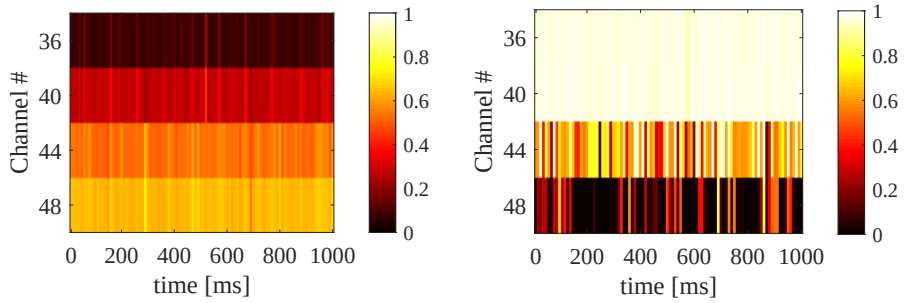
3.4.4 Controlled testbed evaluation

We now measure the activity of 3 overlapping BSSs' with static channel bonding capabilities under different traffic regimes. As shown in Figure 2, we separated the APs from their corresponding `iperf` servers by the same distance $d = 4.8$ m. The WACA platform was placed equidistantly from the central AP and STA also at $d = 4.8$ m.

We analyze two particular setups where none and all the BSSs' saturate. The load of every BSS is 20 and 150 Mbps in the first and second setup, respectively. Table 2 collects the setups' details. Essentially, in the unsaturated scenario, all the BSSs' share the same primary channel (48) but are set with different bonding capabilities. The second setup assigns different primary channels and bandwidth capabilities to each BSS. Note that we changed the primary channels from the previous

Regime	AP	36	40	44	48	Thr. [Mbps]
Unsaturated $\ell = 20$ Mbps	AS				**p**	20
	TP				**p**	20
	LS				**p**	20
Saturated $\ell = 150$ Mbps	AS		**p**			100
	TP	**p**				59
	LS				**p**	16

Table 2: Setup of controlled experiments. Letter p indicates the primary channel whereas colors represent the allocated bandwidth to each AP.



(a) Unsaturated. (b) Saturated.

Figure 4: Spectogram evolution of the controlled scenarios. Slots represent the 10-ms mean occupancy.

experiments due to AP hardware restrictions. Nonetheless, we also confirmed that the external interference at the new band of interest was negligible.

WACA collected 1 second of consecutive RSSI samples during the execution of each setup to measure the spectrum occupancy. Figure 4 shows the occupancy evolution at each basic channel for the unsaturated and saturated setups. We confirm that the former setup does not reach full occupancy in any channel, whereas it is the contrary for the later. Indeed, we observe that the load ($\ell = 150$ Mbps) is high enough to make channels 36 and 40 occupied almost all the time (i.e., occupancy at the first 40-MHz band is always close to 1) while not being able to successfully deliver all the load as indicated by the throughput values in Table 2. This has an important consequence for the LS AP. Essentially, since static channel bonding is applied, the LS AP does not benefit from having its primary channel (48) far from the rest of APs. Consequently, its throughput performance is drastically reduced given that AS and TP introduce inadmissible activity in the first 40 MHz band.

4 Measurement Campaigns

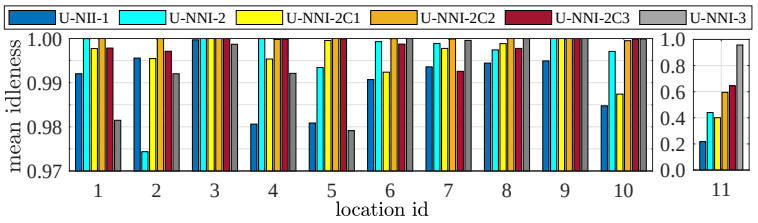
Using WACA, we perform extensive measurement campaigns covering two continents, dense urban areas, and multiple hours of samples in places of interest such as university campuses, apartment buildings, shopping malls, hotels and the Futbol Club Barcelona (a.k.a Barça) stadium (Camp Nou), one of the largest sports stadiums in the world. The measurements were taken from February to August 2019 in Houston, TX, USA, and Barcelona, Spain. The shortest experiment took 20 minutes and the longest covered more than 1 week. The list of locations is shown in Table 3.

Here, we overview the complete dataset: there are 153,033 iterations of 1 second accounting for 42 hours, 30 minutes, and 3 seconds of actual measurements in the 24 20-MHz channels of the 5 GHz WiFi band. We first assess the entire activity record of each location at band $\mathcal{B}$, where $\mathcal{B}$ is a predefined set of contiguous channels. For instance, U-NII-1 band is defined as $\mathcal{B} = \{1, 2, 3, 4\}$, corresponding to basic channels 36, 40, 44, and 48 (see Figure 1a).[6] Figure 5a shows the normalized mean idleness of each band, i.e., the mean number of samples that were found idle in each channel of band $\mathcal{B}$. We observe that the spectrum is idle most of the time in all scenarios except the stadium, indicating that the 5 GHz band is still profoundly underutilized even in densely populated areas.

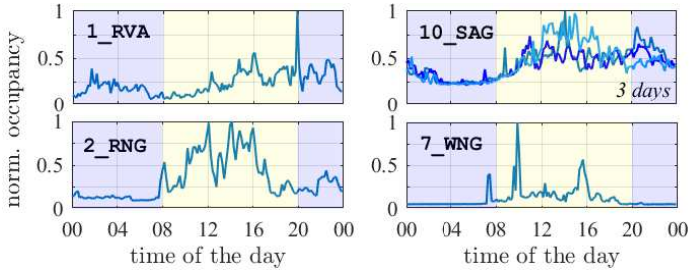
[6]The rest of bands are sequentially composed of groups of four consecutive 20-MHz channels. So, the next band is U-NNI-2 with channels $\{5, 6, 7, 8\}$ (basic channels 52, 56, 60, and 64), and the last one is U-NNI-3 with channels $\{21, 22, 23, 24\}$ (basic channels 149, 153, 157, 161).

Id	Location	Type	Duration
1_RVA	Rice Village Apart., HOU	Apartment	1 day
2_RNG	RNG lab at Rice, HOU	Campus	1 day
3_TFA	Technology for All, HOU	Com. center	1 day
4_FLO	Flo Paris, HOU	Cafe	1 hour
5_VIL	Rice Village, HOU	Shopping mall	20 min
6_FEL	La Sagrera, BCN	Apartment	1 week
7_WNO	WN group office, BCN	Campus	1 day
8_22@	22@ area, BCN	Office area	1 day
9_GAL	Hotel Gallery, BCN	Hotel	1 day
10_SAG	Sagrada Familia, BCN	Apartment	4 days
11_FCB	Camp Nou, BCN	Stadium	5 hours

Table 3: Measurement locations.



(a) Mean idleness per band.



(b) Temporal evolution of the normalized occupation for apartments (upper) and universities (lower) highlighting day (yellow) and night (blue).

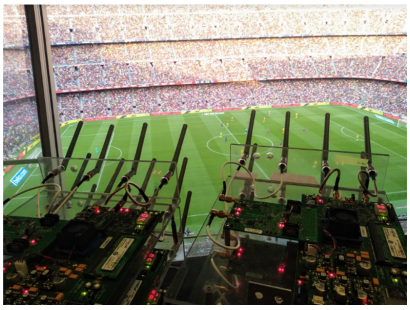
Figure 5: General spectrum occupancy trends.

Figure 5b shows the daily temporal evolution of 4 example locations (2 apartments in the upper subplots, and 2 university campuses in the lower subplots). For the sake of representation, we plot the normalized occupancy of the whole band averaged in periods of 10 minutes. Concretely, we normalize with respect to the highest 10-minutes average occupancy encountered in each location. We clearly observe higher activity at working hours in the campus locations and a much less variable pattern in the apartment locations. In any case, from the low spectrum utilization observed in Figure 5a, significant opportunities for channel bonding can be expected.

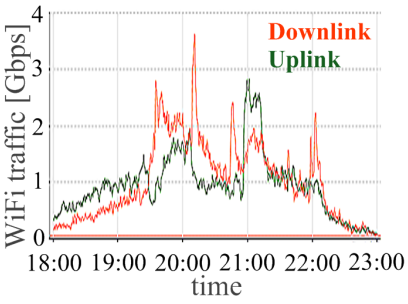
Figure 6a shows a photograph of the WACA setup in the Futbol Club Barcelona's stadium. We deployed WACA in the press box of the stadium during a football game with over 98,000 spectators present. Measurements were taken on August 4, 2019, from 17:24 to 22:30 accounting for a total duration of 5 hours and 6 minutes. On that date, the Joan Gamper trophy was held, which pitted the local club (Futbol Club Barcelona) against the visiting club (Arsenal Football Club). Free Wi-Fi was provided to the audience through thousands of APs primarily installed beneath the seats. We also obtained data from the stadium's network management team which indicated that up to 12,000 Wi-Fi clients were simultaneously connected. Downlink and uplink traffic is depicted in Figure 6b[7] and the spectogram in Figure 6c. Each slot in the spectogram represents the occupancy of each channel averaged in 1-second periods. We observe that most channels were highly occupied during the measurements. Moreover, we can observe the users' behavior induced nonstationarity of the traces as the match progressed. Namely, while activity is always high, there is a notable reduction

[7]Aggregated downlink and uplink traffic was provided by Futbol Club Barcelona's IT management.

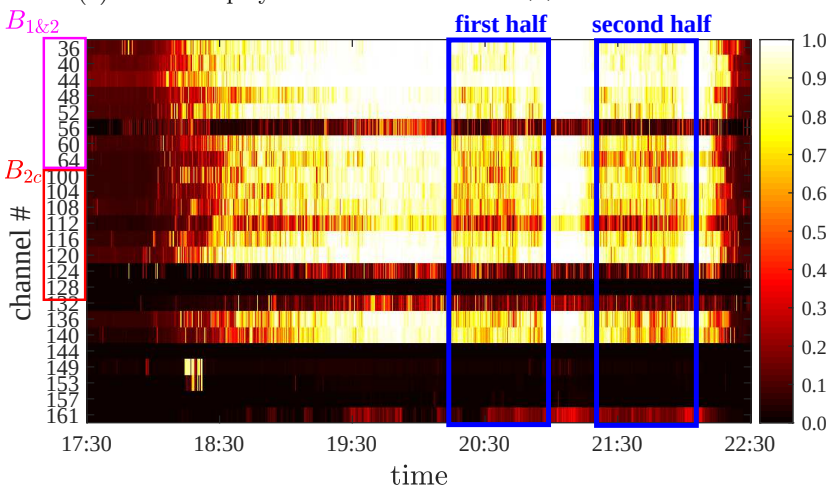
during play (first and second half) compared to activity before, between halfs, and right after the game time. We also observe that the majority of the channels are crowded most of the time with periods reaching mean band occupancy values rising up to 99%. In fact, 22% of the periods are above 80% occupancy.



(a) WACA deployment.



(b) Traffic evolution.



(c) Spectrum occupancy evolution.

Figure 6: Wi-Fi activity at the Camp Nou stadium: a) WACA deployment in the press box, b) downlink and uplink Wi-Fi traffic evolution (source: stadium's IT management), and c) spectrum occupancy evolution.

5 Evaluation of bonding gains

In this section, we study the performance of channel bonding against the band occupancy. Notice that, due to the disparate nature of the stadium scenario (recall the contrasting mean idleness in Figure 5a), we focus our study on the non-stadium measurement campaigns to provide insights on a more homogeneous dataset. The analysis of the stadium campaign will be covered in future work.

5.1 Evaluation Methodology

5.1.1 Overview

Our objective is to study the throughput performance that a fully backlogged channel-bonding BSS, w, would obtain if it encountered the channels recorded in the measurement campaigns described above. An schematic of the system model is illustrated in Figure 7. BSS w consists of an access point (AP) and a one or multiple stations (STAs) sufficiently close to assume that they perceive the same RSSI measurements.

The throughput is a function of a number of factors such as which primary channel the bonding BSS selects, which channel bonding policy it employs, and the spectrum occupancy perceived while operating. So, here we describe the trace-driven framework we use. The RSSI measurements captured by WACA at scenario s are represented by a 2-dimensional matrix Y^s of size $(N^s_{\text{it}} \times n_{\text{s}}) \times |\mathcal{R}|$, where N^s_{it} is the number of iterations of scenario s, and any element $y^s_{t,c}$ represents the power value at temporal sample t in basic channel c. From Y^s, we generate a binary matrix X^s of same size

through an occupancy indicator function, where any element $x^s_{t,c}$ represents whether channel c was occupied at temporal sample t (1) or not (0). Formally, $x^s_{t,c} = (y^s_{t,c} > \text{CCA} : 1, 0), \forall t, c$, where the clear channel assessment (CCA) is set to -83.5 dBm (or 150 10-bit RSSI units), corresponding to the common CCA threshold -82 dBm plus a safety margin of -1.5 dBm. While IEEE 802.11ac/11ax introduce different CCA levels for the primary and secondary channels, in this work, we consider a more restrictive approach by assuming the same threshold in order to fairly compare different channel bonding policies. The mean occupancy at a certain band $\mathcal{B}$ in scenario s is simply defined as

$$\bar{o}^s_{\mathcal{B}} = \frac{\sum_t \sum_{c \in \mathcal{B}} x^s_{t,c}}{N^s_{\text{it}} n_{\text{s}} |\mathcal{B}|}. \tag{1}$$

To provide meaningful experiments, we separately consider two 160-MHz bands composed of 8 basic channels: the U-NII-1&2 and part of the U-NII-2c sub-bands, $\mathcal{B}_{1\&2} = \{1, 2, 3, ..., 8\}$ and $\mathcal{B}_{2\text{c}} = \{9, 10, 11, ..., 16\}$, respectively. These sub-bands cover from channel 36 to 64 and from channel 100 to 128, respectively (see Figure 1a). Notice that these are the only sub-bands that allow to perform 160-MHz transmissions in the IEEE 802.11ac/ax channelization. Moreover, we focus on epochs (or periods) of duration $T_{\text{per}} = 100$ ms for which the mean occupancy at such sub-bands is at least 5%, i.e., $\bar{o}_{\mathcal{B}} \geq 0.05$, where $\mathcal{B} \in \{\mathcal{B}_{1\&2}, \mathcal{B}_{2\text{c}}\}$.

5.1.2 State Machine

We develop a discrete state machine that characterizes how the channel-bonding BSS responds to each power sample (or temporal sample) t according to the current state $S(t)$, and channel-bonding policy π, following the 802.11 standard. The set of possible states is $\mathcal{S} = \{Busy, DIFS, BO, TX/RX\}$. State *Busy* indicates that the primary channel is busy, *DIFS* represents the period before initiating the backoff process, the backoff counter is decreased during *BO* state, and *TX/RX* represents the actual frame transmission-reception (including the control frames RTS, CTS, and ACK, the DATA frame, and the SIFS periods in between). We represent the channel-bonding BSS w as an AP and one or multiple clients that would perceive exactly the same spectrum activity as WACA captured in the measurement campaigns and must contend accordingly.

The set of basic channels selected for transmitting a frame depends both on the spectrum occupancy and on the selected channel-bonding policy π. Figure 8 illustrates an example of the transitions between states. Empty slots have a duration $T_{\text{slot}} = 10\,\mu$s and we consider $T_{\text{slot}} = T_{\text{s}} = 10\,\mu$s rather than $9\,\mu$s (802.11's default value) to align the duration of an idle backoff slot with the sample duration. Hence, whenever the channel-bonding BSS is in the backoff process at state *BO*, every idle sample at the primary channel p results in a backoff counter decrease of one empty slot. We use Wi-Fi parameters according to IEEE 802.11ax [3] and assume 256-QAM modulation coding scheme regardless of the transmission bandwidth. The setup parameters are listed in Table 4. After running the state-machine through all the temporal samples in the epoch, we compute the throughput Γ as the number of data packets sent divided by the duration of the epoch T_{per}.

5.1.3 Channel Bonding Policies and Response

A channel-bonding policy π selects the set of basic channels to aggregate at the end of the backoff provided that the primary channel is available. Namely, *contiguous* channel bonding can select a set of channels both above and below the primary channel, provided they are consecutive. In contrast,

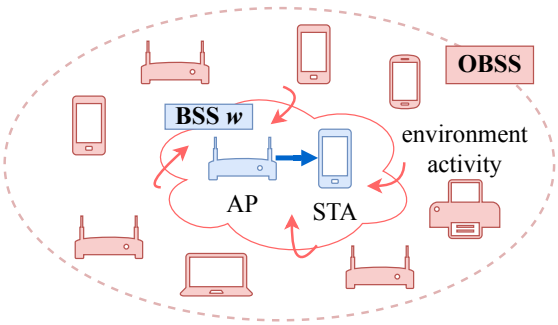


Figure 7: Diagram of the system model. The fully backlogged channel-bonding BSS, w, operates under instances of the same environment – or overlapping BSS (OBSS) – that WACA perceived at the measurement campaigns.

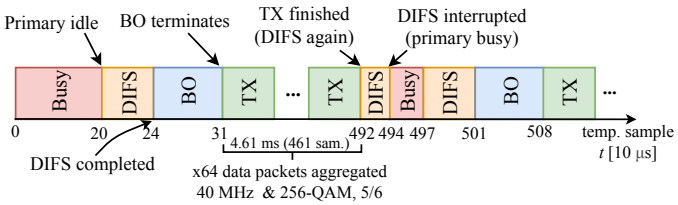


Figure 8: Example of the transitions between states.

Param.	Description	Value
CCA	CCA threshold	-83.5 dBm
MCS	MCS index	9 (256-QAM 5/6)
b	*basic* channel bandwidth	20 MHz
L_{d}	Length of a data packet	12000 bit
$\mathcal{B}$	Allocated set of *basic* channels	$\mathcal{B} \subseteq \mathcal{R}$
p	Primary channel	$p \in \mathcal{B}$
N_{a}	Max. no. of agg. packets per frame	64
T_{e}	Duration of an empty slot	10 µs
T_{SIFS}	SIFS duration	20 µs
T_{DIFS}	DIFS duration	30 µs
T_{PIFS}	PIFS duration	30 µs
T_{RTS}	RTS duration	50 µs
T_{CTS}	CTS duration	40 µs
T_{BACK}	Block ACK duration	50 µs
TXOP	Max. duration of a TXOP	5 ms
$\mathrm{CW}_{\mathrm{min}}$	Min. contention window	16
m	No. of backoff stages	5

Table 4: Trace-driven setup.

non-contiguous channel bonding can combine all available channels at the time the primary channel becomes available. We compare both policies in §5.2.

How will other BSS's respond to the channel bonding BSS? We consider that they will defer their transmissions. Namely, the channel bonding BSS needs the channels to be available only when its countdown timer expires. If the bonding BSS does transmit but the trace indicates that a channel would have been occupied at some point during the transmission, we consider that such other BSS's will sense the bonding BSS and defer.

5.2 Contiguous vs. Non-Contiguous

For each transmission, non-contiguous channel bonding can utilize additional channels as compared to contiguous, by "skipping over" the busy channels to find the next unused one. Here, we explore the gains of this flexibility as well as (in rare cases) the losses by comparing the throughput of contiguous and non-contiguous channel bonding in three load regimes: low ($\bar{o}_{\mathcal{B}} \leq 0.1$), medium ($0.1 < \bar{o}_{\mathcal{B}} \leq 0.2$) and high ($\bar{o}_{\mathcal{B}} > 0.2$), respectively.

Figure 9 shows the the throughput ratio of contiguous to non-contiguous channel bonding $\Gamma_{\mathrm{CO}}(p)/\Gamma_{\mathrm{NC}}(p)$, where $\Gamma_{\pi}(p)$ is the throughput achieved by policy π when selecting primary p in a given period. We plot the ratio for all possible primaries in $\mathcal{B}_{1\&2}$ and $\mathcal{B}_{2\mathrm{c}}$.

The data reveals two remarkable phenomenon. First, contiguous outperforms non-contiguous in 2.5% of the cases (albeit with a modest throughput difference of 1.9%). But since non-contiguous is more flexible, how can it ever do worse? The answer is that the two policies result in different instants for transmission attempts. The contiguous policy occasionally (and quite randomly since the traces are the same) ends up with more favorable attempt instants. Nonetheless, in most cases, non-contiguous obtains higher throughput. For example, in many periods, at least one 20-MHz channel is idle during the whole period, which will always yield a gain for non-contiguous, but only sometimes yields a gain for contiguous bonding. In some cases, the difference can be quite high (e.g., a ratio of approximately 0.2 observed in low load). The origins of such extreme cases are the selection of the primary channel. Second, the bar chart inside Figure 9 reveals that both contiguous and non-contiguous channel bonding perform quite close on average for all occupancy regimes (low, medium, and high), and especially for the latter, as high load results in far fewer

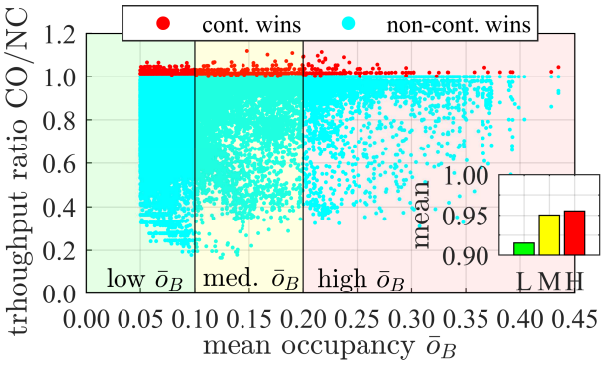


Figure 9: Throughput ratio of contiguous vs. non-contiguous channel bonding. The bar chart inset depicts the mean aggregated ratio for low (L), medium (M), and high (H) occupancy regimes.

bonding opportunities overall.

Finding: Non-contiguous almost always outperforms contiguous channel bonding and their throughput differences are occasionally over a factor of 5. Nonetheless, their average throughputs are quite similar, which may ultimately favor contiguous channel bonding, since it is simpler to implement.

6 Conclusion

In this paper, we introduce WACA, an all-channel Wi-Fi spectrum analyzer for simultaneous measurement of all 24 20 MHz channels that allow channel bonding at 5 GHz. We present extensive measurement campaigns covering two continents, diverse areas, and many hours of signal strength samples. The gathered dataset is a unique asset to find insights otherwise not possible to study. We leave the analysis of key factors like spectral correlation or bandwidth prevention as future work.

Acknowledgements

We appreciate the generosity of Futbol Club Barcelona for letting us take measurements in the Camp Nou stadium. The authors would also like to thank Chia-Yi Yeh, Albert Bel, Álvaro López, and Marc Carrascosa for their contribution in taking the measurements. The work from S. Barrachina-Muñoz and B. Bellalta has been partially supported by CISCO (CG#890107-SVCF), Maria de Maeztu Units of Excellence Programme (MDM-2015-0502), WINDMAL PGC2018-099959-B-I00 (MCIU/AEI/FEDER,UE), and SGR-2017-1188. The research of E. Knightly was supported by Cisco, Intel, and by NSF grants CNS-1955075 and CNS-1824529.

References

[1] IEEE 802.11n enhancements for Higher Throughput, 2009.

[2] IEEE 802.11ac Very High Throughput., 2013.

[3] IEEE P802.11ax/D4.0 High Efficiency WLAN, 2019.

[4] IEEE 802.11be Extremely High Throughput, 802.11-18/1231r6, 2019.

[5] Michael Rademacher, Karl Jonas, and Mathias Kretschmer. Quantifying the spectrum occupancy in an outdoor 5 GHz WiFi network with directional antennas. In *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6. IEEE, 2018.

[6] Tanim Taher, Ryan Attard, Ali Riaz, Dennis Roberson, Jesse Taylor, Kenneth Zdunek, Juhani Hallio, Reijo Ekman, Jarkko Paavola, Jaakko Suutala, et al. Global spectrum observatory network setup and initial findings. In *2014 9th International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM)*, pages 79–88. IEEE, 2014.

[7] Sriram Subramaniam, Hector Reyes, and Naima Kaabouch. Spectrum occupancy measurement: An autocorrelation based scanning technique using USRP. In *2015 IEEE 16th Annual Wireless and Microwave Technology Conference (WAMICON)*, pages 1–5. IEEE, 2015.

[8] Matthias Wellens, Janne RiihijäRvi, and Petri MäHöNen. Empirical time and frequency domain models of spectrum use. *Physical Communication*, 2(1-2):10–32, 2009.

[9] Vinod Kone, Lei Yang, Xue Yang, Ben Y Zhao, and Haitao Zheng. The effectiveness of opportunistic spectrum access: A measurement study. *IEEE/ACM Transactions on Networking*, 20(6):2005–2016, 2012.

[10] Simon W Day and Adrian Wagstaff. Activity in the WiFi bands - An Objective Approach to Estimation. In *ARMMS Conference*. Mass, 2014.

[11] Fatima Salahdine and Hassan El Ghazi. A real time spectrum scanning technique based on compressive sensing for cognitive radio networks. In *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*, pages 506–511. IEEE, 2017.

[12] Sanjit Biswas, John Bicket, Edmund Wong, Raluca Musaloiu-e, Apurv Bhartia, and Dan Aguayo. Large-scale measurements of wireless network behavior. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, pages 153–165, 2015.

[13] M. López-Benítez and F. Casadevall. Modeling and simulation of time-correlation properties of spectrum use in cognitive radio. In *2011 6th International ICST Conference on Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM)*, pages 326–330, 2011.

[14] Vinod Kone, Lei Yang, Xue Yang, Ben Y Zhao, and Haitao Zheng. On the feasibility of effective opportunistic spectrum access. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, pages 151–164, 2010.

[15] Marko Höyhtyä, Aarne Mämmelä, Marina Eskola, Marja Matinmikko, Juha Kalliovaara, Jaakko Ojaniemi, Jaakko Suutala, Reijo Ekman, Roger Bacchus, and Dennis Roberson. Spectrum occupancy measurements: A survey and use of interference maps. *IEEE Communications Surveys & Tutorials*, 18(4):2386–2414, 2016.

[16] Salim A Hanna and John Sydor. Distributed sensing of spectrum occupancy and interference in outdoor 2.4 GHz Wi-Fi networks. In *2012 IEEE Global Communications Conference (GLOBECOM)*, pages 1453–1459. IEEE, 2012.

[17] Salim A Hanna. A 3-state hypothesis test model for cognitive radio systems. In *2014 IEEE International Symposium on Dynamic Spectrum Access Networks (DYSPAN)*, pages 291–302. IEEE, 2014.

[18] Sixing Yin, Dawei Chen, Qian Zhang, Mingyan Liu, and Shufang Li. Mining spectrum usage data: a large-scale spectrum measurement study. *IEEE Transactions on Mobile Computing*, 11(6):1033–1046, 2012.

[19] Lara Deek, Eduard Garcia-Villegas, Elizabeth Belding, Sung-Ju Lee, and Kevin Almeroth. The impact of channel bonding on 802.11n network management. In *Proceedings of the Seventh COnference on emerging Networking EXperiments and Technologies*, pages 1–12, 2011.

[20] Lara Deek, Eduard Garcia-Villegas, Elizabeth Belding, Sung-Ju Lee, and Kevin Almeroth. Intelligent channel bonding in 802.11n WLANs. *IEEE Transactions on Mobile Computing*, 13(6):1242–1255, 2014.

[21] Mustafa Y Arslan, Konstantinos Pelechrinis, Ioannis Broustis, Srikanth V Krishnamurthy, Sateesh Addepalli, and Konstantina Papagiannaki. Auto-configuration of 802.11n WLANs. In *Proceedings of the 6th International Conference*, pages 1–12, 2010.

[22] Mustafa Y Arslan, Konstantinos Pelechrinis, Ioannis Broustis, Shailendra Singh, Srikanth V Krishnamurthy, Sateesh Addepalli, and Konstantina Papagiannaki. ACORN: An auto-configuration framework for 802.11n WLANs. *IEEE/ACM Transactions on Networking (TON)*, 21(3):896–909, 2013.

[23] Minyoung Park. IEEE 802.11ac: Dynamic bandwidth channel access. In *Communications (ICC), 2011 IEEE International Conference on*, pages 1–5. IEEE, 2011.

[24] Michelle X Gong, Brian Hart, Liangfu Xia, and Roy Want. Channel bounding and mac protection mechanisms for 802.11ac. In *2011 IEEE Global Telecommunications Conference-GLOBECOM 2011*, pages 1–5. IEEE, 2011.

[25] Boris Bellalta, Alessandro Checco, Alessandro Zocca, and Jaume Barcelo. On the interactions between multiple overlapping wlans using channel bonding. *IEEE Transactions on Vehicular Technology*, 65(2):796–812, 2015.

[26] Mohand Yazid and Adlen Ksentini. Stochastic modeling of the static and dynamic multichannel access methods enabling 40/80/160 MHz channel bonding in the VHT WLANs. *IEEE Communications Letters*, 23(8):1437–1440, 2019.

[27] Mun-Suk Kim, Tanguy Ropitault, Sukyoung Lee, and Nada Golmie. A throughput study for channel bonding in ieee 802.11ac networks. *IEEE Communications Letters*, 21(12):2682–2685, 2017.

[28] Sergio Barrachina-Muñoz, Francesc Wilhelmi, and Boris Bellalta. To overlap or not to overlap: Enabling channel bonding in high-density WLANs. *Computer Networks*, 152:40–53, 2019.

[29] Sergio Barrachina-Muñoz, Francesc Wilhelmi, and Boris Bellalta. Dynamic channel bonding in spatially distributed high-density WLANs. *IEEE Transactions on Mobile Computing*, 19(4):821–835, 2020.

[30] Giorgio Buttazzo, Carmelo Di Franco, and Mauro Marinoni. Design and analysis of target-sensitive real-time systems. *Software: Practice and Experience*, 46(9):1181–1200, 2016.

[31] Leili Farzinvash and Mehdi Kargahi. A scheduling algorithm for execution-instant sensitive real-time systems. In *2009 15th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, pages 511–518. IEEE, 2009.

[32] Narendra Anand, Ehsan Aryafar, and Edward W Knightly. WARPlab: a flexible framework for rapid physical layer design. In *Proceedings of the 2010 ACM workshop on Wireless of the students, by the students, for the students*, pages 53–56. ACM, 2010.

[33] Rice University Wireless Open Access Research Platform. `http://warpproject.org`, last accessed January 30, 2020.

[34] Seowoo Jang, Kang G Shin, and Saewoong Bahk. Post-CCA and reinforcement learning based bandwidth adaptation in 802.11ac networks. *IEEE Transactions on Mobile Computing*, 17(2):419–432, 2018.

8.6

Wi-Fi Channel Bonding: an All-Channel System and Experimental Study from Urban Hotspots to a Sold-Out Stadium

Sergio Barrachina-Muñoz, Boris Bellalta, and Edward Knightly

Abstract

In this paper, we present WACA, the first system to simultaneously measure all 24 Wi-Fi channels that allow channel bonding at 5 GHz with microsecond scale granularity. With WACA, we perform a first-of-its-kind measurement study in areas including urban hotspots, residential neighborhoods, universities, and even a game in Futbol Club Barcelona's Camp Nou, a sold-out stadium with 98,000 fans and 12,000 simultaneous Wi-Fi connections. We study channel bonding in this environment, and our experimental findings reveal the underpinning factors controlling throughput gain, including channel bonding policy and spectrum occupancy statistics. We then show the significance of the gathered dataset for finding insights, which would not be possible otherwise, given that simple channel occupancy models severely underestimate the available gains. Likewise, we characterize the risks of channel bonding due to other BSS's, including their missed transmission opportunities and potential collisions due to imperfect sensing of bonded transmissions. We explore 802.11ax which imposes constraints on bonded channels to avoid fragmentation and defines different modes that can trade implementation complexity for throughput. Lastly, we show that the stadium, while seemingly too highly occupied for channel bonding gains, has transient gaps yielding impressive gains.

1 Introduction

Channel bonding is a key mechanism for increasing Wi-Fi data rates as the maximum data rate increases in proportion to the total channel bandwidth. In Wi-Fi, while the *basic* channel width remains 20 MHz, the maximum *bonded* channel width has increased from 40 MHz in 802.11n to 160 MHz in 802.11ac/ax [1], and 320 MHz in 802.11be [2]. During this time, the standard has evolved to not only support wider bandwidths, but also to enable more sophisticated channel bonding policies: in 802.11n, only "static" channel bonding was allowed in which a fixed group of pre-configured channels must always be bonded. Today, the standard enables a far richer set of capabilities including dynamic selection of channel width as well as bonding both contiguous and non-contiguous channels.

In this paper, we provide the first experimental study of channel bonding policies and performance factors with a high-resolution all-channel measurement study.[1] In particular, we make the following contributions. First, we develop WACA as the first system to simultaneously measure all 24 Wi-Fi channels at 5 GHz with a 10 μsec sampling rate. WACA employs multiple synchronized WARP software defined radios (SDRs) and has 24 antennas and 24 radio frequency (RF) chains to match the number of Wi-Fi channels. Our approach contrasts with prior work that uses one or several RF chains, thereby encountering *second* scale channel switching delays. Thus, no prior methods capture all channels at the channel-access μsec timescale.

Second, using WACA, we conduct extensive measurement campaigns covering two continents, dense urban areas, and places of interest such as university campuses, apartment buildings, shopping malls, and hotels. We also perform measurements in one of the largest sports stadiums in the world, Futbol Club Barcelona's Camp Nou. The shortest campaign took 20 minutes and the longest covered more than 1 week, and the total number of samples exceeds 10^{11}. In all cases, we record signal strength on all channels at 10 μsec sample rate, from which we infer the epochs for which each channel is occupied.[2][3] Notice that the traces compose an unprecedented dataset to find insights otherwise not possible. We highlight Wi-Fi as the main research application of the dataset but

[1] This paper is an extension of Wi-Fi All-Channel Analyzer [3].

[2] The WACA dataset v1 can be found at `https://www.upf.edu/web/wnrg/wn-datasets`.

[3] All of the source code of WACA is open, encouraging sharing of algorithms between contributors and providing the ability for people to improve on the work of others under the GNU General Public License v3.0. The repository can be found at `https://github.com/sergiobarra/WACA_WiFiAnalyzer`.

others can also be taken into account, e.g., cognitive radios or coexistence between Wi-Fi and LTE and/or Internet of Things (IoT) technologies, to mention a few examples.

Third, we evaluate the performance of a channel bonding basic service set (BSS) using the aforementioned high-resolution traces of channel activity. As the channel occupancies are highly dynamic, we group epochs according to their average utilization. While the stadium measurements have band occupancies as high as 99%, even dense urban scenarios with many competing BSS's yielded maximum occupancies near 45%. Beginning with the non stadium scenarios, we first experimentally study the foundational elements controlling the gains of channel bonding.

We find that the increased flexibility of non-contiguous channel bonding compared to contiguous can yield a throughput improvement of up to 5×. Unfortunately, the scenarios for these gains occur quite rarely yielding modest average gains of less than 10%. Moreover, selection of the primary channel is critical and improves average throughput by over 60%. This is because the primary channel must be available before bonding can occur, and for contiguous channel bonding, it defines the crucially important neighboring channels. We next study correlation among channel occupancies: while one may expect minimal correlation with different BSS's using different channels, we observed that some measured BSS's employ channel bonding thereby inducing correlation. We show that this correlation, observed over all traffic loads, aids further channel bonding. Namely, when the primary channel becomes available, it increases the likelihood that other channels are available too. We also show that simple channel occupancy models severely underestimate the available gains for channel bonding. Thus, modeling occupancy behavior by introducing inter-channel correlation remains an important avenue for future work.

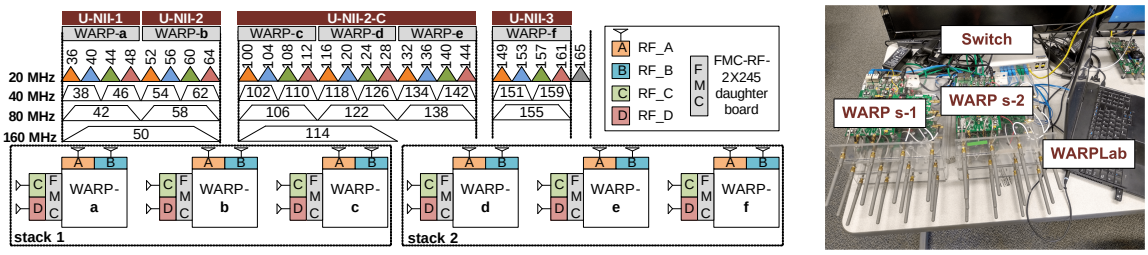
After understanding the sources and factors controlling the *gains* of channel bonding, we next turn to potential detrimental effects involving other in-range BSS's. First, when one BSS channel bonds, others must defer: is channel bonding simply a zero-sum-game transferring bandwidth from one BSS to another? We fortunately find that this is not the case, as the experiments indicate that the channel occupancy dynamics enable the channel bonder to gain at least 45 times what the other BSS's lose. Likewise, hidden nodes could also reduce the performance of channel bonding, as failed transmissions can occur if neighboring BSS's do not sense and defer to a bonded transmission. Our analysis shows that such a scenario of hidden other-BSS traffic would be quite costly, and would often result in single channel transmission outperforming channel bonding, even under relatively low channel occupancy.

Next, we consider the channelization restrictions imposed by IEEE 802.11ax along with its three channel bonding modes: static channel bonding, dynamic channel bonding, and preamble puncturing. Unfortunately, we find that static channel bonding gives up the majority of the available throughput gains. Namely, while simple to implement, this "all or nothing" strategy is too often waiting for all channels to become available which comes at a very high cost of missed opportunities to use more narrow bandwidths. Dynamic channel bonding and preamble puncturing are special cases of the above contiguous and non-contiguous strategies but are subject to further channelization constraints designed to simplify implementation and avoid channel fragmentation. We find that 802.11ax's channelization restrictions reduce throughput by up to 19% for dynamic channel bonding. In contrast, the increased flexibility of preamble puncturing limits the reduction to less than 6%.

Finally, we deploy WACA in a sold-out sports event with over 98,000 attendees. Partnering with the stadium's network management team, we find that up to 12,000 Wi-Fi clients were simultaneously connected via thousands of access points (APs). Despite a peak occupancy over all channels of 99%, which would yield negligible margin for channel-bonding gains, we observed significant epochs with opportunities. Namely, transient occupancy lulls of less than 30% can yield gains of up to 5.3×. Qualitatively, we find that the lulls in network traffic are inversely related to the match's lulls in action, e.g., half time yields a traffic boom. Nonetheless, shorter time-scale lulls in occupancy, which benefit channel bonding, can be observed at all times.

2 Related work

Prior work performed spectrum measurements for Wi-Fi traffic, e.g., [4–14]. Example objectives include creating interference maps [14], assessing interference behavior [15], surveying Wi-Fi usage [9], quantifying spectrum occupancy in outdoor testbeds [4, 5], designing efficient scanning methods [6, 10], modeling spectrum use [7], opportunistic spectrum access [8, 13], dynamic channel selection [16], and assessing real-world network behavior by examining data from thousands APs [11]. Unfortunately, no such prior work provided simultaneous measurement across the entire 5 GHz band, which we require for our channel bonding study. While some papers do provide multi-channel mea-



(a) IEEE 802.11ac/ax channelization and assignation per RF.

(b) Deployment schematic.

Figure 1: The WACA all WiFi channels spectrum analyzer.

surements, e.g., [4, 8, 13, 17], they do so via sequential scanning, thus taking on the order of seconds to change from one channel to the next, orders of magnitude beyond the transmission time scale for channel bonding. Namely, WACA measures all channels simultaneously using SDRs having a sampling rate of 10μsec. Moreover, prior work does not consider (for example) stadiums, the potential costs of other BSS's deferring or colliding, the impact of fine-grained channel correlation, nor a diverse set of policies.

Throughput gains of channel bonding have been demonstrated previously in testbeds. In particular, IEEE 802.11n static channel bonding has been shown to be affected not only by link signal quality, but also by the power and rates of neighboring links [18]. Likewise, intelligent channel bonding management was shown to benefit from identifying the signal strength of neighboring links and interference patterns [19]. High bandwidths were shown to be vulnerable to increased thermal noise or the power per Hertz reduction [20, 21]. Finally, an experimental study of IEEE 802.11ac channel bonding (both static and dynamic) showed that unplanned primary channel selection and bandwidth allocation may severely degrade the throughput of links operating at wider channels [22]. Nonetheless, existing experimental results have targeted only on one or few controlled links at most. In contrast, we develop WACA as a monitoring system in order to measure multiple BSS's in various operational settings, thus tackling how a channel bonding BSS interacts with surrounding BSS's in a broad set of channel bonding policies and scenarios.

Simulation studies and analytical models have also been employed to study channel bonding, e.g., early simulation studies demonstrated throughput gains of channel bonding compared to single-channel transmission [23, 24]. Likewise, analytical models have been proposed to study channel bonding, especially through Markov chains [25–29]. For instance, high-density deployments are evaluated in [26], showing exposure to unfairness in groups of overlapping BSS's. Analytical models for unsaturated traffic have been also proposed [28, 30]. As well, different channel bonding policies were introduced and modeled for spatially distributed scenarios [30, 31]. Lastly, primary channel selection was studied in [32–35], traffic demands in [36], and QoS constraints in [37]. However, such work does not have an experimental validation as presented here. In this regard, WACA datasets can be used for validation of the aforementioned methods under real-world occupancies in future work.

3 Wi-Fi All-Channel Analyzer

3.1 Overview

Our objective is to simultaneously capture activity on all Wi-Fi channels, i.e., all 24 basic (non aggregated) 20 MHz channels in the 5 GHz band that permit channel bonding. In principle, this could be achieved with an off-the-shelf spectrum analyzer. However, most spectrum analyzers are not capable of dealing with the required bandwidth of this objective, i.e., they cannot simultaneously measure the entire Wi-Fi 5-GHz band: 645 MHz ranging from channel 36 to 161 (i.e., from 5170 to 5815 MHz). Moreover, wide-band spectrum analyzers that can cover this bandwidth lack resolution to analyze basic channels within the band. Likewise, one could envision a system comprising 24 off-the shelf Wi-Fi cards as sniffers, one per basic channel. Unfortunately, such a system would be quite unwieldy and would introduce a challenge of ensuring synchronicity among wireless cards: restricting the delay between channel measurements to the order of nano/microseconds is unfeasible due to the hardware interrupt latency and jitter from the different peripherals [38, 39].

Thus, we design WACA to simultaneously measure power (and I/Q signals if required) on all 5-GHz basic channels. Key benefits of WACA include the simplicity of experimental procedures

(from deployment to post-processing), a dedicated RF chain per channel (covering the whole band and easing hardware failure detection), and the ease of configuration empowered by the WARPLAb framework [40].

3.2 Building Blocks

The key building blocks of WACA are (*i*) six WARP v3 programmable wireless SDRs [41], (*ii*) six FMC-RF-2X245 dual-radio FMC daughterboards,[4] (*iii*) 24 5-GHz antennas (one per RF chain), and (*iv*) one Ethernet switch to enable communication between the WARPLab host (e.g., PC) and the WARP boards. The preeminent building block is WARP, a scalable and extensible programmable wireless platform to prototype advanced wireless networks. The FMC-RF-2X245 module dual-radio FPGA Mezzanine Card (FMC) daughterboard extends the capability of WARP v3 boards from 2 to 4 RF chains. Therefore, by combining 2 stacks of 3 WARP boards each with their corresponding FMC-RF-2X245 daughterboards, we realize 24 RF chains (with one 5-GHz antenna each) enabling us to assign a single RF chain per basic channel. Finally, the Ethernet switch enables the communication from the WARP nodes to the WARPLab host. Fig. 1a shows the assignment of the RF chains to each basic channel allowed for bonding and Fig. 1b depicts the physical realization of WACA.

3.3 Measurement Methodology

An iterative procedure is followed for collecting power samples. Namely, in each iteration, WACA first simultaneously measures the power at each basic channel during T_{m} and then takes T_{proc} to process and forward the measurements to the WARPLab host. Both tasks are sequentially performed until the end of the measurement campaign. Table 1 shows the main WARPLab parameters used for measurements. Notice that all parameters are fixed except N_{it}, used for determining the duration of the measurement campaign.

Parameter	**Value**
Active RFs (channels 36 to 161), $\mathcal{R}$	$\{1, 2, ..., 24\}$
Iteration's measurement duration, T_{m}	1 s
Iteration's processing duration, T_{proc}	~ 9 s
Duration of a complete iteration, T_{it}	~ 10 s
Original no. of power samples per channel per second, n_{s}^*	10^7
Downsampled no. of power samples per channel per second, n_{s}	10^5
Time between consecutive samples, T_{s}	10 µs
Number of iterations, N_{it}	125-59500

Table 1: WARPLab setup.

WARP boards install the MAX2829 transceiver, which has a fixed 10 Msps received signal strength sampling rate. Accordingly, since the measurement duration in an iteration is $T_{\mathrm{m}} = 1$ second, the number of consecutive samples captured per basic channel per iteration is $n_{\mathrm{s}}^* = 10^7$. Then, in each iteration, we store $|\mathcal{R}| \times n_{\mathrm{s}}^* = 24 \times 10^7$ samples. Nonetheless, to decrease the amount of required memory, we downsample the gathered samples in each iteration by a $100\times$ factor, thus reducing the data size per iteration from 60 MB to 600 KB. Essentially, while the transceiver measures 1 power sample every 100 ns by default, we keep just 1 sample every $T_s = 10$ µs. Notice that the resulting time scale is also suitable given Wi-Fi timings. Indeed, the short interframe space (SIFS) is the smallest interframe space and takes 16 µs ($> T_s$).

As for the duration of processing and forwarding (period in which no data is collected), T_{proc} entails a significant yet unavoidable delay overhead with respect to the total duration of an iteration $T_{\mathrm{it}} = T_{\mathrm{m}} + T_{\mathrm{proc}}$. Specifically, for the host PC used in all the scenarios in the dataset (Intel Core i5-4300U CPU @ 1.9 GHz x 4 and 7.7 GiB memory) and the parameters listed in Table 1, $T_{\mathrm{proc}} \approx 9$ s for $T_{\mathrm{m}} = 1$ s. Once initiated, WACA operates by itself and no human intervention is required.

[4]FMC-RF-2X245 datasheet: `https://mangocomm.com/products/modules/fmc-rf-2x245`, retrieved January 30, 2020.

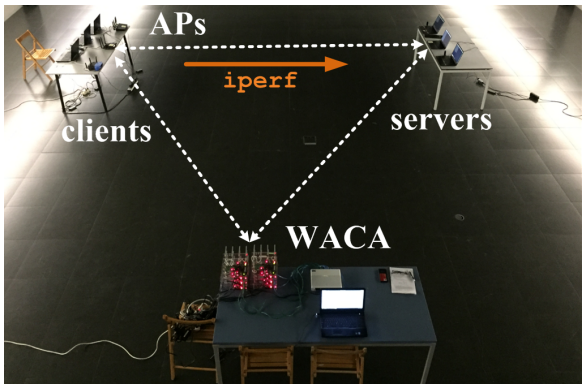


Figure 2: Validation testbed at the main events room of Universitat Pompeu Fabra's Communication campus.

3.4 Validation

Before deploying WACA for the measurement campaigns, we perform an extensive set of in-lab controlled experiments for validation, comprising over 6×10^8 power samples collected in WACA cross validated with controlled and known transmissions from commercial APs. We conduct the validation of all the boards by first measuring the power perceived against transmission bandwidth.[5] Then, we explore the spectrum behavior when setting up different off-the-shelf channel bonding configurations. The measurements were gathered in an empty and large event room (about 300 m^2). Figure 2 shows the complete deployment. In particular, we deploy WACA including the PC hosting the WARPLab framework (lower part of the figure), 3 PCs receiving `iperf` traffic (right part of the figure), and 3 PCs sending `iperf` traffic with the corresponding 3 APs enabling the `iperf` connections (left part of the figure). The AP models we use are Asus RT-AC87U (AS), TP-Link AC1750 (TP), and Linksys WRT 3200 ACM (LS). Remarkably, all of these APs are only capable of performing static channel bonding.

3.4.1 RSSI vs. bandwidth

We measure the RSSI in dBm[6] at a fixed distance (3.6 m) in four basic channels at the same time for 20, 40, and 80 MHz transmissions. The general procedure followed in the experiment can be summarized in 4 steps: *i)* the AS AP is set up with bandwidth $b \in \{20, 40, 80\}$ MHz and primary channel $p = 124$ since negligible activity was detected; *ii)* an `iperf` client is placed close to the AP and associates to it; *iii*) an *iperf* server is located at distance $d = 3.6$ m from the AP and also associates to it; finally, *iv*) the `iperf` communication is triggered and WACA captures the resulting RSSI at channels 116, 120, 124, and 128, thus covering 80 MHz. Notice that it is enough to use just one WARP board with the corresponding daughterboard to cover the 4 basic channels of interest. Theoretically, when doubling the bandwidth, the transmission power gets reduced by a half (3 dB). We corroborate this fact by looking at the similar reduction factor in the power detected per Hz in Fig. 3.

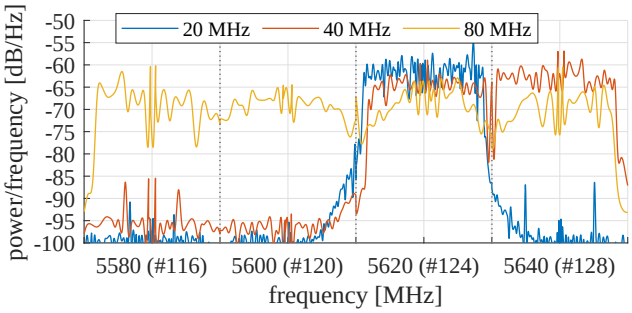
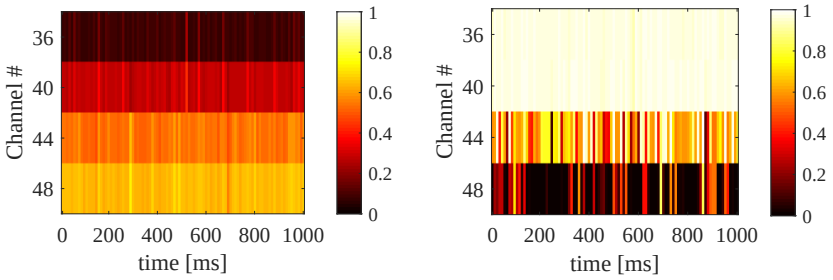


Figure 3: Transmission bandwidth effect on power.

[5] Further validations on power vs. central frequency (and RF) and power vs. distance can be found at [3].
[6] We follow the MAX2829 transceiver data sheet to convert from 10-bit RSSI values to dBm values as done in [42].



(a) Unsaturated. (b) Saturated.

Figure 4: Spectogram evolution of the controlled scenarios. Slots represent the 10-ms mean occupancy.

Regime	AP	36	40	44	48	Thr. [Mbps]
Unsaturated $\ell = 20$ Mbps	AS				**p**	20
	TP				**p**	20
	LS				**p**	20
Saturated $\ell = 150$ Mbps	AS		**p**			100
	TP	**p**				59
	LS				**p**	16

Table 2: Setup of controlled experiments. Letter **p** indicates the primary channel whereas colors represent the allocated bandwidth to each AP.

3.4.2 Controlled testbed evaluation

We now measure the activity of 3 overlapping BSSs' with static channel bonding capabilities under different traffic regimes. As shown in Fig. 2, we separate the APs from their corresponding `iperf` servers by $d = 4.8$ m. The WACA platform was placed equidistantly from the central AP and STA also at $d = 4.8$ m. We analyze two particular setups where none and all the BSSs' saturate. The load of every BSS is 20 and 150 Mbps in the first and second setup, respectively. Table 2 collects the setups' details. Essentially, in the unsaturated scenario, all the BSSs' share the same primary channel (48) but are set with different bonding capabilities. The second setup assigns different primary channels and bandwidth capabilities to each BSS. Note that we changed the primary channels from the previous experiments due to AP hardware restrictions. Nonetheless, we also confirmed that the external interference at the new band of interest was negligible.

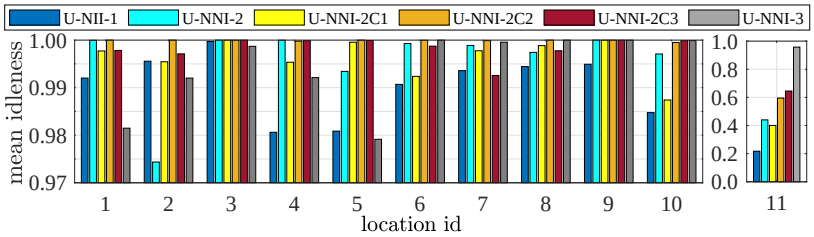
WACA collected 1 second of consecutive RSSI samples during the execution of each setup to measure the spectrum occupancy. Namely, we just compare the RSSI against the CCA level and say that a given channel is occupied `iff` RSSI $>$ CCA. Note that, throughout this paper, we compare the RSSI directly with the CCA no matter the type of signal being received. This approach is much more restrictive than relying on carrier sense since CCA thresholds are much smaller than energy detection (-82 and -62 dBm, respectively). This lets us simplify the experiments and consider worst-case occupancy values. Fig. 4 shows the occupancy evolution at each basic channel for the unsaturated and saturated setups. We confirm that the former setup does not reach full occupancy in any channel, whereas it is the contrary for the later. Indeed, we observe that the load ($\ell = 150$ Mbps) is high enough to make channels 36 and 40 occupied almost all the time (i.e., occupancy at the first 40-MHz band is always close to 1) while not being able to successfully deliver all the load as indicated by the throughput values in Table 2. This has an important consequence for the LS AP. Essentially, since static channel bonding is applied, the LS AP does not benefit from having its primary channel (48) far from the rest of APs. Consequently, its throughput is drastically reduced given that AS and TP introduce inadmissible activity in the first 40 MHz band.

4 Measurement Campaigns

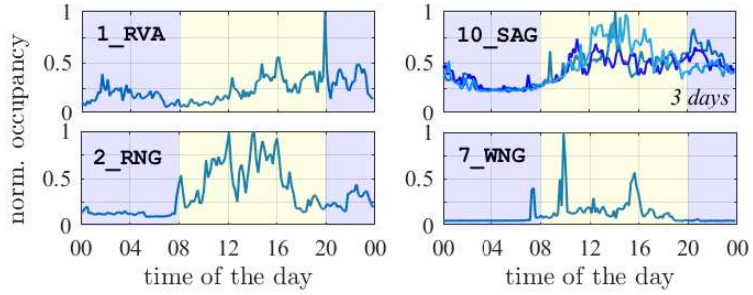
Using WACA, we perform extensive measurement campaigns covering two continents, dense urban areas, and multiple hours of samples in places of interest such as university campuses, apartment buildings, shopping malls, hotels and the Futbol Club Barcelona (a.k.a Barça) stadium (Camp Nou), one of the largest sports stadiums in the world. The measurements were taken from February to

Id	Location	Type	Duration
1_RVA	Rice Village Apart., HOU	Apartment	1 day
2_RNG	RNG lab at Rice, HOU	Campus	1 day
3_TFA	Technology for All, HOU	Com. center	1 day
4_FLO	Flo Paris, HOU	Cafe	1 hour
5_VIL	Rice Village, HOU	Shopping mall	20 min
6_FEL	La Sagrera, BCN	Apartment	1 week
7_WNO	WN group office, BCN	Campus	1 day
8_22@	22@ area, BCN	Office area	1 day
9_GAL	Hotel Gallery, BCN	Hotel	1 day
10_SAG	Sagrada Familia, BCN	Apartment	3 days
11_FCB	Camp Nou, BCN	Stadium	5 hours

Table 3: Measurement locations.



(a) Mean idleness per band.



(b) Runchart of the normalized occupation for apartments (upper) and universities (lower) highlighting day (yellow) and night (blue).

Figure 5: General spectrum occupancy trends.

August 2019 in Houston, TX, USA, and Barcelona, Spain. The shortest experiment took 20 minutes and the longest covered more than 1 week. The list of locations is shown in Table 3.

Here, we overview the complete dataset: there are 153,033 1-second iterations accounting for 42 hours, 30 minutes, and 3 seconds of actual measurements in the 24 20-MHz channels of the 5 GHz WiFi band. We first assess the entire activity record of each location at band $\mathcal{B}$, where $\mathcal{B}$ is a predefined set of contiguous channels. For instance, U-NII-1 band is defined as $\mathcal{B} = \{1, 2, 3, 4\}$, corresponding to basic channels 36, 40, 44, and 48 (see Figure 1a).[7] Fig. 5a shows the normalized mean idleness of each band, i.e., the mean number of samples that were found idle in each channel of band $\mathcal{B}$. We observe that the spectrum is idle most of the time in all scenarios except the stadium, indicating that the 5 GHz band is still profoundly underutilized even in densely populated areas.

Fig. 5b shows the daily temporal evolution of 4 example locations (2 apartments in the upper subplots, and 2 university campuses in the lower subplots). For the sake of representation, we plot the normalized occupancy of the whole band averaged in periods of 10 minutes. Concretely, we normalize with respect to the highest 10-minutes average occupancy encountered in each location. We clearly observe higher activity at working hours in the campus locations and a much less variable pattern in the apartment locations. In any case, from the low spectrum utilization observed in Fig. 5a, significant opportunities for channel bonding can be expected.

[7]The rest of bands are sequentially composed of groups of four consecutive 20-MHz channels. So, the next band is U-NNI-2 with channels $\{5, 6, 7, 8\}$ (basic channels 52, 56, 60, and 64), and the last one is U-NNI-3 with channels $\{21, 22, 23, 24\}$ (basic channels 149, 153, 157, 161).

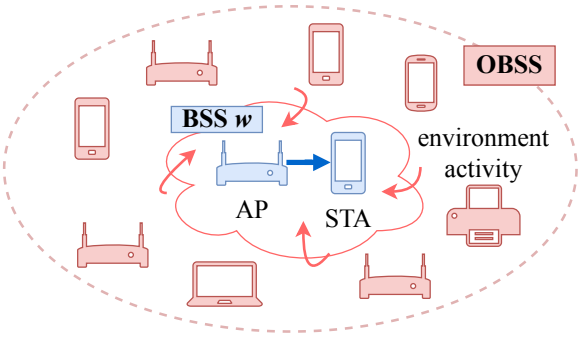


Figure 6: Diagram of the system model.

5 Evaluation Methodology

5.1 Overview

Our objective is to study the performance that a fully backlogged channel-bonding BSS, w, would obtain if it encountered the channels recorded in the measurement campaigns described above (see Fig. 6). The performance is a function of a number of factors such as which primary channel the bonding BSS selects, which channel bonding policy it employs, and the spectrum occupancy. Such factors are explored in subsequent sections. Here, we describe the overall methodology for all experiments.

The datasets captured by WACA at scenario s are represented by a 2-dimensional matrix Y^s of size $(N^s_{\text{it}} \times n_{\text{s}}) \times |\mathcal{R}|$, where N^s_{it} is the number of iterations of scenario s, and any element $y^s_{t,c}$ represents the power value at temporal sample t in basic channel c. From Y^s, we generate a binary matrix X^s of same size through an occupancy indicator function, where any element $x^s_{t,c}$ represents whether channel c was occupied at temporal sample t (1) or not (0). Formally, $x^s_{t,c} = (y^s_{t,c} > \text{CCA} : 1, 0), \forall t, c$, where the clear channel assessment (CCA) is set to -83.5 dBm (or 150 10-bit RSSI units), corresponding to the common CCA threshold -82 dBm plus a safety margin of -1.5 dBm. While IEEE 802.11ac/11ax introduce different CCA levels for the primary and secondary channels, in this work, we consider a more restrictive approach by assuming the same threshold in order to fairly compare different channel bonding policies. The mean occupancy at band $\mathcal{B}$ in scenario s is simply defined as

$$\bar{o}^s_{\mathcal{B}} = \frac{\sum_t \sum_{c \in \mathcal{B}} x^s_{t,c}}{N^s_{\text{it}} n_{\text{s}} |\mathcal{B}|}. \tag{1}$$

To provide meaningful experiments, we separately consider two 160-MHz bands composed of 8 basic channels: the U-NII-1&2 and part of the U-NII-2c sub-bands, $\mathcal{B}_{1\&2} = \{1, 2, 3, ..., 8\}$ and $\mathcal{B}_{2\text{c}} = \{9, 10, 11, ..., 16\}$, respectively. These sub-bands cover from channel 36 to 64 and from channel 100 to 128, respectively (see Fig. 1a). Notice that these are the only sub-bands that allow to perform 160-MHz transmissions in the IEEE 802.11ac/ax channelization. Moreover, we focus on epochs (or periods) of duration $T_{\text{per}} = 100$ ms (containing 10^4 temporal samples each) for which the mean occupancy at such sub-bands is at least 5%, i.e., $\bar{o}_{\mathcal{B}} \geq 0.05$, where $\mathcal{B} \in \{\mathcal{B}_{1\&2}, \mathcal{B}_{2\text{c}}\}$.

5.2 State Machine

We develop a discrete state machine that characterizes how the channel-bonding BSS responds to each power sample (or temporal sample) t according to the current state $S(t)$, and channel-bonding policy π, following the 802.11 standard. Given that the channel-bonding BSS is fully backlogged, the set of possible states is $\mathcal{S} = \{Busy, DIFS, BO, TX/RX\}$. State *Busy* indicates that the primary channel is busy, *DIFS* represents the period before initiating the backoff process, the backoff counter is decreased during *BO* state, and *TX/RX* represents the actual frame transmission-reception (including the control frames RTS, CTS, and ACK, the DATA frame, and the SIFS periods in between). We represent the channel-bonding BSS w as an AP and one or multiple clients that would perceive exactly the same spectrum activity as WACA captured in the measurement campaigns and must contend accordingly. To focus on channel bonding effects, we do not consider collisions within the channel bonding BSS, but only collisions that can occur due to other BSS's (see §7.2).

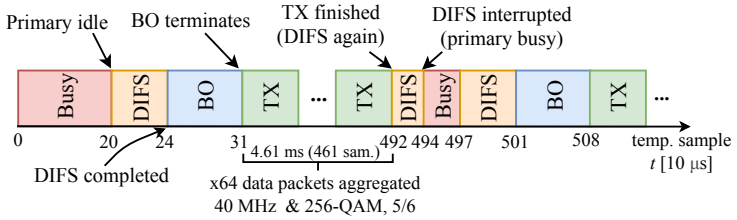


Figure 7: Example of the transitions between states.

Param.	Description	Value
CCA	CCA threshold	-83.5 dBm
MCS	MCS index	9 (256-QAM 5/6)
b	basic channel bandwidth	20 MHz
L_{d}	Length of a data packet	12000 bit
$\mathcal{B}$	Allocated set of basic channels	$\mathcal{B} \subseteq \mathcal{R}$
p	Primary channel	$p \in \mathcal{B}$
N_{a}	Max. no. of agg. packets per frame	64
T_{e}	Duration of an empty slot	10 µs
T_{SIFS}	SIFS duration	20 µs
T_{DIFS}	DIFS duration	30 µs
T_{PIFS}	PIFS duration	30 µs
T_{RTS}	RTS duration	50 µs
T_{CTS}	CTS duration	40 µs
T_{BACK}	Block ACK duration	50 µs
TXOP	Max. duration of a TXOP	5 ms
$\mathrm{CW}_{\mathrm{min}}$	Min. contention window	16
m	No. of backoff stages	5

Table 4: Trace-driven setup.

The set of basic channels selected for transmitting a frame depends both on the spectrum occupancy and on the selected channel-bonding policy π. Fig. 7 illustrates an example of the transitions between states. Empty slots have a duration $T_{\mathrm{slot}} = 10\,\mu\mathrm{s}$ and we consider $T_{\mathrm{slot}} = T_{\mathrm{s}} = 10\,\mu\mathrm{s}$ rather than 9 µs (802.11's default value) to align the duration of an idle backoff slot with the sample duration. Hence, whenever the channel-bonding BSS is in the backoff process at state *BO*, every idle sample at the primary channel p results in a backoff counter decrease of one empty slot. We use Wi-Fi parameters according to IEEE 802.11ax as shown in Table 4. After running the state-machine through all the temporal samples in the epoch, we compute the throughput Γ as the number of bytes in the successfully transmitted data packets n_{d} divided by the duration of the epoch, i.e., $\Gamma = (n_{\mathrm{d}} L_{\mathrm{d}})/T_{\mathrm{per}}$.

5.3 Channel Bonding Policies and Response

A channel-bonding policy π selects the set of basic channels to aggregate at the end of the backoff provided that the primary channel is available. Namely, *contiguous* channel bonding can select a set of channels both above and below the primary channel, provided they are consecutive. In contrast, *non-contiguous channel bonding* can combine all available channels at the time the primary channel becomes available. In §8, we consider additional channelization constraints imposed by 802.11ax.

How will other BSS's respond to the channel bonding BSS? For most experiments, we consider that they will defer their transmissions. Namely, the channel bonding BSS needs the channels to be available only when its countdown timer expires. If the bonding BSS does transmit but the trace indicates that a channel would have been occupied at some point during the transmission, we consider that such other BSS's will sense the bonding BSS and defer. The exception is in §7.2 in which we consider that other BSS's are hidden, do not sense the channel bonder, and cause a collision.

6 Evaluation of Bonding Gains and Origins

Here, we experimentally study the foundational elements of channel bonding including contiguous (CO) vs. non-contiguous (NC) channel bonding, load, primary channel selection, and cross-channel correlation. Due to the disparate nature of the stadium scenario, we focus our study on the non-stadium measurement campaigns to provide insights on a more homogeneous dataset. We leave the stadium's study for a dedicated section (§9).

6.1 Contiguous vs. Non-Contiguous Aggregation

For each transmission, non-contiguous channel bonding can utilize additional channels as compared to contiguous, by "skipping over" the busy channels to find the next unused one. Here, we explore the gains of this flexibility as well as (in rare cases) the losses by comparing the throughput of contiguous and non-contiguous channel bonding in three load regimes: low ($\bar{o}_{\mathcal{B}} \leq 0.1$), medium ($0.1 < \bar{o}_{\mathcal{B}} \leq 0.2$) and high ($\bar{o}_{\mathcal{B}} > 0.2$), respectively.

Fig. 8 shows the the throughput ratio of contiguous to non-contiguous channel bonding $\Gamma_{\text{CO}}(p)/\Gamma_{\text{NC}}(p)$, where $\Gamma_{\pi}(p)$ is the throughput achieved by policy π when selecting primary p in a given period. We plot the ratio for all possible primaries in $\mathcal{B}_{1\&2}$ and $\mathcal{B}_{2c}$.

The data reveals two remarkable phenomenon. First, contiguous outperforms non-contiguous in 2.5% of the cases (albeit with a modest throughput difference of 1.9%). But since non-contiguous is more flexible, how can it ever do worse? The answer is that the two policies result in different instants for transmission attempts. The contiguous policy occasionally (and quite randomly since the traces are the same) ends up with more favorable attempt instants. Nonetheless, in most cases, non-contiguous obtains higher throughput. For example, in many periods, at least one 20-MHz channel is idle during the whole period, which will always yield a gain for non-contiguous, but only sometimes yields a gain for contiguous bonding. In some cases, the difference can be quite high (e.g., a ratio of approximately 0.2 observed in low load). The origins of such extreme cases are the selection of the primary channel which we explore next. Second, the bar chart inside Fig. 8 reveals that both contiguous and non-contiguous channel bonding perform quite close on average for all occupancy regimes (low, medium, and high), and especially for the latter, as high load results in far fewer bonding opportunities overall.

Finding: Non-contiguous almost always outperforms contiguous channel bonding and their throughput differences are occasionally over a factor of 5. Nonetheless, their average throughputs are quite similar, which may ultimately favor contiguous channel bonding, since it is simpler to implement.

6.2 Primary Channel Selection

Selection of the primary channel is critical for channel bonding as it is where the backoff procedure is run. Namely, the transmitter must wait for a transmission opportunity on the primary channel and only then can explore adding channels. For contiguous channel bonding, there is an additional "edge effect". For example, having the first channel as primary only allows bonding of higher numbered channels, whereas having a middle channel allows both higher and lower, provided they are contiguous in both cases. To explore this issue, we define the best-throughput $\Gamma^*(\pi)$ of policy

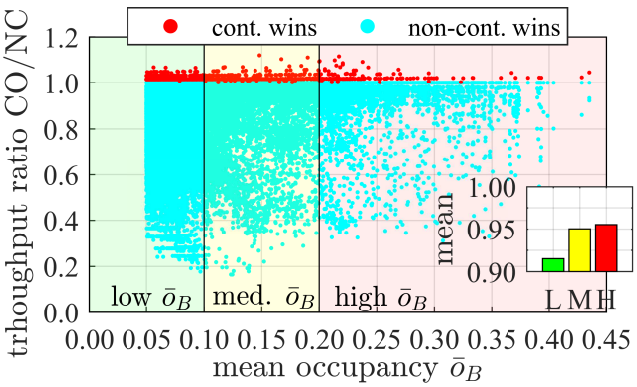


Figure 8: Throughput ratio of contiguous vs. non-contiguous channel bonding. The bar chart inset depicts the mean aggregated ratio for low (L), medium (M), and high (H) occupancy regimes.

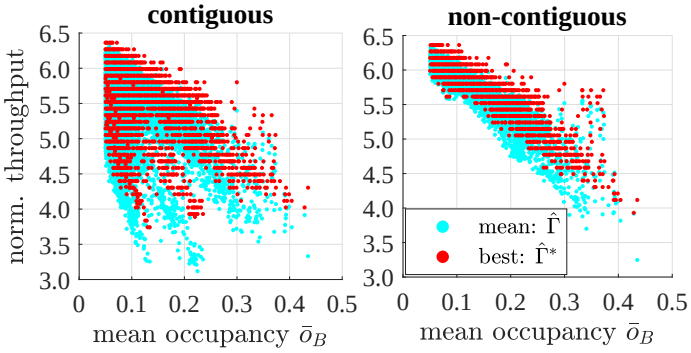


Figure 9: Normalized throughput of the best primary channel (*best*) and mean throughput for every primary channel (*mean*).

π as the throughput achieved when selecting the best primary channel $p^* \in \mathcal{B}$, i.e., the primary channel that maximizes throughput in each period when implementing π. So, apart from assessing the throughput when considering all possible primary channels, we also evaluate an upper bound on performance by focusing only on the best primary channel.

Figure 9 shows the throughput gain with respect to the throughput achieved by single channel transmission, i.e., $\hat{\Gamma}_\pi = \Gamma_\pi / \Gamma_{\text{SC}}$, where Γ_{SC} and Γ_π are the throughput achieved by single channel transmission and channel bonding policy π, respectively. Such normalized throughput is presented in two ways for each period: the *mean* value $\hat{\Gamma}$ in the legend refers to the mean normalized throughput achieved by selecting each of the possible 8 primary channels per sub-band, and the *best* value $\hat{\Gamma}^*$, to the normalized throughput achieved by the primary p^* providing the highest throughput.

First, observe that both contiguous and non-contiguous channel bonding outperform single channel by a factor of at least 3 to over 6. Second, observe the region where contiguous channel bonding has the lowest gains. Strikingly, while selecting the best primary channel boosts throughput, these lower-gain cases cannot be entirely eliminated with better primary channel selection. Thus, in these cases, the available spectrum does not have a consistent structure and only non-contiguous bonding can exploit the gaps (and there are indeed many gaps as the load is low here). Next, observe the high variance of the throughput due to the spectrum activity distribution. In particular, throughput differences for similar occupancy values can be up to 1.9× and 1.7× for contiguous and non-contiguous, respectively. Lastly, despite these first two findings, selecting the best primary channel indeed provides substantial gains and can raise throughput by up to 68% and 64% for contiguous and non-contiguous, respectively. While non-contiguous channel bonding might seem impervious to primary channel selection as it can bond any channels, recall that even non-contiguous requires an idle primary channel to begin transmission.

Finding: Selection of the best primary channel improves average throughput by over 60%. But for contiguous channel bonding, it cannot overcome high variance and lower gain scenarios due to the channel occupancy dynamics of the spectrum band.

6.3 Inter-Channel Occupancy Correlation

When a transmitter finds its primary channel available for transmission and attempts to bond channels, it aims to find secondary channels available. Thus, correlation among channel occupancies can be expected to help channel bonding performance. Here we study the spectral correlation of the best primary channel with other channels and compute the mean correlation coefficient as

$$\begin{aligned} \xi &= \sum_{p \in \mathcal{B}, p \neq p^*} \rho(p^*, p) / \big(|\mathcal{B}| - 1\big), \text{ with} \\ \rho(p^*, p) &= \frac{\mathbb{E}[(p^* - \mu_{p^*})(p - \mu_p)]}{\sigma_{p^*} \sigma_p} \end{aligned} \tag{2}$$

where $\mathcal{B} \in \{\mathcal{B}_{1\&2}, \mathcal{B}_{2c}\}$, thus $|\mathcal{B}| = 8$, $\rho(p^*, p)$ is the Pearson correlation coefficient of the temporal occupancy of channels p^* and p, as a function of the means, μ_{p^*} and μ_p, and standard deviations, σ_{p^*} and σ_p, of channel p^* and p, respectively. Notice that this assessment of spectral correlation by simultaneously measuring all channels is achieved for the first time by WACA.

Fig. 10 shows both the distribution of ξ vs. mean occupancy $\bar{o}_{\mathcal{B}}$ and the empirical cumulative distribution function (CDF) of ξ. We observe that ξ ranges approximately from -0.10 to 0.60, so

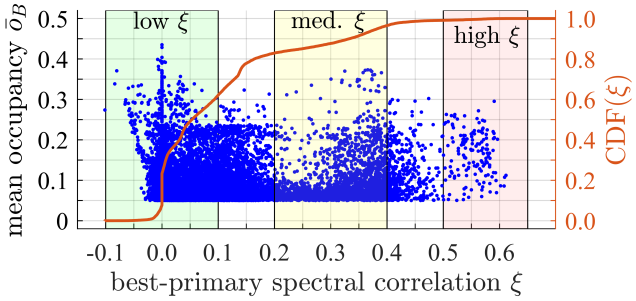
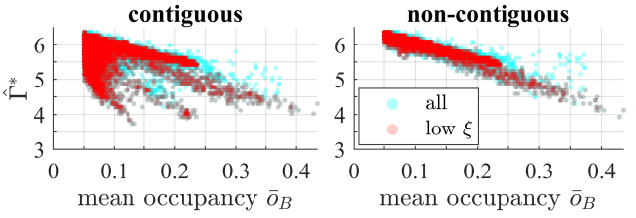
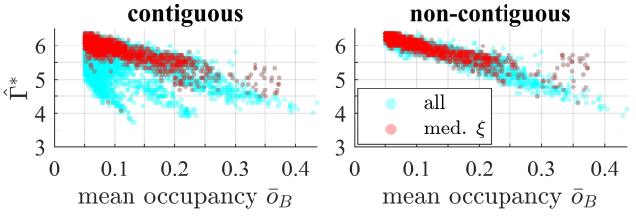


Figure 10: Distribution and empirical CDF of the best primary channel correlation ξ.

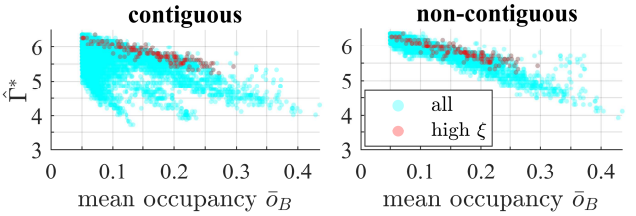
we define three correlation levels for ease of analysis: low ($\xi \leq 0.1$), medium ($0.2 \leq \xi < 0.4$), and high ($\xi \geq 0.5$). While we find that most of the periods (62%) fall inside the low correlation range, 17% of them present medium or higher correlation, indicating a significant amount of correlated epochs. A key origin of correlation among channels in the data set may be that the APs are already using channel bonding. Indeed, even though the dataset does not provide header information, some subsets of traces have multiple contiguous channels having almost identical busy/idle evolution. This suggests that channel bonding was employed during some portions of the measurement campaign.



(a) Low-correlation periods.



(b) Medium-correlation periods.

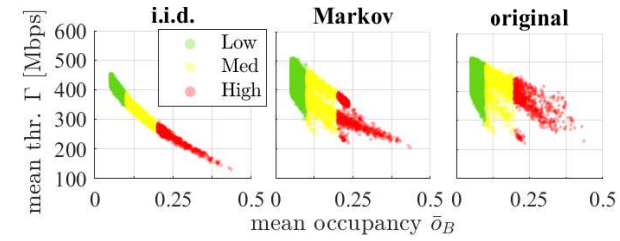


(c) High-correlation periods.

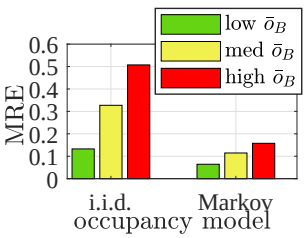
Figure 11: Evolution of the best-primary normalized throughput vs. correlation. We use transparencies at the points to highlight denser regions.

Figure 11 shows throughput given selection of the best primary channel, with the sub-figures highlighting intervals with low, medium, and high correlation. We observe that most of the measured channel occupancies show minimal inter-channel correlation. However, the figures for periods of medium and high correlation indicate that with higher correlation among channel occupancy, the lower performance periods are increasingly avoided, especially for contiguous channel bonding. Interestingly, this effect holds regardless of the mean occupancy $\bar{o}_{\mathcal{B}}$.

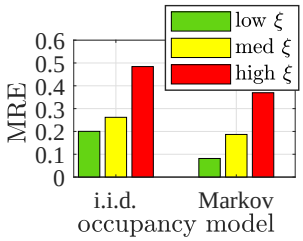
Finding: There are significant periods (17%) presenting medium to high correlation among channel occupancies. Such periods improve the performance of channel bonding regardless of the load. This can provide a hint as to what ideally would happen if multiple BSS's employ channel bonding, since channel bonding itself creates channel-occupancy correlation.



(a) Throughput vs. occupancy



(b) Error vs. occupancy.



(c) Error vs. correlation.

Figure 12: Original vs. synthetic traces.

6.4 A Markov Model for Channel Occupancy?

In §6.3, we found that even though channels could be considered to be uncorrelated on average since activity is expected to be normally from separate BSS's, traces exhibited a significant amount of epochs of inter-channel occupancy correlation. With this experiment, we aim at pointing out the importance of capturing inter-channel correlation by comparing the original traces against two simple models. In particular, we first consider a Markov model that characterizes each channel as an independent two state (occupied, not occupied) Markov chain with exponential holding times for each state. We compute the mean transition rates (and hence mean occupied and not-occupied time) for each channel from the traces. We then compare performance between the parameter-matched Markov model and the actual traces. As a baseline, we also consider a *uniform i.i.d.* model in which each temporal sample is occupied or not according to an i.i.d. Bernoulli distribution, with each channel again having the mean band occupancy matched from data. Our objective is not to develop a statistical occupancy model, but rather to study the extent to which simple models can or cannot characterize the behavior we observe.

We study below downlink throughput for these two models as compared to the traces when applying contiguous channel bonding. Since the throughput is impacted by the primary channel selection, we evaluate the selection of the 16 possible primary channels: 8 in sub-band $\mathcal{B}_{1\&2}$ and 8 in $\mathcal{B}_{2c}$.

Figure 12a shows the raw throughput (considering every possible primary) of the original traces and the synthetic model-generated occupancy. While both models capture the general trend of throughput decaying with increasing occupancy, the difference between datasets is evident in the form of the scatter plots.

Fig. 12b and Fig. 12c depict the mean relative error (MRE) of the models' throughput with respect to the original traces grouping by load and correlation regime, defined by the thresholds presented in §6.1 and §6.3, respectively. We observe that the i.i.d model is completely misleading with a mean error around 50% for high loads and/or high correlation. Unfortunately, although the Markov model is able to capture the mean on and off times for each channel, we observe that the mean error is still significant (up to 16% and 37% on average for high load and high correlated periods, respectively). Worse, particular periods assessed through the Markov model lead to large errors up to 33%, 58%, and 62% for low, medium and high occupancy regimes, respectively.

While we expected the higher relative errors to arise for high load periods, it is surprisingly critical the impact of not capturing correlation on both models. In fact, in contrast to the correlation distribution of the original traces shown in Fig. 10, both models present a distribution similar to a delta function centered at $\xi = 0$. Given that the original mean occupancy is kept for all the periods generated by the models, results suggest that missing correlation is the main cause of outraging the models' accuracy.

Finding: unfortunately, simple channel occupancy models severely underestimate the available gains for channel bonding. Thus, modeling occupancy behavior by introducing inter-channel corre-

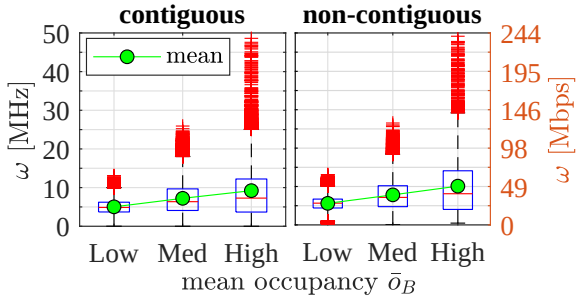


Figure 13: Distribution of bandwidth and data rate lost for other BSS's for every primary channel.

lation remains an important avenue for future work.

7 Whither Other BSS's

Thus far, we have evaluated how much throughput is gained from channel bonding. However, when one BSS gains access to multiple channels, nearby BSS's operating on other channels will have to defer or could yield hidden terminal collisions. We consider both cases as follows.

7.1 How Much Are Others Hindered?

When other BSS's defer to a channel bonder, it will add to their channel access latency and could decrease their throughput. Here, we study how much all other BSS's are hindered due to channel bonding and introduce a bandwidth deprivation metric ω as the aggregate bandwidth that would have been active (i.e., was being used by neighbor BSS's) during the transmissions of the channel bonding BSS. This metric can serve as an upper bound to how much others are hindered by viewing that all of these attempts by other BSS's would be lost vs. deferred. More formally, let

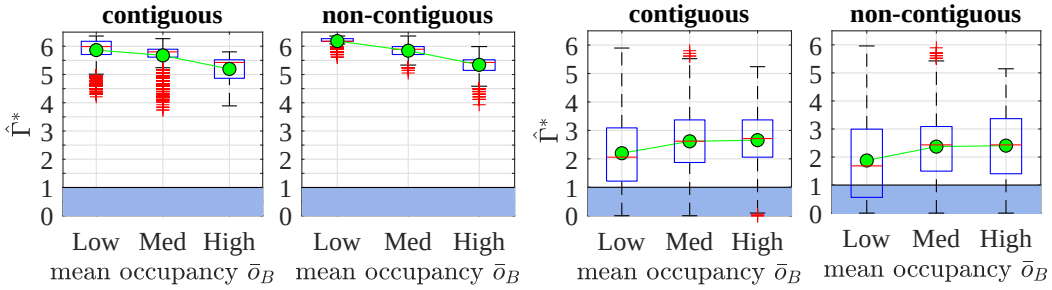
$$\omega = \frac{\left(\sum_{f=1}^{n_f} \sum_{t \in \mathcal{T}_f} \sum_{c \in \mathcal{B}_{\text{tx},f}} x_{t,c}\right) b T_\text{s}}{T_\text{per}}, \tag{3}$$

where n_f is the number of frames transmitted, $\mathcal{T}_f$ is the set of consecutive temporal samples used for transmitting frame f, $\mathcal{B}_{\text{tx},f}$ is the set of channels used for transmitting frame f, T_s is the duration of a temporal sample (10 µs), $b = 20$ MHz is the bandwidth of a basic channel, and $T_\text{per} = 100$ ms is the observation duration. We normalize by $T_\text{per} = 100$ ms rather than the transmission time of each policy to provide a fair comparison. That is, ω represents how much raw bandwidth is deprived to surrounding BSSs in absolute terms [MHz], without considering how many transmissions are performed by each policy.

Fig. 13 shows the distribution (boxplot) of the bandwidth deprivation in MHz and Mbps for every primary channel using the occupancy categories defined in §6.1. In order to estimate the corresponding data rate, we assume that all occupied samples are comprised of 20-MHz data frames transmitted also at MCS 9 The figure illustrates the intrinsic consequences of bonding channels without considering how the others might use the spectrum during the transmission. Namely, a channel bonder may find that most channels are free at backoff termination, thus subsequently occupying most (or all) of them. However, it may occur that right after starting the transmission, external activity appears in some of the channels being used, thus hindering surrounding BSS's.

Second, we observe that the differences in bandwidth deprivation for non-contiguous vs. contiguous channel bonding are quite modest, even when non-contiguous channel bonding achieves higher throughput. The key reason is that most high occupancy periods concentrate activity in some 20 MHz channels, often leaving at least one 20 MHz channel idle during nearly the entire period. In contrast to contiguous, non-contiguous channel bonding can always realize small throughput gains by bonding those idle channels regardless of the primary channel allocation. Also, despite the low average impact, there are multiple outliers in which the bandwidth lost by other BSS's is significantly higher than average, thus manifesting the worst-case external effect.

Finally, we can ask if channel bonding is simply a zero-sum-game: do the gains of a bonding BSS simply subtract from other BSS's throughput? To address this question, we assess whether the average throughput gained by channel bonding exceeds the worst-case prevented transmissions by



(a) Hindering.

(b) Hidden terminals.

Figure 14: Distribution of the norm. best-primary-throughput.

other BSS's, using single channel transmission as a baseline. Thus, we define the ratio

$$\kappa_\pi = \frac{\bar{\Gamma}_\pi - \bar{\Gamma}_{\text{SC}}}{\bar{\omega}_\pi - \bar{\omega}_{\text{SC}}}, \tag{4}$$

where $\bar{\Gamma}_\pi$ and $\bar{\Gamma}_{\text{SC}}$ are the mean throughput of policy π and SC, and $\bar{\omega}_\pi$ and $\bar{\omega}_{\text{SC}}$ are the corresponding mean throughput deprivation. A value $\kappa_\pi = 1$ would indicate that the trade-off between the bonding gain and external throughput deprivation is actually a zero-sum-game, whereas $\kappa_\pi > 1$ would indicate net gains. The results indicate that both contiguous and non-contiguous channel bonding yield substantial net gains over all occupancy regimes. Specifically, $\kappa_{\text{CO}} = 95.2, 68.9, 49.4$ for contiguous, and $\kappa_{\text{NC}} = 91.9, 65.7, 45.5$ for non-contiguous, under low, medium, and high occupancy regimes, respectively.

Finding: In the worst case, a channel bonding BSS can hinder neighbors by 100's of Mb/sec. However, the *average* hindrance is quite low and neighbors can defer instead of not transmitting to reduce the impact. Moreover, the data rate gained by bonding far exceeds the worst-case data rate lost by neighbors, since neighbors are not always backlogged during bonding epochs.

7.2 The Hidden Cost of Hidden Nodes

Here, we continue our study of potential detrimental effects of channel bonding by considering the case of hidden nodes. We again impose a worst-case scenario on the measurements as follows: channel access occurs as previously with contention occurring on the primary channel and channel bonding adding channels according to availability and the policy, contiguous or non-contiguous. However, here we reconsider the measured activity that occurs during the bonded transmission. While in the previous subsection, we considered that the other BSS's would have deferred, here, we consider that other BSS's are hidden from the bonded transmission and still transmit and result in frame loss. Thus, this scenario considers that the RTS/CTS mechanism is not used or fails and disallows capture effects in which the higher signal-to-noise ratio receiver correctly decodes, e.g., [43]. Thus, in these two ways, this experiment provides an empirical upper bound as to the damage that hidden terminals could do to a channel bonded transmission.

In particular, we assume that a frame f is lost if temporal samples are found active in any of the channels used for transmitting f. Specifically, f is lost if the following expression holds,

$$\sum_{t \in \mathcal{T}_f} \Big(\big(\sum_{c \in \mathcal{B}_{\text{tx},f}} x_{t,c} \big) \geq 1 : 1, 0 \Big) \geq \alpha |\mathcal{T}_f|, \tag{5}$$

where $\mathcal{T}_f$ is the set of consecutive temporal samples required for transmitting frame f, and the term $\sum_{c \in \mathcal{B}_{\text{tx},f}} x_{t,c}$ denotes the number of occupied channels at time t. We apply a factor $\alpha = 0.01$ to exclude spurious noise as cause of loss. Thus, we avoid for instance losing an entire frame when just a single temporal sample was found active during the transmission.

Fig. 14a and Fig. 14b depict the throughput distribution under hindering (without hidden terminals) and hidden-terminal scenarios, respectively. First, observe that throughput is significantly reduced in comparison to the case with no hidden terminals. For example, for contiguous channel bonding in high load, mean throughput gains over single-channel transmission are reduced from 5.2× to 2.6× (50%).

Second, there are significant periods where channel bonding results in extremely low performance, even for low occupancy values. In fact, there are scenarios in which single-channel access outperforms

bonding (blue areas where $\hat{\Gamma}^* < 1$ in the figure). In particular, single channel outperforms contiguous channel bonding in 18%, 8% and 8% of the cases for low, medium and high occupancy, respectively. Even worse, single channel outperforms non-contiguous channel bonding in 38%, 14%, and 15% of cases respectively. This illustrates that external traffic patterns may lead to accentuated chances of incurring collisions, thus impairing the performance of the channel bonder.

Third, due to its more aggressive nature, non-contiguous channel bonding performs slightly worse than contiguous both in terms of average throughput and how often it is outperformed by single channel transmission. This, together with the fact that contiguous performs similarly on average to non-contiguous without hidden terminals, indicates that gains provided by non-contiguous can be relatively small considering its greater risk and more complex design.

Finally, also in contrast to Fig. 14a, Fig. 14b indicates that the decaying trend of mean normalized best-throughput with load when neighbors defer does not hold with hidden terminals. Indeed, we observe a slight increase of the normalized best-throughput as the occupancy increases as fewer bonding opportunities also diminish the risk of collisions.

Finding: While channel bonding outperforms single-channel transmission in most cases, it is vulnerable to hidden node interference, with non-contiguous channel bonding being impacted the most. To avoid the aforementioned worst-case throughput penalties, nodes can use RTS/CTS, implement capture (to better receive even while a hidden terminal transmission overlaps in time), or increase channel sensing sensitivity, i.e., reduce the threshold for sensing channel activity to reduce the frequency of hidden terminal transmissions.

8 IEEE 802.11ax Constraints

Here, we consider additional constraints imposed by 802.11ax on channel bonding. Namely, 802.11ax defines three modes of channel bonding, each of which restrict channel bonding to a subset of channels [1]. For example, U-NNI-1 and U-NNI-2 restrict channel bonding groups to $\mathcal{C}_{\mathcal{B}_{1\&2}} = \{\{1,2\},\{3,4\},\{5,6\},\{7,8\},\{1,2,3,4\},\{5,6,7,8\},\{1,...,8\}\}$ so that (for example) a 40 MHz bonded channel that combines channels 2 and 3 is not allowed. Notice that we do not use the IEEE 802.11 channel indices (e.g., 36, 40, 44,...) in the example above for ease of exposition. While reducing throughput when only a single BSS bonds, such restrictions can reduce fragmentation if multiple BSS's bond. Moreover, such restrictions simplify implementation. Thus, here we study the cost of 802.11ax's channelization restrictions when a single BSS bonds and compare to the non-restricted cases explored this far.

Static Channel Bonding (SCB): employs a *fixed* channel width and therefore must always use the same set of channels. It can be viewed as an"all-or-nothing" policy: for example, if static channel bonding uses 8 consecutive basic channels (160 MHz), it must wait until all 8 channels are free in order to transmit using the entire 160 MHz. Although simple to implement, it is vulnerable to excessive deferral while waiting for all channels to become available.

Dynamic Channel Bonding (DCB): picks the largest available set of contiguous basic channels found free subject to both the primary channel being included and the total set of channels being an allowed set under the restrictions such as above. For example, if channel 4 is selected as a primary channel, dynamic channel bonding allows bonded transmissions with channels $\{3,4\}$, $\{1,2,3,4\}$, or $\{1,...,8\}$, as well as single channel transmission on channel 4. Other combinations such as $\{4,5\}$ are not permitted.

Preamble Puncturing (PP): allows puncturing wide channels, i.e., aggregating non-contiguous basic channels. Even though the total channel is still subjected to the above channelization constraints, preamble puncturing is more flexible than dynamic channel bonding since it enables non-contiguous channel combinations, provided the primary channel is included [1, 44]. For example, a transmission using primary channel 3 could transmit on 60 MHz using channels $\{1,3,4\}$, which would be a punctured subset of $\{1,2,3,4\}$. Nonetheless, not every combination of punctured channels is allowed in the standard. For instance, $\{1,3\}$ is not permitted.

Observe that both static and dynamic channel bonding are restricted special cases of contiguous channel bonding studied previously, whereas preamble puncturing is a restricted special case of non-contiguous channel bonding.

Here, we compare the throughput of both the (non-restricted) contiguous and non-contiguous polices with the three 802.11ax policies, all using selection of the best primary channel. Fig. 15 shows the resulting 802.11ax throughput normalized to the unrestricted case for the respective category of contiguous or non-contiguous.

First, observe that 802.11ax static channel bonding suffers severe throughput degradation com-

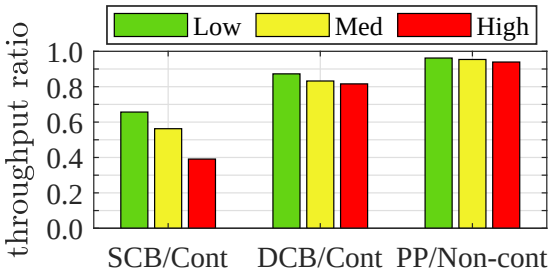


Figure 15: Best primary throughput ratio of the IEEE 802.11ax policies with respect to contiguous and non-contiguous channel bonding.

pared to contiguous channel bonding. Namely, the desirable simplicity of static channel bonding comes with a high performance cost. For example, under heavy load, static channel bonding only obtains 39% of the throughput compared to contiguous channel bonding, indicating that too much time is spent deferring to realize significant gains from wide channels. Moreover, static channel bonding also encounters poorly performing outliers, well below the mean. For example, the 1%-percentile performance is less than 10% for medium and high loads, even less than single channel allocation.

Second, comparing 802.11ax dynamic channel bonding to contiguous channel bonding (middle bars), we observe that it achieves 87% to 81% of the average throughput of contiguous channel bonding for low and high load respectively, with 1%-percentiles over 65% for any load regime.

Finally, comparing 802.11ax preamble puncturing to non-contiguous channel bonding (right set of bars), the former obtains 96% to 94% of the throughput of the latter in low and high load respectively. Further, throughput differences are kept smaller also in the most divergent scenarios, reaching 1%-percentiles over 77% for all the loads.

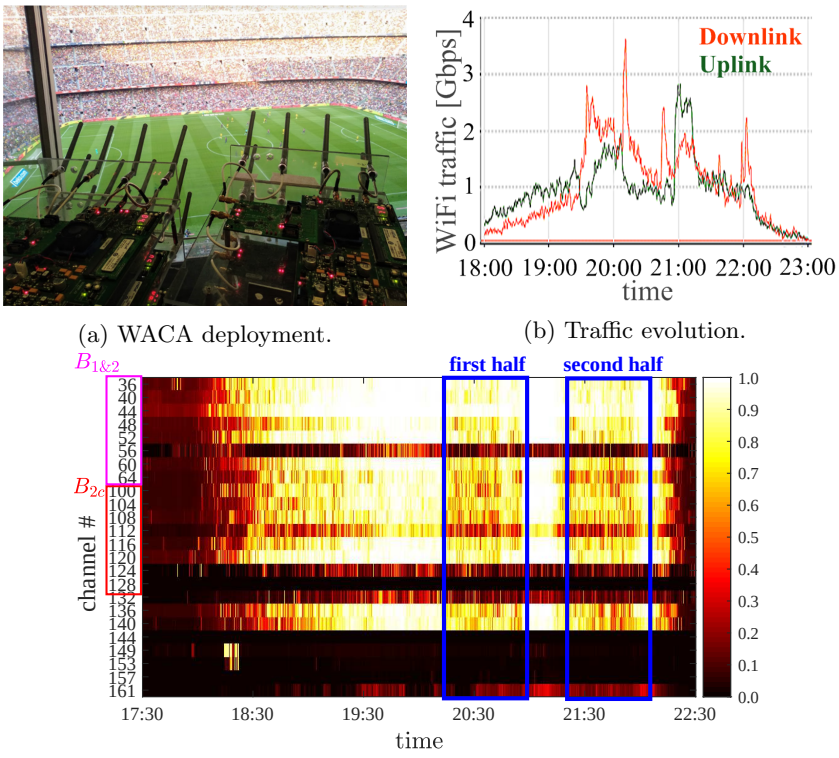
Finding: IEEE 802.11ax channelization restrictions have a modest throughput cost compared to non-restricted. For example, standard-compliant preamble puncturing obtains over 90% of the throughput of (non-restricted) non-contiguous channel bonding. Unfortunately, the simplest standard policy, static channel bonding, performs quite poorly.

9 A Sold Out Stadium

Finally, we assess if channel bonding can provide throughput gains in areas with persistently high load over all channels. To study such a scenario, we deployed WACA in the press box of the Camp Nou stadium during a football game with over 98,000 spectators present. Measurements were taken on August 4, 2019, from 17:24 to 22:30 accounting for a total duration of 5 hours and 6 minutes. On that date, the Joan Gamper trophy was held, which pitted the local club (Futbol Club Barcelona) against the visiting club (Arsenal Football Club). Fig. 16a shows a photograph of the setup.

We also obtained data from the stadium's network management team which indicated that up to 12,000 Wi-Fi clients were simultaneously connected. Downlink and uplink traffic (provided by Camp Nou's management) is depicted in Fig. 16b and the spectogram in Fig. 16c. Each slot in the spectogram represents the occupancy of each channel averaged in 1-second periods. We observe that most channels were highly occupied during the measurements. Moreover, we can observe the users' behavior induced nonstationarity of the traces as the match progressed. Namely, while activity is always high, there is a notable reduction during play (first and second half) compared to activity before, between halfs, and right after the game time. We also observe that the majority of the channels are crowded most of the time with periods reaching mean band occupancy values $\bar{o}_{\mathcal{B}}$ rising up to 99%. In fact, 22% of the periods have $\bar{o}_{\mathcal{B}} > 0.8$. Nonetheless, we pose the question as to whether the remaining occupancy gaps can be exploited by channel bonding to boost aggregate throughput.

Fig. 17 shows normalized throughput under best primary channel selection when applying contiguous and non-contiguous channel bonding. Under loads exceeding 90%, both policies have throughput close to single channel transmission, indicating that there is simply minimal remaining margin for gain. Nonetheless, in transient epochs where load is low, both policies exploit throughput gains. For example, in the stadium during epochs of 20-30% load, the gain over single channel throughput is 3.69× and 5.30× for contiguous and non-contiguous respectively. In fact, we observe that non-contiguous significantly outperforms contiguous in a wide regime of "non-extreme" loads



(a) WACA deployment.

(b) Traffic evolution.

(c) Spectrum occupancy evolution.

Figure 16: Wi-Fi activity at the Camp Nou stadium: a) WACA deployment in the press box, b) downlink and uplink Wi-Fi traffic evolution (source: stadium's IT management), and c) spectrum occupancy evolution.

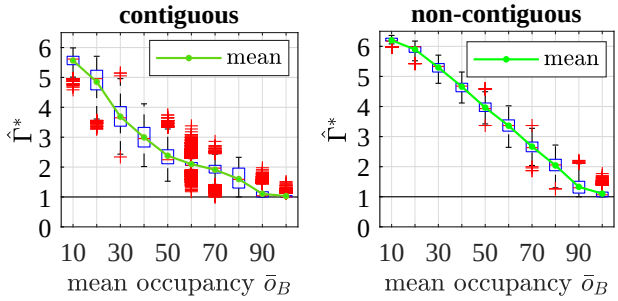


Figure 17: Gains for the bonding BSS assuming others defer.

given the heterogeneous activity distribution over channels. For example, at 40-50% average occupancy, contiguous channel bonding has average throughput gain of 2.38× whereas non-contiguous has 3.97×, a gain of 67%. Recall that for the non-stadium experiments in Figure 14a, the gains during epochs of 20-30% load (considered inside "high" load for the urban scenarios) were 5.19× and 5.34× for contiguous and non-contiguous bonding. Comparing to the stadium, non-contiguous channel bonding performs remarkably similarly (5.30 in stadium vs. 5.34 otherwise). In contrast, contiguous channel bonding performs worse in the stadium (3.69 in stadium vs. 5.19 otherwise) even under the same average load. The reason may be the minimal inter-channel correlation at the stadium. Indeed, the stadium management confirmed that no channel bonding was implemented in any of the stadium's APs.

Lastly, as shown in §7.2, the story can turn quite negative if competing BSS's do not defer to the channel bonder, i.e., if other BSS's are hidden. In such a worst case, all bonded transmissions with subsequent activity are considered lost. Figure 18 shows the results of this scenario for the stadium traces. As indicated by the blue shadowed areas, most of the periods achieve higher throughput with single channel transmission rather than using channel bonding. Indeed, single channel transmission outperforms both channel bonding policies in at least 80% of the epochs over all occupancy regimes. Moreover, comparing similar loads to the other data sets (e.g., 20-30% occupancy), the stadium

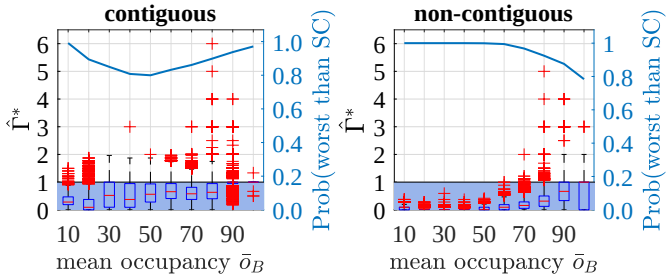


Figure 18: Hidden terminal scenario with the ratio of epochs that single channel outperforms bonding shown by the blue curve.

performs far worse with hidden terminals. For instance, comparing Figure 18 and Figure 14b, we find that for 20-30% average occupancy, even with worst-case hidden terminals, the average throughput gain of contiguous channel bonding was 2.66×, whereas for the stadium, the gain is reduced to 0.59×, i.e., throughput is lost. This indicates that the stadium traffic and channel occupancies yield greater hidden terminal risk than urban environments, even for the same traffic load. Thus, similarly to the hindering case, the results suggest that negligible correlation is detrimental in the presence of hidden terminals.

Finding: Even under the stadium's extremely high average load, short durations of lower load can be exploited to yield significant throughput gains. However, the risk is high as hidden terminals could drive the throughput to levels even worse than without any channel bonding.

10 Conclusion

In this paper, we introduce WACA, an all-channel Wi-Fi spectrum analyzer for simultaneous measurement of all 24 20 MHz channels that allow channel bonding at 5 GHz. We present extensive measurement campaigns covering two continents, diverse areas, and many hours of signal strength samples. We use the measurements to explore key aspects of channel bonding spanning from statistics of the channel occupancies to channel bonding standards and policies.

References

[1] "IEEE P802.11ax/D4.0 High Efficiency WLAN," pp. 1–682, 2019.

[2] "IEEE 802.11be Extremely High Throughput, 802.11-18/1231r6," 2019. [Online]. Available: http://www.ieee802.org/11/Reports/ehtsg_update.htm

[3] S. Barrachina-Muñoz, B. Bellalta, and E. Knightly, "Wi-Fi All-Channel Analyzer," in *14th International Workshop on Wireless Network Testbeds, Experimental evaluation & Characterization (WiNTECH'20)*. ACM, 2020.

[4] M. Rademacher, K. Jonas, and M. Kretschmer, "Quantifying the spectrum occupancy in an outdoor 5 GHz WiFi network with directional antennas," in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2018, pp. 1–6.

[5] T. Taher, R. Attard, A. Riaz, D. Roberson, J. Taylor, K. Zdunek, J. Hallio, R. Ekman, J. Paavola, J. Suutala *et al.*, "Global spectrum observatory network setup and initial findings," in *2014 9th International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM)*. IEEE, 2014, pp. 79–88.

[6] S. Subramaniam, H. Reyes, and N. Kaabouch, "Spectrum occupancy measurement: An autocorrelation based scanning technique using USRP," in *2015 IEEE 16th Annual Wireless and Microwave Technology Conference (WAMICON)*. IEEE, 2015, pp. 1–5.

[7] M. Wellens, J. RiihijäRvi, and P. MäHöNen, "Empirical time and frequency domain models of spectrum use," *Physical Communication*, vol. 2, no. 1-2, pp. 10–32, 2009.

[8] V. Kone, L. Yang, X. Yang, B. Y. Zhao, and H. Zheng, "The effectiveness of opportunistic spectrum access: A measurement study," *IEEE/ACM Transactions on Networking*, vol. 20, no. 6, pp. 2005–2016, 2012.

[9] S. W. Day and A. Wagstaff, "Activity in the WiFi bands - An Objective Approach to Estimation," in *ARMMS Conference*. Mass, 2014.

[10] F. Salahdine and H. El Ghazi, "A real time spectrum scanning technique based on compressive sensing for cognitive radio networks," in *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*. IEEE, 2017, pp. 506–511.

[11] S. Biswas, J. Bicket, E. Wong, R. Musaloiu-e, A. Bhartia, and D. Aguayo, "Large-scale measurements of wireless network behavior," in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, 2015, pp. 153–165.

[12] M. López-Benítez and F. Casadevall, "Modeling and simulation of time-correlation properties of spectrum use in cognitive radio," in *2011 6th International ICST Conference on Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM)*, 2011, pp. 326–330.

[13] V. Kone, L. Yang, X. Yang, B. Y. Zhao, and H. Zheng, "On the feasibility of effective opportunistic spectrum access," in *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, 2010, pp. 151–164.

[14] M. Höyhtyä, A. Mämmelä, M. Eskola, M. Matinmikko, J. Kalliovaara, J. Ojaniemi, J. Suutala, R. Ekman, R. Bacchus, and D. Roberson, "Spectrum occupancy measurements: A survey and use of interference maps," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2386–2414, 2016.

[15] S. A. Hanna and J. Sydor, "Distributed sensing of spectrum occupancy and interference in outdoor 2.4 GHz Wi-Fi networks," in *Global Communications Conference (GLOBECOM)*. IEEE, 2012, pp. 1453–1459.

[16] S. A. Hanna, "A 3-state hypothesis test model for cognitive radio systems," in *2014 IEEE International Symposium on Dynamic Spectrum Access Networks (DYSPAN)*. IEEE, 2014, pp. 291–302.

[17] S. Yin, D. Chen, Q. Zhang, M. Liu, and S. Li, "Mining spectrum usage data: a large-scale spectrum measurement study," *IEEE Transactions on Mobile Computing*, vol. 11, no. 6, pp. 1033–1046, 2012.

[18] L. Deek, E. Garcia-Villegas, E. Belding, S.-J. Lee, and K. Almeroth, "The impact of channel bonding on 802.11n network management," in *Proceedings of the Seventh COnference on emerging Networking EXperiments and Technologies*, 2011, pp. 1–12.

[19] L. Deek, E. Garcia-Villegas, E. Belding, S.-J. Lee, and K. Almerot, "Intelligent channel bonding in 802.11n WLANs," *IEEE Transactions on Mobile Computing*, vol. 13, no. 6, pp. 1242–1255, 2014.

[20] M. Y. Arslan, K. Pelechrinis, I. Broustis, S. V. Krishnamurthy, S. Addepalli, and K. Papagiannaki, "Auto-configuration of 802.11n WLANs," in *Proceedings of the 6th International COnference*, 2010, pp. 1–12.

[21] M. Y. Arslan, K. Pelechrinis, I. Broustis, S. Singh, S. V. Krishnamurthy, S. Addepalli, and K. Papagiannaki, "ACORN: An auto-configuration framework for 802.11n WLANs," *IEEE/ACM Transactions on Networking (TON)*, vol. 21, no. 3, pp. 896–909, 2013.

[22] Y. Zeng, P. H. Pathak, and P. Mohapatra, "A first look at 802.11ac in action: Energy efficiency and interference characterization," in *2014 IFIP Networking Conference*. IEEE, 2014, pp. 1–9.

[23] M. Park, "IEEE 802.11ac: Dynamic bandwidth channel access," in *Communications (ICC), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1–5.

[24] M. X. Gong, B. Hart, L. Xia, and R. Want, "Channel bounding and mac protection mechanisms for 802.11ac," in *2011 IEEE Global Telecommunications Conference-GLOBECOM 2011*. IEEE, 2011, pp. 1–5.

[25] B. Bellalta, A. Faridi, J. Barcelo, A. Checco, and P. Chatzimisios, "Channel bonding in short-range WLANs," in *European Wireless 2014; 20th European Wireless Conference*. VDE, 2014, pp. 1–7.

[26] B. Bellalta, A. Checco, A. Zocca, and J. Barcelo, "On the interactions between multiple overlapping wlans using channel bonding," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 2, pp. 796–812, 2015.

[27] A. Faridi, B. Bellalta, and A. Checco, "Analysis of dynamic channel bonding in dense networks of wlans," *IEEE Transactions on Mobile Computing*, vol. 16, no. 8, pp. 2118–2131, 2016.

[28] M.-S. Kim, T. Ropitault, S. Lee, and N. Golmie, "A throughput study for channel bonding in ieee 802.11ac networks," *IEEE Communications Letters*, vol. 21, no. 12, pp. 2682–2685, 2017.

[29] M. Yazid and A. Ksentini, "Stochastic modeling of the static and dynamic multichannel access methods enabling 40/80/160 MHz channel bonding in the VHT WLANs," *IEEE Communications Letters*, vol. 23, no. 8, pp. 1437–1440, 2019.

[30] S. Barrachina-Muñoz, F. Wilhelmi, and B. Bellalta, "To overlap or not to overlap: Enabling channel bonding in high-density WLANs," *Computer Networks*, vol. 152, pp. 40–53, 2019.

[31] S. Barrachina-Muñoz, F. Wilhelmi, and B. Bellalta, "Dynamic channel bonding in spatially distributed high-density WLANs," *IEEE Transactions on Mobile Computing*, vol. 19, no. 4, pp. 821–835, 2020.

[32] S. Khairy, M. Han, L. X. Cai, Y. Cheng, and Z. Han, "A renewal theory based analytical model for multi-channel random access in IEEE 802.11ac/ax," *IEEE Transactions on Mobile Computing*, 2018.

[33] A. Zakrzewska and L. Ho, "Dynamic channel bandwidth use through efficient channel assignment in ieee 802.11ac networks," in *2019 IEEE 90th Vehicular Technology Conference*. IEEE, 2019, pp. 1–6.

[34] S. Barrachina-Muñoz, F. Wilhelmi, and B. Bellalta, "Online primary channel selection for dynamic channel Bonding in high-density WLANs," *IEEE Wireless Communications Letters*, vol. 9, no. 2, pp. 258–262, 2020.

[35] Z. Khan and J. J. Lehtomäki, "Interactive trial and error learning method for distributed channel bonding: model, prototype implementation, and evaluation," *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 2, pp. 206–223, 2019.

[36] S.-s. Lee, T. Kim, S. Lee, K. Kim, Y. H. Kim, and N. Golmie, "Dynamic channel bonding algorithm for densely deployed 802.11ac networks," *IEEE Transactions on Communications*, vol. 67, no. 12, pp. 8517–8531, 2019.

[37] M. Han, S. Khairy, L. X. Cai, Y. Cheng, and F. Hou, "Capacity analysis of opportunistic channel bonding over multi-channel WLANs under unsaturated traffic," *IEEE Transactions on Communications*, 2019.

[38] G. Buttazzo, C. Di Franco, and M. Marinoni, "Design and analysis of target-sensitive real-time systems," *Software: Practice and Experience*, vol. 46, no. 9, pp. 1181–1200, 2016.

[39] L. Farzinvash and M. Kargahi, "A scheduling algorithm for execution-instant sensitive real-time systems," in *2009 15th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*. IEEE, 2009, pp. 511–518.

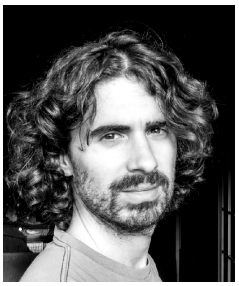
[40] N. Anand, E. Aryafar, and E. W. Knightly, "WARPlab: a flexible framework for rapid physical layer design," in *Proceedings of the 2010 ACM workshop on Wireless of the students, by the students, for the students*. ACM, 2010, pp. 53–56.

[41] "Rice University Wireless Open Access Research Platform," http://warpproject.org, last accessed January 30, 2020.

[42] S. Jang, K. G. Shin, and S. Bahk, "Post-CCA and reinforcement learning based bandwidth adaptation in 802.11ac networks," *IEEE Transactions on Mobile Computing*, vol. 17, no. 2, pp. 419–432, 2018.

[43] J. Lee, W. Kim, S.-J. Lee, D. Jo, J. Ryu, T. Kwon, and Y. Choi, "An experimental study on the capture effect in 802.11a networks," in *Proceedings of the second ACM international workshop on Wireless network testbeds, experimental evaluation and characterization*, 2007, pp. 19–26.

[44] Q. Qu, B. Li, M. Yang, Z. Yan, A. Yang, D.-J. Deng, and K.-C. Chen, "Survey and performance evaluation of the upcoming next generation WLAN standard - IEEE 802.11ax," *Mobile Networks and Applications*, vol. 24, no. 5, pp. 1461–1474, 2019.



Sergio Barrachina-Muñoz is a Ph.D. candidate at Universitat Pompeu Fabra (UPF). He received his B.Sc. in Telematics Engineering and M.Sc. in Intelligent Interactive Systems in 2015 and 2016, respectively, both from UPF. He joined the Wireless Networking Research Group in 2015 under the supervision of Dr. Boris Bellalta. His main research interests are focused on machine learning for next-generation WLANs. Sergio is a recipient of a FI grant from the Generalitat de Catalunya.

Boris Bellalta is an Associate Professor in the Department of Information and Communication Technologies (DTIC) at Universitat Pompeu Fabra (UPF). He obtained his Ph.D. in Information and Communication Technologies from UPF in 2007. His research interests are in the area of wireless networks, with emphasis on the design and performance evaluation of new architectures and protocols.

Edward Knightly (S'92–M'96–SM'04–F'09) received the B.S. degree from Auburn University and the M.S. and Ph.D. degrees from the University of California at Berkeley. He is currently the Sheafor-Lindsay Professor and the Chair of the Department of Electrical and Computer Engineering, Rice University. His research interests are in experimental wireless networking, urban-scale testbeds, 60-GHz, terahertz, and VLC bands, wireless security, and performance evaluation. He is a Sloan Fellow. He was a recipient of the NSF CAREER Award. He received the Best Paper Award from the ACM MobiCom, the ACM MobiHoc, and the IEEE SECON.