Aalto University
School of Science
Degree Programme in Mathematics and Operations Research

Antti Pöllänen

Optimizing Dense Wi-Fi Deployments Using Markov Chain Models and Simulated Annealing

Master's Thesis
Espoo, July 3, 2019

Supervisor:	Professor Lasse Leskelä
Advisor:	M.Sc. (Tech.) Janne Tervonen

Aalto University
School of Science
Degree Programme in Mathematics and Operations Research

ABSTRACT OF MASTER'S THESIS

Author:	Antti Pöllänen		
Title: Optimizing Dense Wi-Fi Deployments Using Markov Chain Models and Simulated Annealing			
Date:	July 3, 2019	**Pages:**	viii + 98
Major:	Mathematics	**Code:**	SCI3054
Supervisor:	Professor Lasse Leskelä		
Advisor:	M.Sc. (Tech.) Janne Tervonen		

Currently, the demand for wireless communication capacity is rising rapidly due to challenging applications such as video streaming and the emerging Internet of things. In meeting these ambitious requirements, the most important factor is predicted to be network densification, which refers to increasing the geographical density of simultaneously communicating devices. A natural choice for implementing dense networks is the wireless local area network technology Wi-Fi, characterized by being cheap and easy to deploy.

Network density aggravates the harmful effects of interference and causes scarcity of free transmission bandwidth. To counter this, dense networks need radio resource management algorithms.

This thesis presents a Wi-Fi radio resource management algorithm which jointly optimizes access point channels, user association and transmission power. It estimates future throughput using a continuous time Markov chain based model, and finds solutions maximizing this estimate via a discrete search metaheuristic called simulated annealing.

The algorithm is validated through a wide range of simulations where for instance network density is varied. The algorithm is found to be highly versatile, yielding good performance in all scenarios. Moreover, the general design approach places few restrictions on further algorithm improvement and extension. Markov chain modeling, although accurate in an idealized setting, turns out to be inaccurate with real-world Wi-Fi, with a simpler model offering similar accuracy but lighter computational load.

Keywords:	IEEE 802.11, WLAN, Wi-Fi, radio resource management, channel allocation, power control, user association, simulated annealing, continuous time Markov chain
Language:	English

Aalto-yliopisto
Perustieteiden korkeakoulu
Matematiikan ja systeemianalyysin koulutusohjelma

DIPLOMITYÖN TIIVISTELMÄ

Tekijä:	Antti Pöllänen		
Työn nimi:			
Tiheiden Wi-Fi verkkojen optimointi Markov-ketjumallien ja simuloidun jäähdytyksen avulla			
Päiväys:	3. heinäkuuta 2019	**Sivumäärä:**	viii + 98
Pääaine:	Matematiikka	**Koodi:**	SCI3054
Valvoja:	Professori Lasse Leskelä		
Ohjaaja:	Diplomi-insinööri Janne Tervonen		

Nykyisin vaatimukset langattoman tiedonsiirron kapasiteetille ovat voimakkaassa kasvussa johtuen haastavista sovelluksista kuten videon suoratoistosta ja tulossa olevasta esineiden Internetistä. Näiden vaatimusten täyttämiseksi tärkein keino on langattomien tiedonsiirtoverkkojen tihentäminen, mikä tarkoittaa yhtäaikaa samalla maantieteellisellä alueella kommunikoivien laitteiden määrän kasvattamista. Luonnollinen valinta tiheiden verkkojen toteuttamiseen on langattomien lähiverkkojen teknologia Wi-Fi, jonka etuja ovat edullisuus ja asennuksen helppous.

Langattoman verkon tiheys lisää haitallista interferenssiä ja aikaansaa pulaa vapaista lähetystaajuuksista. Näiden ongelmien ratkaisemiseksi tarvitaan radioresurssien hallinta-algoritmeja.

Tässä työssä suunnitellaan Wi-Fiä varten radioresurssien hallinta-algoritmi, joka optimoi samanaikaisesti tukiasemien kanavia, käyttäjien allokaatiota tukiasemille sekä lähetystehoja. Se estimoi tulevia tiedonsiirtonopeuksia jatkuvan ajan Markov-ketjuihin pohjautuvan mallin avulla ja löytää tämän estimaatin maksimoivia ratkaisuja hyödyntämällä diskreettiä hakumenetelmää nimeltä simuloitu jäähdytys.

Algoritmi validoidaan käyttäen monipuolista joukkoa simulaatioita, jossa vaihtelee esimerkiksi verkon tiheys. Algoritmi osoittautuu erittäin monipuoliseksi, sillä sen suorituskyky on hyvä kaikissa simulaatioskenaarioissa. Käytetyn lähestymistavan etuna on myös se, että se asettaa varsin vähän rajoituksia algoritmin jatkokehitykselle. Markov-ketjumallit osoittautuvat todellisen Wi-Fin tapauksessa epätarkoiksi, vaikka ne idealisoidussa ympäristössä ovatkin tarkkoja. Käy ilmi, että yksinkertaisemmalla mallilla saadaan vastaava tarkkuus, mutta laskentatehoa tarvitaan vähemmän.

Asiasanat:	IEEE 802.11, WLAN, Wi-Fi, radioresurssien hallinta, kanavanvalinta, lähetystehon säätö, tukiaseman valinta, simuloitu jäähdytys, jatkuvan ajan Markov-ketju
Kieli:	Englanti

Acknowledgments

This Master's thesis is based on work carried out at Nokia Bell Labs between spring 2017 and summer 2018. I am grateful for the resources provided, especially for the opportunity to use the Nokia proprietary Wi-Fi simulator Wenla.

I wish to thank Olli Alanen, my colleague at Nokia, for the work he did to support my thesis. He programmed the simulator interface that my algorithm uses. I am also grateful for the time he used to instruct me in using the simulator, and for the countless insightful conversations we had.

I also want to thank my supervisor professor Lasse Leskelä and advisor Janne Tervonen at Nokia for the invaluable support and guidance they have provided me during my thesis project.

There are many other people at Nokia who deserve my gratitude, including Lauri Kuru, Janne Aho, Marko Teittinen and Jari Mustajärvi for encouragement and insightful discussions, as well as Mads Brix for technical support with the simulator. Also, I want to thank professor Riku Jäntti at Aalto Department of Communications and Networking for his advice.

Finally, I would like to give special thanks to my family and friends for supporting me throughout my studies. You have made my time at Aalto University an experience I will remember with fondness.

Espoo, July 3, 2019

Antti Pöllänen

Abbreviations and acronyms

ACK	Acknowledgment Frame
AP	Access Point
CCA	Clear Channel Assessment
CDF	Cumulative Distribution Function
CTMC	Continuous Time Markov Chain
DCF	Distributed Coordination Function
DTMC	Discrete Time Markov Chain
FSPL	Free Space Path Loss
HCF	Hybrid Coordination Function
LCCS	Least Congested Channel Search
MAC	Medium Access Control layer
MCS	Modulation and Coding Scheme
MIMO	Multiple Input Multiple Output
MPDU	MAC Protocol Data Unit
MSDU	MAC Service Data Unit
NAV	Network Allocation Vector
OFDM	Orthogonal Frequency-Division Multiplexing
PCF	Point Coordination Function
PHY	Physical layer
RRM	Radio Resource Management
SIFS	Short Inter-Frame Space
SINR	Signal to Interference and Noise Ratio
SISO	Single Input Single Output
SIR	Signal to Interference Ratio
SNR	Signal to Noise Ratio
WLAN	Wireless Local Area Network

Mathematical notation

$\mathbb{N}$	Set of natural numbers $\{1, 2, 3, ...\}$
$\mathbb{N}_0$	Set of natural numbers augmented with zero $\{0, 1, 2, 3, ...\}$
$\mathbb{R}$	Set of real numbers
$\|X\|$	Number of elements in set X
$[n]$	For any $n \in \mathbb{N}$, the set $\{1, 2, ..., n\} \subseteq \mathbb{N}$
2^A	For a set A, the power set $\{X \mid X \subseteq A\}$
$\mathbb{P}(E)$	Probability of event E
$\mathbb{E}(X)$	Expected value of random variable X
$X \times Y$	Cartesian product $\{(x, y) \mid x \in X, y \in Y\}$
$\lfloor x \rfloor$	For $x \in \mathbb{R}$, the largest integer n such that $n \leq x$

Contents

Chapter 1

Introduction

In the current era, demands for wireless capacity are ever growing due to the increasing use of mobile hand held devices and challenging applications such as video streaming, online gaming and augmented reality. The emerging Internet of things may further dramatically increase the amount of communicating devices. Events with large masses of people, such as sports events, represent an especially challenging scenario. To satisfy these needs, the mobile industry is aiming to increase wireless capacity by a factor of 100x over the next 20 years [1], with the most ambitious visions targeting even an increase of 1000x [2].

In wireless communications, the means of increasing performance can roughly be divided into the three categories of enhancing *spatial reuse*, improving *spectral efficiency* and using more *bandwidth* [1]. This thesis focuses on the first approach, improved spatial reuse, which is predicted to be the main driver of future network capacity increase [1]. It is focused on improving system-wide performance, while the other two consider single-link performance.

Spatial reuse is defined as having multiple simultaneously communicating links on overlapping channels. Spatial reuse is closely associated with *network density*, which roughly refers to the density of links on overlapping channels, measured in number of links per square meter. Network capacity can be enhanced by increasing the number of simultaneous links while limiting the harmful effects of interference which network density naturally aggravates. Another drawback of network densification is increased network complexity and thus costs.

The second contributor to wireless performance, spectral efficiency, refers to the ratio of throughput and bandwidth usage for a single link, measured in bits/second/Hz. It depends on the signal processing techniques used, including modulation and coding schemes. For a single-input-single-output (SISO) link, it is upper bounded by the channel capacity given by the Shannon-Hartley theorem. Current technologies are already approaching this capacity, so no much additional performance is obtainable by improving spectral efficiency of SISO links. However, multiple-input-multiple-output (MIMO) schemes, including beamforming, offer much more room for improvement, and are currently a subject of active research and development. As a drawback, they require more complex signal processing techniques and more accurate knowledge of channel state information [1].

The last approach, increasing bandwidth usage, is naturally limited for lower frequencies (500-2600 MHz) by the fact that they are already to a large extent reserved by existing technologies. Higher bandwidths on the other hand are less crowded, but suffer from higher equipment cost and higher signal attenuation. The latter is both a drawback and an advantage: On the other hand, received signal strength is reduced, but on the other, dense networks naturally suffer less from attenuation due to short distances. They even benefit from it via attenuation reducing the interference from neighboring links (defined as transmitter-receiver pairs).

Multiple technologies could potentially contribute to achieving the 1000x capacity goal for wireless communication. This thesis studies Wi-Fi, a technology for wireless local area networks (WLANs). Compared to its alternatives (mainly 5G cellular networks that use small cells), it is cheap and easy to deploy, but provides less control over network operation, thus providing less means to ensure quality and fairness. Usual Wi-Fi networks consist of *access points* (APs) and *client devices* (abbreviated *clients*), where APs act as wireless hubs between the clients and wired Internet. Each client is *associated* to one AP at a time, and each AP and its associated clients transmit on some set range of frequencies, called the *channel*.

1.1 Problem statement and motivation

The goal of this thesis is to develop a radio resource management (RRM) algorithm for ultra dense Wi-Fi deployments, up to any density that is reasonable for practical applications. RRM in the context of Wi-Fi refers to managing transmission channels, user association (i.e. client association to APs) and transmission powers. RRM algorithms are vital for a dense network, since these RRM parameters determine the interference between transmitting links and thus the maximum extent of channel reuse.

Optimizing channel reuse resembles a packing problem: Very roughly speaking, we try to pack as many transmissions as possible in a space with two physical and one frequency dimension, with the constraint that transmissions on overlapping frequencies must not be too close to each other physically in order that mutual interference stays tolerable. In the time dimension, Wi-Fi protocols, instead of the RRM algorithm, determine how transmissions are allocated.

There is a need for new algorithm development, since most existing Wi-Fi RRM algorithms in the literature make assumptions that render them invalid for practical ultra dense networks: they for instance assume that every AP is always in operation or require currently infeasible modifications to hardware. Most existing algorithms are also to a large extent based on heuristics, and in experimental studies are only compared against rudimentary algorithms which are easily outperformed. Finally, they optimize each type of RRM parameter separately, whereas for optimal results, the parameters should be optimized together [3].

Thus we obtain the requirements for the algorithm developed in this work: Firstly, it must be suitable for ultra dense deployments. This means that RRM parameters must be tuned *dynamically*, i.e. as a response to the setup of clients currently needing service. Secondly, its hardware and standard requirements must be such that they could reasonably be satisfied in the near future by a commercial product. Thirdly, it must jointly optimize all RRM parameters which are: AP primary channels, channel widths, association of users to APs and transmit power levels.

The algorithm is designed to be run on a centralized WLAN controller that manages all the APs in the network, as opposed to be run on each AP in a distributed fashion. The motivation behind this choice is that the fact that all information is available in the same place simplifies algorithm design and expands the limits of what an algorithm can achieve. Also, many practical ultra-dense implementations will likely have a single owner and administrator, who may thus also manage the centralized controller.

The algorithm is developed for WLAN standard version 802.11ac since it is the most recent version that has found commercial use. The upcoming 802.11ax could also be considered, but it is omitted due to the following reasons: Firstly, there is little existing research on 802.11ax, which makes it considerably harder to study. Secondly, although there are some major differences between the standards, the results on 802.11ac will likely still be to a large extent applicable for 802.11ax. This is partially due to the fact that 802.11ax APs are required to be backwards compatible with legacy clients designed for previous standard versions. Thirdly, 802.11ac might remain commercially relevant in the near future due to only a gradual transition to 802.11ax.

1.2 Thesis contribution and organization

This work presents an algorithm, called the *dynamic algorithm*, that satisfies the requirements in the previous section and is based on the following building blocks: Firstly, it contains a mathematical model for estimating client throughput. A central part of it is a continuous time Markov chain based model for estimating the average fraction of time each device is transmitting. Secondly, it finds RRM parameters that optimize the estimated throughputs using simulated annealing, a search metaheuristic for discrete optimization problems. The optimum does not mean just maximizing total throughput, since fairness must be considered as well. Fairness means that there are no large discrepancies in the capacity that is allocated for each client.

The organization of the thesis is the following: The second chapter reviews how Wi-Fi operates. The mathematical model used by the algorithm is explained in the third chapter. The fourth chapter contains a literature review on existing RRM algorithms. The fifth chapter gives a precise description of the dynamic algorithm and a justification for its design choices. In the sixth chapter, the simulation are described. Finally, in the last chapter we present and discuss the simulation results, evaluate the methods used and conclude this work.

Chapter 2

Wi-Fi wireless networks

Wi-Fi is a technology for wireless local area networks (WLANs) that follows the IEEE 802.11 standard. The original 802.11 standard from 1997 is periodically updated by new amendments that introduce improvements and new features. They are denoted by lower-case letters such as 802.11e or 802.11ac.

To ensure interoperability, Wi-Fi products must additionally be tested and certified by the Wi-Fi Alliance, a global nonprofit industry association. Wi-Fi is characterized by ease of deployment and cost-efficiency, operating mostly in the 2.4 GHz Industrial, Scientific and Medical (ISM) frequency band and the three 5 GHz Unlicensed National Information Infrastructure (U-NII) bands. Unless indicated otherwise, the material in this chapter is based on [4] and [5], where the reader can find further information.

2.1 Basics of wireless communication over radio frequencies

Wireless communication in Wi-Fi is performed over radio frequencies, i.e. using electromagnetic waves in the radio frequency portion of the electromagnetic spectrum. A *carrier signal* is used to transmit the bits. The signal starts as an alternating current (AC) signal, and information, i.e. bits, are included in the signal via modulation techniques, including amplitude, frequency and phase modulation. The signal is then directed to a transmitting antenna, from which it is radiated as radio frequency electromagnetic waves. This signal is picked up by the receiver antenna, converting it to an AC signal, after which it is demodulated to obtain the original information bits.

Signal power is a crucial concept, as sufficient power at the receiver is required to correctly decode the message. Instead of the usual linear unit of power watts (W), the logarithmic unit of power decibel-milliwatts (dBm) is usually preferred. This is because there are huge differences in the magnitudes of power levels encountered in RF communication. If P_{mW} is the power in mW (milliwatts), then the power in dBm is obtained as

$$P_{\mathrm{dBm}} = 10 \log_{10} P_{\mathrm{mW}}, \tag{2.1}$$

or conversely,

$$P_{\mathrm{mW}} = 10^{\frac{P_{\mathrm{dBm}}}{10}}. \tag{2.2}$$

We say that the power level P is x dB larger than Q if $P_{\mathrm{dBm}} = x + Q_{\mathrm{dBm}}$, where the power levels were expressed using dBm. Equivalently, using mW (or any other linear unit) for power, we have $P_{\mathrm{mW}} = 10^{x/10} Q_{\mathrm{mW}}$. Note that power given in dBm expresses a value for a physical quantity, whereas dB is only used for comparison (at least in the context of wireless communications).

Signal power is lost between the transmitter and receiver, which is called *loss* or *attenuation*. Firstly, the signal suffers loss in the transmitter between the signal generator and the antenna due to imperfectness of the transmitter design. However, the antenna usually amplifies the signal, referred to as *antenna gain* (however, the gain might be negative as well). While propagating through the wireless medium, the signal suffers *free space path loss* (FSPL), given in dB by the formula

$$\mathrm{FSPL} = 32.44 + 20 \log_{10}(f) + 20 \log_{10}(D), \tag{2.3}$$

where f is the signal frequency in MHz and D is the distance in kilometers between the antennas.

In addition to FSPL, attenuation is caused by *absorption*, which refers to the signal losing power due to propagating through matter, for example walls, windows or water. For instance, the loss might be about 12 dB for a concrete wall and about 3 dB for a window. Air does not cause significant absorption in WLAN settings. The amount of absorption is larger for signals with a higher frequency, with the difference between the 2.4 GHz and 5 GHz bands being moderately significant in practice. Note that absorption is separate from FSPL: The frequency term in equation (2.3) results from antennas being less effective at receiving power from signals with higher frequency, not from absorption.

In addition to FSPL and absorption, signal propagation is affected by a number of phenomena, including reflection, scattering, refraction and diffraction. In most cases, these deteriorate signal quality. An exception is that reflection, via causing multipath propagation, actually benefits multiple-input-multiple-output (MIMO) transmissions, covered in section 2.2.

The variation of attenuation over time is called *fading*. It can be classified as either *fast fading* or *slow fading*. Roughly speaking, the former is fading which is fast enough for significant change to occur between measurement and the point of time where it actually affects the application in question. Slow fading is the opposite. This means that for fast fading there is not enough time to take individual values of the fading into account, and thus it can only be treated as a statistical phenomenon. On the contrary, individual values of slow fading can be measured and reacted to in time. Fast fading is for instance caused by rapid variation in multipath propagation conditions for moving receivers. Slow fading is for example caused by *shadowing*, i.e. objects moving in and out of the path between transmitter and receiver.

A major cause of interference is *noise*, which refers to ubiquitous background RF waves. The power of this noise is called the *noise floor*, and is usually about -100 dBm per a bandwidth of 20 MHz. However it may be higher in certain situations, for instance about -90 dBm in manufacturing plants due to the machinery producing additional noise. The noise floor tends to be lower in the 5 GHz band than in the 2.4 GHz band due to the former being less crowded.

Higher levels of noise make it more difficult for the receiver to correctly decode a transmission, while higher received power makes it easier. A good measure of signal quality is the *signal-to-noise ratio* (SNR), defined as the ratio between the received signal power and the noise power, when using watts, and as the corresponding difference when using dB. A SNR of at least 25 dB is usually considered good signal quality, while a SNR of less than 10 dB is considered poor signal quality.

Often the term *interference* is used as a contrast to the term noise to refer to interference which has a clear source and whose power is significantly higher than that of background noise. Potential sources include other radio equipment and microwave ovens. In this context the term *signal-to-interference-and-noise ratio* (SINR) refers to the ratio of transmitted signal power to the sum of the powers of interference and noise. However, the strength of the background noise is often minimal compared to that of the interference (when present), in which case examining the *signal-to-interference ratio* (SIR) suffices.

A fundamental challenge of RF communication is that even though all *communication links* (abbreviated as *links*, defined as transmitter-receiver pairs) are prone to error, oftentimes none can be tolerated. Even the flipping of a single bit may e.g. cause a program to crash. Luckily, measures exist to make the probability of error negligible. Firstly, a more robust, more error tolerant modulation method may be chosen. Secondly, error detection or correction codes are usually used. A particular choice of both a modulation and an error detection/correction coding technique is called a *modulation and coding scheme* (MCS).

2.2 Optimizing wireless performance

When designing wireless technologies, the aim is to maximize performance. A number of question arise: What exactly is performance in this context? How can it be increased? What are its fundamental limits? This subsection aims to provide some basic insight into these questions.

Perhaps the most common measure of wireless performance is *throughput*, which refers to the average amount of information (in bits) transmitted per unit of time (in seconds), while ensuring that the probability of error remains negligible. For a single communication link, in a theoretical context, the maximum value for this quantity is called *channel capacity* [6]. Channel capacity for a SISO single link, in a channel suffering from additive white gaussian noise, is obtained from the Shannon-Hartley theorem [6]

$$C = B \log_2 \left(\frac{S}{N} \right), \tag{2.4}$$

where C is the capacity in bits per second, B is bandwidth in hertz (Hz), S is the signal power in watts, and N is the noise power, also in watts.

Achieving channel capacity is in practice impossible, but it is possible to come relatively close to it with optimal signal processing, modulation and coding [7]. Increasing total throughput roughly linearly with bandwidth is fairly simple, so the essential task is to optimize *spectral efficiency*, i.e. throughput divided by bandwidth, measured in bits/(s · Hz). The goal is to achieve $\log_2\left(\frac{S}{N}\right)$ as in equation (2.4).

In addition to striving for achieving the capacity in equation (2.4), we may increase throughput by increasing the capacity itself. Firstly, we may increase the amount of bandwidth used and secondly, we may increase signal power by either increasing transmission power or antenna gain. However, limits exist for both in the form of regulations, imposed by different national organizations. These limits exist as in practice a plethora of devices compete for spectrum and may interfere with each other if using overlapping bandwidth and too large a transmission power. Also, the returns from increased received power diminish quickly as it grows larger relative to the noise power, due to the logarithm in equation (2.4).

The exposition has so far considered transmissions with a single transmitting and receiving antenna, called *single input single output* (SISO). However, significant improvement to performance can be obtained by using *multiple input multiple output* (MIMO) technology, which utilizes multiple transmitter and receiver antennas in combination with complex signal processing techniques. Equation (2.4) only gives an upper bound for SISO performance, and using MIMO link performance may be further increased.

The improvement is obtained by using multiple data streams, which in turn requires at least as many transmitter and receiver antennas as there are streams. Interference between the streams can be reduced via beamforming technologies. Contrary to SISO, in MIMO multipath propagation is a benefit, and in optimal multipath propagation conditions throughput improves linearly with the number of data streams.

On the level of a whole communication network that contains multiple links, performance may be measured as total network throughput. In addition to enhancing individual links, it can be improved by increasing the number of simultaneously communicating links. This is limited by interference between the transmitting links: Equation (2.4) may be modified to the form

$$C = B\log_2\left(\frac{S}{N+I}\right), \tag{2.5}$$

which takes into account I, the power of interference from neighboring links, in addition to the noise power N.

Interference attenuates with distance, so provided that the receivers are far enough from each other, multiple links may transmit on overlapping bandwidth. This is called *spatial reuse* or *channel reuse*. When is it efficient? Roughly speaking, the number of simultaneous transmitters can be increased linearly with available area, e.g. by just replicating a network design pattern. Thus to measure the efficiency of spatial reuse we should consider capacity per unit of area, or more precisely, *area spectral efficiency*. It is defined as throughput divided by bandwidth and area, and measured in bits/(s$\cdot$Hz$\cdot$m^2). Note that interference may be mitigated and area spectral efficiency improved by a propagation medium causing attenuation, for instance by office walls.

In addition to total throughput, goals for communication networks also include low *delay* and *fairness*: The former refers to the time it takes from queuing a transmission to the time it is finished, while a fair network is one where there are no large discrepancies between the performances offered for individual users.

2.3 The Internet protocol stack

Wi-Fi operates on the two lowest layers of the Internet protocol stack, which we will briefly cover in order to place Wi-Fi in a context and obtain necessary terminology. Internet uses a layered architecture, where protocols are organized hierarchically by layer with the purpose of a lower layer protocol providing services for (and only for) the layer directly above it (hence the name protocol stack). The benefit is modularization: one does not need to be aware of the complete Internet architecture when examining one layer, only of the interfaces to the layers directly above and below. These layers are, from the top to bottom: *application*, *transport*, *network*, *link* and the *physical layer*.

The physical layer provides the service of transferring a link layer *frame* (also called MPDU for *MAC Protocol Data Unit*) from one node to its neighbor over the physical medium, for instance the air, or over a cable. To achieve this, the physical layer protocol at the transmitter codes and modulates the bits of the frame to a signal and transmits it. The received signal is then demodulated and decoded by the physical layer at the receiver. Wi-Fi (i.e. IEEE Std. 802.11) defines several physical layer protocols corresponding to different radio technologies. As another example, Ethernet (i.e. IEEE Std. 802.3) defines a physical layer protocol for transmissions over wire.

The purpose of the link layer is to transfer a network layer *datagram* (also called MSDU for *MAC Service Data Unit*) from one node (host or router) to a neighbor. This is performed utilizing the physical layer, by encapsulating the datagram to a link layer frame, which additionally contains header information. The link layer is further divided into the upper logical link control (LLC) and lower medium access control (MAC) layers. The LLC layer may provide services such as multiplexing between different network protocols, flow control and automatic repeat requests. The MAC layer coordinates transmissions of multiple transmitters over a shared medium, as well as ensures reliable delivery of datagrams. Wi-Fi (as well as Ethernet) defines one MAC-layer protocol.

Wi-Fi does not operate on the upper layers, but for completeness, the network layer transports information from host (e.g. computer) to host, the transport layer transports information between applications, and the application layer consists of application specific communication protocols. For more information of the Internet protocol stack, the reader is referred to [8], which also acts as the source of the material in this section.

2.4 Wi-Fi protocol

The main component of a Wi-Fi network is a *station* (abbreviated STA in technical literature), which is simply any device with a radio that satisfies the requirements of the standard. A station may be either an *access point* (AP) or a *client station*, abbreviated hereafter as *client*. APs function as Ethernet hubs, gateways to the wider Internet, in addition to containing functionality related to controlling the network.

The IEEE 802.11 network topology is based around *service sets*, which are sets of stations communicating with each other. Four types exist, of which we cover the two most common. The *basic service set* (BSS) is used to implement wireless local area networks (WLANs). It consist of a single AP and an arbitrary number of clients that communicate with the Internet through the AP. We say that the clients are *associated to* their AP or (equivalently) BSS.

An extended basic service set (EBSS) consists of multiple basic service sets, usually with the APs being connected via a wired medium. The purpose of the EBSS is to provide seamless mobility for the clients between BSSes. The client switching the BSS/AP it is associated to is called a *handover*. The APs of an EBSS may be controlled by a common WLAN controller.

Each BSS operates on some *channel*, i.e. (mostly) continuous range of bandwidth. Available channel widths are 20, 40, 80 and 160 MHz. We review

Wi-Fi channelization in more detail in its own section at the end of the chapter.

Wi-Fi radios are *half-duplex*, meaning that they may either transmit or receive, but not both at the same time. This is an important feature that determines much of the design of the algorithms used by Wi-Fi.

2.4.1 PHY-layer

The physical layer of modern Wi-Fi uses two technologies: orthogonal frequency division multiplexing (OFDM) and direct-sequence spread spectrum (DSSS). OFDM is one of the most popular communication technologies, used both in wired and wireless communication, including for example LTE cellular networks. OFDM is a spread spectrum technology which enables a tight subcarrier packing by using mutually orthogonal subcarriers.

The Wi-Fi PHY-layer uses *automatic rate adaptation*, which is a means to try to find the optimal MCS for the current channel conditions. The specific algorithms are proprietary, but simply put, they tune the robustness of the MCS used based on the success or failure of past frame transmissions. If transmissions fail, a more robust MCS is adopted, and on the other hand MCSes with higher data rates are carefully tried if transmission errors are rare.

Starting from amendment 802.11n, Wi-Fi allows single user MIMO transmissions. Multi user MIMO is planned for the upcoming 802.11ax. In contrast to single user MIMO, it allows simultaneous transmission between more than two stations in one BSS, on the same channel.

2.4.2 MAC-layer

Wi-Fi MAC-layer protocol focuses on resolving medium contention between radios on overlapping channels. The two protocols in use are called the *distributed coordination function* (DCF) and the *hybrid coordination function* (HCF). We will first cover DCF as it is the fundamental technique of which HCF is an extension. The IEEE 802.11 also defines a third MAC-layer protocol, *point coordination function* (PCF), but it is not used in commercial products and therefore will not be covered here.

DCF is based on the CSMA/CA -protocol (carrier sense multiple access with collision avoidance). It is a channel access / multiple access method where only one radio transmits via the shared medium (channel) at one time, and radios contend for opportunities to transmit. *Collisions* (multiple radios transmitting simultaneously) are avoided via *carrier sensing*, i.e. radios listening for whether the channel is reserved.

With DCF, when a station wins contention, it is allowed to transmit one frame. The amount of information in the frame is constant, and thus the duration of the transmission depends on the rate (i.e. the MCS) used. In contrast, with HCF, the victor of the contention receives a TXOP (transmission opportunity), which means that the station has the opportunity to transmit as many frames as it has time for during a set amount of time, the TXOP length.

Wi-Fi stations use *half-duplex* radios, meaning that the stations can not transmit and receive simultaneously. This means that if a collision occurs, it is detected only after the transmission is complete. Therefore, DCF emphasizes collision avoidance (CA). As a contrast, IEEE 802.3 Ethernet wired networks use full-duplex communication, where simultaneous transmission and reception is possible. There, collisions may be immediately detected which results in significantly less wasted transmission time for each collision. Thus Ethernet networks use a different MAC-protocol, called CSMA/CD, where CD stands for collision detection.

Frames sent between radios are divided into *data frames*, *control frames* and *management frames*. Data frames are used for transmitting network level datagrams between radios. Management frames are used for the purpose of authentication and associating clients to BSSes/APs. They include for instance beacons sent by the AP to advertise its existence as well as association requests and confirmations sent between the client and the AP.

Control frames are used for controlling transmissions. The most important control frame is the ACK (acknowledgment) frame which the receiver sends to the transmitter upon successfully receiving a data frame, after first waiting for the duration of a SIFS (short inter frame space). If the transmitter does not receive an ACK, it attempts retransmission. Control frames also include the RTS (request-to-send)and CTS (clear-to-send) frames, explained later.

The DCF consists of multiple techniques, namely *inter frame spaces* (IFSes), a *binary exponential back-off counter* (BEB) and both *physical* and *virtual carrier sensing*. The different types of IFSes are mentioned when reviewing the situation where they are used. The BEB and carrier sensing will be covered next.

2.4.3 Carrier sensing

IEEE 802.11 networks use two parallel protocols for carrier sensing: physical layer carrier sensing, also known as *clear channel assessment* (CCA), and MAC-layer carrier sensing, also known as *virtual carrier sensing* or as the *network allocation vector* (NAV). We say that the channel is *sensed idle* when both CCA and the NAV indicate that the channel is idle. The purpose of carrier sensing is to enable other stations to stay silent when one transmits.

CCA consists of two different types of channel sensing while listening to the channel: Firstly, stations try to decode headers of incoming PPDUs (PLCP protocol data units, a type of frame used internally by the physical layer) in order to read, among others, the duration field, which describes the duration of the PPDU. If this is successful for a PPDU whose received signal strength is at least -82 dBm, CCA indicates a busy channel for the duration read. Secondly, irrespective of the nature of an incoming signal, CCA indicates a busy channel if the signal strength surpasses -62 dBm. This latter threshold is always used for non-Wi-Fi signals, and also for Wi-Fi signals in the case that decoding the PPDU header fails (usually due to interference).

Virtual carrier sensing is implemented via the NAV, which is a timer that counts towards zero. A station senses the channel busy when the counter has a positive value. For every successfully demodulated frame for which the station is not the intended recipient, the station sets its NAV to the duration value contained in the frame header. This duration encompasses the transmit time of the frame in addition to one SIFS (short inter frame space) and the duration of an ACK. It is important to note that virtual carrier sensing only functions for frames received with signal quality high enough for decoding. The required SINR level depends on the MCS that the received frame employs, i.e. on the data rate of the sensed transmission.

2.4.4 Binary exponential back-off counter

The *binary exponential back-off counter* (BEB) is a counter whose value is gradually decremented when the channel is sensed idle, and upon the counter reaching zero, transmission is initiated. In more detail, when a station wants to transmit a data frame, it first listens to the channel. After the channel has been sensed idle continuously for a duration of a distributed inter frame space (DIFS), the station initializes the BEB with a random initial non-negative integer value. If the value is initially zero, the station immediately starts its transmission. Otherwise, the station listens to the channel for a duration of a *slot time* at a time. The back-off counter is decremented at the end of each slot time during which the channel was sensed idle. If the counter reaches zero, the station begins transmitting its frame.

The purpose of the random back-off counter is to differentiate the stations so that only one may start transmitting at a certain time. Of course, the mechanism is not perfect since a collision may occur if multiple stations decrement their counter to zero during a single slot time.

The maximum value for the initial value of the exponential back-off counter is called the *contention window*, and it is obtained from the equation $2^x - 1$, where x starts at 0 and is incremented for each consecutive failed transmission. It is reset to zero upon a successful transmission. The initial value of the counter is uniformly distributed, with a minimum of zero. The exponential nature of the back-off counter may cause serious performance degradation in congested networks if frequent collisions cause the back-off times to grow very large.

2.4.5 Hidden and exposed node problems, RTS/CTS

Channel sensing in CSMA/CA is not perfect, because interference is measured at the transmitter instead of at the receiver where it actually matters. A *hidden node problem* refers to a situation where the transmitter transmits due to sensing the channel being idle, while the receiver senses the channel to be busy. This happens when the signal from an interferer is attenuated more on its path to the transmitter than to the receiver. This may result from differences in physical distance or attenuating obstacles between the interferer and transmitter. In the worst case this causes no data to be correctly decoded at the receiver.

The hidden interferer may prevent successful transmission in two ways [9]: Firstly, the SINR at the receiver may become too low due to the interference. Secondly, even if the SINR is sufficient, the receiver may be locked to listen to a frame from the interferer, in which case it does not receive its intended packet. To alleviate the second type of problem, there is a feature defined in IEEE 802.11 called *restart mode* (RS) [9]. Using it, a receiver currently receiving a frame may change to listen to a transmission whose received power is significantly higher. This prevents the second type of hidden node problem. However, as a drawback, the ACK frame from the receiver might corrupt the transmission of the interferer. For this reason, most commercial products do not use RS by default [9].

A dual problem of the hidden node problem is the *exposed node problem*. Here, the transmitter senses the channel to be busy when the interference at the receiver would in actuality be so much lower as to allow a successful transmission. This leads to unnecessary contention in the network and thus suboptimal network performance. However, the issue is much less severe than the hidden node problem.

RTS/CTS is a protocol in 802.11 for alleviating the hidden node problem. A station T that wants to transmit a data frame to R transmits a RTS (request-to-send) frame first, instead of the data frame. If the intended recipient R correctly receives the RTS frame, after a SIFS it responds with a CTS (clear-to-send) frame, upon reception of which (after a SIFS) T sends the actual data frame. Both the RTS and CTS frame set NAV timers of neighboring other stations to a value that encompasses the transmission of the data frame, in order to protect it from interference. Even if an interfering node is not in range to sense transmissions from T, it has likely received the CTS frame from R.

It is possible that the reception of the RTS frame at R fails due to interference. However, this is less likely to happen than in the case of T sending a data frame, as the RTS frame uses a more robust MCS. Moreover, the RTS frame is shorter than a data frame, so failed reception results in less wasted transmission time.

Another benefit of RTS/CTS is that it may reduce performance impairment from collisions, as it results in less wasted transmission per collision. Thus it is especially beneficial in dense crowded networks, where collisions are more prevalent [10].

The decision whether to use RTS/CTS for a frame is made based on frame length only. Each station has a length parameter defined, and frames surpassing this length are transmitted using RTS/CTS.

2.4.6 Wi-Fi channelization

The term *channel bonding* refers to using several 20 MHz channels simultaneously for transmission. The standard version 802.11n allows for two adjacent 20 MHz to be combined to one 40 MHz channel. Version 802.11ac allows even wider channels, up to 40, 80 or even 160 MHz. A 80 MHz channel is formed from two adjacent 40 MHz channels while the two 80 MHz channels constituting a 160 MHz channels need not be adjacent. Bonded channels of

80 MHz or 160 MHz are only available in the 5 GHz band, due to there being only 3 orthogonal 20 MHz channels in the 2.4 GHz band.

Any channel, independently of its width, has a constituent 20 MHz channel that is called the *primary channel.* The other constituent 20 MHz channels are called *secondary channels.* For any choice of a primary channel, the standard determines the secondary channels for any chosen channel width. The secondary channels are chosen such that two channels can not partially overlap: either there is interference on every 20 MHz subchannel of the narrower channel, or the channels are orthogonal. The purpose of this is to utilize bandwidth more efficiently, reducing interference in crowded networks. In the 5 GHz band, the number of 20 MHz channels varies based on country. A typical number however is 24.

Channel bonding complicates the rules of channel contention significantly. There are two modes of contention defined in the 802.11 standard: *static access* and *dynamic access* [11]. In both cases, a station uses the standard contention method on its primary channel. Before the back-off counter reaches zero, each secondary channel is sensed for a duration of a PIFS (PCF inter frame space). The two contention methods differ in regards to what occurs when the counter reaches zero: In static access, transmission is only initiated if all channels are sensed free, and all the subchannels are always used in a transmission. In dynamic access however, the transmission is always initiated, but the bandwidth used is only the maximum value (from 20, 40, 80 or 160 MHz) for which the corresponding subchannels are free.

Carrier sensing thresholds for power are dependent on channel width in such a way that the threshold in terms of spectral power density stays constant. This means that the threshold in power is increased by 3 dB for every time channel width is doubled.

Chapter 3

Mathematical models of Wi-Fi networks

This chapter covers the mathematical models that this thesis uses for modeling Wi-Fi networks. First however, we review preliminaries on graphs and Markov chains.

3.1 Preliminaries

3.1.1 Graphs

The notation and definitions in this section follow those of [12], in which the reader may also find further information. An *undirected graph* is an ordered pair $G = (V, E)$, where $E \subseteq \{\{a, b\} \mid a, b \in V\}$. The elements of the set V are called the *vertices* of the graph, and the elements of E the *edges* of the graph G. Most graphs in this text will be undirected, so we will refer to undirected graphs merely as *graphs*. Moreover, we assume that the graphs are finite, i.e. $|V| \in \mathbb{N}$, and that they contain no loops, defined as edges $\{v, v\} \in V$ from a vertex v to itself.

Two vertices $a, b \in V$ are called *adjacent* if $\{a, b\} \in E$, otherwise they are *independent.* A set $I \in V$ for which all vertices are pairwise independent is called an *independent set.* Similarly, a set $C \subseteq V$ for which all vertices are pairwise adjacent is called a *clique.* An independent set I (resp. a clique C) is called *maximal* if there is no vertex $v \in V \setminus I$ (resp. $v \in V \setminus C$) such that $I \cup \{v\}$ is an independent set (resp. $C \cup \{v\}$ is a clique). An independent set

I (resp. a clique C) is called *maximum* if there is no independent set $I' \subseteq V$ with $|I'| > |I|$ (resp. clique $C' \subseteq V$ with $|C'| > |C|$).

The *complement* $\bar{G}$ of graph $G = (V, E)$ is the graph

$$\bar{G} = (V, \{\{a, b\} \subseteq V \mid a \neq b\} \setminus E)\,. \tag{3.1}$$

We easily notice that a set $W \subseteq V$ is an independent set of G if and only if W is a clique of $\bar{G}$. Thus the maximal (resp. maximum) independent sets of G equal the maximal (resp. maximum) cliques of $\bar{G}$.

A graph $H = (V', E')$ is called a *subgraph* of $G = (V, E)$ if $V' \subseteq V$ and $E' \subseteq E$. The graph H is called the *induced subgraph* of G on V' if $E' = \{\{a, b\} \in E \mid a, b \in V'\}$.

The *distance* $d(x, y)$ between two vertices $x, y \in V(G)$ is defined as the shortest path between them, counted as the number of edges traversed. The *radius* of a graph is defined as $\min_{a \in V(G)} \max_{b \in V(G)} d(a, b)$. The *diameter* of a graph is defined as $\max_{a,b \in V(G)} d(a, b)$.

3.1.2 Continuous time Markov chains

Markov chains are stochastic processes which satisfy the *Markov property* which states that the future is only dependent on the past via the current state of the process. Markov chains can be either discrete or continuous. This exposition is based on [13], [14] and [15], where the reader can find more information and the proofs that are omitted.

Definition 3.1.1. A discrete time Markov chain (DTMC) with a finite state space S is a random sequence $(X_0, X_1, X_2, ...)$ such that for any $x, y \in S$, any $t \in \mathbb{N}_0$ and any event of the form $H_{t-} = \{X_0 = x_0, X_1 = x_1, X_{t-1} = x_{t-1}\}$,

$$\mathbb{P}(X_{t+1} = y \mid X_t = x, H_{t-}) = \mathbb{P}(X_{t+1} = y \mid X_t = x). \tag{3.2}$$

Definition 3.1.2. A continuous time Markov chain (CTMC) with a finite state space S is a random function X_t, $t \in \mathbb{R}, t \geq 0$ such that for any $x, y \in S$, any $t, h \geq 0$ and any event of the form $H_{t-} = \{X_{t_0} = x_0, X_{t_1} = x_1, ..., X_{t_n} = x_n\}$ such that $0 \leq t_0 < t_1 < ... < t_n$,

$$\mathbb{P}(X_{t+h} = y \mid X_t = x, H_{t-}) = \mathbb{P}(X_{t+h} = y \mid X_t = x). \tag{3.3}$$

The probabilities $\mathbb{P}(X_{t+1} = y \mid X_t = x)$ for the discrete and $\mathbb{P}(X_{t+h} = y \mid X_t = x)$ for the continuous case are called *transition probabilities.* If they do not depend on time t, the chain is called *time homogeneous.* In this case, we denote

$$P(x, y) = \mathbb{P}(X_{t+1} = y \mid X_t = x) = \mathbb{P}(X_1 = y \mid X_0 = x) \tag{3.4}$$

and

$$P_h(x, y) = \mathbb{P}(X_{t+h} = y \mid X_t = x) = \mathbb{P}(X_h = y \mid X_0 = x), \tag{3.5}$$

where P and P_h are called *transition matrices.*

All Markov chains in this text are assumed to be time homogeneous. Moreover, we assume that the state space S is always finite. For convenience, we take S to consist of the first $|S|$ natural numbers.

For both the discrete and continuous case, the *state distribution* of a chain X_t at time t is defined as the *row* vector μ_t for which $\mu_t(x) = \mathbb{P}(X_t = x)$ for every $x \in S$. Obviously now $\sum_{x \in S} \mu_t(x) = 1$ for every $t \geq 0$. The state distribution is obtained using the *initial distribution* μ_0 and the transition matrix as $\mu_t = \mu_0 P^t$ for the discrete and as $\mu_t = \mu_0 P_t$ for the continuous case. These are obtained by conditioning on the initial state, e.g. in the continuous case

$$\mu_t(y) = \sum_{x \in S} \mathbb{P}(X_0 = x)\mathbb{P}(X_t = y \mid X_0 = x) = (\mu_0 P_t)(y). \tag{3.6}$$

The CTMCs that we will use to model Wi-Fi networks belong to a very general class of CTMCs that we will now construct. This class contains all CTMCs that are right continuous (i.e. the outcome of X_t is always right continuous) and have a bounded jump rate (i.e. the CTMC always changes state a finite number of times in a finite time).

The chain is constructed as follows: It spends in the starting state $s_0 \in S$ an exponentially distributed time with rate parameter λ_{s_0}. Then it jumps to the next state s_i following a probability distribution $P(s_0, s_1)$ (where $P(s, s) = 0$ for all $s \in S$). Hereafter the chain continues this behavior indefinitely, spending an $\text{Exp}(\lambda_s)$-distributed time in state s, after which it jumps to the next state s' with probability $P(s, s')$. The Markov property follows from the memorylessness of the exponential distribution.

Let us define *transition rates* $Q(i, j)$ via $Q(i, j) = \lambda(i)P(i, j)$ when $i \neq j$, and $Q(i, j) = -\lambda(i)$ when $i = j$. Assuming the states in the state space S are the natural numbers from 1 to the size of S, the transition rates can be presented as a *generator matrix* Q, whose all rows sum to zero due to they way it is defined. Intuitively, the transition rate between states i and j (when $i \neq j$) describes the rate at which probability mass transfers from one state to another, relative to the probability mass currently at i.

Now, the behavior of the chain is characterized by the following proposition:

Proposition 3.1.1. *The time t transition matrix of a CTMC with generator matrix Q is given by*

$$P_t = e^{tQ}. \tag{3.7}$$

Moreover, we obtain the following equations, called Kolmogorov's backward and forward equations:

$$\frac{d}{dt}P_t = P_tQ \tag{3.8}$$

$$\frac{d}{dt}P_t = QP_t \tag{3.9}$$

Here, the matrix exponential notation e^A is defined by

$$e^A = \sum_{n=0}^{\infty} \frac{A^n}{n!}. \tag{3.10}$$

A DTMC is *connected* if for every pair of states $a, b \in S$ there exists $t \in \mathbb{N}$ such that $P^t(a, b) > 0$. Correspondingly, a CTMC is connected if for every pair of states $a, b \in S$ there exists $t \geq 0$ such that $P_t(a, b) > 0$. In other words, for both cases, this essentially means that for any pair of states a, b, it is possible that the chain, starting from a, eventually visits b as time progresses.

For a discrete Markov chain, a *stationary distribution* is a state distribution π such that $\pi = \pi P$, i.e. a state distribution that remains unchanged in time. For CTMCs, we analogously require that $\pi = \pi P_t$ for all $t \geq 0$, which is a much harder requirement to verify due to having to check over all transition matrices P_t. Consequently, we will here review an easier method for finding stationary distributions for CTMCs.

First however, consider a special case of stationary distributions called *limit distributions*. For both the discrete and continuous case, a limit distribution is a state distribution π satisfying $\pi = \lim_{t\to\infty} \mu_t$ where μ_t is the state distribution of the chain at time t. All limit distributions are also stationary distributions. A stationary distribution exists and is unique if the chain is connected. All CTMCs converge to some limit distribution, whereas the situation is slightly more complicated for DTMCs due to possible periodicity. If a CTMC is connected, it converges to a unique stationary distribution irrespective of its initial distribution. Proofs for these assertions can be found for instance in [13].

The limit distribution thus describes the long time average behavior of the chain, and is consequently useful for applications. The following proposition gives a method for finding it.

Proposition 3.1.2. *Assuming the generator matrix Q of the chain X_t is bounded, distribution π is a stationary distribution if and only if $\pi Q = 0$.*

As a special case, π is a stationary distribution if for every $i, j \in S$,

$$\pi(i)Q(i,j) = \pi(j)Q(j,i). \tag{3.11}$$

3.2 Modeling a full buffer Wi-Fi network

The goal of this section is to obtain a model for Wi-Fi throughput, to be used by the dynamic algorithm developed in this work. We consider a set of nodes $[S]$, where node may mean anything that can be considered a transmitter, i.e. the nodes may be all the stations, just the APs, or the set of BSSes (which makes sense because only one device in a BSS transmits at one time). A *full buffer* situation is modeled, i.e. every node always has something to send.

We denote the average throughput of node $i \in [S]$ by T_i, and it is obtained as

$$T_i = R_i x_i. \tag{3.12}$$

Here R_i is the data rate of node i when it is transmitting, and x_i is the long term average fraction of time it is transmitting, called *channel share* (also *normalized throughput* in the literature).

All traffic is assumed to be user data, i.e. the amount of control traffic is regarded insignificant. For simplicity we are also not taking into account collisions, which in reality reduce throughput depending on how congested the network is. They could be modeled for instance using the method in [16]. This would help make the model accurate even when the number of collisions rises dramatically due to large numbers of nodes contending with each other.

3.2.1 Data rate

For estimating data rate R_i, we use the Shannon-Hartley theorem (equation (2.5)). However the following simplifications are made: First, since the dynamic algorithm is designed for dense networks, the background noise N is ignored since it is assumed to be much smaller than the interference from other Wi-Fi nodes. Second, we assume that the density of the network causes the interference to be roughly equal to the channel sensing threshold of Wi-Fi. This is the maximum value that does not prevent transmission altogether.

We will for convenience write the formula using power spectral density (abbreviated *power density*) instead of power. Also, we will switch to using decibels for power density. Precisely, the unit for power density we use is such that power density D is

$$D = P - 3\log_2\left(\frac{B}{20}\right), \tag{3.13}$$

where power P is in dBm and bandwidth B in MHz. This means that for 20 MHz Wi-Fi channels, power and power density are equal, and every time channel width is doubled, power is increases by 3 dB relative to power density.

With the aforementioned adjustments, the rate R_i of transmitting node i is obtained as

$$R_i = B_i \log_2\left(1 + \sqrt{10}^{D_i - p(i,r(i)) - D_{\text{CST}}}\right), \tag{3.14}$$

where B_i is the bandwidth, D_i the transmission power density of node i, $p(i, r(i))$ the pathloss between node i and its receiver $r(i)$ and finally $D_{\text{CST}} = -82$ the channel sensing threshold given using power density. When given as a power density level, this threshold does not depend on bandwidth, as explained in section 2.4.6.

The choice to use Shannon capacity as an approximation relies on the assumption that it is up to a constant factor a tight upper bound for modern wireless SISO performance, which this work focuses on. A more accurate estimate could be obtained by using measurements to find a mapping from the SINR level to data rate. In any case, MIMO throughput is much harder to predict due to unpredictable multipath propagation characteristics [17].

3.2.2 Estimating channel share

To estimate channel share, a model based on CTMCs is used, roughly following the exposition in [18]. The use of CTMC models for CSMA/CA networks was first developed in [19] and adapted for IEEE 802.11 networks in [20], [21], [16], [22] and [23] among others. These also contain simulations and experimental studies that demonstrate the accuracy of the models.

Channel shares of nodes are primarily determined by MAC-layer channel contention. We model it via a *contention graph* $G = ([S], E)$, where the set of nodes $[S]$ acts as the set of vertices, and there is an edge $\{i, j\} \in E$ between nodes i and j if i and j are within carrier sensing range (physical or virtual) of each other and i and j are on overlapping channels. We assume this relation to be symmetric, even though it is not always so in reality.

We assume a node is always either *transmitting*, *contending* or *blocked*. This implies that each node has always something to send, i.e. a *full buffer* model is used. A node is blocked exactly when at least one of its neighbors in the contention graph G is transmitting. We describe the state of the network as the set of nodes that are currently transmitting. A state is *valid* if it contains no two nodes that are connected by an edge in the contention graph G. Denote the set of valid network states by $\Omega \subseteq 2^{[S]}$.

Transmission time in the context of this model means the length of the continuous period of transmission after a station wins contention and before it has to contend again. *Contention time* refers to the length of time between two transmissions of one station during which the station is decrementing its backoff, i.e. it is contending. This period of time is often not continuous because some other transmission interrupts the decrementing of the backoff.

To obtain the model, some additional idealizing assumptions are made:

1. There are no hidden nodes.

2. The backoff counter is continuous instead of discrete.
3. Channel sensing is instant, so no collisions occur.
4. The transmission and contention times of each node i are exponentially distributed with rate parameters μ_i and λ_i, respectively.

Now we show that this model yields a CTMC. First however, the following proposition is needed:

Proposition 3.2.1. *Let $X_1, X_2, ..., X_n$ be exponentially distributed random variables with rates $\lambda_1, \lambda_2, ..., \lambda_n$, respectively. Then the random variable $\min_{i\in[n]} X_i$ is exponentially distributed with rate $\sum_{i=1}^n \lambda_i$.*

Moreover, for any $i \in [n]$,

$$\mathbb{P}(X_i = \min_{j\in[n]} X_j) \;=\; \frac{\lambda_i}{\sum_{j=1}^n \lambda_j}. \tag{3.15}$$

The proof can be found for instance in section 2.1 of [13].

Note that due to the memorylessness of the exponential distribution, the backoff counters and remaining transmission durations can equivalently be re-initialized at any time. Let us assume that all the backoffs are initialized at the time of the n:th state transition (a transition happens when a transmission starts or ends), which results in state s. Denote the new values for the backoffs by the random variables X_i for each $i \in [S] \setminus s \setminus b$, where b is the set of blocked nodes. Denote the remaining transmission times by Y_i for each $i \in s$. Then the next state transition occurs after time $\min(\{X_i \mid i \in [S] \setminus s \setminus b\} \cup \{Y_i \mid i \in s\})$, which is according to proposition 3.2.1 exponentially distributed with rate parameter $\sum_{i\in[S]\setminus s\setminus b} \lambda_i + \sum_{i\in s} \mu_i$. The same proposition shows that the probability for the state change being due to node j changing transmission status is

$$\frac{\mu_j}{\sum_{i\in[S]\setminus s\setminus b} \lambda_i + \sum_{i\in s} \mu_i} \tag{3.16}$$

if node j is transmitting,

$$\frac{\lambda_j}{\sum_{i\in[S]\setminus s\setminus b} \lambda_i + \sum_{i\in s} \mu_i} \tag{3.17}$$

if it is contending and zero if it is blocked.

Thus we obtain a CTMC as constructed in section 3.1.2 with the following transition rates for when $s \neq s'$:

$$q(s, s') = \begin{cases} \lambda_i & \text{if } s' = s \cup \{i\} \text{ with } i \in [S] \text{ and } s' \in \Omega, \\ \mu_i & \text{if } s' = s \setminus \{i\} \text{ with } i \in s, \\ 0 & \text{otherwise.} \end{cases} \tag{3.18}$$

The chain has a stationary distribution π, where each state $s \in \Omega$ has the probability mass

$$\pi_s = \frac{\prod_{j \in s} \theta_j}{\sum_{s \in \Omega} \prod_{j \in s} \theta_j}, \tag{3.19}$$

where θ_i is the *activity ratio* of node i defined as $\theta_i = \lambda_i / \mu_i$. We can verify this using Proposition 3.1.2.

Since the chain is connected (every state is linked to the state $s = \emptyset$ where no node transmits), the stationary distribution is a unique limit distribution and thus gives the long term averages the network spends in each state.

Using this stationary distribution we may compute the channel share x_i for any node $i \in [S]$ as

$$x_i = \sum_{s \ni i} \pi_s = \frac{\sum_{s \ni i} \prod_{j \in s} \theta_j}{\sum_{s \in \Omega} \prod_{j \in s} \theta_j}. \tag{3.20}$$

Despite the idealizing assumptions made, this model for channel share has been demonstrated to be remarkably accurate for instance in [20] and [23]. Assuming exponentially distributed transmission and contention times does not seem to affect channel share. This result, called the *insensitivity result*, has been analytically derived in [20] which shows specifically that channel share does not depend on the distributions of transmission and contention times, given their means.

However, there is uncertainty whether some of the other assumptions make the model deviate from reality, since the simulators used for verification themselves make some of the same idealizing assumptions that the model makes. In particular, consider the assumption that channel contention is defined by a contention graph G that stays constant over time. One reason this does not hold in reality is fast fading, which rapidly changes received power. This is taken into account in many simulators. However, another deviation from the assumption stems from the fact that Wi-Fi has two physical carrier sensing thresholds, -82 and -62 dBm, of which the former is only used when the header of the received frame is successfully decoded (see section 2.4.3). Virtual carrier sensing also depends on whether the received transmission could be decoded. In this work, we will use a simulator that models Wi-Fi physical and virtual carrier sensing properly, and we will notice that assuming only one threshold -82 dBm makes the model significantly deviate from the more accurate simulation.

3.2.2.1 Obtaining model parameters

The following is a description on how to obtain model parameters intended to match real-world Wi-Fi.

In the contention graph G, we take there being and edge between nodes i and j if the nodes transmit on overlapping channels and the average received power in either node when the other is transmitting is above the channel sensing threshold -82 dBm. If a node is taken to be a whole BSS, we consider whether the received powers of AP-AP transmissions, with the idea that traffic is mostly downlink, i.e. the AP is transmitting.

We first derive the rates for contention and transmission times when nodes are single stations. In the model, the times are exponentially distributed, so their rate parameters are obtained as inverses of their averages. Justified by the insensitivity result, the model averages are set to match real averages. Thus the rate for contention time of node i is obtained as

$$\lambda_i = \frac{1}{E(T_{\text{contention}})} = \frac{1}{T_{\text{slot}} \cdot E(B_{\text{init}})} = \frac{2}{T_{\text{slot}} \cdot \text{cw}}, \tag{3.21}$$

where $T_{\text{contention}}$ is a random variable for real contention time, T_{slot} is slot time, B_{init} is the initial value of the backoff (an uniformly distributed random variable) and cw is the contention window, i.e. maximum value for B_{init}. As cw we use the minimum contention window value, i.e. the one that has not been increased due to collisions.

We model modern Wi-Fi that uses HCF, so transmission length is the same as TXOP length. Thus the rate parameter for transmission time of node i is obtained simply as

$$\mu_i = \frac{1}{E(T_{\text{transmission}})} = \frac{1}{T_{\text{TXOP}}} \tag{3.22}$$

where T_{TXOP} is TXOP length.

When nodes are whole BSSes, the rate for transmission time is obtained in the same way, since transmission time stays the same. This is assuming that the TXOP length is the same for all stations in the BSS.

The rate of the contention time of a BSS can be estimated by two simple methods. The first is to continue with the assumption that all stations, both the AP and clients, are always saturated. In this case proposition 3.2.1 implies that the contention time of the BSS is exponentially distributed with a rate parameter that is the sum of the rates of the contention times of the individual stations. The second is to assume that always only one station in the BSS is saturated at a time, and that this station is mostly the AP. In this case, we can approximate that the rate of contention of the BSS is the same as that of its AP. In this work the latter method will be used.

3.2.3 Further approximations for channel share

Calculating channel share from equation (3.20) has an exponential time complexity since finding all independent sets of a graph is an NP-complete problem [24]. Thus with increasing network size we quickly encounter a situation where calculating channel share from equation (3.20) is no longer feasible. In order to increase the size of the network for which these calculations can be performed, we cover some further approximation techniques that reduce computational cost.

The simplest of these methods is to reduce the number of nodes by defining nodes to be BSSes instead of individual stations. However, exponential time complexity still remains.

Another method is obtained in [20] by noting that in general contention times are short compared to transmission times, which implies that the activity ratio θ_i of each node i is a large number. Denote $\theta = \min_{i\in[S]}\{\theta_i\}$ and write $\theta_i = \alpha_i\theta$ for every $i \in [S]$. Note that $\max_{s\in\Omega}|s| = m$ for an m that is the size of a maximum independent set of G. Now the channel shares x_i are approximated by their limits at $\theta \to \infty$ as

$$\begin{aligned} x_i \approx \lim_{\theta\to\infty} x_i &= \lim_{\theta\to\infty} \frac{\sum_{s\ni i}\prod_{j\in s}\alpha_j\theta}{\sum_{s\in\Omega}\prod_{j\in s}\alpha_j\theta} \\ &= \lim_{\theta\to\infty} \frac{\sum_{s\ni i}\theta^{|s|-m}\prod_{j\in s}\alpha_j}{\sum_{s\in\Omega}\theta^{|s|-m}\prod_{j\in s}\alpha_j} \\ &= \frac{\sum_{s\ni i,|s|=m}\prod_{j\in s}\alpha_j}{\sum_{s\in\Omega,|s|=m}\prod_{j\in s}\alpha_j} \end{aligned} \tag{3.23}$$

In the case that the backoff rates λ_i are all equal (implying that $\alpha_i = 1$ for every BSS b_i), we obtain that

$$x_i \approx \lim_{\theta\to\infty} x_i = \frac{|\{s \in \Omega : i \in s, |s| = m\}|}{|\{s \in \Omega : |s| = m\}|} \tag{3.24}$$

This means that the channel share of node i is the fraction of maximum independent sets of G that i belongs to. Finding all maximum independent sets is still an NP-complete problem [24], so the approximation just somewhat increases the network size limit for which computations remain feasible.

This approximation technique should only be used for at most moderately sized graphs. For large graphs, the number of independent sets of size $m-1$ might be so many times larger than those of size m that in equation (3.20) the former matter too despite large activity ratios θ_i.

To obtain proper scalability, channel share can be approximated locally. Following [25], this is done by calculating the channel share for a node i on a local graph $G_i(s)$ centered on node i. The graph $G_i(s)$ is the induced subgraph of all nodes whose distance to i is at most $s+1$. In addition, there is an edge between every pair of nodes for which both have the distance $s+1$ to i. The parameter s is called *span*, and by adjusting it we can control the trade-off between accuracy and computational cost. Reference [25] demonstrates that increasing span indeed increases average accuracy. However, the variance in error is significant, and one can even for a large span easily construct scenarios where a node gets close to zero channel share without local approximation but with it we mistakenly obtain a channel share of at least 0.5.

Chapter 4

Literature review

In this literature review, we review existing radio resource management algorithms and assess their viability for ultra-dense Wi-Fi networks.

4.1 RRM algorithms

Radio resource algorithms have been studied extensively in the literature. Thus we will not attempt to exhaustively review existing research, but present a few examples of state-of-the-art algorithms of each type, i.e. algorithms managing channels, user association or transmit power. Many of the algorithms in the literature only manage one type of parameter at a time, but there are some combined approaches as well. In what follows, the algorithms are grouped by the combination of parameter types they manage.

In this section *channel assignment* means assigning channels which all have the same width. When variable widths are considered, we speak about *channel bonding.*

Channel assignment

A basic static channel assignment scheme is suggested in [26]. It is based on vertex coloring: It constructs a graph that has the APs as nodes and edges between the APs that can channel sense each other. With channels corresponding to colors, it finds a coloring for the graph using a minimum amount of colors (in a coloring no two adjacent vertices may have the same color). The problem is NP-complete, but efficient heuristics are presented for almost optimal solutions.

Also, an extended algorithm that minimizes partial overlap in the 2.4 GHz band is given for the case when there are not enough non-overlapping channels present to reach a coloring. Both algorithms are *centralized* in the sense that global knowledge (i.e. the structure of the whole interference graph) is required to compute a channel even for a single AP.

In *distributed* schemes each AP runs computations independently based on only locally available information. A simple but commonly used example is *least congested channel search* (LCCS): each AP measures activity on each channel, and chooses the channel that is the least loaded. Load can be measured e.g. by the number of active client devices, or the amount of frames sent during a certain period of time [27].

A more advanced distributed algorithm in [28] minimizes the sum of interference (in watts) over all interferers and APs. This yields a utility function whose form is suitable for a *Gibbs sampler* to be used for optimization. Roughly, a Gibbs sampler is a stochastic optimization method where nodes (APs) change parameters randomly over time preferring values that correspond to high local utility. With a local utility function of the correct form, maximizing it also leads to maximization of the global utility that we actually are interested in.

User association

A basic user association scheme commonly used in commercial applications is that clients simply associate to the AP from which the received signal strength is the highest. This has the problem that clients may be concentrated on a few APs in a suboptimal manner.

A centralized graph-based algorithm for user association in [29] maximizes throughput using max-min fairness, taking into account upper limits for throughput demands of clients. It finds a fractional optimal solution by iteratively applying a linear program to find association for the "worst" client. An integral solution can be found which is at most a constant factor away from the optimal fractional one.

A distributed algorithm for user association is proposed in [28]. It minimizes the inverses of estimated full buffer throughputs summed over all clients. The utility function obtained is of suitable form for a Gibbs sampler to be used for the optimization process.

Channel assignment and user association

In [30], a centralized approach is suggested to simultaneously optimize both channel assignment and user association. The scheme minimizes the number of client devices each client conflicts with, summed over the clients. In its model of network topology, it takes into account clients in addition to APs. A benefit of this is for instance that neighboring APs do not need to be assigned different channels if the other AP has no clients. The approach not only considers AP-AP conflicts, but conflicts between AP and client or just between clients as well.

For optimization of channels, the algorithm uses a variant of the search metaheuristic *hill climb*. A hill climb search starts with some initial solution, chosen e.g. by random. Then, the solution is gradually improved by iteratively making small changes of some clearly defined type. In the algorithm of [30], such a change means changing the channel of a single AP. The algorithm never makes clients perform handovers and initially associates them with the AP that is in conflict with a minimum number of devices.

Reference [31] presents a centralized dynamic algorithm designed specifically for dense networks. The algorithm uses the performance metric *available capacity*, which is the product of expected data rate and free air time, i.e. the fraction of time the channel is idle. These are approximated directly based on data a WLAN controller obtains from APs. In order to minimize disruptions from channel and association changes, the algorithm determines a channel for an AP only when it goes active by obtaining its first client. The AP and channel chosen are the ones which give most estimated available capacity to the client. Client re-association is only performed when an AP becomes really loaded. Finally, the algorithm is specifically designed not to require modifications in legacy clients. The paper concludes that power control with variable power levels is not worth it, and maximum power should always be used.

Power control

Power control in CSMA/CA networks is challenging due to variable transmit powers easily leading to unfair MAC-layer contention due to one station carrier sensing the transmission of the other, but the other not sensing transmissions from the first one [32] [31]. The authors in [31] state that they have experimentally tested a power control algorithm with variable transmit powers and concluded that it was detrimental for network performance.

Reference [32] suggests an approach of simultaneously tuning carrier sensing thresholds so that carrier sensing remains symmetric and unfairness is avoided. They present a distributed algorithm for this purpose which optimizes inverses of average throughputs summed over the clients. This allows formulating an utility function in a form which allows the use of a Gibbs sampler for distributed optimization.

Channel assignment, user association and power control

The paper [33] criticizes earlier work with Wi-Fi RRM algorithms for failing to address the unreliability of propagation models and requiring modifications to clients, and presents a dynamic, centralized algorithm without these shortcomings. The algorithm is based on maximizing total utility, which is a weighted sum of individual utilities corresponding to different network characteristics, namely client throughput and interference. Optimization is based around a variant of hill climb search.

Channel assignment and channel bonding

In [18] an algorithm is presented that simultaneously considers primary channel assignment and channel bonding. It uses graph coloring similarly to [26] to find a set of channels so that preferably no APs conflict with each other. If more bandwidth than 20 MHz can be used without conflict, it is distributed in a proportional fair manner among the color classes, inside which each AP has the same primary channel and same channel width.

Channel assignment, channel bonding and user association

A centralized algorithm is presented in [34] that optimizes channel assignment, bandwidth and user association. It relies on APs estimating channel access time and transmission delay with user association being determined only when users join. Then, an AP is chosen such that approximated total throughput is maximized. For choosing AP channels and bandwidths, the algorithm uses steepest descent hill climb.

4.2 Evaluation of the algorithms

None of the reviewed algorithms satisfy all the requirements set for an RRM algorithm in this thesis. All the reviewed algorithms have the drawback that they lead to RRM parameters being optimized separately instead of together. Most of them require knowledge of values for RRM parameters that they do not manage. If other algorithms are used to manage the rest of the parameters, iterative optimization results, where the algorithms are run in turns until convergence. However, there are no guarantees that the local optimum found is optimal globally, studied for instance in [3]. The performance of the algorithms is only verified by simulations that compare them to simple alternatives whose performance should be easy to beat.

Many of the reviewed algorithms assume that all APs are in use. These include the algorithms in [26], [28] and [18] as well as LCCS. Many of the algorithms are also based on heuristics, lacking mathematical justification. These include the algorithms in [26], [28], [30] and [33].

The most promising seem to be the algorithms in [31] and [34]. Their only clear drawback is that they do not optimize all of the RRM parameters together, with no obvious way to extend the algorithms to do that.

Chapter 5

Dynamic RRM algorithm

This chapter describes the dynamic RRM algorithm developed in this thesis, called simply the *dynamic algorithm*, as well as explains the reasoning behind its design choices.

5.1 General setting

Although the algorithm is only implemented for a simulation, this implementation is based on a view on how the algorithm would operate in a real-world setting, and what restrictions this scenario imposes on algorithm design.

Physically, the algorithm would be run either on a local WLAN controller or in the cloud. APs would be connected to the controller or Internet via Ethernet cable or equivalent. We assume that this allows plenty of throughput and a relatively low latency for the control traffic between the APs and the device on which the algorithm is run.

In the case of a local controller, the computational resources available for the algorithm would be of the scale of a basic desktop computer. If the algorithm was run in the cloud, more resources could be available, but the former is still a reasonable goal as we would like to save costs.

We assume that the APs measure mutual pathloss, as well as pathloss between clients and APs. How to practically implement these measurements is to a large extent left out of the scope of this work. However, we note that pathloss may be inferred from the received power level of any packet when the transmit power level is known. Thus pathloss between APs could be measured either from regularly sent beacon frames or from testing frames sent specifically for this purpose at times when the network has only little load. The hardware and software modifications required for these measurements seem commercially viable as they only concern the APs that are managed by the network administrator.

However, measuring pathloss between client and AP is more complicated, as usually the network would need to be able to serve all kinds of client devices. Working only with features included in the current IEEE 802.11ac standard, prompting a client to send testing frames at desired times should be possible via RTS/CTS exchanges or by simply sending any data frame to the client and measuring the received power of the ACK frame the client responds with. However, inferring pathloss would additionally require knowledge of the transmit power level of the client, which is a feature planned for future 802.11 amendments.

These measurements would have to be repeated periodically since the pathlosses change due to fast and slow fading. It is left as an open question as to how often or in what situations measurements should be performed, and how consecutive measurements should be averaged. Instead, in the simulations, the pathlosses without taking into account fast fading (i.e. averages with fast fading) are idealistically constant and precisely known.

The algorithm manages BSS primary channels, channel bandwidths, transmit powers and user association. With a *full run* we refer to an algorithm run that may alter any of these parameters in the network, consisting of the WLAN controller obtaining the most recent measurement results, performing the algorithm computations, and implementing the resulting network changes in the APs. Such a run could be performed either regularly or as a reaction to performance degradation. In this work, the former option is chosen for simplicity.

The algorithm computations in a full run consist of running one or more simulated annealing searches to find a network setup that maximizes a utility function built around full buffer throughput estimated using the model in section 3.2. Possible stopping criteria for the search are real time used, CPU time used and the number of search moves made. This work uses the last option in order to eliminate random change between simulation runs.

We use full buffer models, i.e. more precisely the model assumes that at all time instants, some station in each BSS has something in its buffer to send. These models are used because they offer simplicity and closed form throughput formulas which are needed for fast enough utility computations.

The drawback is that the full buffer assumption might not in reality hold. However, we could firstly try to circumvent this by re-evaluating with fairly short intervals whether the client is "active", i.e. requests enough capacity to be considered full-buffer. In the negative case, the client would be invisible to the model, even if it is technically associated to some BSS in the network. We leave the details of this as an open problem. A second method is suggested in [35], where a network where clients download large files using TCP is modeled via an equivalent full-buffer model. As a bottom line, the full buffer assumption should work decently because if a network is optimized to handle a maximally congested scenario, then it should perform at least satisfactorily in less congested cases.

In addition to full runs that may take a non-negligible amount of time, we would like the controller to be able to quickly respond to new clients joining the network by recommending an initial AP for them. In this case, the other network parameters are not optimized, but the algorithm simply recommends to the client an AP so that total network utility is maximized.

It is worth noting that the 802.11 standard does not currently include any mandatory feature for enabling the network to manage client association. Amendment 802.11k contains a feature where the client may request a *neighbor report* from its AP which ranks APs neighboring the client from best to worst [4]. However, most clients do not yet support this feature and the client anyways makes the final decisions on potential handovers [4]. In addition, [31] contains methods of managing association using only mandatory features. These are based on only exposing the desired AP to each client by setting SSID fields to null in beacon frames from non-desired APs and only responding to client probe requests from the desired AP. Further possibilities of managing association may become available with further amendments of the 802.11 standard. In any case, the dynamic algorithm can naturally be implemented so that managing the network otherwise is not prevented even though the association of some clients could not be managed. This happens simply by removing the BSS of those clients from the list of optimization variables that simulated annealing searches over.

Finally, the model that the dynamic algorithm uses assumes that traffic is downlink only, i.e. that the transmitter is always an AP. This assumption is done for simplicity, with the justification that for a regular mobile Internet user most traffic is downlink.

5.2 Algorithm design considerations

In this section, we will discuss some critical matters related to RRM algorithm design, related to power control and fairness.

5.2.1 Power control

As discussed in [32], variable transmit powers easily lead to asymmetric channel contention in the network. This in turn leads to significant unfairness, which we want to avoid.

When considering channels with multiple widths simultaneously, we note that variable *transmit power spectral density* levels is what more precisely speaking causes asymmetry. This is because in Wi-Fi, channel sensing thresholds for power are adjusted with channel width such that the threshold for power density remains constant.

Thus in order to avoid node starvation from asymmetric channel contention, the dynamic algorithm uses a constant transmit power density for all devices. In principle, this level could be different for different orthogonal channels, or for disconnected parts of the network, while still causing no asymmetry. However, the simplicity of having just one power density level for the whole network is preferred.

A potential problem arises from the fact that national regulations set an upper limit specifically on the transmission power for Wi-Fi devices, not on power density. We circumvent this by allowing channel width only to be set to values that keep total transmission power under the limit.

The network can set the transmit powers of clients by using the transmit power control (TPC) feature introduced in amendment 802.11h for the 5 GHz band and in 802.11k for both bands [4]. Of course, currently not all client devices are 802.11h or 802.11k capable, which in practice could prevent enforcing equal transmit power density.

5.2.2 Fairness

Simply maximizing total network throughput is not feasible, since it may leave some clients with barely any service at all. In fact, this easily happens in Wi-Fi for instance due to hidden node problems. Also, in the CTMC model (equation (3.20)) we notice that it easily occurs that some node gets barely any throughput because it is not contained in any maximum independent set. Thus *fairness* needs to be ensured, i.e. that there are no large differences between the throughputs offered to different clients.

The following three types of fairness commonly appear in the literature [36]: *Proportional fairness*, *max-min fairness* and minimization of *potential delay*. For this section, let C be the set of clients and let t_c be the throughput of client $c \in C$. In proportional fairness, we maximize the quantity $\sum_{c \in C} \log(t_c)$, which is equivalent to maximizing $\prod_{c \in C} t_c$. Max-min fairness simply maximizes the worst throughput any client obtains, i.e. the function $\min_{c \in C} t_C$. Potential delay is the inverse of throughput, so we minimize the function $\sum_{c \in C} 1/t_c$.

Of these, proportional fairness is used in this work, with minimization of potential delay being another good option. We consider max-min fairness to be too strict since it is only concerned with the performance of the worst client. This is a clear drawback in a situation where bad service for one client is unavoidable, in which situation we would still like to optimize the service for the other clients.

Table 5.1: Parameters defining the problem instance.

S	Number of stations
$[S]$	Set of all stations
A	Number of APs ($A \leq S$)
$[A]$	Set of all APs ($[A] \subseteq [S]$)
C	Number of available primary channels
$[C]$	Set of available primary channels
$P_{\max}$	Maximum allowed power level, in dBm
$d(i, j)$	path loss between stations i and j
$\mathcal{C}' = (c'_1, .., c'_A)$	Existing allocation of primary channels
$\mathcal{W}' = (w'_1, .., w'_A)$	Existing allocation of channel widths
$\mathcal{A}' = (a'_{A+1}, ..., a'_S)$	APs each client is currently associated to
d'	Current transmit power density level used by all stations

Table 5.2: Optimization variables.

$\mathcal{C} = (c_1, .., c_A)$	Allocation of primary channels
$\mathcal{W} = (w_1, .., w_A)$	Allocation of channel widths
$\mathcal{A} = (a_{A+1}, ..., a_S)$	APs each client is associated to
d	Transmit power density level used by all stations

Table 5.3: Other mathematical symbols used.

p_i	Transmit power of AP i
T_i	Throughput of station i
U	Utility of the current network setup

5.3 Formulation as an optimization problem

Let us give a precise mathematical description of the optimization problem the algorithm needs to solve. The Tables 5.1, 5.2 and 5.3 list the parameters defining a problem instance, the optimization variables and other mathematical symbols used, respectively. All the symbols in Table 5.3 are actually functions of the parameters and optimization variables, but function notation is omitted for notational simplicity.

5.3.1 Notation and conventions

Let $[C]$ denote the available primary channels, where C is their number. This is, the primary channels are assumed to be consecutive. Let $[A]$ be the set of APs and A their number. Similarly, denote the set of stations by $[S]$, where S is the number of them. We assume that the first A stations are the APs.

We denote the global allocation of primary channels with $\mathcal{C} = (c_1, .., c_A) \in [C]^A$, where c_i is the channel of AP i. Similarly, $\mathcal{W} = (w_1, .., w_A) \in \{1, 2, 4, 8\}^A$ denotes the channel widths of each AP, where the number indicates the amount of bonded 20 MHz channels. The primary channel and channel width suffice to determine which primary channels are bonded: for primary channel c_i and channel width w_i of AP i, the range of 20 MHz channels used is

$$R_i = \left\{ \left\lfloor \frac{c_i - 1}{w_i} \right\rfloor w_i + 1, \ldots, \left\lfloor \frac{c_i - 1}{w_i} \right\rfloor w_i + w_i \right\} \tag{5.1}$$

when these channels exist, otherwise the channel width in question is not available. The actual channelization in Wi-Fi is different from this in that a 160 MHz channel may consist of two 80 MHz channels that are not consecutive, however this is overlooked here for simplicity.

For any station $i \in [S]$, let a_i denote the AP of the BSS the station i belongs to. Denote the scheme by which clients are associated to APs by $\mathcal{A} = (a_{A+1}, ..., a_S)$.

The primary channel allocation $\mathcal{C}$, bandwidth allocation $\mathcal{W}$, association scheme $\mathcal{A}$ and power density level d together form the set of optimization variables for the problem.

Changing the network parameters ($\mathcal{C}$, $\mathcal{W}$, $\mathcal{A}$, d) might cause impairment to network performance. This is why we include the existing set of network parameters $\mathcal{C}' = (c'_1, .., c'_A)$, $\mathcal{W}' = (w'_1, .., w'_A)$, $\mathring{A}' = (a'_{A+1}, ..., a'_S)$ and d' as problem instance defining parameters.

The transmit power level p_i of an AP i is obtained as

$$p_i = d + 3 \log_2 (w_i) \,. \tag{5.2}$$

5.3.2 Utility function

We choose to optimize the throughputs of stations assuming a full buffer downlink only traffic model. These throughputs, denoted by T_i for client i are obtained from equation (3.12). The channel shares π_i in it are obtained from estimates in section 3.2.2 and 3.2.3. We model channel share on the level of BSSes, e.g. take the nodes in the model to be BSSes. Most simulations use the maximum independent set approximation, obtaining channel share from equation (3.24), and do not use the local approximation technique of section 3.2.3. An exception are the simulations where these are purposely varied in order for us to observe the effects.

To obtain an utility function, we need to combine the throughputs T_i of individual clients into a single-valued function. We do this via proportional fairness (described in section 5.2.2) to obtain

$$U = \sum_{i \in [S] \backslash [A]} \log(T_i) - I_c N_c - I_a N_a - I_p N_p, \tag{5.3}$$

where the last three terms describe the penalty resulting from making changes to the network configuration. The functions N_c, N_a and N_p give the numbers of clients who are affected by a channel, association or power change, respectively. The client i is affected by an association change if $a_i \neq a'_i$. The penalty from a channel change occurs if the client is not affected by an association change and if $c_{a_i} \neq c'_{a_i}$ or $w_{a_i} \neq w'_{a_i}$. The client suffers from a transmit power change if it is not affected by the previous two types of changes and $p'_{a_i} \neq p_{a_i}$. The constants I_c, I_a and I_p reflect the severity of the performance impairment resulting from each type of configuration change.

The form of the utility function is justified as follows. Assume that the utility U_i of a single client i at time $t = 0$ is of the form

$$U_i = \int_{t=0}^{\infty} w(t) \mathbb{E}[\mathcal{R}_i(t)] dt, \tag{5.4}$$

where $\mathcal{R}_i$ is the real instantaneous data rate of the client i at time t and $w(t)$ is some decreasing non-negative weight function such that the integral always converges and $\int_{t=0}^{\infty} w(t)dt = 1$. The reason to use a decreasing weight function is that later values of expected data rate matter less due to increased uncertainty.

What do we know about $\mathbb{E}[\mathcal{R}_i(t)]$? If the client undergoes a change in channel, association or power, its service is interrupted for a time t', during which $\mathbb{E}[\mathcal{R}_i(t)] = 0$. Otherwise, for $t \geq t'$, no other information is available besides the throughput estimate T_i, so we set

$$\begin{aligned} U_i &= \int_{t=0}^{\infty} w(t) \mathbb{E}[\mathcal{R}_i(t)] dt \\ &= \int_{t=t'}^{\infty} w(t) T_i dt \\ &= p(t') T_i, \end{aligned} \tag{5.5}$$

where $p(t')$ denotes a penalty multiplier from network parameter changes, defined by $p(t') = \int_{t=t'}^{\infty} w(t)dt$. We notice that we can now write

$$\log U_i = \log(T_i) - I_c \mathbb{1}_{c,i} - I_a \mathbb{1}_{a,i} - I_p \mathbb{1}_{p,1}, \tag{5.6}$$

where $\mathbb{1}_{c,i}$, $\mathbb{1}_{a,i}$ and $\mathbb{1}_{p,1}$ return 1 if the client i is affected by a channel, association or power level change, respectively. Otherwise they return 0. Notice that because of how "affected by" was defined previously, the client can only be affected by one type of change, with an association change taking highest priority due to the longest service interruption and a power change the lowest, due to the shortest service interruption. Summing over the clients, we obtain the utility function in equation (5.3).

5.3.3 Optimizing the computation of utility

We need to devise some more efficient way to evaluate utility than naive depth-first evaluation. This is because in the simulated annealing search, mostly just one optimization variable changes between consecutive utility evaluations, so we do not want to redo the whole computation each time. Moreover, the naive method would carry out multiple times the computationally heaviest part, the independent set search in the contention graph G, even during a single evaluation of utility.

The more efficient computation scheme used by the algorithm is a form of lazy evaluation. The whole utility function is represented as a computational DAG (directed acyclic graph), where the nodes represent simpler functions and there is an edge from node/function a to b if the function in b uses the result of a as an argument. The only node which has only incoming edges is the node for total utility (equation (5.3)). There are also nodes for the optimization variables, and these have only outgoing edges.

When updating some optimization variable, we recursively mark as outdated all the nodes/functions in the computational DAG that depend on the variable. When evaluating utility, we then only need to re-compute values for the outdated nodes. An exception are functions whose value depicts whether two nodes *in the contention graph* G contend, called *edge functions*. These are updated immediately after marked outdated, after which the outdated status is propagated further only if the binary contention status changed. Without this special treatment, the edge functions would commonly propagate outdated status even if their value did not change, for instance causing a full utility re-evaluation even from a small power density change. Immediate recomputation means that the value of an edge function might be recomputed multiple times per parameter update. This is however not a significant drawback since the edge functions are quite close to the optimization variables in the computational DAG.

For computations, the independent set searches are formulated equivalently as clique searches, utilizing the fact that an independent set of a graph G is a clique of the complement graph $\bar{G}$. For clique searching, the software Cliquer [37] is used.

5.4 Simulated annealing

Simulated annealing is a metaheuristic for finding global minima (equivalently maxima) in a discrete search space, proposed in [38] and [39]. It is inspired by the physical process where a solid gradually cools, such that it finds a minimum energy configuration when the cooling is complete [40].

We do not have an especially good reason to use simulated annealing over other search methods such as hill climb or *tabu search* [41]. A goal for further research could be to compare different search heuristics to try to find the most efficient and practical one.

Simulated annealing was initially chosen because it seemed easy to implement. However, it turned out that there are some complications: for instance choosing good parameters for simulated annealing is nontrivial, and it needs to be done automatically for new problem instances. Fortunately, optimality is not required and simply finding satisfactory values suffices.

Pseudocode (modified from that in [42]) for the simulated annealing search heuristic is depicted as Algorithm 1. The algorithm searches for a global minimum in a finite search space X, where the utility of each element $x \in X$ is given by a function $U : X \to \mathbb{R}$.

Simulated annealing requires that we define a neighborhood structure $N : X \to 2^X$ for the search space. Then, the search essentially proceeds as a random walk in X, where the successor x' for a state $x \in X$ is chosen among $N(x)$.

At each step of the algorithm, a neighbor $x' \in N(x)$ for the current state $x \in X$ is chosen randomly according to some probability distribution $p_x : N(X) \to \mathbb{R}$. If the utility U of x' is larger than that of x, we move to

x' (called *accepting* x'). Otherwise, x' is accepted only with probability $\exp((U(x') - U(x))/T)$, where T is called the *temperature*.

The temperature T initially has the value of the chosen starting temperature T_{init}. For each value of the temperature T, `sweep`(T) steps are performed, after which the temperature is lowered according to the function `lower`(T). The search is terminated when the temperature reaches some preset lower bound T_{final}. One "cooling", i.e. one execution of the loop on line 4 of Algorithm 1, is in this text called one *cooling round* as opposed to a simulated annealing full run, which may consist of multiple cooling rounds.

As is apparent from the above description, there are many parameters involved whose values are important design decisions. These are the search space X, the neighborhood structure N, the probability distribution p_x (unique for each current state $x \in X$), the initial and final temperatures T_{init} and T_{final}, the starting location x_{init} (or method used to choose it) and the functions `sweep` and `lower`.

The search space X is named as a parameter because it might make sense to ignore some potential solutions of the original problem if good enough solutions remain in X. This might be the case if a class of solutions is clearly suboptimal, or if it is unnecessary to consider them due to symmetries.

In rough terms, it seems that the neighborhood structure should preferably be chosen so that neighboring solutions have similar values for the utility U. This allows the algorithm to spend more time exploring regions of the search space that are likely to contain good solutions. One the other hand, N should be such that the search space is connected enough for the search not to get stuck too easily in a small part of the search space. On the other, N should be sparse enough that the algorithm has time to find a local optimum at the end of the cooling.

The initial temperature T_{init} should be chosen such that the algorithm can relatively freely wander in the search space at the start of the search, in order to find a region with good solutions without getting stuck at locally optimal regions too early. The final temperature T_{final} should be small enough for the search to finally converge at a local optimum.

The functions `sweep` and `lower` together determine the *cooling schedule*, i.e. determine in what manner and how fast the temperature is lowered. The optimal manner to do this is non-trivial. There exists theoretical results that with a sufficiently slow cooling schedule, the algorithm is guaranteed to find the global optimum. However, this kind of a provably optimum finding cooling schedule is required to be exceedingly slow to the extent that it is not viable in practice. Instead, in practical applications, a common choice is to have lower$(T) = \alpha T$ for some constant $0 < \alpha < 1$, chosen relatively close to 1.

Algorithm 1 Simulated annealing

```
$T \leftarrow T_{\text{init}}$
$x \leftarrow x_{\text{init}}$
$x^* \leftarrow x$
while $T > T_{\text{final}}$ do
    $L \leftarrow \text{sweep}(T)$
    for $L$ times do
        choose $x' \in N(x)$ uniformly at random
        $\Delta U \leftarrow U(x') - U(x)$
        if $\Delta U \geq 0$ then
            $x \leftarrow x'$
        else
            choose $r \in [0, 1)$ uniformly at random
            if $r \leq \exp(\Delta U / T)$ then
                $x \leftarrow x'$
        if $U(x) > U(x^*)$ then
            $x^* \leftarrow x$
    $T \leftarrow \text{lower}(T)$
return $x^*$
```

5.5 Choosing simulated annealing parameters

Let us review how values for the simulated annealing parameters are chosen in this work. They are X, N, p_x, $T_{\text{init}}, T_{\text{final}}$, x_{init}, `sweep` and `lower`, defined in the previous section.

As the search space X we the set of all possible tuples

$$(\mathcal{C}, \mathcal{A}, d) \in [C]^A \times [A]^{S-A} \times D$$

is used, where D is a set of relevant power density levels $D = L_1 \cup L_2$, where

$$\begin{aligned} L_1 =& \{d' \in \mathbb{R} \mid d' \leq P_{\max} \text{ and } \exists i, j \in [A] \text{ such that } d' + p(i, j) = -82 - \epsilon\}, \\ L_2 =& \{P_{\max}, P_{\max} - 3, P_{\max} - 6, P_{\max} - 9\} \end{aligned} \tag{5.7}$$

for some very small positive $\epsilon \in \mathbb{R}$. The maximum allowed power level for a 20 MHz channel, in dBm, is denoted by $P_{\max}$. These are the only power levels where maximum utility as of equation (5.3) can be attained, since utility U is increasing with power density except at points where the contention graph C or the maximum allowed bandwidth changes. Of these classes, the former corresponds to L_1 and the latter to L_2 in equation (5.7).

For the sake of rigor we note that the maximum of U relative to power density does not exist if L_2 does not contain a maximum point: In this case the supremum of U is obtained for some point in L_1 when power density approaches the point from the left. Since ϵ is small, L_1 contains a point where utility is so close to the supremum that we say that maximum utility is attained there.

The optimization variable $\mathcal{W}$ is not included in a solution since for each AP i the channel width is calculated from the closed form formula

$$\begin{aligned} w_i = \max\{w \in \{1,2,4,8\} \mid w \leq W_{\max}(d) \text{ and } c_j \notin R_i(w) \text{ for every} \\ j \in [A] \text{ such that } \{i,j\} \in E(G)\}. \end{aligned} \tag{5.8}$$

Here G is the contention graph from section 3.2.2, $R_i(w)$ is obtained from equation (5.1) and $W_{\max}(d)$ is the largest allowed channel width for power density level d such that transmit power does not surpass its maximum $P_{\max}$.

The point in choosing channel width as in equation (5.8) is that it prevents contention between the primary channel of one AP and a secondary channel of another AP. This kind of *primary-secondary contention* has been deemed harmful in [11] where it was found to cause unfair contention due to a higher CCA threshold being used for secondary channels.

Using a closed form formula for bandwidth is obviously advantageous because it reduces the size of the search space X and thus makes the search more efficient. Also, assuming that primary-secondary channel contention is not allowed, the bandwidths from equation (5.8) maximize the term $\sum_{i\in[S]\setminus[A]} \log(T_i)$ of utility U in equation (5.3). However, as a disadvantage they may not maximize the penalty terms, which might lead to some optimal or close to optimal solutions being excluded from the search space.

The neighborhood $N(x)$ of a solution x is chosen to consist of the configurations that can be obtained from x by either changing the primary channel of one AP, having one client perform a handover to another AP or changing the power density level. The magnitude of the change in power density is limited to be at most 1/4 of the difference between the highest and lowest power level in D (but changing power density to the closest smaller and higher value is still always permitted to avoid getting stuck). Additionally, when the step is a client handover, if the new AP is idle previously, the algorithm finds and sets the optimal channel for that AP as well. By optimal we here mean optimal w.r.t the utility function in equation (5.3).

The probability distribution p_x is for simplicity chosen to be the uniform distribution for each state $x \in X$, except that moves are memorized and a move tried before can only be retried after the algorithm has iterated over all possible moves. The temperature T_t at time step t is obtained from the formula

$$T_t = \alpha^t T_{\text{init}}(t), \tag{5.9}$$

where $T_t^{\text{init}}(t)$ is obtained from a heuristic formula as a constant multiplied by the diameter of the contention graph G multiplied by the difference between the maximum and minimum utility encountered so far. Thus $T_{\text{init}}(t)$ is in this scheme updated as time progresses. We use $\alpha = 0.95$ (with `sweep` $=$ 1) and the cooling continues until the search is stuck, which we notice has happened when the algorithm has iterated over all possible moves without an improvement occurring. The initial configuration x_{init} is chosen to be that in which the network currently is.

Chapter 6

Simulations

In this chapter, we describe the simulations run in this work. The simulations are mostly concerned with evaluating the performance of the dynamic algorithm in different scenarios and with different features of it being turned on or off. In addition, a few small simulations are run to evaluate the CTMC model used internally by the dynamic algorithm.

6.1 The Wenla simulator

For the simulations in this work, the simulator Wenla is used. It is a proprietary step-based system level simulator for Wi-Fi networks developed at Nokia. It follows the the 802.11 standard, supporting both versions 802.11n and 802.11ac, including HCF and client mobility.

The physical layer is simulated as follows: The IEEE TGac model [43] (extension of TGn [44]) is used to model path loss, attenuation caused by the medium, shadowing and fast fading and to calculate received signal strengths and SINRs at receivers. These are calculated separately for each OFDM symbol in the time axis and for each 20 MHz channel, and used to calculate an effective SINR via the Exponential Effective SINR Metric (EESM) [45]. Finally, the effective SINR is mapped to block error rates (BLER) via an actual value interface table obtained from separate link level simulations.

6.2 Default parameters for the simulations

We simulate a variety of different scenarios, but some of their properties and parameters are common to several or all of them, hence listed here as default parameters.

We use three kinds of simulation worlds that differ in the number and location of the APs. These worlds, having 8, 64 and 256 APs, respectively, are pictured in Figure 6.1. Red squares represent APs while black lines are walls. The dimension of each world is 32×32 meters, with 8×8 meter rooms. The distance between an AP and any of its four closest neighbors is 16 m, 4 m, and 2 m, for each of the world types, respectively. APs are located in the ceiling of the rooms, at a height of 3 meters. A signal attenuates by 6 dB for each wall it passes through. The simulations use 64 APs by default.

We use worlds with different AP densities in order to test how the dynamic algorithm performs in networks of different density. A world where the APs are in a regular grid was chosen to reflect the fact that dense networks would most likely be planned, thus having a regular structure. A random setup was also considered, but left outside the scope of this work.

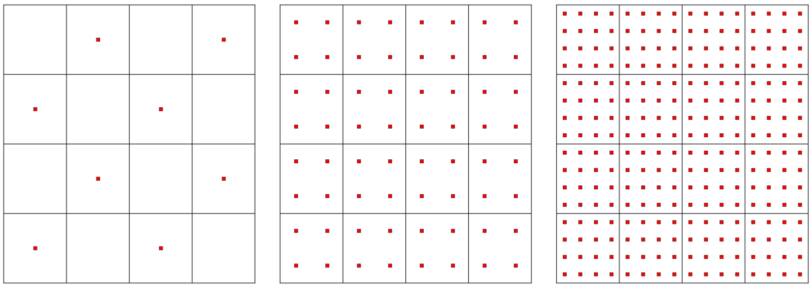

Figure 6.1: Simulation worlds with 8 APs, 64 APs and 256 APs, from left to right. By default, 64 APs are used.

Traffic in the simulations is downlink only, except for CTS and ACK frames sent by clients. APs always have a full buffer of client data to send to each client. APs choose the receiver for each TXOP by cycling through their clients in a round-robin manner.

Client stations spawn in the simulation following a three dimensional homogeneous Poisson point process, with two spatial and one temporal dimension, with a default average density of 60 new clients per second. Essentially this means that each physical location in the horizontal plane in each time instant is equally likely to spawn a client, and these events are mutually independent. The spawning height is a fixed 1.5 meters. We refer the reader to [46] for a proper definition and analysis on Poisson point processes. The clients stay in the simulation for times that are independent and exponentially distributed with a mean value of 0.4 seconds, after which they exit permanently. The simulated real world time is 4 seconds per simulation instance. When the dynamic algorithm is used, a full run is performed between every 0.08 seconds. The run is instant, i.e. the algorithm output is produced at the same instant of simulated time as input was collected. However, it might take a short while (usually at most 0.2 seconds) to implement the network parameter changes indicated in the output.

These times are not necessarily intended to be realistic, but the goal is simply to simulate each consecutive network configuration (i.e. set of active clients and network parameters) sufficiently long so that throughputs for the clients stabilize to their long term averages. It was determined by tests that no significant changes in simulation results are caused by increasing time intervals between network configuration changes such that they remain proportionally the same. (This is achieved by altering total simulation length while proportionally increasing client lifetime and algorithm full run interval as well as inversely proportionally decreasing average client spawning density.) The only observed difference was that the clients with lowest performance perform somewhat better with less dense events, possibly due to overheads from network parameter changes mattering less then.

There are 24 available 20 MHz channels, which are adjacent. Bonded channels always consist of consecutive 20 MHz channels (even for 160 MHz, contrary to the standard) according to equation (5.1). The available channel widths are 20, 40, 80 and 160 MHz as in the standard. This channelization resembles the actual one permitted in most countries, except that the number of available channels for each bandwidth is one less in most cases. This implies that in reality not every 20 MHz channel may be part of a bonded channel of any width, a situation which would unnecessarily complicate this work.

Dynamic bandwidth selection is used, i.e. the largest bandwidth for which all 20 MHz channels are sensed free is used. Still, the bandwidth can be no more than the maximum bandwidth, which is 80 MHz by default.

Proper implementation for control plane messaging regarding network configuration changes is not implemented in Wenla, the requested changes instead happening almost instantly. This means that the penalty on network operation from network changes is negligible in the simulator even though it usually is considerable in practice. Properly simulating impairment from network changes was left out of the scope of this work as the actualized penalties are highly implementation dependent. Thus, we will only study how much impairment free throughput suffers if the algorithm is incentivized to perform less changes by increasing the change penalties in the utility function in equation (5.3). By default, only small penalties were used (penalty multiplier 0.25), sufficient to somewhat reduce the amount of changes requested by the algorithm, without harming throughput noticeably.

RTS/CTS is used for all data frames. Single user MIMO with two transmit data streams is used. Fast fading, referring to the rapid variation of signal attenuation, is modeled according to the IEEE TGac channel model (see [43]). Slow fading is not included in the simulations. When not otherwise stated, 30 parallel instances are run that are otherwise identical, but have a different seed for the generation of pseudo-random numbers. This results in an expected amount of 7200 clients, i.e. data points, for each simulation parameter combination.

By default, all features of the dynamic algorithm are in use, which is: In addition to producing in full runs new configurations for primary channels, bandwidths, transmit power density and association, the algorithm between runs recommends initial APs for new clients. In the simulated annealing search, the probability of each trial move to alter a primary channel, try a handover or alter the common transmit power density (all mutually exclusive), are 0.4, 0.4 and 0.2, respectively. In the clique search to determine channel share, no local approximation is used and only maximum cliques are taken into account (as in equation (3.24)). This is because the graph radii remain quite small in all simulations, and the cliques with non-maximum weight were demonstrated to have negligible impact.

6.3 RRM algorithms compared

In the simulations, three different RRM algorithms or schemes are compared, called `Random`, `Static` and `Dynamic`, where `Dynamic` is the algorithm developed in this thesis. All of these are concerned with managing AP channels, client initial association and handovers, transmit powers as well as bandwidths, together called the *network configuration*. However, only `Dynamic` manages all of these dynamically, whereas the others may use static values for some or all of them.

Table 6.1: The subvariants of `Dynamic` with probabilities for each type of search move.

Name/Prob. of changing	channel	association	power density
`Dynamic`	0.4	0.4	0.2
`NoChannels`	0	0.7	0.3
`NoAssociation`	0.7	0	0.3
`NoPowerDensity`	0.5	0.5	0
`OnlyChannels`	1.0	0	0
`OnlyAssociation`	0	1.0	0
`OnlyPowerDensity`	0	0	1.0

With `Static`, primary channels, bandwidths and transmit powers stay constant throughout the simulation, with bandwidth and transmit powers being set to their maximum allowed values given in the previous section. Clients always associate to the AP from which the received power is the highest and perform no handovers. Primary channels are given to each AP one after another so that each AP is set to the channel where its closest same channel neighbor (that has already been assigned a channel) is as far as possible regarding radio distance (defined as signal attenuation in dBm). The set of allowed primary channels is chosen such that a secondary channel of one AP is never the primary channel of another.

The algorithm `Random` is otherwise similar to `Static`, except that the primary channels are every 0.08 seconds randomized according to the uniform distribution. The same rules apply for choosing the available primary channels to avoid primary-secondary contention.

For the algorithm `Dynamic`, several subvariants are tested, where each subvariant differs in which aspects of the network configuration it manages dynamically, the three options being AP channels, client association and the common transmit power density level. Managing client association includes both ordering handovers as a result of algorithm full runs as well as choosing the initial AP for a new clients. Properties not managed dynamically are handled similarly as in `Static`. In this sense `Static` is a subvariant of `Dynamic` where nothing is managed dynamically.

Table 6.1 lists the subvariants and the probabilities of performing each type of move during each trial step of the simulated annealing search. A probability of zero means that the algorithm variant does not try to manage the corresponding property.

This choice of algorithms to compare is a compromise between research interest and the amount of time required to implement and test further algorithms. The different versions of Dynamic are tested in order to gain information on how useful and important each feature of Dynamic is in each simulated setting, as well as to obtain insight on the performance of the algorithm. Each combination possible is covered, where Static can be seen as a subvariant of Dynamic where none of the network properties are managed dynamically. Static was chosen as a basic point of reference as it closely resembles many state-of-the art algorithms, where APs upon being turned on choose the channel via LCCS. Random represents a situation where no intelligent network management is performed.

It would be interesting to additionally simulate different algorithms from the literature to compare their performance to that of Dynamic. However, this is left out of the scope of this work due to the time requirements of programming those algorithms.

6.4 Simulation types

In general, choosing the simulation scenarios is a highly nontrivial problem, as on one hand relative algorithm performance depends heavily on the scenario, so a range of scenarios needs to be simulated for any sort of reliability, but on the other hand the whole range of different scenarios is too large for every scenario to be included. Therefore this thesis adopts an approach where a reasonable default scenario (defined by the default parameters of the previous section) is constructed and only one parameter type of the simulation altered at a time so that the total number of simulations remains reasonable (however the number of clients and APs are altered simultaneously).

In these simulations, the main interest is client average throughput. To present it we use CDFs (cumulative distribution functions) where one client is one data point. We are interested in the distribution (and not just averages) because it shows the fairness of the network in addition to total capacity.

Further result statistics that help in interpreting the results are presented in Appendix A. In addition, there is a separate set of simulations for evaluating the accuracy of the channel share estimate of the CTMC model.

The first set of simulations is concerned with evaluating the performance of the dynamic algorithm by comparing the subvariants of Dynamic with each other and by comparing Dynamic to Static and Random. The parameters varied in these are the number of APs (with values 8, 64 and 256 APs), client density (with 15, 30, 60 and 120 new clients per second), maximum bandwidth (20, 40, 80 and 160 MHz) and wall attenuation (0, 6, 12 and 18 dB). All combinations of AP amount and client density are simulated, while bandwidth and wall attenuation are varied alone. When varying the amount of clients and APs, there are only 6 simulation instances per parameter combination, but the durations of the simulations are longer so that the number of clients (i.e. data points) remains similar to the other simulations. When varying maximum bandwidth, `Static` and `Random` always use the maximum value as their bandwidth.

A simulation is also included for studying the effect that different change penalties have on the performance of `Dynamic`. The relative change penalty amounts simulated are 0, 0.1, 0.25, 0.5, 1, 2 and 1000. The point is to obtain some insight into how real-world impairment from network changes could affect performance of the dynamic algorithm.

For testing the performance of simulated annealing, a set of simulations is included where the maximum number of search moves is varied. The amounts simulated are 40, 80, 160, 320 and 640 thousand search moves. The actual amount of moves is higher than this since the algorithm finishes the current cooling round when it reaches the move limit. Two client densities, 60 and 240 clients per second are simulated, and the amount of APs is 256.

Finally, simulations for evaluating the CTMC model are included. The first group of them directly evaluate the accuracy of the CTMC channel share estimate. Simple simulations are used where four uniformly distributed clients stay for the whole simulation duration and all use the same primary channel. Four different locations of the clients are studied. In addition, simulations are run with CCA energy detection threshold values of both -62 dBm (used in Wi-Fi and therefore in the rest of the simulations) and -82 dBm, which makes eight simulations in total.

Second, the CTMC model is evaluated by simulating how the dynamic algorithm performance is affected by what CTMC model simplifications are in use. Two simplifications are tested: activity rate being infinite (equation (3.24)) and the local approximation (in section 3.2.3). Only the extreme cases, with graph radii 1 and infinite (the whole graph), are tested. These tests are performed in two kinds of scenarios: In the first case, client density is 60 new clients per second and wall attenuation is 6 dB. In the second, client density is 240 new clients per second and wall attenuation is 24 dB. In both cases, the algorithm does not perform search moves altering power density and moves changing channel or association are equally likely.

The first case is similar to the other simulations in this work. The second case was designed so that the connected components of the contention graphs would be more complex with a larger diameter (largest distance between any two vertices) so that having a large graph radius setting in the algorithm could potentially make a difference.

Chapter 7

Results and discussion

In this chapter, we present and analyze the simulation results. We assess the dynamic algorithm by comparing its performance to algorithm `Static` as well as analyze individually the performance of each of its components i.e. throughput model, simulated annealing search and general algorithm form. We also discuss what the results indicate about the usefulness of managing each type of RRM parameter. Moreover, we evaluate the research methods of this work, i.e. how the limitations of the simulation setup impact the relevance of the results. Finally, we present conclusions on how well the research goals of this thesis were met as well as discuss possible future work.

7.1 Performance of the dynamic algorithm

To evaluate the performance of the dynamic algorithm, we first look at the results of the simulations where the amount of APs, client density, maximum bandwidth and wall attenuation were varied. These are presented in Figures 7.1–7.10.

It appears that transmit power density is not lowered in any of the simulations with 8 APs (in Figures 7.1 and 7.2, see also Figure A.1 in Appendix A), so the curves for `Static` and `OnlyPowerDensity`, `OnlyChannels` and `NoAssociation` as well as `Dynamic` and `NoPowerDensity` are identical or almost identical in these cases, and therefore the other curve may not be showing. The curves are similarly in some cases very close to each other with 15 clients per second and 64 or 256 APs (in Figures 7.3 and 7.5).

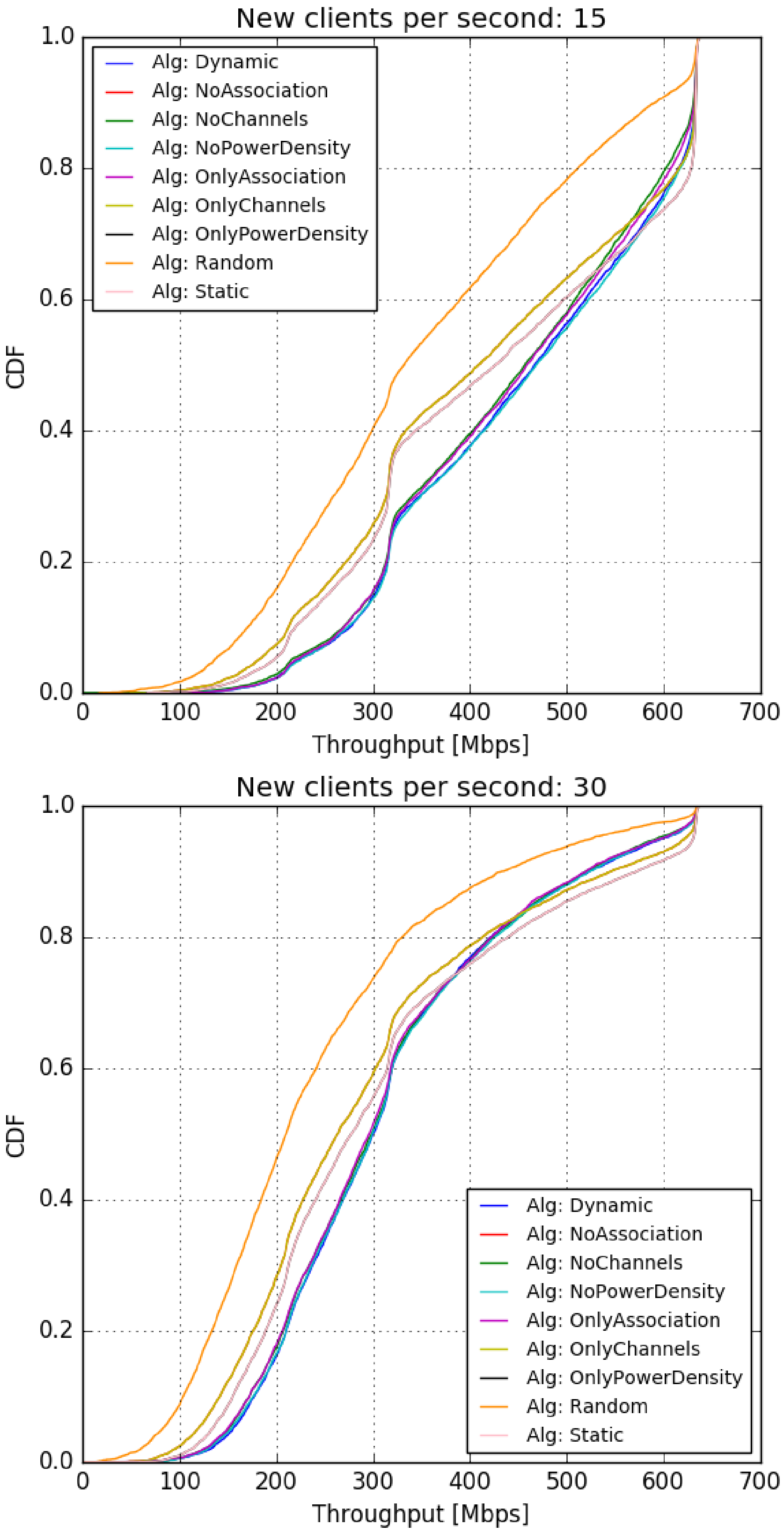

Figure 7.1: Throughputs for different numbers of clients with 8 APs.

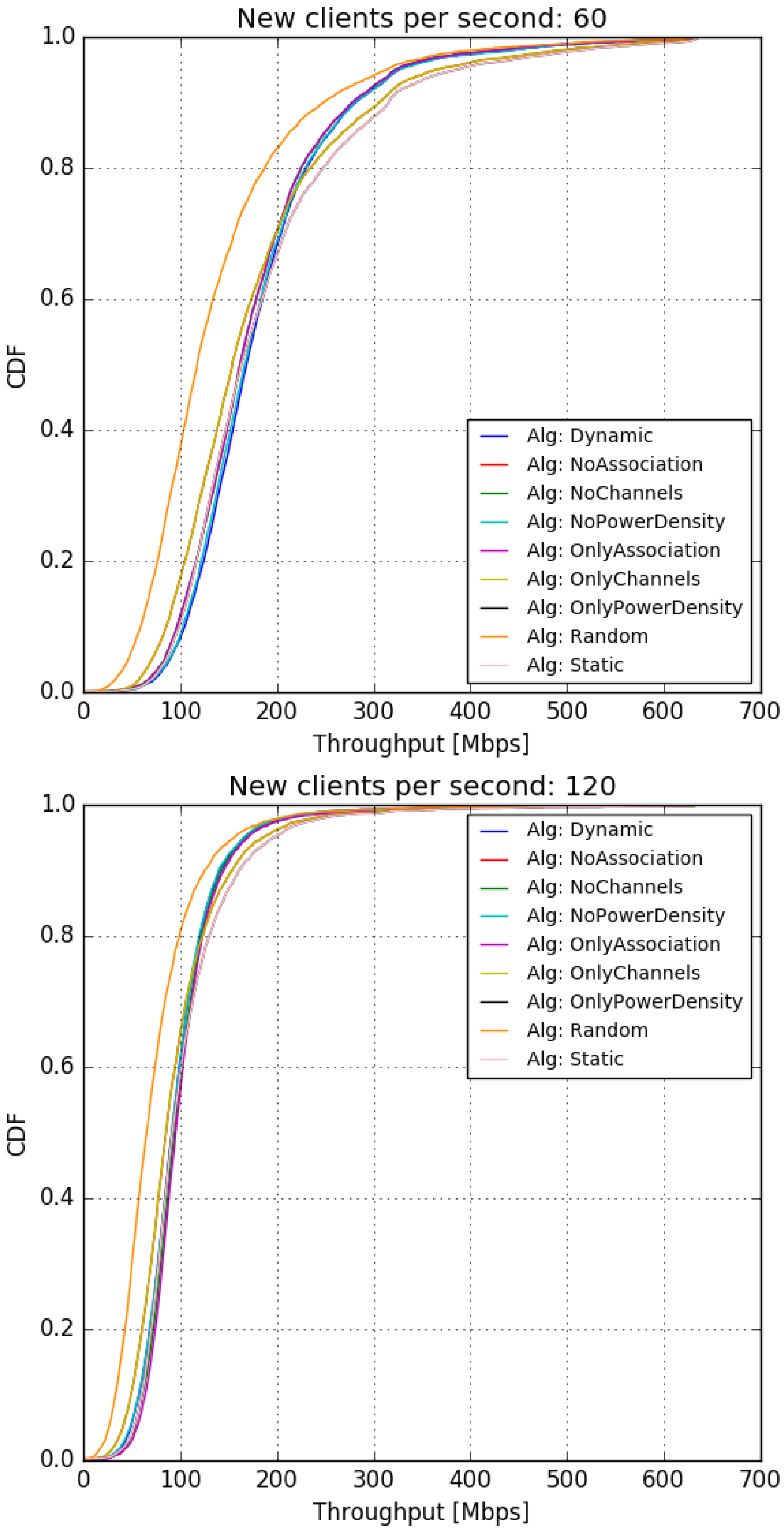

Figure 7.2: Throughputs for different numbers of clients with 8 APs.

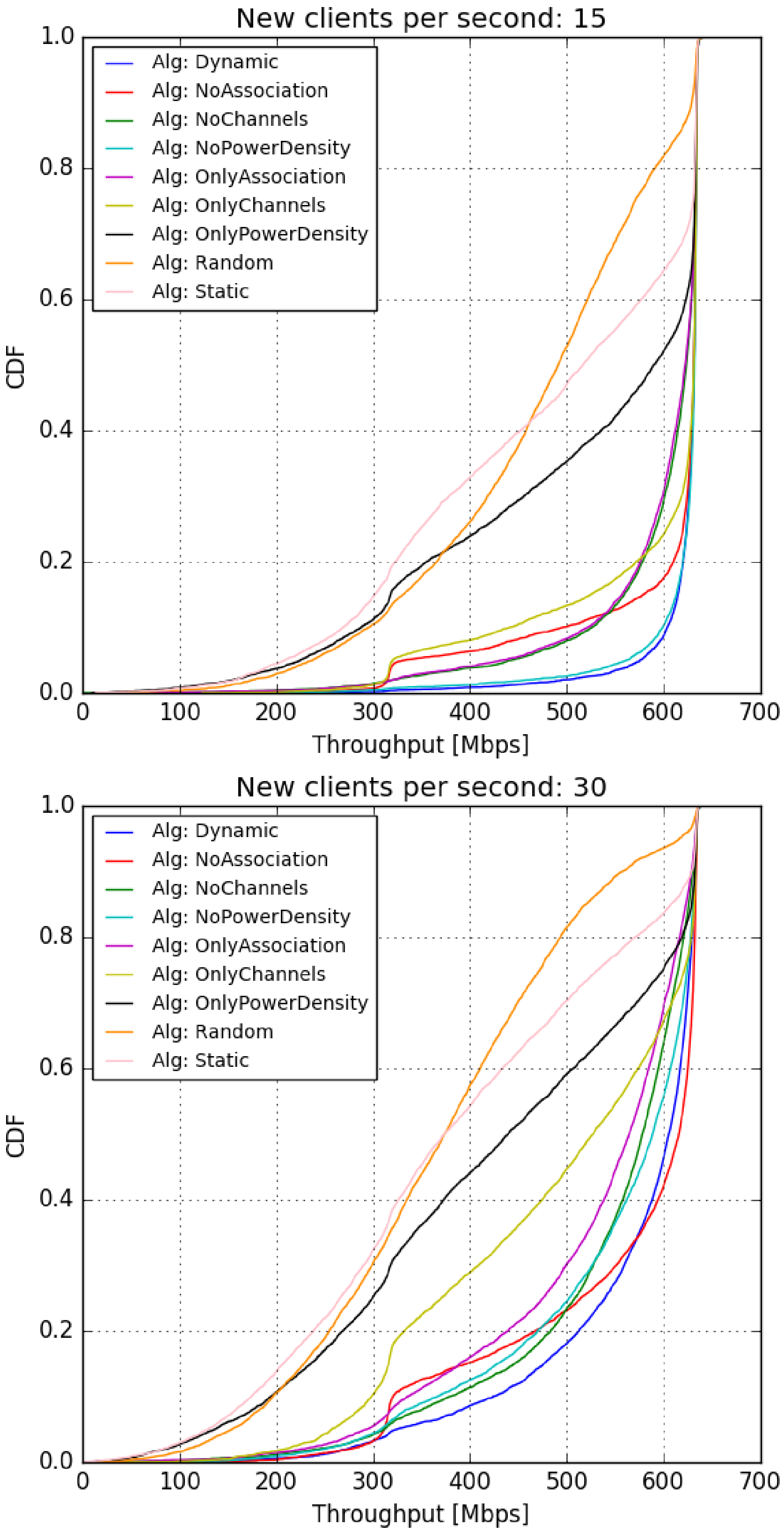

Figure 7.3: Throughputs for different numbers of clients with 64 APs.

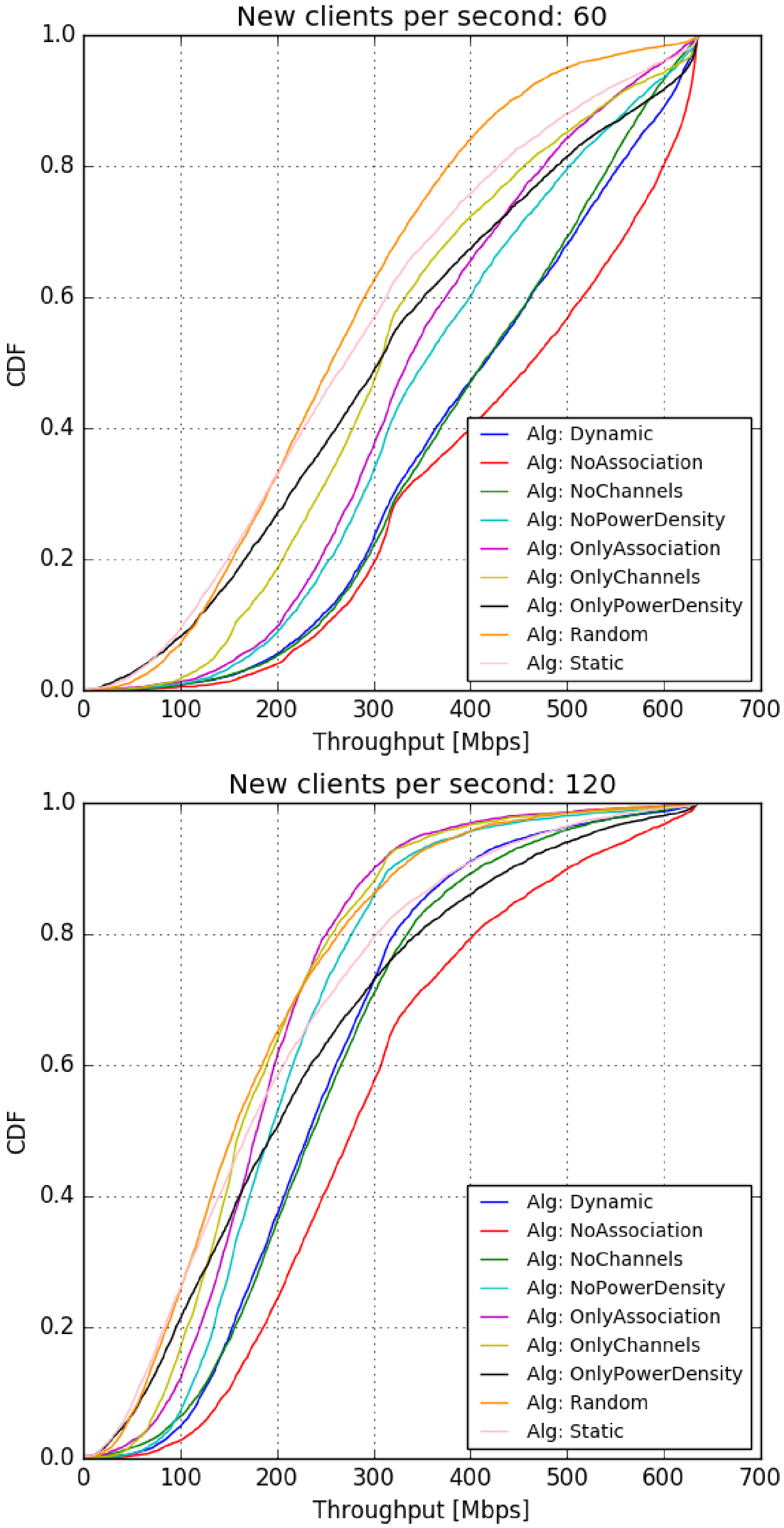

Figure 7.4: Throughputs for different numbers of clients with 64 APs.

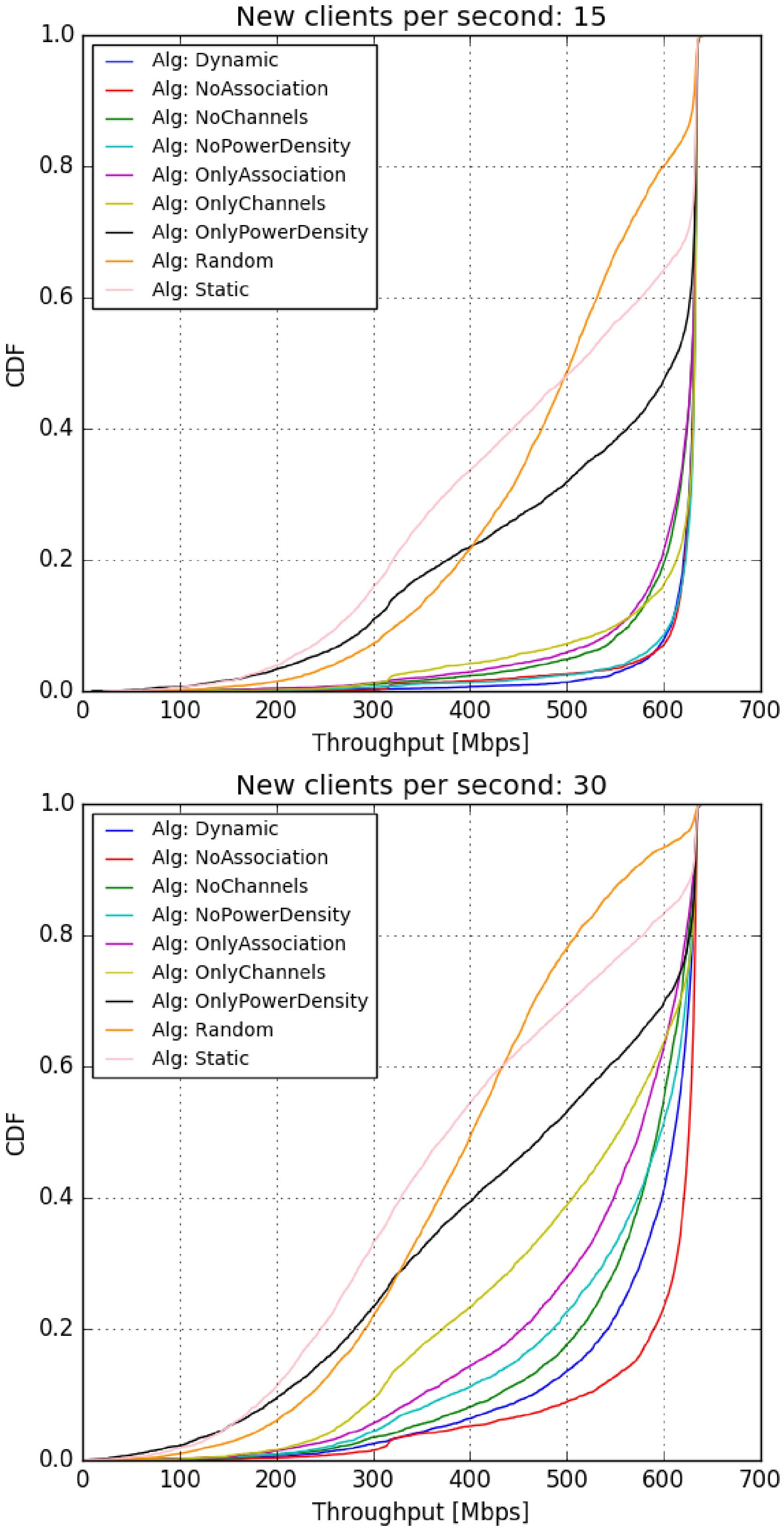

Figure 7.5: Throughputs for different numbers of clients with 256 APs.

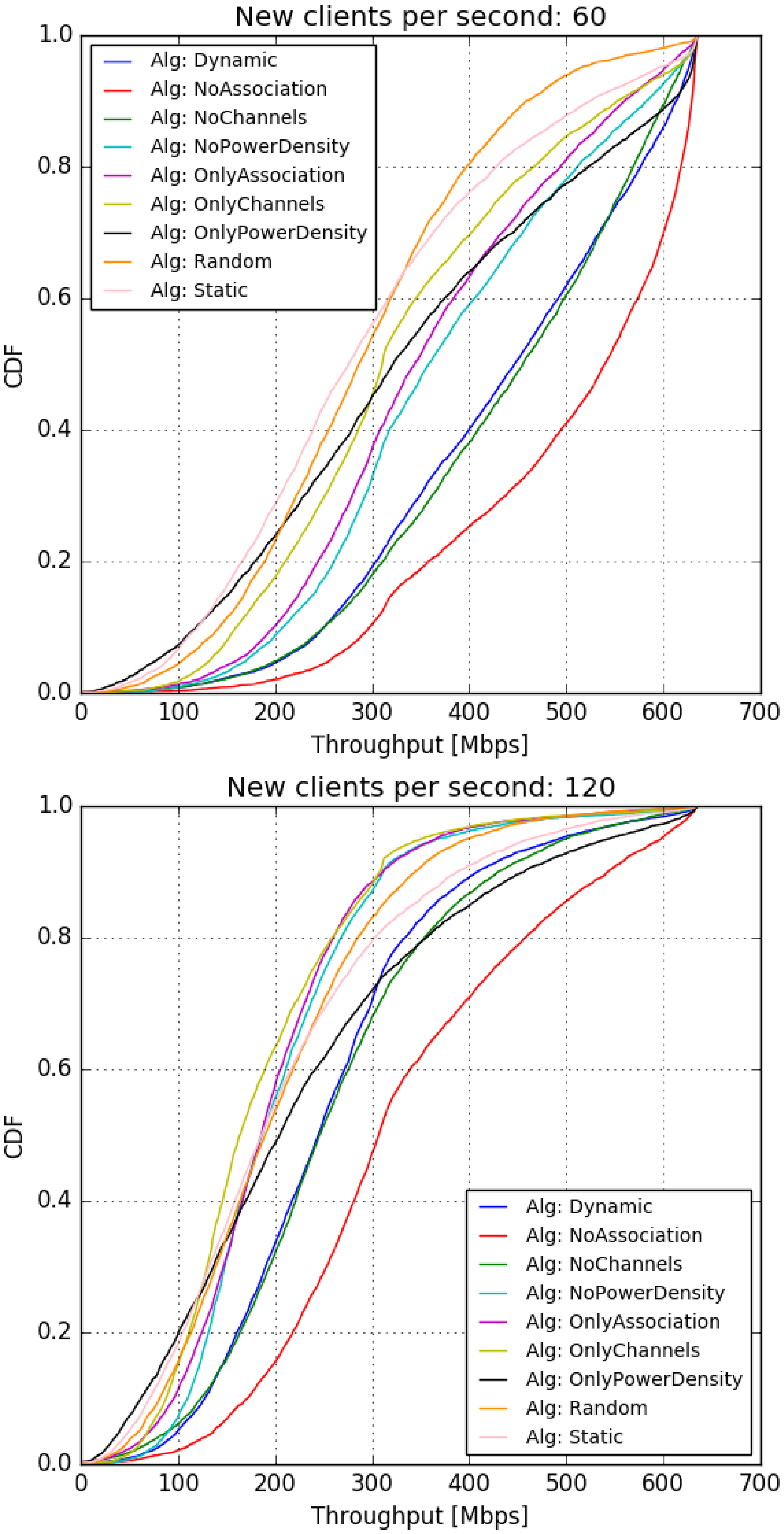

Figure 7.6: Throughputs for different numbers of clients with 256 APs.

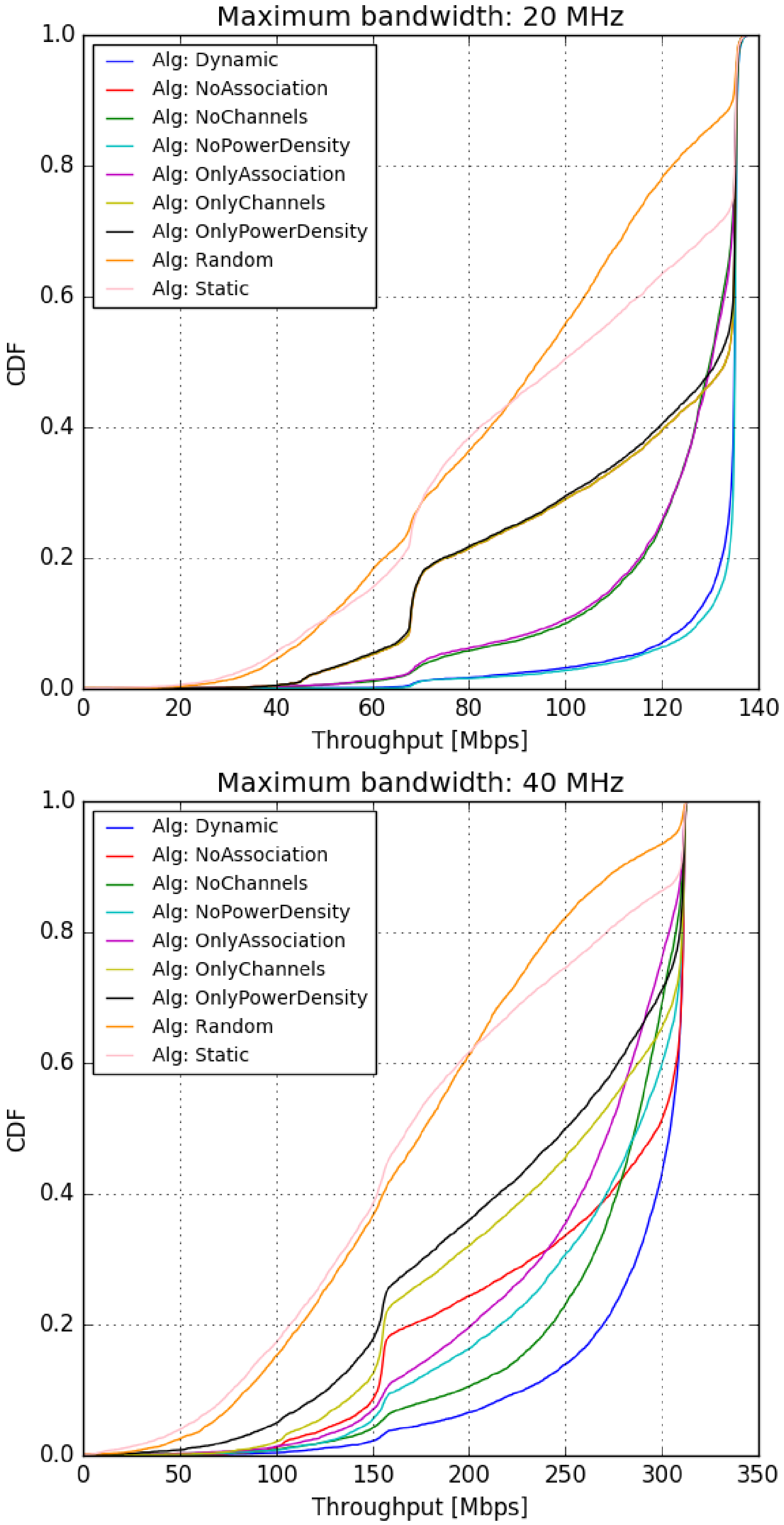

Figure 7.7: Throughputs with maximum bandwidths 20 MHz and 40 MHz.

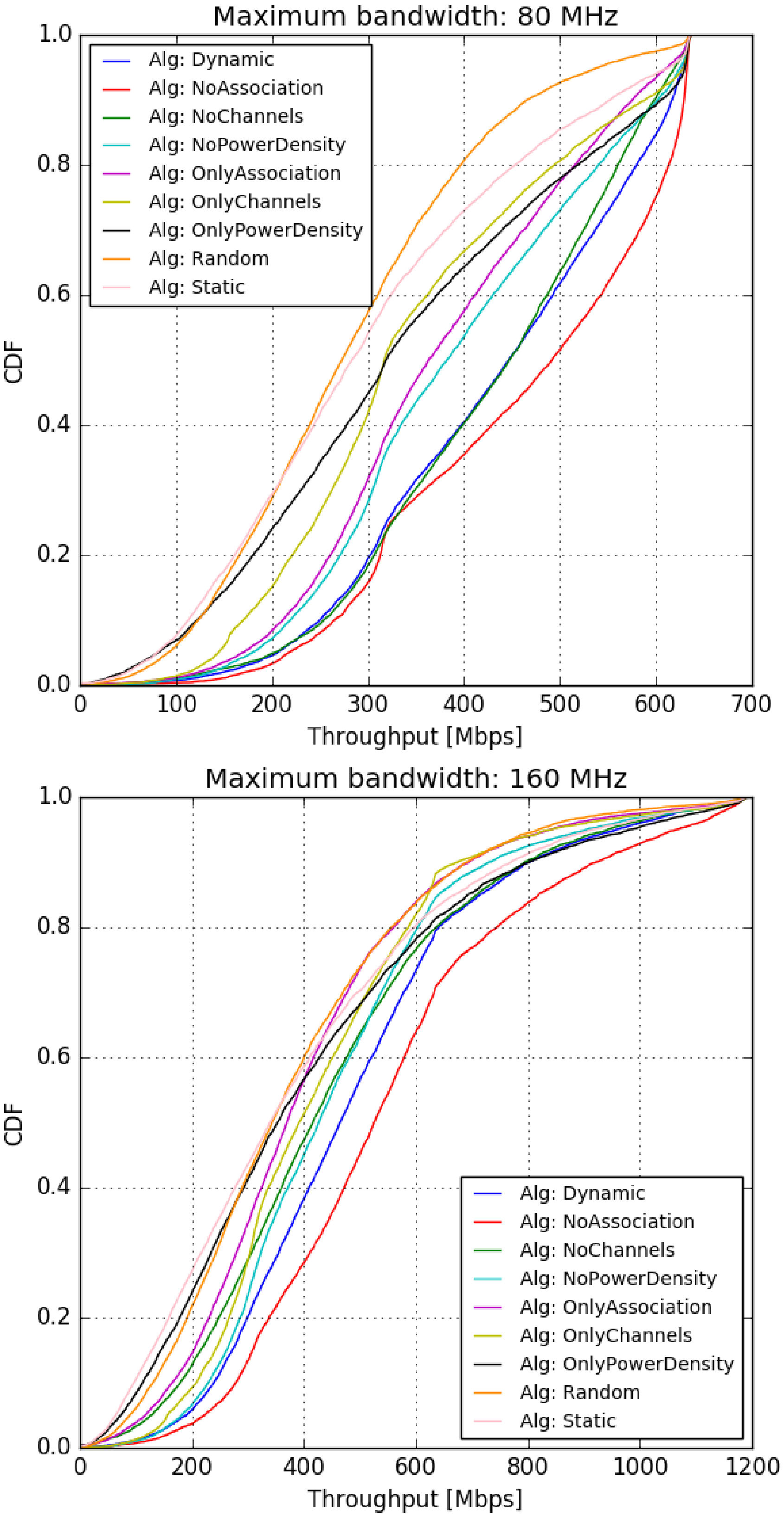

Figure 7.8: Throughputs with maximum bandwidths 80 MHz and 160 MHz.

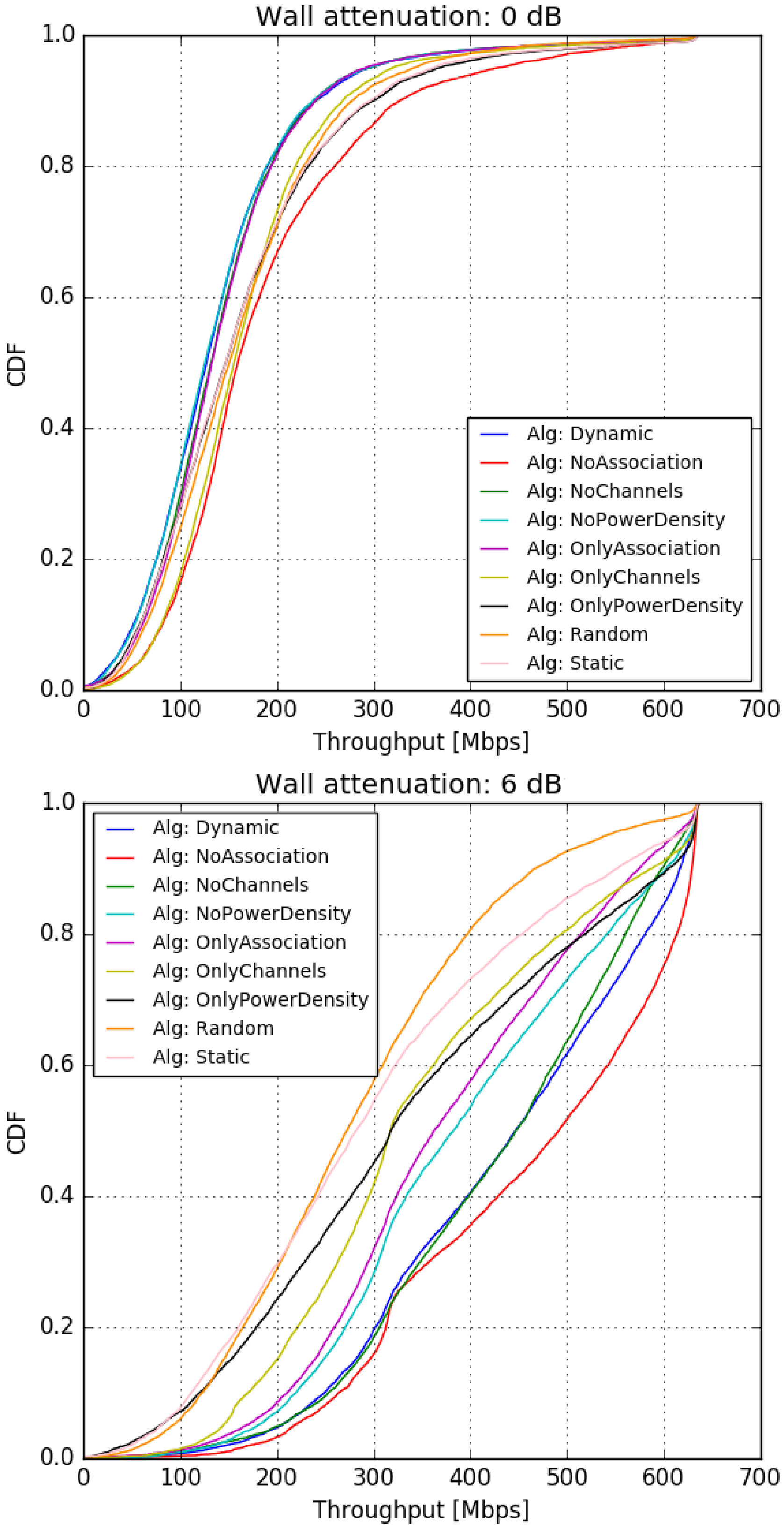

Figure 7.9: Throughputs with wall attenuation 0 dB and 6 dB.

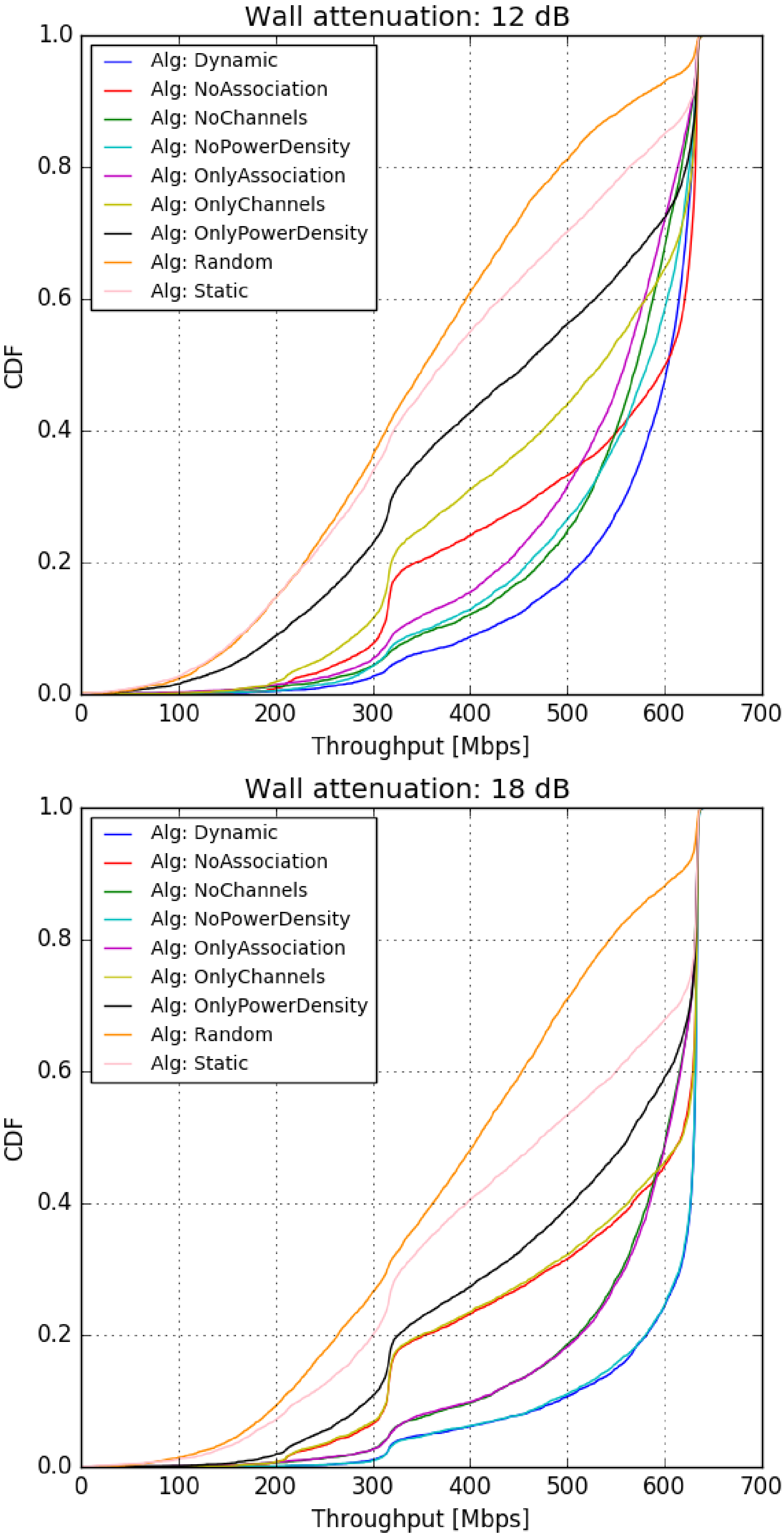

Figure 7.10: Throughputs with wall attenuation 12 dB and 18 dB.

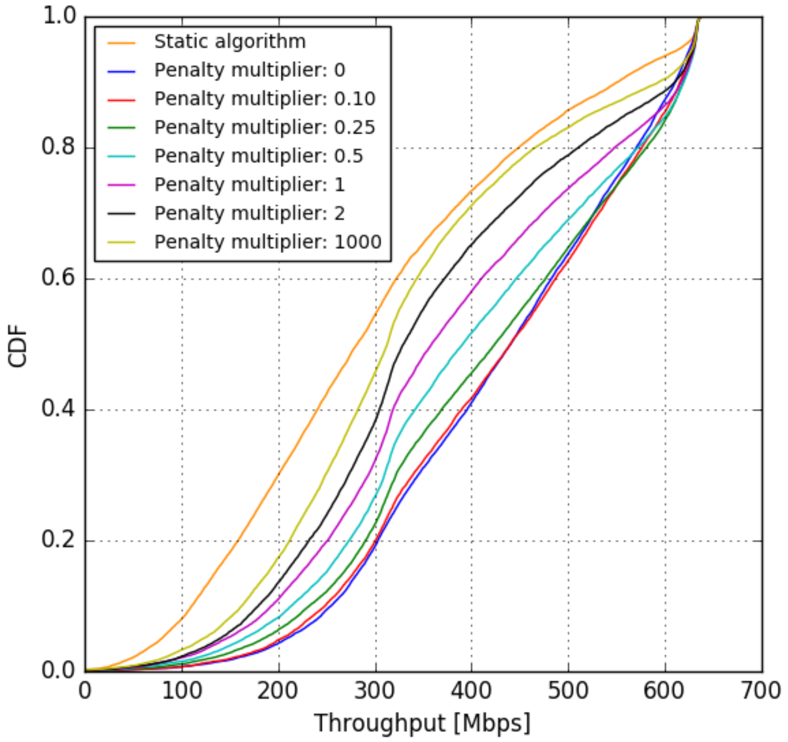

Figure 7.11: Throughputs for each change penalty amount.

Looking at the throughput results in Figures 7.1-7.10, variants of the algorithm `Dynamic` perform really well in comparison to `Static` and `Random`, both in terms of total throughput and fairness. A significant advantage of the approach is its generality: Either `Dynamic` or `NoAssociation` performs significantly better in almost all situations. The only exception is the scenario with the minimum number of APs and maximum numbers of clients, where all algorithms except `Random` perform roughly equally. This is because here all algorithms find an optimal channel allocation (one without co-channel interference), and user association is not needed since a large number of clients is distributed roughly evenly (i.e. optimally) even when associating to the closest AP.

A weakness of these results is that most simulations were run with a change penalty amount that likely permits more configuration changes than what would be viable in practice. However, in the simulations where change penalties were varied (Figures 7.11 and 7.12) we observe that the benefits of `Dynamic` remain even when the frequency of network parameter changes decreases with rising penalties.

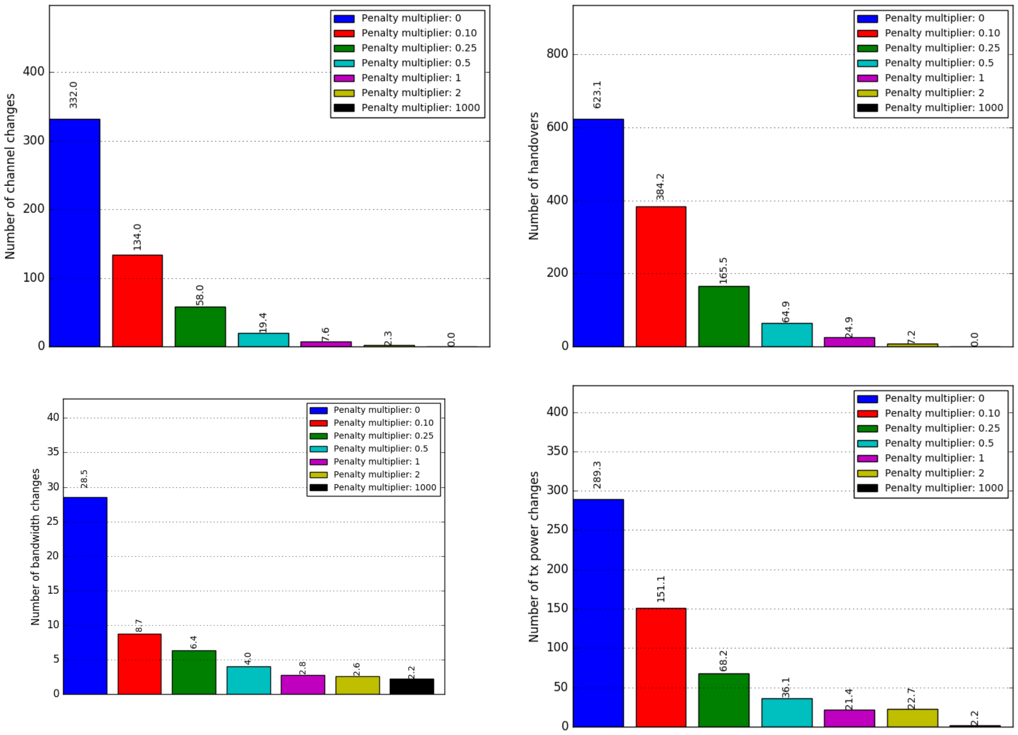

Figure 7.12: Amounts of network changes for different parameter types and change penalty amounts.

With 64 or 256 APs, there is no large difference in total throughput between `Random` and `Static`. This seems to be the case because only a fraction of the APs is in use with `Static`, so the channels of the APs in use actually get randomly chosen. This leads to the conclusion that in a dense network with clearly more APs than clients, only dynamic channel allocation schemes are useful.

Unfortunately there is no experimental data for comparing `Dynamic` to an optimal RRM scheme or any other dynamic RRM scheme for that matter. Still, we can make a number of observations on how, why and to what extent `Dynamic` fails to be optimal, by considering its components individually.

The first limitation of `Dynamic` comes naturally from the general framework of how the algorithm operates: Reconfiguration of the whole network is only done between regular intervals, during which only greedy optimization for new clients is performed. Alternatively, network optimization could be triggered based on for instance sufficiently decreased utilities or sufficient changes in path losses between stations. Another limitation is the full buffer assumption, which will be discussed in section 7.3.

An important component of the algorithm is the form of its utility function (equation (5.3)). The simulation results do not offer much insight into how optimal it is. We can note that it performs at least satisfactorily and seems justified by the argument in section 5.3.2.

In addition to the general operating model and the form of the utility function, `Dynamic` consists of a model for estimating instantaneous full buffer throughput as well as the simulated annealing search algorithm used for finding good solutions.

7.1.1 Performance of simulated annealing

Good algorithm performance implies that simulated annealing functions decently as a search algorithm. A benefit of it is that it is not very sensitive to the amount of search time or moves available, as shown by the results in Figure 7.13, where the search move limit is varied.

We see from the results in Appendix A that in many simulation scenarios the search did not reach solutions that are close to optimal utility. This is shown most clearly in the simulations with the highest client density 120 per second (in Figures A.2 and A.3), where the algorithm `NoAssociation`

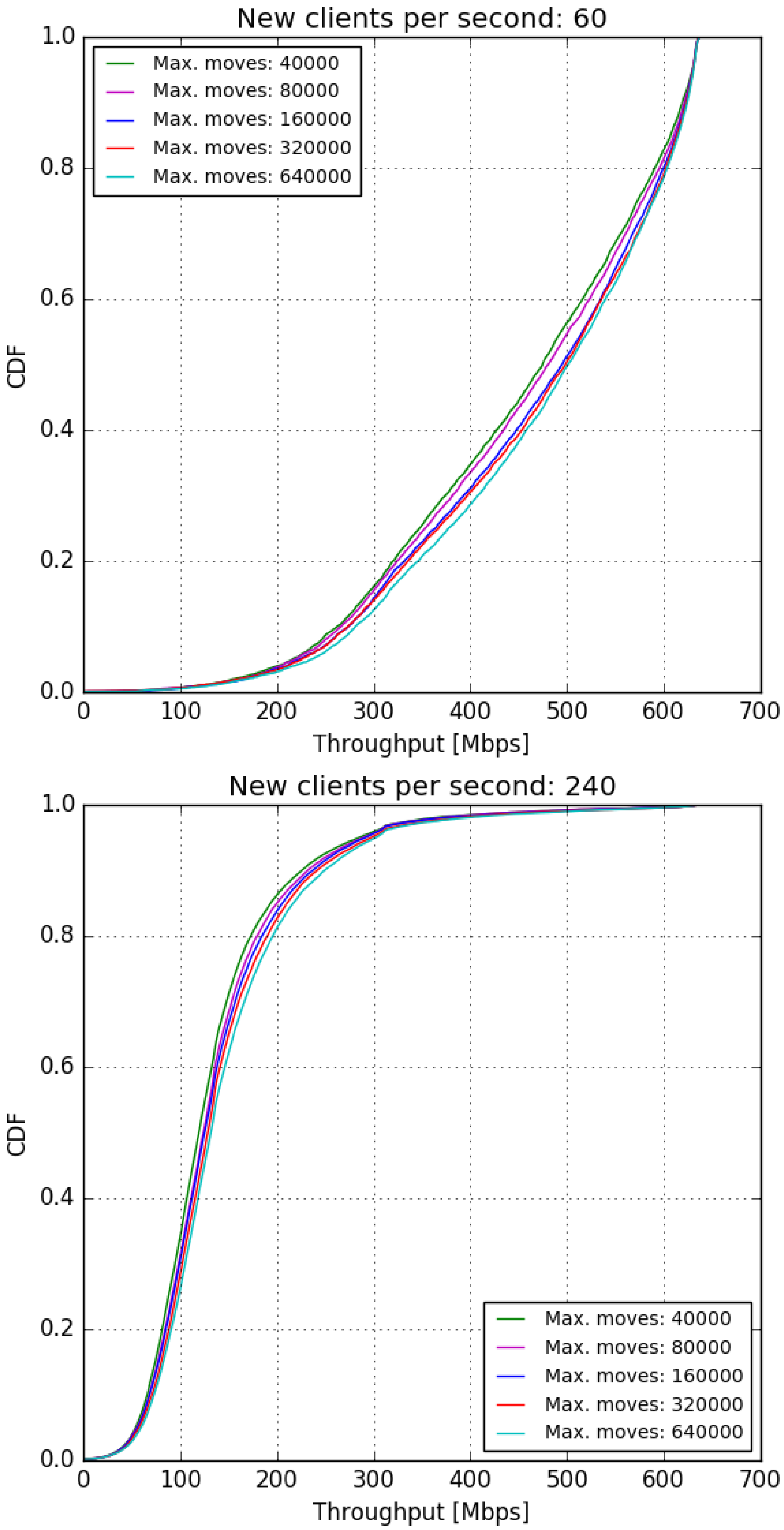

Figure 7.13: Throughputs with different amounts of search moves, with 60 new clients per second above and 240 new clients per second below.

achieves better utility than `Dynamic`, even though the search space of the former is a subset of that of the latter.

The best solutions that the search with `Dynamic` failed to find are those with low transmit power, as indicated by the results on power e.g. in Figures A.2 and A.3. Perhaps it is easier for the search with `NoAssociation` to lower transmit power because clients are always associated to the closest AP, so lowering transmit power easily gives a benefit even when adjusted individually. In contrast, in full `Dynamic` lowering transmit power individually might be harmful if the clients are associated to more far away APs, from which signal quality is weaker.

The results in Appendix A for when the move limit is varied (Figure A.7) point towards the same conclusion: More search time specifically leads to simulated annealing finding solutions with lower transmit power.

The second problem of the simulated annealing approach is finding good values for its hyperparameters. Different values are needed for different scenarios, so the values need to be chosen automatically. Only very vague heuristics were available, and thus the values used might be significantly suboptimal. This could also contribute to simulated annealing finding notably suboptimal solutions in many scenarios.

In section 1.1 we set the goal that the algorithm must only require computational resources so that it can be run on a regular desktop computer. For this end simulated annealing running times are presented in Figure 7.14, where the time is CPU time. The upper picture represents the time of a single cooling round, while the lower shows the time of a whole full run. For these results, as for all simulations by default, the search move limit for simulated annealing is 160 000. Average client lifetime is 0.4 seconds so the average numbers of clients present are 6, 24 and 96 for 15, 60 and 240 new clients per second, respectively.

At least one cooling round must be run, so we observe that simulated annealing is not efficient enough computationally for a network with the highest client density, 240 new clients per second. For 15 and 60 new clients per second, the running time seems to be of a tolerable magnitude. We note that full run CPU computation durations are less critical, since the individual cooling rounds may easily be distributed for multiple CPUs, and the results are not very sensitive to their number, as shown in Figure 7.13.

Finding an AP for a newly joined client took usually at most 0.2 seconds for 60 new clients per second, and at most 6 seconds for 240 new clients per second. The conclusion for both full run durations and new client association times is that with a network of 60 new clients per second (24 existing simultaneously on average) they were satisfactory, but the computations did not scale well for 240 new clients per second (96 average simultaneous clients).

7.1.2 Accuracy of the throughput model

For evaluating the throughput model, let us first focus on CTMC approximations of channel share. The results on the accuracy of these are pictured in Figures 7.15 and 7.16. The uppermost pictures show the locations of the clients with blue dots. A line connects APs with clients if the APs can sense each other, i.e. the signal from the neighboring AP is received at least at -82 dBm per 20 MHz channel, excluding momentary variations due to fast fading. Below each of these pictures, the channel shares are pictured for the respective scenario, for CCA energy detection threshold -62 dBm in the middle and -82 dBm in the lowest pictures.

The channel share realized in the simulations is defined as the fraction of time the AP is either transmitting or receiving a transmission from a client in its BSS. The channel share estimated by the algorithm is that obtained from equation (3.20). Here, the maximum independent set approximation is not used.

In Figures 7.15 and 7.16 we notice that there is a significant amount of disparity between the actualized and predicted channel shares. For -62 dBm, this was the case except for scenario 3, and a major reason for this was found to be that stations failed to decode headers of frames that arrived with a low received power, especially in the presence of interference. Thus duration fields for physical and virtual carrier sensing could not be read, hence the -62 dBm CCA threshold for energy was used in these cases. This was confirmed from simulator logs that detail every frame sent and received (not included in this work). This happens despite the physical layer headers being decoded with a low (i.e. robust) MCS.

The effects of this can be clearly seen in scenario 1: APs 32 and 49 sense each other properly, as they are separated only by one wall, thus hearing each other above the energy detection threshold -62 dBm. However, AP 61 is regularly not able to decode headers from AP 49, thus being able to freely transmit almost at any time. On the other hand, in scenario 3, we observe a situation where there is no third interferer close by, so even channel sensing using the -82 dBm threshold works properly.

Another way to confirm that channel sensing with the -82 dBm regularly fails

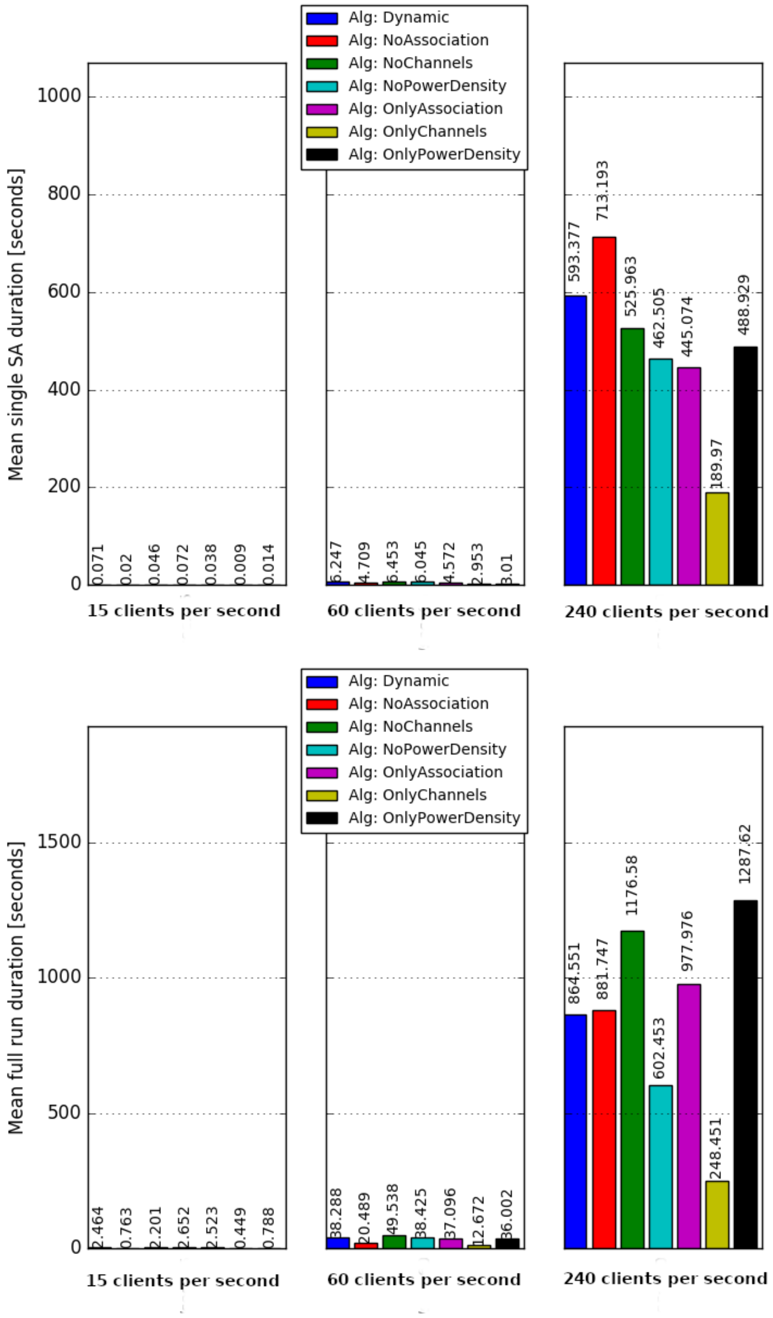

Figure 7.14: Simulated annealing running times.

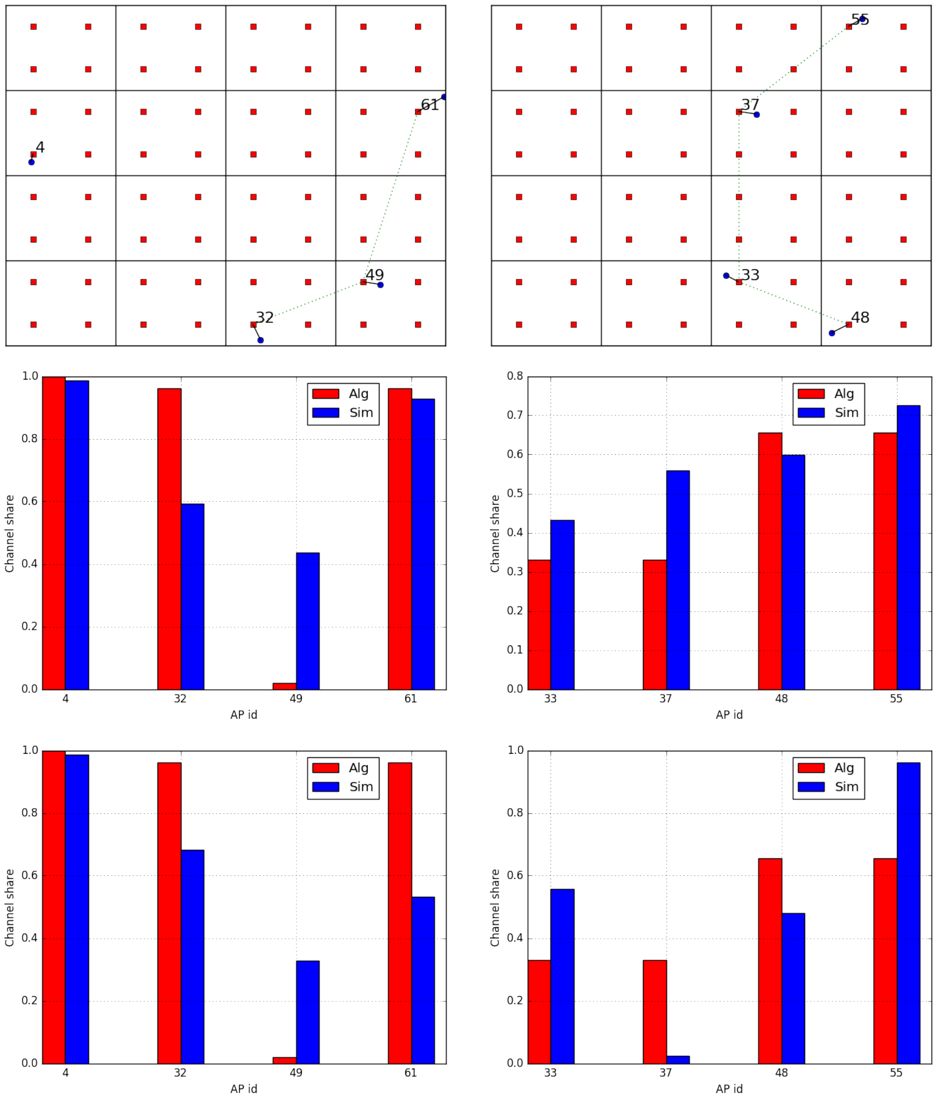

Figure 7.15: Comparison of estimated and simulated channel share. Scenario 1 (left) and 2 (right).

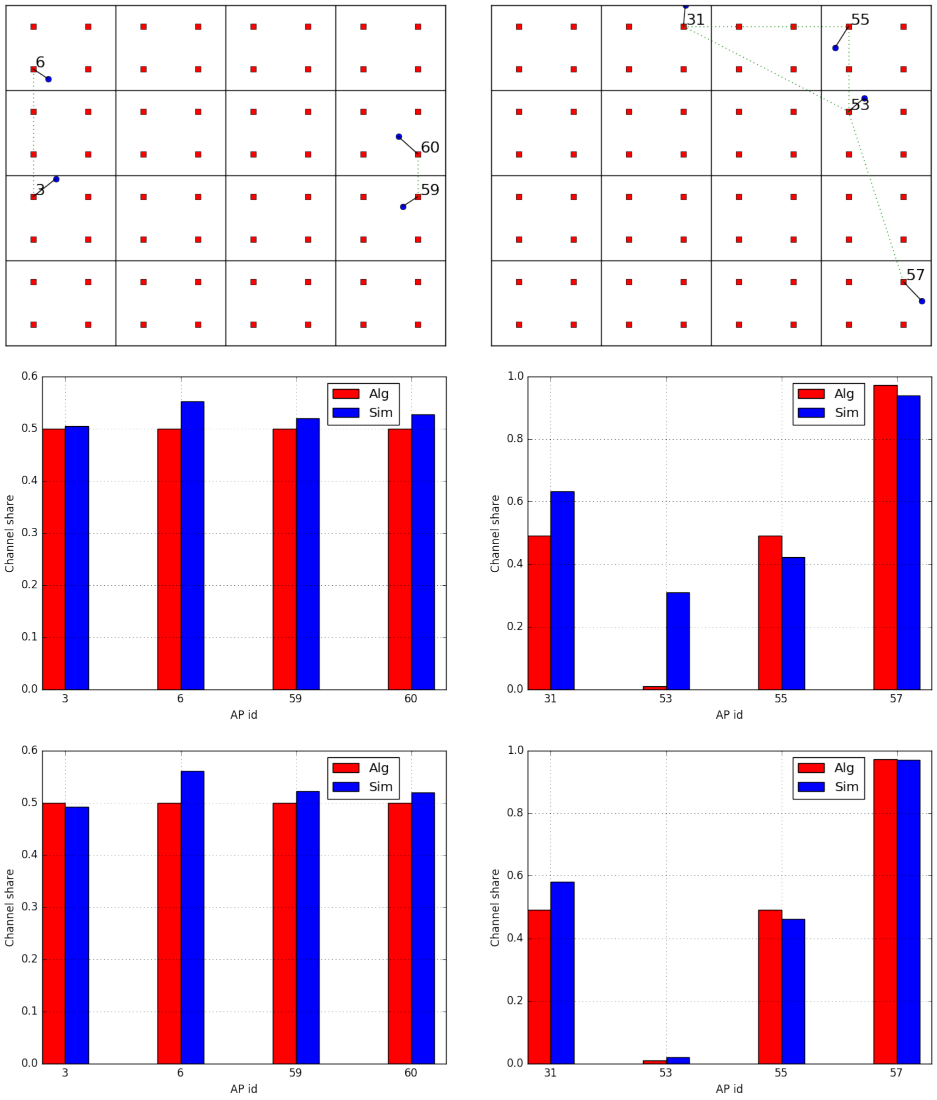

Figure 7.16: Comparison of estimated and simulated channel share. Scenario 3 (left) and 4 (right).

is to notice that the sum of channel shares, i.e. the average number of simultaneous transmitters, is larger than what the CTMC model predicts. This prediction is after all a theoretical approximate maximum for any instantaneous number of simultaneous transmitters, assuming that channel sensing works perfectly and there are no collisions. In these simulations, the number of mutually contending stations is so small that significant numbers of collisions do not occur.

The simulations using energy detection threshold -82 dBm were run to find out further reasons for the CTMC model failing, as sensing using the energy threshold is substantially more reliable than that based on decoding headers. Now, the predictions in scenarios 3 and 4 are reasonably accurate, while in scenarios 1 and 2 the predictions fail. In scenario 1 the reason for this is that ACK frames from the client of AP 4 are received at over -82 dBm by AP 61 and its client. This is possible because ACK frames have a higher power spectral density due to being sent at the same power level as data and RTS/CTS frames, but only using 20 MHz bandwidth instead of 80 MHz.

In scenario 2 the reason was found out to be that the received signal power between AP 48 and the client of AP 37 is actually slightly over -82 dBm, which due to using RTS/CTS means that the APs 37 and 48 effectively contend with each other. Adding an edge to the contention graph to reflect this would result in a reasonably accurate channel share estimate.

To summarize, the channel share estimation was found to be remarkably inaccurate, and the reasons for this were the following three matters that the CTMC model for channel share does not take into account: Firstly, carrier sensing based on decoding frame headers often fails in the presence of interference, so the actual power density threshold used varies unpredictably between -82 and -62 dBm. Secondly, ACK frames are sent at a higher power spectral density than other frames, which means that the contention range for these frames is larger. Finally, the model ignores the effect of clients when assessing contention.

Only four individual scenarios were examined here to limit the already large scope of this work. However, further simulations run by the author, not presented in this work, further corroborate the findings that the inaccuracy of the CTMC model used is significant especially with energy detection threshold -62 dBm (value used by Wi-Fi and in the simulations of this work). No further factors causing it besides the three presented here were found.

There are also simulations for evaluating the CTMC model based on how its parameters affect throughput. These results are presented in Figure 7.17

where graph radius n corresponds to span $n-1$ as defined in section 3.2.3. The results also point towards the CTMC model being useless, since it shows that using maximum clique radius does not give any improvement over simply estimating the channel share of a node to be the inverse of the number of nodes competing with it, including the node itself (this is what follows from having graph radius 1). However, the inaccuracy of the CTMC model is not the only explanation for these results, as we notice by examining simulation results not included in this thesis that the algorithm mostly allocates channels in such a way that actualized contention graphs only have radius 1. In this case, using graph radius 1 gives the same estimate as the full CTMC model.

In addition to estimating channel share, the throughput model includes estimation of instantaneous data rate, for which the Shannon-Hartley theorem is used (equation (2.5)). Modern OFDM-based wireless communication systems come reasonably close to this bound (see [7]). Thus it does seem like a good approximation, especially as correct decision making only requires the estimates to be proportionally correct. However, the estimation of interference as a constant -82 dBm seems fairly inaccurate as it was above concluded that the carrier sensing threshold -82 dBm may fail in which case the threshold -62 dBm is used. This would suggest that the algorithm in loaded settings does not see enough harm in poor link quality, and may thus have a tendency to overemphasize channel reuse against link quality.

How much does the model accuracy suffer from not taking into account collisions? To estimate the number of collisions occurring we look at the fraction of failed MPDUs (included in the results of Appendix A) instead of the fraction of failed RTS/CTS exchanges, for two reasons. Firstly, since RTS/CTS exchanges are short compared to transmission of the actual user data frames, the impact of failed RTS/CTSes is limited. Secondly, RTS/CTS failure rate is to the best knowledge of the author an unreliable measure of the actual number of collisions, since in the simulations it is fairly common that RTS/CTS is successful despite a collision that causes MPDU failure. (This is explained by RTS/CTS being sent with a more robust MCS than the user data frame.)

From the results in Appendix A we see that the percentage of failed MPDUs stays below 5% except for the most crowded scenarios, i.e. those with the maximum client density or bandwidth, or with zero wall attenuation. This indicates that not estimating collisions only skews model predictions by a limited amount, except for those most crowded scenarios. The extreme case is the simulation with zero wall attenuation, where the percentage of collisions is over 35 for some algorithms, which indicates a need to estimate the impact of collisions in this scenario. With collision estimation, the algo-

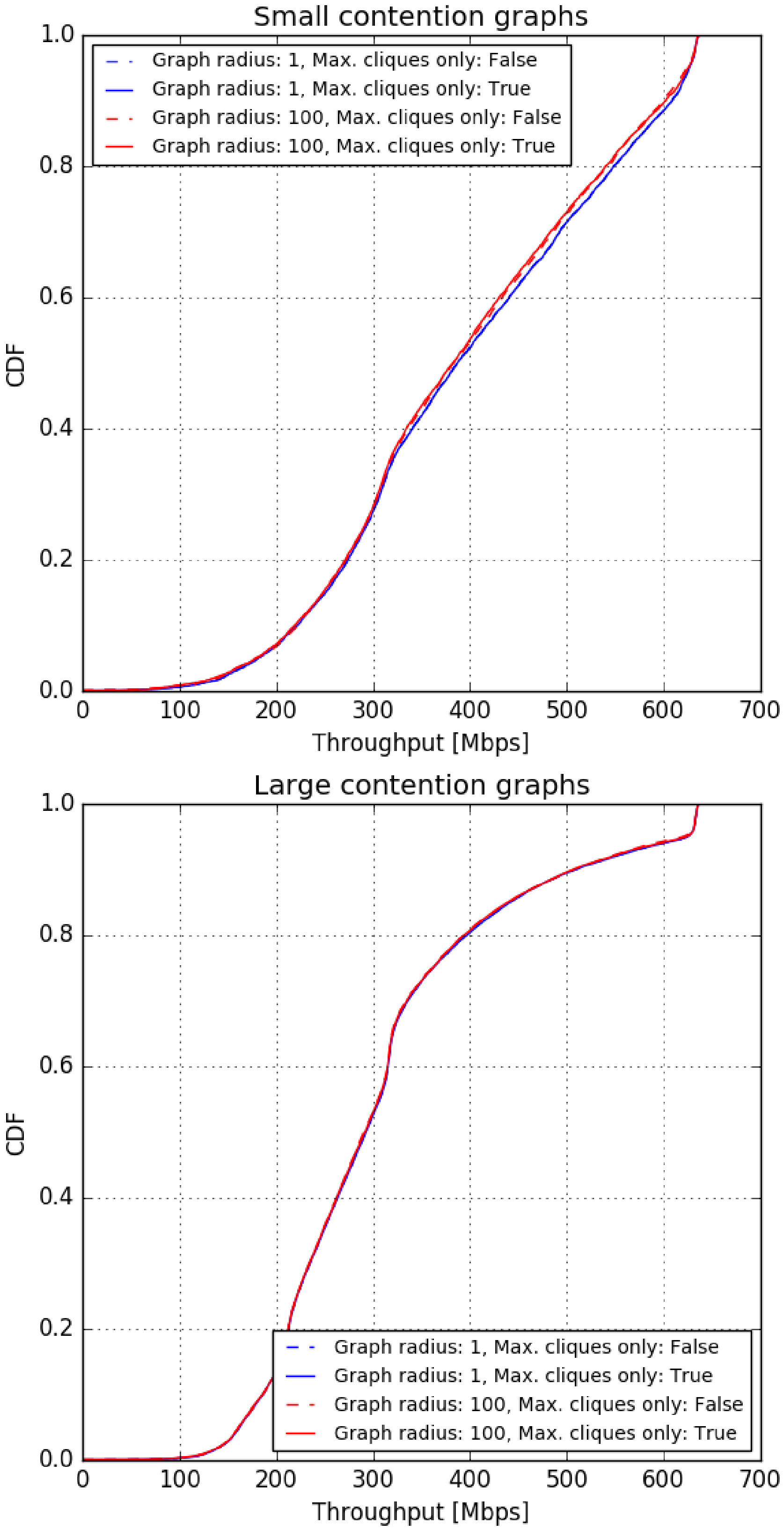

Figure 7.17: Testing different channel share approximations.

rithm could in that case improve performance by decreasing the number of mutually competing stations. It would do this by allocating channels with smaller bandwidth, by reducing transmit power or by focusing the clients to be associated to a smaller number of APs.

Estimating the effects of collisions could give the algorithm a more clear criterion for managing bandwidth. In its current form, there are, with the exception of lower bandwidth allowing higher power density, no explicit criteria based on which to choose bandwidth: for instance, the model predicts equal throughput regardless of whether there are four devices contending on one 160 MHz channel, or 4 devices each on a separate 20 MHz channel, if power density is the same in both scenarios.

Currently, `Dynamic` uses mostly maximum bandwidth except for the scenario where it is 160 MHz, or when the client density is high. Maybe this results simply from this kind of solutions being easy to find for the simulated annealing search? In addition to more collisions favoring lower bandwidths future criteria that would improve the choice of bandwidth could be the following two: Firstly, larger bandwidths result in slightly higher spectral density, since they allow slightly more subcarriers in proportion to the bandwidth used. Secondly, if the algorithm is evaluated in an environment where the full buffer assumption does not hold, higher bandwidth and thus more contending users could mean less wasted resources when a subset of the stations is idle.

Surprisingly, in many scenarios `NoAssociation` performs better than full `Dynamic`. This implies that the algorithm `Dynamic` works in a suboptimal manner, either by the throughput model being inaccurate or simulated annealing not finding near optimal solutions.

The better performance of `NoAssociation` to `Dynamic` is not always explained by suboptimal SA performance however, which we notice from the simulations with 256 APs and 30 clients per second (Figure 7.5), 64 APs and 60 clients per second (Figure 7.4), wall attenuation 6 dB (Figure 7.9), and maximum bandwidth 80 MHz (Figure 7.8). In these cases, as well as the ones where `NoAssociation` achieves better utility, its better performance results from more simultaneous transmitters achieved via a smaller transmit power density, slightly better error rates and using MCSes that have slightly higher rates on average. These were observed from the results in Appendix A.

One potential reason for `NoAssociation` doing better might be the fact that interference is often higher than the estimated -82 dBm due to channel sensing at that threshold failing, as discussed. This benefits `NoAssociation` due to its better SINRs that results from clients always associating to the closest AP.

7.2 Usefulness of dynamically managing each RRM parameter type

Since the results include simulations for every subvariant of the dynamic algorithm, we may conveniently observe how different network parameters affect the usefulness of dynamically managing each type of RRM parameter.

7.2.1 Managing transmit power density

From Figures 7.1–7.10 and A.1–A.5 (in Appendix A) it appears that generally the more dense or crowded a network is, the more transmit power density is decreased from its maximum value and the more benefit this yields. With 8 APs co-channel interference can be completely avoided even without lowering power density at all, and the largest benefit is obtained with 256 APs. Parameter changes that increase network crowdedness (via increased co-channel interference) and thus make power density management more beneficial, are increased client density, higher bandwidth and a smaller wall attenuation. An exception is wall attenuation 0 dB (Figure 7.9) where the radio distance between APs is so small that only `NoAssociation` manages to increase channel reuse by drastically lowering transmit power density. A probable cause for power density control being more beneficial for more dense or crowded networks is that then there is more channel contention and thus also more room to alleviate it by decreasing transmit power density.

In general, a much larger advantage in power control is gained without dynamic user association. This seems to be due to the fact that on the other hand a better signal quality, resulting from clients associating to the closest AP, enables power density to be lowered without signal quality suffering too much. Conversely, a low transmit power density makes it more harmful for clients to associate to a more far away AP.

7.2.2 Managing association

Again, we examine the results in Figures 7.1–7.10 and A.1–A.5. We see that generally, managing association is the less useful the more power density is decreased. The probable cause for this is the same as was discussed in the previous section.

Another reason for association management correlating negatively with network density and crowdedness might be that with a less loaded network, more APs are needed to use up all transmission bandwidth everywhere, and thus user association is needed to take more APs into use. This explains why managing association is not useful with 8 APs even with maximum client density (Figure 7.2): The 8 APs are all in use already even with clients associating to the closest AP.

Finally, we notice that managing association seems in general to improve fairness more than managing channels does. This is likely at least partially explained by the fact that association management prevents very uneven numbers of clients for different APs.

7.2.3 Managing channel assignment

With 8 APs (Figures 7.1, 7.2 and A.1) dynamic channel management is not useful due to static channel allocation being sufficient to completely prevent co-channel interference. Dynamic channel allocation is even somewhat harmful because the channel share model of the dynamic algorithm does not take into account clients contending with STAs that are not in the contention range of the AP. Simulation data confirms this as there is over double the amount of unanswered RTSes with `NoAssociation` compared to `Static`.

With 64 and 256 APs (Figures 7.3–7.6, A.2 and A.3) dynamic channel allocation is generally useful, except less so with a large client density when dynamic user association is already in use. The latter might result from the fact that user association in a dense network already allows some degree of channel management by associating users to APs with the desired channel.

A large maximum bandwidth (Figures 7.7, 7.8 and A.4) in general seems to make dynamic channel management more useful. A probable reason for this is that with larger bandwidths the alternative static channel allocation is less effective at preventing co-channel interference.

Dynamic channel management brings benefits with a moderate wall attenuation, but not so much with minimum and maximum attenuation (Figures 7.9, 7.10 and A.5). With minimum attenuation zero dB this results from contention being impossible to prevent even with good channel allocation (with the exception of algorithm `NoAssociation` which combines channel management with drastically reducing transmit power density). With maximum wall attenuation, the static allocation manages to always avoid co-channel interference, so dynamic allocation is not needed.

7.3 Evaluation of research methods

The practical value of the results is to some extent limited by the idealizing assumptions made in the simulations: firstly, that traffic is downlink only and secondly that the APs with clients associated always have full buffers. However, the first assumption can be justified by the fact that in practice for regular Internet users downlink traffic comprises the majority of the total traffic. The second limitation could possibly be circumvented in a real world implementation by the algorithm rapidly updating its knowledge on which APs have something in their buffer to send, and only consider those APs in its model.

A second limitation of the methods used is the Wi-Fi version used, 802.11ac. We observe that the channel sensing threshold -82 dBm is too strict to be practical for dense networks, as performance is actually improved due to channel sensing at -82 dBm failing and the -62 dBm threshold being used instead. Future ultra-dense Wi-Fi networks would likely be implemented with an evolved standard with less strict channel sensing thresholds. They would likely also be capable of multi-user MIMO, not taken into account by the dynamic algorithm. A benefit of the algorithm is however that it has a lot of parameters to tune, and it seems there is a good chance that the algorithm can be developed alongside Wi-Fi.

Unfortunately, the algorithm could not be compared to state-of-the-art competitors, due to lack of time to implement them. However, the algorithm was validated using a wide range of different simulations to ensure that the algorithm performs well generally and not just in some specific circumstances.

A benefit of this study is the accuracy of the simulator used. It yielded the insight that the CTMC model might not be so accurate at all for real-world scenarios, where some idealizing assumptions about contention do not hold.

7.4 Conclusions

This thesis started by motivating the need for RRM algorithms for dense Wi-Fi deployments. We continued by reviewing the Wi-Fi standard. We performed a literature review on existing state-of-the art RRM algorithms and presented a new dynamic algorithm. We validated the algorithm by simulations and discussed the results, including how several network properties affect the benefits of dynamically managing each type of RRM parameter.

In the simulations, the dynamic algorithm turned out to be widely applicable, offering good performance in all scenarios, even in extremely dense ones. Despite this, several suboptimalities were found. Firstly, a notable observation was that the CTMC model for channel share is inaccurate for actual Wi-Fi. In the literature, the model is considered accurate because it has been validated mainly with idealized simulators. Secondly, simulated annealing turned out to be a tricky optimization method for its task, since parameters were required to be found automatically for new search instances and to the author's best knowledge no robust methods exist for the task. Thirdly, there remains a plethora of other model parameters which could not be properly optimized in the scope of this thesis. A conclusion that follows is that the chosen algorithm approach has the drawback of being quite work-heavy.

The goal was to develop an algorithm with hardware, software and standard requirements such that they could realistically be satisfied by commercial Wi-Fi in the near future. In section 5.1 we discussed that the dynamic algorithm mostly succeeds in this. The worst shortcoming of the algorithm in this regard is that it wants to enforce equal transmit power density, which clients might not comply with because they do not support the transmit power control feature or because they want to save power.

The dynamic algorithm was concluded to be computationally efficient enough up to moderately dense networks. However, the algorithm does not scale well into extremely dense networks.

There are many ways the algorithm could be improved in future work. Perhaps most importantly, the CTMC model could be replaced with approximating channel share of a node as being the inverse of the number of neighbors it contends with, including itself. According to the results, this would not decrease model accuracy. In addition to the poor performance of the CTMC model, there is another explanation for this result. It is that the algorithm almost always chooses to form contention graphs with diameter at most 1, in which case the CTMC and the simple model yield an equal estimate. This algorithm behavior seems justified by the fact that contention graphs with a diameter of two or more often lead to some node receiving little channel share because it does not belong to any maximum independent set of the contention graph.

Thus a future improved algorithm could approximate channel share with the simple model, but give a utility penalty for contention graphs with diameter over 1, i.e. contention graphs whose connected components are not cliques. This would simultaneously counter unfairness and make the estimates of the model accurate. Solutions with such contention graphs seem to be readily available due to the large number of channels in the 5 GHz band. Simpler channel share estimation would also make the algorithm significantly lighter computationally, helping it scale to even more dense settings.

To solve the difficulties with simulated annealing, other local discrete search metaheuristics could be tried. These include hill climb with restarts as well as tabu search, and they seem to include less hyperparameters that need to be adjusted depending on the problem instance.

Another possible direction of future work is to expand the validation setting. This could be done by abandoning the full buffer assumption or even by moving on from simulations to use a real Wi-Fi testbed.

Altogether, the algorithm approach chosen in this work seems very promising. It has a lot of potential to be developed further and to grow alongside the evolving Wi-Fi standard.

Bibliography

[1] D. López-Pérez, M. Ding, H. Claussen, and A. H. Jafari, "Towards 1 Gbps/UE in cellular systems: Understanding ultra-dense small cell deployments," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2078–2101, 2015.

[2] Qualcomm staff, "Rising to meet the 1000x mobile data challenge [online]." Available: `https://www.qualcomm.com/media/documents/files/rising-to-meet-the-1000x-mobile-data-challenge.pdf` [Accessed: July 1, 2019], 2012.

[3] I. Broustis, K. Papagiannaki, S. V. Krishnamurthy, M. Faloutsos, and V. P. Mhatre, "Measurement-driven guidelines for 802.11 WLAN design," *IEEE/ACM Transactions on Networking*, vol. 18, no. 3, pp. 722–735, 2010.

[4] D. D. Coleman and D. A. Westcott, *CWNA: Certified Wireless Network Administrator Official Study Guide.* Sybex, 3 ed., 2014.

[5] D. A. Westcott, D. D. Coleman, P. Mackenzie, and B. Miller, *CWAP: Certified Wireless Analysis Professional Official Study Guide.* Sybex, 2011.

[6] B. P. Lathi, *Modern Digital and Analog Communication Systems.* Oxford University Press, 4 ed., 2010.

[7] P. Mogensen, W. Na, I. Z. Kovács, F. Frederiksen, A. Pokhariyal, K. I. Pedersen, T. Kolding, K. Hugl, and M. Kuusela, "LTE capacity compared to the Shannon bound," in *Proceedings of the IEEE 65th Vehicular Technology Conference (VTC)*, pp. 1234–1238, IEEE, 2007.

[8] J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach.* Addison Wesley, 4 ed., 2013.

[9] L. B. Jiang and S. C. Liew, "Hidden-node removal and its application in cellular WiFi networks," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 5, pp. 2641–2654, 2007.

[10] 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.

[11] S. Jang and S. Bahk, "A channel allocation algorithm for reducing the channel sensing/reserving asymmetry in 802.11ac networks," *IEEE Transactions on Mobile Computing*, vol. 14, no. 3, pp. 458–472, 2015.

[12] R. Diestel, *Graph Theory*, vol. 173 of *Graduate Texts in Mathematics*. Springer, 5 ed., 2016.

[13] R. Durrett, *Essentials of Stochastic Processes*. Springer Texts in Statistics, Springer, 2 ed., 2012.

[14] L. Leskelä, "Stokastiset prosessit [lecture notes]." Available: `http://math.aalto.fi/~lleskela/papers/Leskela_2015-10-14_Stokastiset_prosessit.pdf` [Accessed: July 1, 2019], 2015. Aalto University course MS-C2111 Stokastiset prosessit.

[15] F. P. Kelly, *Reversibility and Stochastic Networks*. Cambridge University Press, revised ed., 2011.

[16] B. Nardelli and E. W. Knightly, "Closed-form throughput expressions for CSMA networks with collisions and hidden terminals," in *Proceedings of the 31st Annual IEEE International Conference on Computer Communications (INFOCOM)*, pp. 2309–2317, IEEE, 2012.

[17] 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 International Conference on Emerging Networking Experiments and Technologies (CoNEXT)*, pp. 11:1–11:12, ACM, 2011.

[18] 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.

[19] R. Boorstyn, A. Kershenbaum, B. Maglaris, and V. Sahin, "Throughput analysis in multihop CSMA packet radio networks," *IEEE Transactions on Communications*, vol. 35, no. 3, pp. 267–274, 1987.

[20] 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*, vol. 9, no. 9, pp. 1319–1331, 2010.

[21] M. Durvy and P. Thiran, “A packing approach to compare slotted and non-slotted medium access control,” in *Proceedings of the 25st Annual IEEE International Conference on Computer Communications (INFOCOM)*, pp. 1–12, IEEE, 2006.

[22] R. Laufer and L. Kleinrock, “On the capacity of wireless CSMA/CA multihop networks,” in *Proceedings of the 32st Annual IEEE International Conference on Computer Communications (INFOCOM)*, pp. 1312–1320, IEEE, 2013.

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

[24] I. M. Bomze, M. Budinich, P. M. Pardalos, and M. Pelillo, “The maximum clique problem,” in *Handbook of Combinatorial Optimization*, pp. 1–74, Springer, 1999.

[25] A. Baid and D. Raychaudhuri, “Understanding channel selection dynamics in dense Wi-Fi networks,” *IEEE Communications Magazine*, vol. 53, no. 1, pp. 110–117, 2015.

[26] J. Riihijarvi, M. Petrova, and P. Mahonen, “Frequency allocation for WLANs using graph colouring techniques,” in *Proceedings of the Second Annual Conference on Wireless On-demand Network Systems and Services (WONS)*, pp. 216–222, IEEE, 2005.

[27] S. Chieochan, E. Hossain, and J. Diamond, “Channel assignment schemes for infrastructure-based 802.11 WLANs: A survey,” *IEEE Communications Surveys & Tutorials*, vol. 12, no. 1, pp. 124–136, 2010.

[28] B. Kauffmann, F. Baccelli, A. Chaintreau, V. Mhatre, K. Papagiannaki, and C. Diot, “Measurement-based self organization of interfering 802.11 wireless access networks,” in *Proceedings of the 26th Annual IEEE International Conference on Computer Communications (INFOCOM)*, pp. 1451–1459, IEEE, 2007.

[29] Y. Bejerano, S.-J. Han, and L. E. Li, “Fairness and load balancing in wireless LANs using association control,” in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pp. 315–329, ACM, 2004.

[30] A. Mishra, V. Brik, S. Banerjee, A. Srinivasan, and W. A. Arbaugh, “A client-driven approach for channel management in wireless LANs,” in *Proceedings of the 25st Annual IEEE International Conference on Computer Communications (INFOCOM)*, pp. 1–12, IEEE, 2006.

[31] R. Murty, J. Padhye, R. Chandra, A. Wolman, and B. Zill, “Designing high performance enterprise Wi-Fi networks,” in *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, pp. 73–88, USENIX, 2008.

[32] V. P. Mhatre, K. Papagiannaki, and F. Baccelli, “Interference mitigation through power control in high density 802.11 WLANs,” in *Proceedings of the 26th Annual IEEE International Conference on Computer Communications (INFOCOM)*, pp. 535–543, IEEE, 2007.

[33] N. Ahmed and S. Keshav, “SMARTA: a self-managing architecture for thin access points,” in *Proceedings of the Second Conference on Future Networking Technologies (CoNEXT)*, pp. 9:1–9:12, ACM, 2006.

[34] 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 on Emerging Networking Experiments and Technologies (CoNEXT)*, pp. 27:1–27:12, ACM, 2010.

[35] M. K. Panda and A. Kumar, “Modeling multi-cell IEEE 802.11 WLANs with application to channel assignment,” in *Proceedings of the 7th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, pp. 1–10, IEEE, 2009.

[36] L. Massoulié and J. Roberts, “Bandwidth sharing: objectives and algorithms,” in *Proceedings of the 18th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, vol. 3, pp. 1395–1403, IEEE, 1999.

[37] S. Niskanen and P. R. J. Östergård, “Cliquer user’s guide, version 1.0,” tech. rep., Communications Laboratory, Helsinki University of Technology, Espoo, Finland, 2003.

[38] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[39] V. Černỳ, "Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm," *Journal of Optimization Theory and Applications*, vol. 45, no. 1, pp. 41–51, 1985.

[40] D. Bertsimas and J. Tsitsiklis, "Simulated annealing," *Statistical Science*, vol. 8, no. 1, pp. 10–15, 1993.

[41] F. Glover, "Tabu search - part I," *ORSA Journal on Computing*, vol. 1, no. 3, pp. 190–206, 1989.

[42] S. Tasharrofi, "Lecture 4: Intro to complete and local search methods [slides]." Available: `https://mycourses.aalto.fi/pluginfile.php/273289/course/section/60682/cs-e3200_lect04_slides.pdf` [Accessed: July 1, 2019], 2016. Aalto University course CS-E3200 Discrete Models and Search.

[43] G. Breit *et al.*, "IEEE document 802.11-09/0308r3: TGac channel model addendum [online]." Available: `https://mentor.ieee.org/802.11/dcn/09/11-09-0308-03-00ac-tgac-channel-model-addendum-document.doc` [Accessed: July 1, 2019], 2009.

[44] V. Erceg, L. Schumacher, P. Kyritsi, *et al.*, "IEEE document 802.11-03/940r4: TGn channel models [online]." Available: `https://mentor.ieee.org/802.11/dcn/03/11-03-0940-04-000n-tgn-channel-models.doc` [Accessed: July 1, 2019], 2004.

[45] E. Tuomaala and H. Wang, "Effective SINR approach of link to system mapping in OFDM/multi-carrier mobile network," in *Proceedings of the Second Asia Pacific Conference on Mobile Technology, Applications and Systems*, pp. 5:1–5:5.

[46] S. N. Chiu, D. Stoyan, W. S. Kendall, and J. Mecke, *Stochastic Geometry and its Applications*. John Wiley & Sons, 3 ed., 2013.

Appendix A

Additional results

In this appendix we list additional result data that helps in interpreting the simulation results. This data is presented in figures A.1, A.2, A.3, A.4, A.5, A.6 and A.6. Each of these represents one series of simulations, and there is data describing averages for transmit power, bandwidth, frame failure rate, number of simultaneous 20 MHz transmissions (i.e. the number of active transmissions on a 20 MHz channel summed over all 20 MHz channels), dynamic algorithm utility and throughput.

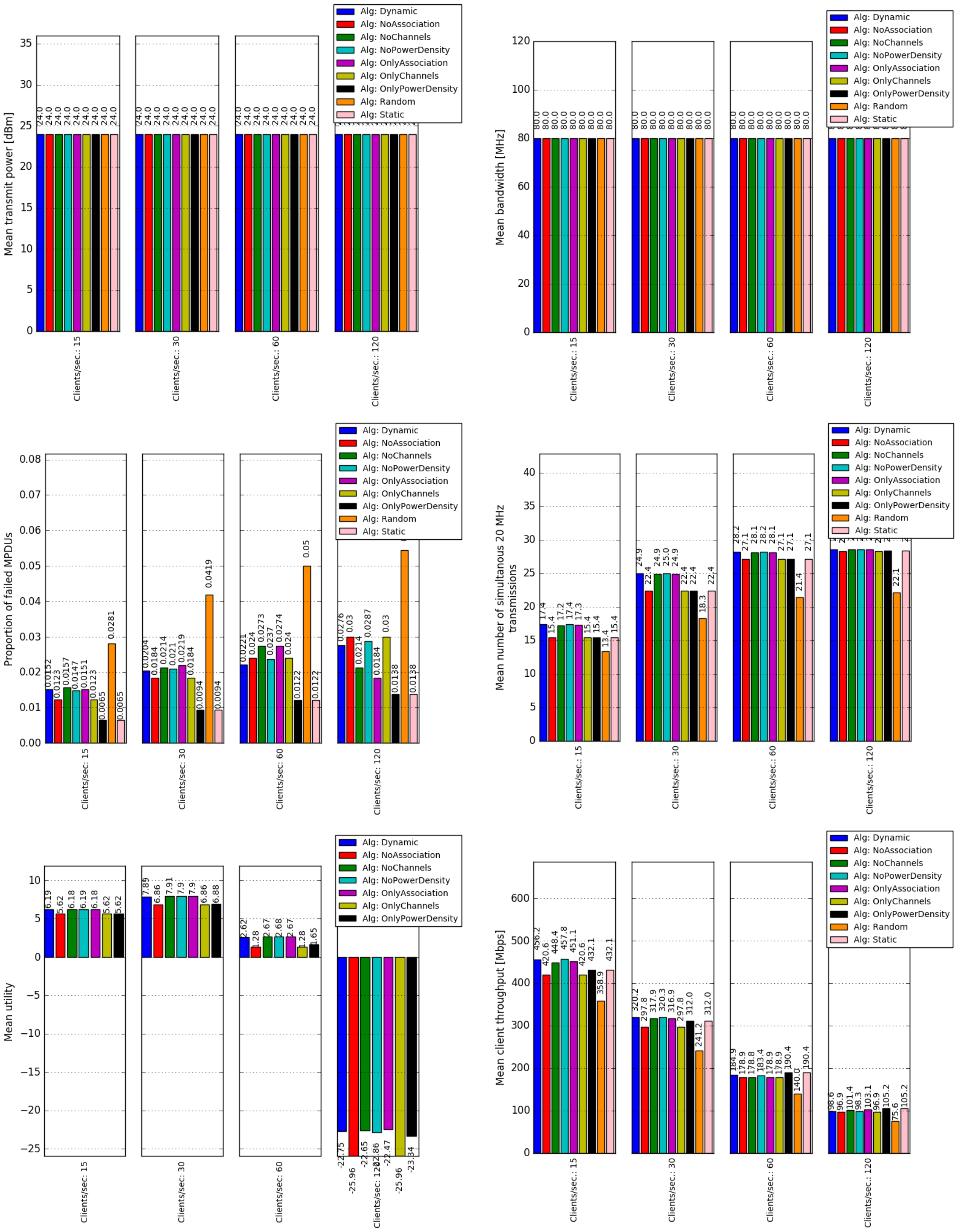

Figure A.1: Additional results for the simulations with 8 APs.

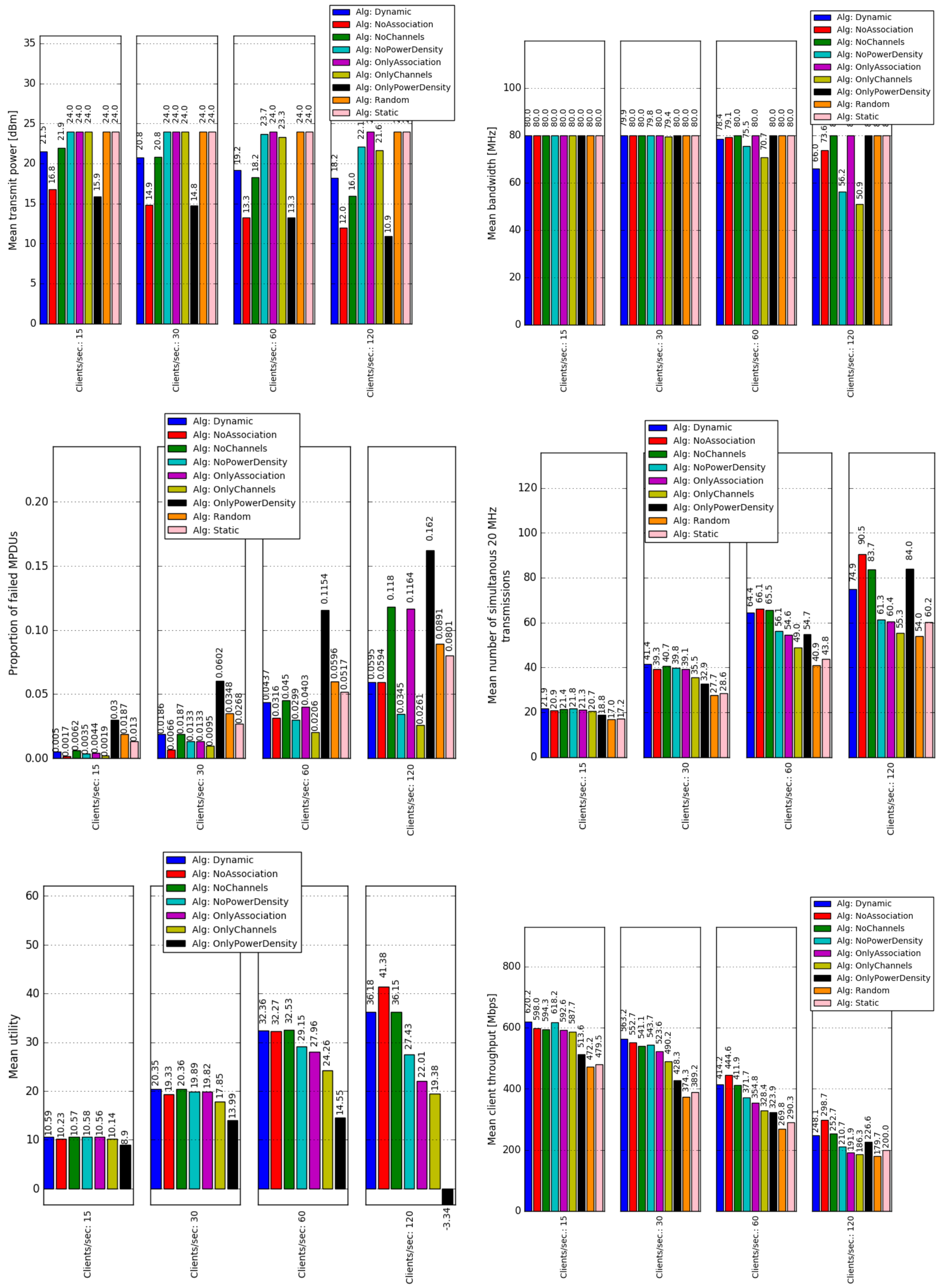

Figure A.2: Additional results for the simulations with 64 APs.

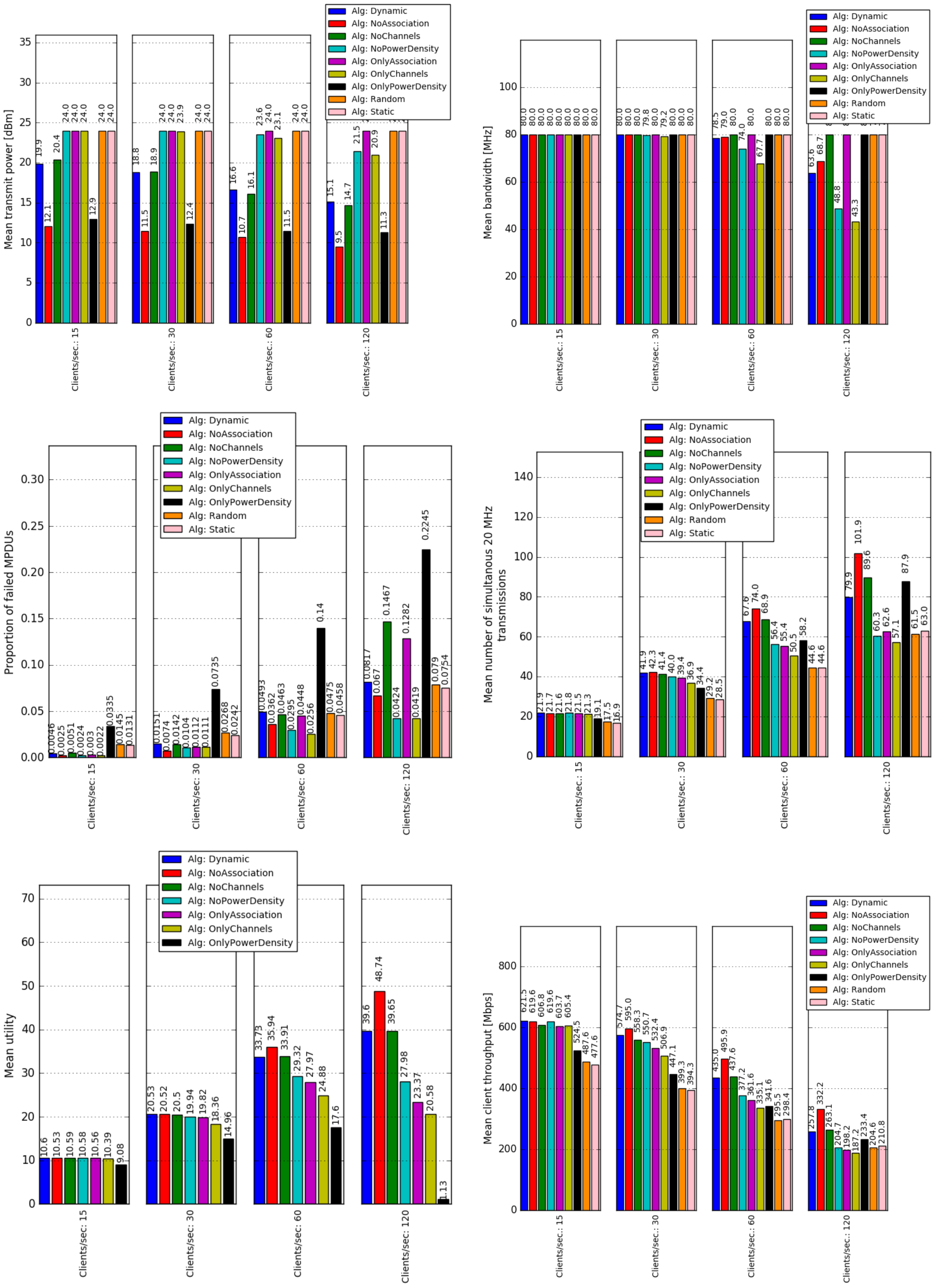

Figure A.3: Additional results for the simulations with 256 APs.

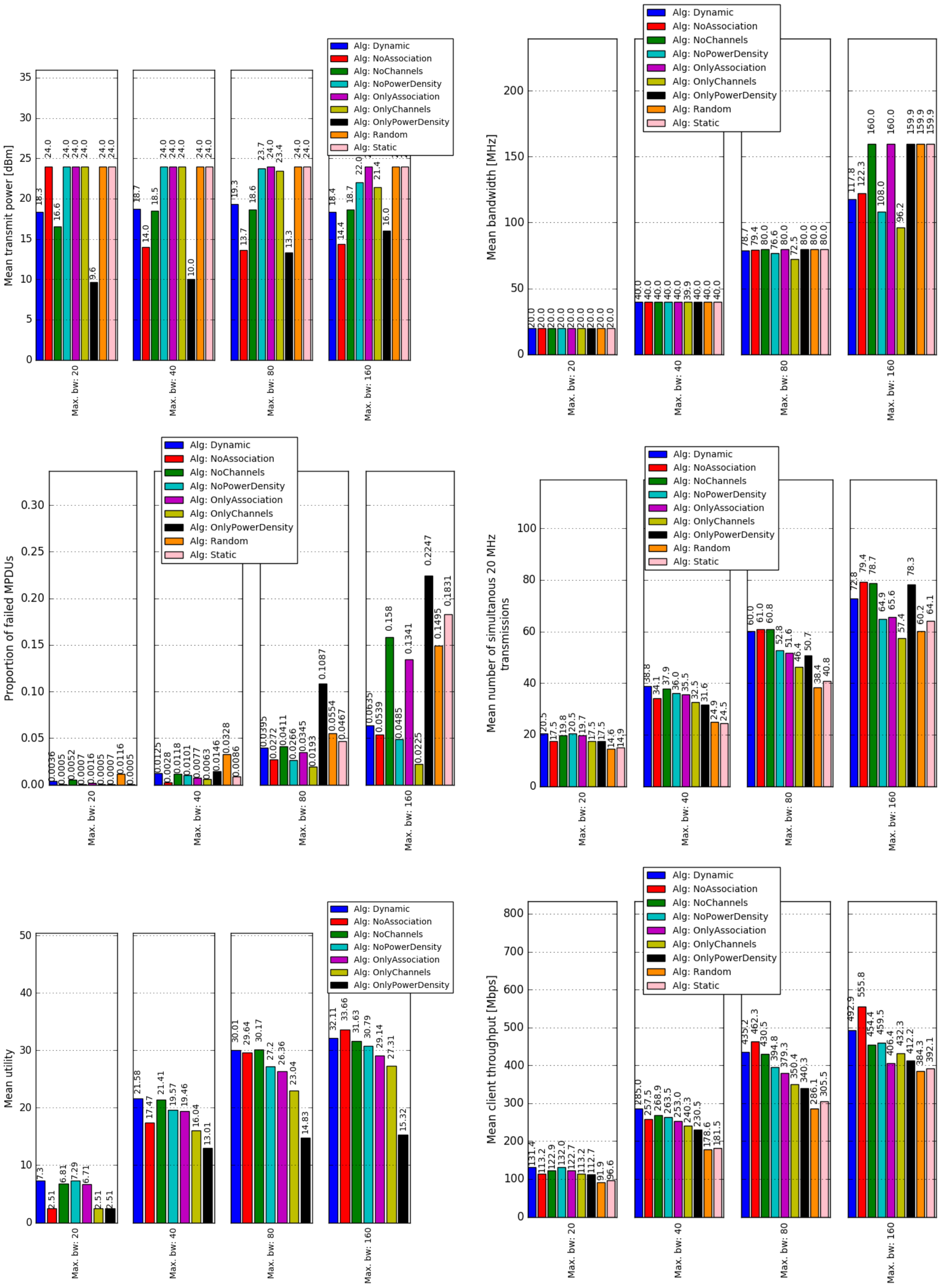

Figure A.4: Additional results for the simulations where bandwidth was varied.

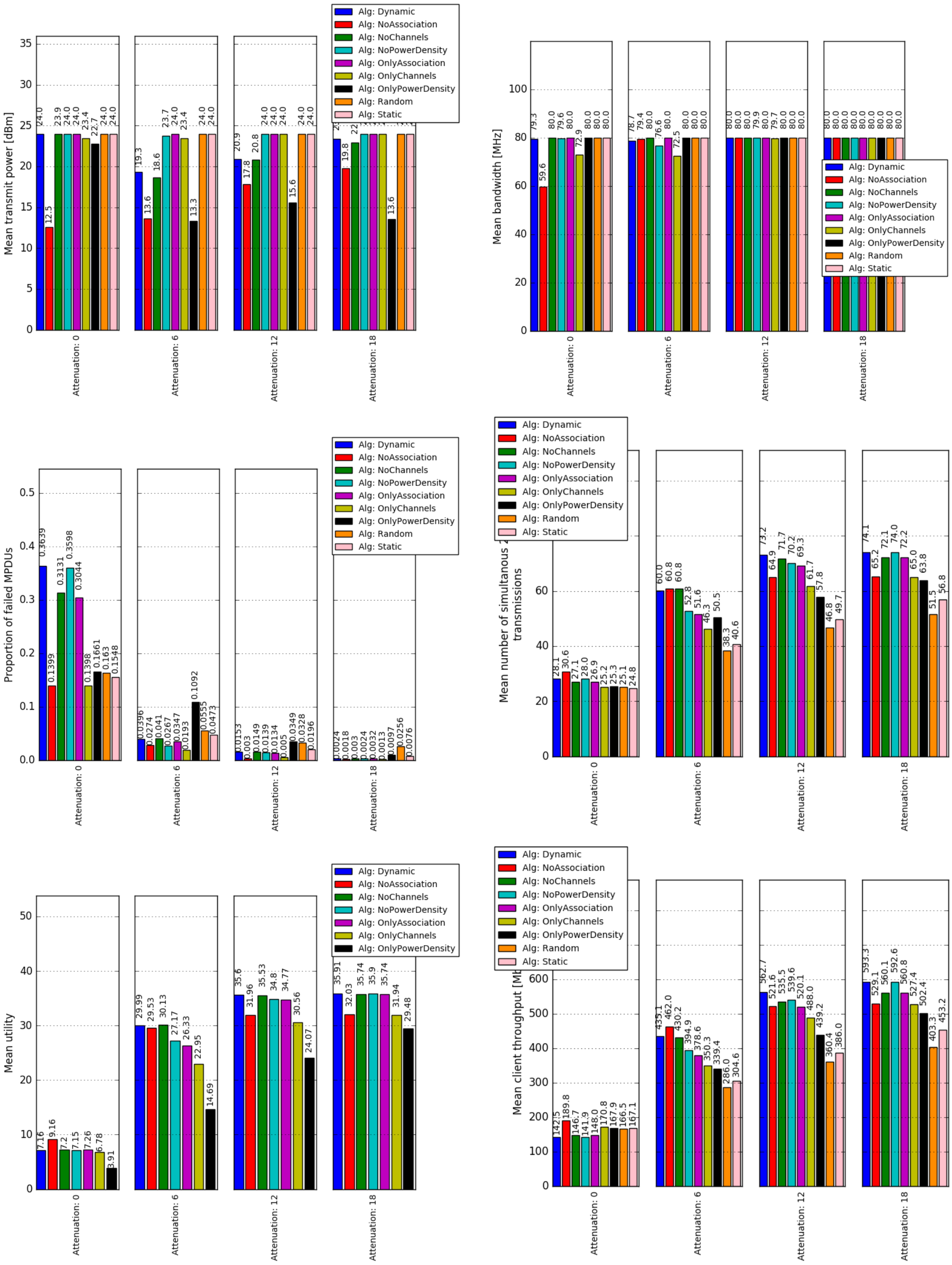

Figure A.5: Additional results for the simulations where wall attenuation was varied.

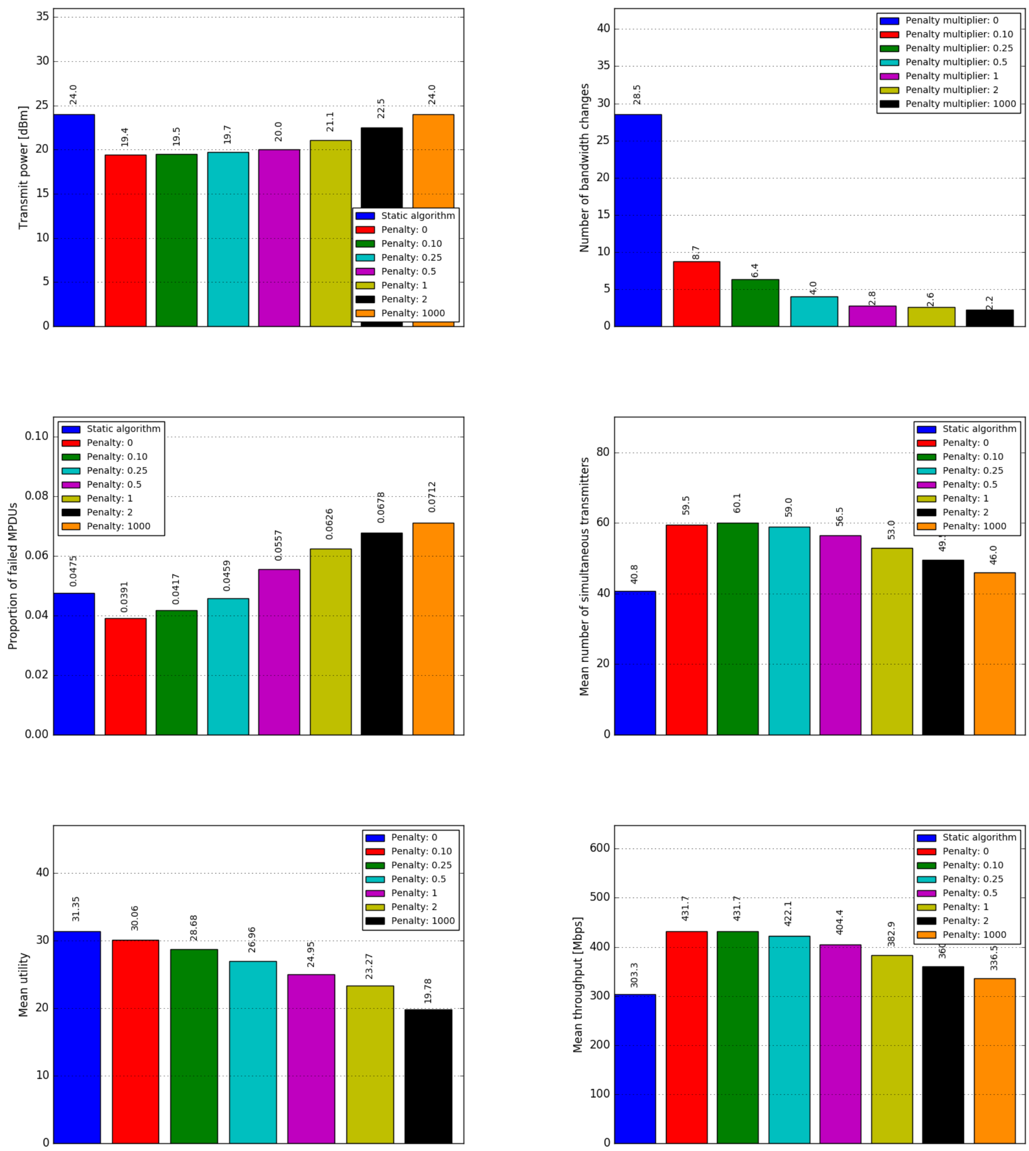

Figure A.6: Additional results for the simulations where change penalties were varied.

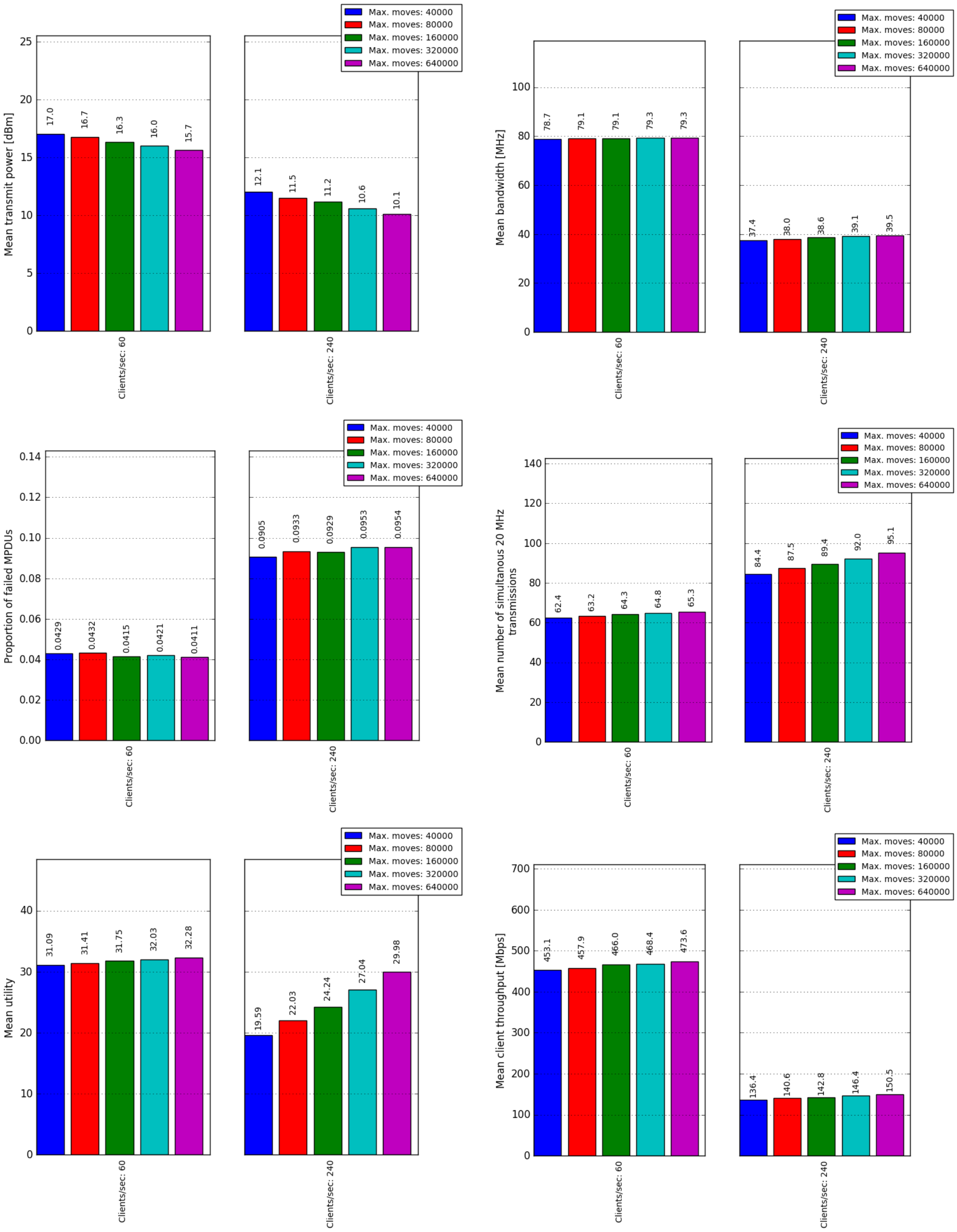

Figure A.7: Additional results for the simulations where the search move limit was varied.