Автономная некоммерческая организация высшего образования
«Университет Иннополис»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(БАКАЛАВРСКАЯ РАБОТА)
по направлению подготовки
09.03.01 - «Информатика и вычислительная техника»

GRADUATION THESIS
(BACHELOR'S GRADUATION THESIS)
Field of Study
09.03.01 – «Computer Science»

Направленность (профиль) образовательной программы
«Информатика и вычислительная техника»
Area of Specialization / Academic Program Title:
«Computer Science»

Тема / Topic	**Проектирование и реализация алгоритма адаптивного управления радиоресурсами в беспроводных сетях стандарта IEEE 802.11 с несколькими точками доступа / Design and Implementation of Adaptive Radio Resource Management algorithm in IEEE 802.11 multi-AP WLANs**

Работу выполнил / Thesis is executed by	**Булгаков Артем Владиславович / Artem Bulgakov**	подпись / signature
Руководитель выпускной квалификационной работы / Supervisor of Graduation Thesis	**Сейтназаров Шынназар Оралбаевич / Shinnazar Seytnazarov**	подпись / signature

Иннополис, Innopolis, 2023

Contents

List of Tables

List of Figures

Abstract

As IEEE 802.11-compatible ("Wi-Fi") Wireless Local Area Networks (WLAN) become ubiquitous, the rising density of WLAN deployments leads to congestion of frequency bands and performance degradation due to interference. Thus, Radio Resource Management (RRM) algorithms for mitigating those problems become in demand, especially for large WLAN deployments.

While several commercial RRM solutions exist, their implementation details remain undisclosed. Moreover, such solutions are incompatible between devices produced by different vendors. At the same time, most of the previous studies have considerable obstacles for production usage.

In this study, I carry out theoretical analysis of *RRMGreedy*, a greedy centralized RRM algorithm used by a major Russian telecommunications vendor. I show that *RRMGreedy* possess a number of flaws and tends to yield suboptimal results. After identifying flaws of the algorithm, I propose several improvements for *RRMGreedy*. The updated version of the algorithm is referred to as *RRMGreedy++*. This algorithm adjusts frequency channel and transmission power parameters of access points based on physical-layer metrics, and uses WLAN Controller (WLC) as a central entity to gather data and perform computations.

Additionally, this study develops a novel framework for the ns-3 network simulator that is used to implement RRMv2 alongside with other RRM algorithms. Simulation results demonstrate that *RRMGreedy++* achieves up to a 12% higher throughput on 2.4 GHz IEEE 802.11n networks and is able to improve signal quality, measured in SNR (Signal-to-Noise Ratio) by 10%.

Chapter 1

Introduction

I Importance of the topic

Nowadays, wireless local area networks (WLAN) implementing IEEE 802.11 standards, commonly known under "Wi-Fi" brand, become an increasingly popular solution for last-mile internet access with a diverse population of users, starting from home Wi-Fi routers up to large campus- and city-scale WLANs with coverage areas reaching several square kilometers. As a result, the density of Wi-Fi network increases, so the frequency band allocated for 802.11 networks becomes more congested, which leads to interference and signal cancellation between different WLANs, resulting in network performance degradation. Moreover, other appliances operating on frequencies that overlap with Wi-Fi band, undermining the performance of WLANs. To meet current bandwidth and latency expectations of modern network applications, such as video streaming, cloud computing, and video conferencing, the wireless network must be able to provide sufficient capacity to all clients. In this light, proper radio resource management becomes crucial for operating wireless networks.

The problem of managing radio resources is studied extensively in the context of **cellular networks**, which are characterized by extensive frequencies reuse, large number of clients and large coverage areas spanning multiple kilometers, so the proper spectrum management is vital for operation of cells. This problem breaks down to the following: given a set of access points (or base stations in cellular terminology) B, which can communicate over a set of channels C, with a maximum transmit power of P_{max}, establish a radio link between a client device and an access point by assigning it a triplet (b, c, p), where $b \in B$, $c \in C$, $p \leq P_{max}$. Essentially, RRM algorithms aim to provide such assignments that maximize the overall network performance.

II   Limitations of existing approaches

A growing demand for Radio Resource Management (RRM) solutions, especially for large enterprise-grade multi-AP WLAN deployments, has led to the development of multiple commercial solutions, such as Cisco Radio Resource Management (RRM) [1], Aruba Adaptive Radio Management (ARM) [2], and others. However, existing solutions offered by major vendors are proprietary, so their source code, used algorithms, and details of operation are not disclosed. In the same time, most studies on RRM have only focused on cellular networks, while RRM in 802.11 networks received much less attention. Many existing works on RRM in 802.11 networks have applicability problems, since real-world hardware poses constraints on what metrics can be retrieved from the wireless interface and which physical and link-level parameters can be adjusted.

III   Research gap

Therefore, I can identify a research gap: the need for an open-source RRM (Radio Resource Management) algorithm that matches the performance of proprietary solutions and is feasible for real-world WLAN deployments.

This study aims to fill this gap by proposing a new RRM algorithm that, unlike existing market solutions, is publicly available, and at least as effective, while being practical and applicable to real-world WLAN deployments.

IV   Contribution and significance of the study

By analyzing theoretical works on 802.11 RRM, I come up with suitable optimization approach that combines both Adaptive Channel Selection (ACS) and Transmit Power Control (TPC) techniques to improve RF spectrum situation EM compatibility and channel reuse, which in turn leads to improved capacity of a wireless network.

The RRM approach that I use adheres with the centralized architecture, introduced by Cisco as Unified Wireless Network (UWN) [3], a highly centralized wired-wireless architecture controlled by a Wireless LAN Controller (WLC). This approach allows for a more efficient spectrum management, as the WLC can collect data from all access points and make decisions based on the global view of the network, which is not possible in a distributed architecture, where each access point makes decisions independently.

The rest of this Thesis is structured as follows: Chapter 2 reviews existing academic research on managing radio resources and publicly available information about proprietary RRM solutions; in Chapter 3 I analyze RRMGreedy, iden-

tify its limitations and propose improvements that result in the new version of the algorithm called *RRMGreedy++*; in Chapter 4, I present implementation and evaluation results using the ns-3 network simulator; Chapter 5 contains the results and discussion.

Chapter 2

Literature Review

The purpose of this chapter is to explore existing approaches on Radio Resource Management (RRM) in IEEE 802.11 networks, including surveying what is proposed as a part of the 802.11 standard itself, what research has been done and what is offered by existing commercial solutions. This chapter is organized as follows:

- Section 2.1 briefly review the fundamentals of Radio Frequency (RF) communications, and IEEE 802.11 standard for wireless LAN;

- Section 2.2 provides an overview how radio resource management is facilitated within the IEEE 802.11 standard;

- Section 2.3 provides a synthesis on previous research in radio resource management;

- Section 2.4 provides an overview of proprietary RRM solutions from major vendors;

- Section 2.5 summarizes the chapter.

Table **??** contains the list of definitions used in this chapter.

I Overview of IEEE 802.11 standard

Throughout this Thesis, I refer to domain-specific terms, whose definitions are given in Table **??**. Below in this chapter, I briefly review IEEE 802.11 standard and highlight the aspects most relevant for this study.

A. *Transmission medium*

The primary medium for communications in IEEE 802.11 are electromagnetic (EM) radio-frequency (RF) waves operating within the microwave range [4].

B. *Frequency band*

Main frequency bands are: 2.4 GHz band introduced by 802.11b, 5 GHz band introduced by 802.11a and 6 GHz, introduced by 802.11ax. This study focuses on first two bands as used the most frequently. Below I describe those bands in more details.

Most of the 802.11 amendments, including b,g,n, and partially ax, operate at unlicensed 2.4 GHz ISM (Industrial, Scientific, Medical) RF band [4], [5]. Using an unlicensed frequency band, however, introduces multiple challenges: the radio spectrum becomes congested with non-802.11 sources, such as microwave ovens, Bluetooth Personal Area Networks (PAN), cordless phones etc. [4], [5]. Moreover, 2.4 GHz signals can propagate through solid obstructions like walls, doors, and windows better than signals operating on higher-frequency ones [4]. This property can provide better coverage and signal quality for clients, although can

cause interference for neighboring WLANs, which will in turn lead to degradation of signal quality.

The 2.4 GHz ISM band is split into 14 channels. Depending on local regulations, number of possible channels can vary, but in general channels 1-11 are available at every region. Assuming 20 MHz channel width, each channel is characterized by its *center frequency*, ± 10 MHz, with 5 MHz width between two adjacent centers, i.e, the channels *overlap*. Channel 1 has central frequency 2.412 GHz, Channel 14 2.484 GHz. Thus, for channels to be non-overlapping, they must have at least 5 channels or 25 MHz in between. Such non-overlapping channels are 1, 6, 11, with central frequencies 2.412, 2.437, and 2.462 MHz, respectively.

The 5 GHz U-NII (Unlicensed National Information Infrastructure) series of bands is used by 802.11a/802.11ac/802.11ax amendments. Unlike the 2.4 GHz band, channels do not overlap. On the end of each band *guard band* as an additional measure to avoid interference. Combined, bands U-NII-1, U-NII-1, U-NII-2A, U-NII-2C, U-NII-3 provide twenty-five 20 MHz or twelve 40 MHz non-overlapping channels [4]. However, some channels may not be available in different regions, since this band can also be used by military and weather radars. The first channel from U-NII-1 band has number 36.

C. Signal quality and its metrics

Thus, the presence of physical obstructions, background noise and interference from other access points urges us to explore possible measurements and metrics for a wireless signal quality. Below, I briefly describe the most widely used quantities:

A measure widely used in RF engineering and employed by Wi-Fi vendors

is **Signal-to-Noise Ratio (SNR)**, which is defined as a ratio between the received signal power and the power of background noise:

$$SNR = \frac{P_{signal}}{P_{noise}} \tag{2.1}$$

Since SNR is essentially a difference in power, which is measured in Watts, in practice it is measured in a relative unit on a logarithmic scale called **decibel (dB)** [1,2]:

$$SNR_{dB} = 10 \log_{10} \frac{P_{signal}}{P_{noise}} \tag{2.2}$$

In recent years, **Signal-to-Interference-Plus-Noise ratio (SINR)** measurement have become a more widespread measurement of wireless networksśignal quality. Similarly, it is defined as:

$$SINR = \frac{P_{signal}}{P_{noise} + P_{interf}} \tag{2.3}$$

where P_{signal} is the power of the signal of interest, and P_{interf} is the power of interfering signals. By considering interference from other 802.11 devices, which is typically a dynamic quantity that changes rapidly over time unlike background noise, SINR describes EM spectrum situation more accurately.

Received Signal Strength Indicator (RSSI) relative measure of signal strength in range from 0 to 255, where 0 is the weakest signal a receiver is able to sense. The exact correspondence between RSSI and received signal power is implementation-specific and is left on behalf of hardware manufacturers [4].

D. Radio Resource Management

The scarcity of available frequency bands in the time of growing demand for wireless connectivity has led to the development of methods called *spectrum management* or *radio resource management (RRM)*. Most of research on RRM is focused on cellular networks, where coverage area of base stations spans across multiple kilometers, and the number of clients for one station can reach several thousands, so proper spectrum management is vital for operation of cells. However, from the physical layer perspective, the radio situation in 802.11 networks is similar. As described in [6], given a wireless network with a set of access points **B**, which can communicate over a set of channels **C**, with a maximum transmit power of P_{max}, establishing a radio link between a client device and an access point requires from the wireless infrastructure to assign:

1. An access point $b \in$ **B**;

2. A frequency channel $c \in$ **C**;

3. A transmission power level $p \leq P_{max}$.

Obviously, channel and transmission power are *global* for a given access point in a sense that all its other clients will have to adjust their parameters correspondingly: switch the operating channel and deal with the new received signal strength from their AP. In Wi-Fi, the first requirement is usually managed by the client itself: user chooses SSID they wish to use, and in case if multiple APs serve the same SSID, a client device associates with AP having the strongest signal available. Later, a client can switch to another access point within the same extended service set via *roaming* methods, such as *Fast Basic Service Set (BSS) Transition* defined in 802.11r **80211r2008IEEE**. The roaming decision is ultimately made

by a client device, which sends a reassociation request to start the roaming process [4]. The access point, however, can force a client to find another access point by sending a deauthentication frame, or moving to another channel without notifying. The second and third requirements are a part of current AP configuration and a subject to change. A client discovers current operating channel of APs by tuning on each available channel in a succession, while transmission power only can be estimated by measuring received signal strength. Thus, **the goal of a radio resource allocation algorithm is to optimize spectrum usage within a WLAN via assigning an operating channel and a transmission power level to each access point in a way that maximizes the overall network performance.**

As it will be shown in Section 2.2, the 802.11 standard does not provide any algorithms for channel and transmission power assignment, however, some amendments introduce methods for measurement, signaling and radio adjustment that can be used for RRM purposes.

Note that related researches and commercial solutions introduce many similar terms for the same procedure of channel change that can use different algorithms and slightly vary according to specifics of their application: *Frequency Selection*, *Frequency Planning*, *Channel Selection*, *Channel Planning*, *Channel Assignment*, etc. Adjustment of transmission power is usually In this study, we use those terms interchangeably.

II   Radio Resource Management in IEEE 802.11

As described in Section 2.1.4, a radio resource management algorithm aims to optimize network capacity through optimizing spectrum usage by adjusting two parameters: frequency and transmission power of each access point. The very

need for an RRM algorithm comes from situation called *Overlapping Basic Service Sets* in the IEEE 802.11 standard, when there are two basic service sets operating on the same channel within reach of each other. This situation leads to co-channel interference, which degrades network performance. In this section, I provide an overview of methods and techniques provided by IEEE 802.11 standard that can assist in radio resource management.

A.   IEEE 802.11h

To comply with legal requirements on transmissions in 5 GHz, 802.11h-2003 [7] amendment (*Spectrum and Transmit Power Management Extensions*) was introduced. Since the goal of this amendment is to prevent legacy 802.11a 5GHz APs from interference with radars, 802.11h is not oriented for optimizing capacity of a wireless network. However, 802.11h introduces [8] spectrum management methods, namely, *Dynamic Frequency Selection (DFS)*, facilitating automatic change of AP's operating frequency, and *Transmit Power Control (TPC)*, adjusting the power of AP's transmitter. Those methods, however, do not implement RRM in the sense of optimizing network capacity, but rather are aimed at preventing interference with radars operating on the 5 GHz band. 802.11h describes procedures for: quieting the channel to detect presence of a radar, switching to another channel if a radar detecting, advertising the channel switch to client stations (STAs). However, the description of a radar detection procuedre is beyond the scope of 802.11 standard [7]. However, the frame types and measurement reporting mechanisms introduced in 802.11h can be considered as a foundation for further research on RRM algorithms.

B. IEEE 802.11k

The amendment 802.11k-2008 **ieee80211k** (*Radio Resource Measurement*, not to be confused with *Radio Resource Management*) improves the performance of roaming. Roaming is a process of station moving from one access point to another [4]. The amendment 802.11k improves roaming by allowing stations to request from access points various reports, such as *neighbor reports* for discovering possible access points that STA can roam to, *link measurements* to estimate how well AP can hear the station etc. This allows stations to reduce power consumption, speed up roaming and decrease power consumption and airtime usage spent on sending probe requests to each channel when trying to find another AP to roam. However, this amendment is also not aimed at optimizing network capacity, but rather at improving roaming performance.

C. IEEE 802.11ax

The amendment 802.11ax **80211ax** (*High Efficiency WLAN*) introduces means to address OBSS problem. The amendment introduces *Basic Service Set (BSS) Coloring*, a method to distinguish between different BSSs operating on the same channel. It works as following: each BSS is assigned with *color*, a 6-bit value spanning from 1 to 63, that resides in the PHY header. Devices belonging to the same BSS check this color when demodulating transmissions (i.e., check if this frame is an *intra-BSS*). If it contains the wrong color (*inter-BSS* frame), further demodulation is not performed, thus saving processing time [9]. If an intra-BSS frame is found to use the same color, AP switches to another color. One can see that such solution completely relies on CSMA/CA media access control (MAC) mechanism and is not able to address hidden terminal problem. That is, if AP

providing overlapping BSS is a hidden terminal, the signal will be disrupted near the destination, so the receiving station will not be able to demodulate the signal to inspect the color tag. Thus, this technique does not solve the problem of co-channel interference and does not provide channel planning, so this cannot be considered as a complete RRM solution.

III   Previous Works

Interference from other Access Points poses a serious obstacle [10] for delivering acceptable quality of service in large 802.11 WLAN deployments. Although the 802.11 carrier-sense MAC protocol is designed to be resilient to interference, interference reduces the available airtime and causes loss of frames already sent. Thus, network capacity and performance tend to degrade drastically [11]. For interference between cells within a single WLAN, such problem in principle can be solved by proper **site surveying**, i.e., planning of geographical placement of cells. However, another and the major source of interference are rogue access points (RAPs), which are operated by third parties and in general are not under control of WLAN administrators. According to [10], interference from rogue APs can introduce up to 50% delays in a WLAN. Moreover, the prevalent amount (more than 70%) of rogue APs are stationary [10], so their radio presence can be considered as a constant factor in the WLAN. Note that *rogue* does not imply that those APs are malicious or posing threats other than congesting channels and occuping airtime as a result of their legitimate operation. In this light, attempts to improve spectrum management via channel assignment and transmit power control adjustment algorithms encompass research on Radio Resource Management algorithms. As shown in Section 2.2, the IEEE 802.11 standard provides a limited set of tools

for radio resource managemens, leaving assignment algorithms and policies on behalf of WLAN equipment vendors. As reported in [10], Cisco's RRM software, shipped with Cisco Aironet APs and Cisco WLAN Controller, was able to improve network performance using Dynamic Channel Selection (DFS) and Transmit Power Control (TPC) so that carrier sense interference was responsible for only 5% of network delays. RRM solutions from Cisco and other vendors will be surveyed in Section 2.4. However, since those technologies are proprietary, their implementation details are not disclosed, so they cannot be properly evaluated in an independent research or adopted by third-party vendors. Moreover, such solutions lack interoperability, so it is difficult to use them with networking products from other vendors, which is a major obstacle for large-scale WLAN deployments and leads to vendor lock-in situations. Thus, research on radio resource management algorithms is important for the industry.

A. *Radio Resource Management Approaches*

Since RRM is a broad topic which does not imply a single methodology, approach, or even a definition, classification of RRM algorithms is a challenging task. Most of the papers with the "Radio Resource Management" keyword are focused on problems of cellular networks, such as LTE or 5G. At the same time, the problem formalization, some optimization objective and algorithms can also be used for research in 802.11 networks, but band usage, client management, deployment and operation specifics make most of the proposals inapplicable for 802.11 networks. To the best of my knowledge, no comprehensive survey on RRM in 802.11 WLAN exist. I will refer to [12], which provides detailed overview of previous research, and [13], describing state-of-the-art on RRM. In [12], authors

classify RRM algorithms into three categories:

- *per-cell* approaches seek to optimize the RF situation within the AP's cell coverage. This means that adjustments of radio parameters applied on a cell scale and will be in effect for all stations within the cell. Such classification can be further divided into:

  - *localized (uncoordinated) per-cell*, where each AP performs RRM decisions independently;

  - *centralized per-cell*, where a central entity, such as WLAN Controller, performs RRM decisions for all APs within a WLAN. Some authors refer to this approach as *super-cell* approach [11];

  - *coordinated per-cell*, employing cooperation between APs for making coordinated RRM decisions.

- *per-link* approaches, which optimize the transmission power for a given station;

- *per-flow* approaches, which employ frequency and AP Tx power adjusting to optimize the QoS to the granularity of a given traffic flow within a station, for example, to the flow of a VoIP application.

In fact, a simple localized per-cell RRM is already widely implemented: almost every home Wi-Fi router has the option to select channel automatically. Typically, in this case access point surveys each channel, makes an estimation how congested it is, then switches to the least congested one. This technique is called *least-congested channel scan*, or *least-congested channel search* (LCCS), and the original design uses the number of associated clients as estimation of channel congestion [14]. As analyzed in [15], LCCS has several limitations:

1. LCCS is unable to accurately identify interference scenarios where clients connected to different Access Points (APs) interfere with each other without the APs themselves causing interference. This issue is particularly prevalent in real-world setups where APs are strategically placed to ensure wide coverage while overlapping minimally to avoid coverage gaps;

2. LCCS also falls short in optimizing channel reuse based on the distribution of clients. It fails to account for the interference experienced by clients, thus missing the opportunity for channel reuse strategies based on client locations and densities.

In general, uncoordinated decision-making like LCCS tends to yield suboptimal results. Consider an extreme case, where a number of APs can sense that channel C_i is not congested and make a decision to switch to that channel. As a result, C_i becomes congested, so APs will seek to switch to another channel C_j, where the problem will reoccur. This situation suggests that using a coordinated RRM policy between APs can improve overall WLAN capacity and thus achieve a global optimum.

Research on Transmit Power Control (TPC) methods, which adjust transmission power P_{Tx} to maintain an acceptable Signal-to-Noise-plus-Interference Ratio (SINR), shows potential for enhancing bandwidth. However, criticisms regarding these studies' simplified models, unrealistic experimental setups, and statistically uncertain outcomes suggest the need for further investigation [16], [17]. The effect of IEEE 802.11 roaming on TPC is underexplored, especially in scenarios where APs are part of the same extended service set (ESS), which is more common in enterprise WLANs.

As shown in [18], per-link TPC considerably improves WLAN performance,

achieves more spatial reuse, increases throughput, and able to avoid channel access asymmetry and receiver-side interference (also known as hidden-node problem). However, such approach has certain hardware requirements, namely, *per-packet transmit power control*, a feature available only for a small selection of 802.11 chipsets. In turn, implementing per-flow RRM in standard 802.11 networks requires an advanced framework for identifying specific traffic flows and assessing their Quality of Service (QoS) demands [12]. Therefore, this thesis will concentrate on solutions that are more practical and applicable to the hardware and software currently available on the market.

B.   *Mathematical Models for Radio Resource Management*

Building on the definition of the Radio Resource Management (RRM) problem introduced in 1, we consider a network composed of a set of base stations (access points) denoted as B, capable of operating over a collection of channels C, each with a maximum transmission power limit P_{max}. The core objective of RRM is to establish a radio link between a client device and an access point by assigning a triplet (b, c, p), where $b \in B$ signifies the base station, $c \in C$ the channel, and $p \leq P_{max}$ the transmission power, so that network capacity is maximized.

The RRM problem, thus, decomposes into three crucial tasks:

- *Client Assignment* — allocating a base station (access point) to a client;

- *Adaptive Channel Selection* — determining the optimal frequency (channel) for client communication;

- *Transmit Power Control* — setting the appropriate transmission power for client communication.

Client Assignment is typically handled by the 802.11 client through roaming decisions, with amendments like 802.11k/r/v designed to enhance and expedite the process of switching to an access point that offers superior service quality. This aspect, therefore, lies beyond the scope of this study.

Channel allocation and transmit power selection, though extensively studied, are often addressed as separate entities in the literature. Combining these factors introduces complexity, as their objectives can conflict. For example, if the objective is to minimize interference, it can be achieved with minimizing the transmission power. However, such outcome probably does not satisfy coverage and quality of service requirements. On the other hand, considering channel and transmit power together can also be troublesome, since change in one variable would change the overall RF situation, and the algorithm would not converge.

Another aspect is choice of metrics. Although the most intuitive and desired metrics are high-level ones like network throughput and capacity, actual values of such metrics cannot be used at the time of RRM computations: a significant time of monitoring is required to estimate how throughput changed, so only past records can be used. Another, simpler approach, followed by many works, is to assume that network throughput or capacity is a function of one or more physical layer metrics, such as interference, Received Signal Strength Indicator (RSSI), Rx signal power etc. Indeed, low interference level and low power signal from other APs implies that less transmission errors tend to happen, and more frames can be transmitted with CSMA/CA MAC mechanism. This reliance on physical layer metrics allows for more immediate adjustments in RF (Radio Frequency) configuration of a WLAN, ensuring the adaptability of the network to immediate environmental changes. However, RRM adjustments in practice can lead to disruptions. Most client devices lack support for the Channel Switch Announce-

ment feature from 802.11h, interpreting a channel switch as if the AP has become unavailable. Therefore, utilizing historical data becomes instrumental in making informed, albeit infrequent and periodic, RRM decisions.

Thus, channel and transmit power settings for each AP should yield optimal value of some given metric for the whole WLAN, such as: interference level, WLAN throughput, WLAN capacity, etc. Thus, most of the works consider RRM as an optimization problem, such as integer linear programming (ILP) [19] [20] or binary quadratic programming (BQP) [13]. Since both ILP and BQP are proven to be NP-hard problems, researchers propose heuristics to reduce search space [11], or apply meta-heuristic methods such as genetic algorithms [21] or deep neural networks [13]. Other approaches, while not solving optimization problem explicitly, aim to keep some target metric, such as SINR (Signal-to-Interference-plus-Noise Ratio), within pre-defined acceptable boundaries [16]. In [15], authors employ a graph model, where APs are represented as nodes and edges connect APs which can potentially interfere. Using such model, each node can be assigned a color representing its channel.

It is important to emphasize that in 802.11 WLANs, all clients connected to a specific Access Point (AP) utilize the same frequency and transmission power settings. Given the limited applicability of per-link (per-client) Transmit Power Control (TPC), as previously discussed, it is assumed that both frequency and transmission power are configured for the entire cell. This means that all clients of a given AP operate on the same frequency, and the AP maintains a consistent transmit power level for communication with all its clients.

IV   Proprietary RRM Solutions

This section surveys proprietary RRM solutions offered by leading vendors in the enterprise WLAN market. I focus on Cisco, Juniper Networks, and Ruckus Networks, since they are the most popular vendors in enterprise WLAN market [22]. To the best of my knowledge, peer-reviewed evaluations of those proprietary RRM efficiency are very limited and scarce, so one could only rely on the claims made by vendor themselves.

A.   Cisco

Cisco's RRM strategy is integral to its Cisco Centralized Architecture, known as the Unified Wireless Network (UWN) [3]. In the UWN framework, a single or multiple Wireless LAN Controllers (WLCs) manage up to several thousand Access Points (APs). These WLCs act as the core of the WLAN architecture, enabling centralized control and the collection of telemetry from all APs within the network. A WLC can be either specialized hardware or a virtual machine hosted in the cloud [23]. Effectively, in UWN, access points can be thought of as Wi-Fi network interface cards for the WLC, providing minimal real-time functionality from 802.11 standard that cannot be carried out to WLC due to propagation and transmission delays. This architectural model has become the de facto standard for large-scale enterprise WLANs and is employed by most major vendors in the industry.

Cisco offers several RRM solutions. First, CleanAir is a flagship technology from Cisco [24] to optimize network performance, avoid jamming, and detect interference sources, including non-802.11 ones. Cisco states that it outperforms

competitors through several features:

- It utilizes specialized hardware for RF analytics. For instance, the Cisco Catalyst 9100 Series Access Points contain a scanning radio for background RF scanning. This functionality allows for continuous service provision to clients without disrupting the main AP radio transceivers. Additionally, the Cisco RF ASIC, a dedicated chip, enables advanced wireless network analytics and spectrum analysis unavailable to conventional Wi-Fi modules;

- Classifying and visualizing interference sources thanks to dedicated RF hardware;

- Comprehensive WLAN-wide radio resource management, supplying both real-time and historical data at varying levels of granularity;

- CleanAir is event-driven, that means it can adapt to changing RF environment and adjust radio parameters in a matter of few minutes, drastically reducing downtime.

However, CleanAir is only available for the higher-end models in the Cisco product line, posing limitations for its large-scale deployment. Furthermore, lack of compatible radio analytics hardware from other vendors and undisclosed implementation details restrict the utility of this technology for integration with non-Cisco equipment. On the other hand, Cisco Catalyst product line of WLAN Controllers also provide "regular" RRM functionality that only requires regular Wi-Fi chipset and can be used with all Cisco APs [1]. The trade-off for this convenience is access to less detailed information about the RF environment and the necessity for Access Points to temporarily switch off their current channel to conduct scanning. In this case, APs collect statistics on their current channel any time they

are not transmitting data. Additionally, periodically APs scan other channels to gather statistics. [23]. At that time, AP is not available to clients, so, scanning introduces latencies for the clients connected to the AP.

Cisco RRM employs a super-cell concept. In such scheme, a group of geographically close APs (forming an *RF Group*) is managed by a designated WLAN Controller (*RF Group Leader*). A subgroup unit within an RF group is called *RF Neighborhood*, and consists of AP that can hear each other at signal strength ≥ 80 *dBm* [23]. Each AP is associated with two lists: RX neighbors, i.e., list of APs that given AP can hear, and TX neighbors, a list of APs that can hear given AP. For each channel, Cisco RRM maintains a *cost metric*, an estimation of channel goodness that is based on RSSI, co-channel interference, non-WiFi interference.

B. Juniper Networks

Juniper Networks offers Mist AI RRM technology to improve network performance. The notable features are [25], [26]:

- Automatic dual-band radio management — if RRM system finds 2.4-GHz radio transmitter to be unused on a given AP, it disables the radio to free airspace for other access points;

- Juniper Mist APs incorporate the Predictive Analytics and Correlation Engine (PACE) "to monitor conditions and make out-of-band adjustments" [26];

- Telemetry is sent to the Juniper Mist Cloud, so that the cloud can periodically fine-tune APs based on historical data and usage statistics;

- Employing a Reinforcement Learning (RL) methodology for the strategic

planning of channel selection and power settings across APs in a WLAN, aiming for optimal network performance [25].

C.   Ruckus Networks

Ruckus Networks offers ChannelFly RRM technology, which provides automatic channel selection. ChannelFly estimates capacity of each channel by continuously monitoring the activity of each channel across the 2.4 and 5GHz bands. Based on this information, ChannelFly develops a statistical model to predict which channel will offer the highest capacity for clients, as detailed by [27]. A key benefit of ChannelFly is its ability to avoid "dead time", defined as the period an AP spends scanning different channels, during which it cannot communicate with clients. This capability implies the inclusion of dedicated scanning radios in Ruckus APs, allowing continuous communication with clients while performing channel assessments.

Additionally, Ruckus offers a "smart adaptive antenna array" technology. This feature enhances the directionality of signals from Ruckus APs, focusing the transmission towards clients to improve the Signal-to-Noise Ratio (SNR).

D.   Aruba Networks

The Adaptive Radio Management (ARM) technology by Aruba represents an earlier approach to Radio Resource Management (RRM), utilizing Adaptive Channel Selection and Transmit Power Control to enhance the RF environment in WLANs. ARM stands out for its algorithmic simplicity and the thoroughness of its documentation provided by Aruba, in contrast with other vendors [28].

Key features of ARM include [29]:

- **Application Awareness**: Addressing the "dead time" caused by APs during channel scanning, ARM throttles the frequency of background scans based on current traffic load, reducing scans under heavy traffic and resuming normal scanning rates when traffic diminishes [2].

- **Mode Awareness**: To mitigate interference in environments with densely installed APs, ARM can switch excessive APs to Air Monitor mode, where they continuously collect and send RRM telemetry to the controller.

- **Band Steering**: Promotes the use 5 GHz band to clients, instead of more congested and higher-range 2.4 GHz band.

- **802.11n HT Mode Support**: ARM can utilize a 40 MHz channel pair for 802.11n networks, selecting the best primary and secondary operating channels automatically.

- **Noise and Error Monitoring**: Distinguishes between 802.11 and non-802.11 noise sources, improving network reliability.

- **Spectrum Load Balancing**: Analyzes client distribution across neighboring APs to direct new connections to less burdened APs, though clients may reconnect to their original choice upon a subsequent attempt.

- **Noise Interference Immunity**: Adjusts the receiver sensitivity threshold to ignore weak and non-802.11 signals, reducing unnecessary decoding efforts and improving network performance.

Reports from system administrators, though, suggest that RRM decisions in Aruba ARM are made by APs rather than controller. Among other user complains

are unnecessary disabling of 2.4GHz radios, erroneous TPC leading to coverage holes [30].

However, ARM is a legacy technology. Its successor, Aruba AirMatch, introduced in recent ArubaOS versions, is a more sophisticated RRM technology, which is based on AI and machine learning and is able to perform channel and power planning on a WLAN-wide scale, suggesting ARM was implemented in a per-cell way and AirMatch is a super-cell solution.

Notable AirMatch features:

- Channel width adjustment based on device density - the more devices are connected to an AP, the narrower channel width is used to allow channel reuse and reduce interference;

- APs measure RF environment for 5 minutes every 30 minutes;

- Decisions based on a 24-hour period analytics unlike instant RF situation snapshots in ARM;

- Elimination of coverage holes based on TPC.

- Configurable thresholds in channel quality improvements to trigger channel and EIRP planning, default threshold is 15%.

- ClientMatch technology that manages clients: performs load balancing between APs, encourages clients to switch to APs providing better signal strength and using higher bands (5 GHz or event 6 GHz in 802.11ax)

Similarly to ARM, Aruba provides more information about AirMatch operating logic than other vendors about their RRM solutions.

According to [28], AirMatch blacklists channel for channel selection if a radar was detected on it (in 5 GHz case) or in case if high noise level was detected on it (for all bands). In those cases, AirMatch will select channel with a minimum interference index.

It is not clear if AirMatch uses the same metrics as ARM, but only ARM metrics are described in the documentation. To make RRM decisions, ARM uses two metrics:

- **Coverage Index** - comprises two components, x and y:

    - x is weighted calculation of Signal-to-Noise Ratio (SNR) for all APs on a given channel;

    - y is the weighted sum of the SNRs that neighboring APs within the group observe on the same channel.

- **Interference Index** - measures co-channel and adjacent-channel interference, calculated as a sum of four quantities a, b, c, d:

    - c is the channel interference the AP neighbors see on the selected channel;

    - d is the interference the AP neighbors see on the adjacent channel.

Additionally, Aruba APs collect several other metrics, including L2 metrics [28]:

- Number of Retry frames, measured in %

- Number of Low-speed frames, measured in %

- Number of Non-unicast frames, measured in %

- Number of Fragmented frames, measured in %

- Number of Bandwidth seen on the channel, measured in kbps

- Number of PHY errors seen on the channel measured in %

- Number of MAC errors seen on the channel measured in %

- Value of noise floor on the specified AP

Aruba documentation indicates that these metrics offer a "snapshot of the current RF health state" [31], suggesting they are informational tools for network administrators rather than being actively used in RRM decision-making.

V   Conclusion

Summarizing the insights from prior sections, I can conclude that the problem of radio resource management in 802.11 WLANs is still relevant, since the IEEE 802.11 standard provides only limited tools for RRM, while existing commercial solutions are proprietary and lack interoperability. Thus, there is a need for a novel RRM algorithm that can be implemented in existing enterprise WLAN infrastructure and improve overall network performance.

I find super-cell approach most fitting for a modern RRM algorithm that can be applied in real-world WLAN deployments. Super-cell algorithms, while being practical and having less obstacles in hardware and current device drivers compared to other approaches, still have the potential to vastly improve RF situation and, thus, WLAN performance.

Centralized management that is typically utilized in super-cell RRM is the standard approach when building modern WLANs, allowing to gather more infor-

mation about RF environment and come up with more optimal allocations compared with local RRM decision-making. Moreover, presence of WLC as centralized entity with orders of magnitude higher computation power and ability to collect and store statistics from all APs all over the WLAN in the long term releases the burden of RRM from Access Points and potentially improves the overall network efficiency.

Despite the promising capabilities of per-link and per-flow radio resource management approaches for optimizing wireless networks in a more fine-grained and application-aware manner, they have considerable limitations that currently prevent from implementing them in production wireless networking solutions.

As a summary of this survey, we can identify the research gap: the problem of radio resource management in 802.11 WLANs is still relevant, since IEEE 802.11 standard does not provide fully-fledged RRM, while existing commercial solutions are proprietary and lack interoperability. Thus, there is a need for an RRM algorithm addressing key issues, including:

- Design for centralized management of enterprise WLAN, working as a part of Wireless LAN Controller;

- Applicability with current hardware and software, namely:

  - Effortless integration with OpenWRT-based access points;

  - Requires data like physical and link-layer statistics that can be obtained using only regular Linux Wi-Fi drivers like `nl80211`, and standard Linux networking tools;

- Performing not worse than existing RRM algorithm *RRMGreedy*, analyzed in Chapter 3.2;

- Able to combine both channel selection and transmit power adjustment to improve RF environment and network performance.

The following chapters will focus on analyzing the limitations of current algorithms and developing a new one.

Chapter 3

Baseline solution analysis and algorithm design

The chapter is organized as follows. Section 3.1 defines the problem of RRM and objectives of an RRM algorithm. Section 3.2 describes the RRM algorithm called RRMGreedy, developed by a Russian telecommunications vendor, analyzes the flow and asymptotic complexity of the algorithm, and identifies flaws and limitations of RRMGreedy. The objective of this study is to create an algorithm outperforming RRMGreedy in terms of network capacity and throughput; in Section 3.3 I propose a new RRM algorithm, called RRMGreedy++, that addresses the limitations of RRMGreedy.

I   Problem statement

Using the RRM problem definition from 2.3.2, I define the problem as follows: given a set of access points $AP = \{AP_1, AP_2, \ldots, AP_n\}$, where each access point AP_i has a set of wireless interfaces $W_i = \{w_{i1}, w_{i2}, \ldots, w_{im}\}$, where each

interface w_{ij} can operate on a set of channels $C = \{c_1, c_2, \ldots, c_k\}$, the goal is to find a channel assignment $f : W \to C$ and transmit power assignment $g : W \to P$ that maximizes network throughput and capacity. The problem can be formulated as a combinatorial optimization problem, where the goal is to find a configuration of channels and transmit powers that maximizes the objective function, which is a function of network throughput and capacity.

II   Baseline solution analysis — RRMGreedy

The greedy RRM algorithm, used by a major telecommunications vendor, which will be further referred to as *RRMGreedy*, is based on background scanning by access points. The goal of this study is to create a new RRM algorithm aligning with requirements and addressing the research gap formulated in Section 2.5. In particular, the novel algorithm should result in network throughput and capacity greater or equal than *RRMGreedy* in corner cases. In this light, I start with describing *RRMGreedy*,

A.   *RRMGreedy: flow description*

This section describes *RRMGreedy* algorithm. As the name suggests, the algorithm tries to achieve optimal radio resource allocation in a greedy way, taking the local optimum for each device in a RRM group. The algorithm operates on the granularity of interfaces rather than access points. I consider it to be reasonable, since practically different interfaces operate on completely different frequency bands, so when are dealing with, for example, 5 GHz band, the whole access point can be considered as a single interface.

In other words, I consider **RRM group** G as a set of **wireless interfaces** w

(that will be further referred to as simply **interfaces**), where several interfaces can belong to a single access point. On a high level, the algorithm consists of two main steps — **Channel Selection** and **Transmit Power Adjustment**:

1. **Channel Selection**:

   (a) compute **group interference** I_G — sum of interference scores for each interface in group:

   $$I_G = \sum_{i=0}^{|G|} OnIfaceInterference(w_i) \tag{3.1}$$

   Initial group interference score is denoted as I_0.

   (b) for each w_i, compute its updated interference score for every channel c_i^k available for that interface:

   $$interf_i = OnIfaceInterference(w_i, c_i^k) \tag{3.2}$$

   If there is a channel c_i^k that reduced $interf_i$, update interface settings to issue change to that channel later.

   (c) **continue while there is an improvement**, i.e., previous group interference score is larger than the latest one: $I_{j-1} > I_j$ for $j > 1$

2. **Transmit Power Adjustment**:

   (a) Compute group interference as in step 1a

   (b) Identify the *worst interface* w_m causing the *worst interference* (i.e.,

whose from-interference score is the largest):

$$w_m = \underset{i}{\arg\max} \; FromIfaceInterference(w_i) \qquad (3.3)$$

(c) Gradually reduce transmit power of w_m with exponential backoff until it stops being the worst interface or reaches minimum Tx power.

ALGORITHM 1: RRMGreedy

Input: *scandata*: scanning data from access points

       G: set of wireless interfaces within the RRM group and its settings

Output: *resultConfig*: final configuration for access points

1 groupState ← PreprocessRRMData(*G*, scandata)

   // Step 1: Start with max Tx power for each interface

2 **for** *each interface in G* **do**

3   groupState[interface].TxDiff ← (*interface.MaxTxPower − interface.TxPower*)

   // Step 2: Greedy Channel Search

4 groupInterference ← GroupInterference(groupState)

5 **repeat**

6   **for** *each interface in G* **do**

7   C_i ← available channels for interface

8   newChannel ← $\underset{ch \in C_i}{\mathrm{argmin}}\ OnIfaceInterference(groupState, interface, ch)$

9   groupState[interface].channel ← newChannel

10   groupInterference ← GroupInterference(scandata)

11 **until** *no improvement in groupInterference*

   // Step 3: Greedy Transmission Power Adjustment (if enabled)

12 **repeat**

     // Identify the "worst interface" *interface$_{worst}$* causing the greatest
       interference for other APs and the maximum RSSI of *interface$_{worst}$* on
       some other interface

13   *interface$_{worst}$*, *RSSI$_{worst}$* ← $\underset{w_i \in G}{\mathrm{argmax}}\ FromIfaceInterference(groupState, w_i)$

     // *RSSI$_{min}$* is a predefined threshold for the minimum RSSI, by default
       set to -100 (dBm)

14    $groupState[interface_{worst}].TxDiff \leftarrow \dfrac{RSSI_{min} - RSSI_{worst}}{2}$

15   groupInterference ← GroupInterference(scandata)

16 **until** *no improvement in groupInterference*

   // Step 4: Send configuration changes to APs

17 *resultConfig* ← getUpdatedConfiguration(groupState)

18 **return** *resultConfig*

After providing a high-level overview of the algorithm, I will analyze the used functions, since they contain the core logic of the algorithm. These functions will be discussed in the order they are executed within the algorithm.

Function *PreprocessRRMData()* does not perform actual RRM logic and is used to preprocess the data for further calculations. Namely, it reorganizes scan results for each interface in a way that for every channel every interface maintains a list of

- Signals originating from access points within the RRM group G. In RRM-Greedy, such signals are called *inner* signals;

- Signals originating from access points outside of the RRM group (i.e., coming from *rogue APs*). In RRMGreedy terms, such signals are called *outer* signals.

The pseudocode for *PreprocessRRMData()* is shown in Algorithm 2.

ALGORITHM 2: PreprocessRRMData

---

1 **foreach** *interface in G* **do**

2   scanFromIface ← scandata[interface]

3   **foreach** *BSSID, signalData in scanFromIface* **do**

4   channel ← signalData.channel

5   **if** *bssid in RRMGroup* **then**

6   interface.InnerBSS[channel].append(signalData)

7   **else**

8   interface.OuterBSS[channel].append(signalData)

The result of PreprocessRRMData is saved in a table called *groupState*. This structure maintains updated channel and transmit power settings for each interface in the RRM group as RRMGreedy progresses.

Function *OnIfaceInterference(groupState, interface, channel)* (Algorithm 3) calculates interference experienced by given access point (and thus by its wireless interface) from outer and inner signals. Essentially, it breaks down to calculating interference as a sum of outer and inner interference scores for a given interface w_i on a given channel c_i:

$$interf_i = OnIfaceInterference(w_i, channel) = interf_i^{outer} + interf_i^{inner} \qquad (3.4)$$

$$interf_i^{outer} = \sum_{j=0}^{|S_i^{outer}|} ChannelInterference(channel, s_i^j.channel) \cdot scale(s_i^j) \qquad (3.5)$$

$$interf_i^{inner} = \sum_{j=0}^{|S_i^{inner}|} ChannelInterference(channel, s_i^j.channel) \cdot scale(s_i^j + w_i.TxDiff)$$

$$(3.6)$$

where:

S_i^{outer}	set of signals sensed by w on from APs $\notin G$;
S_i^{inner}	set of signals sensed by w_i from APs $\in G$;
s_i^j	j-th signal received by w_i, in S_i^{outer} or S_i^{inner};
$w_i.TxDiff$	transmit power adjustment for w_i;
$ChannelInterference(ch_1, ch_2)$	interference between ch_1 and ch_2 (Algorithm 3);
$scale(x)$	normalization function 3.7.

Transmit power is normalized to $[0, 1]$ with min-max normalization using pre-defined RSSI (Received Signal Strength) thresholds $RSSI_{min}$ and $RSSI_{max}$ (3.7):

$$scale(x) = \frac{x - RSSI_{min}}{RSSI_{max} - RSSI_{min}} \qquad (3.7)$$

Interference between channels is estimated with *ChannelInterference()* function(Algorithm 3). It is assumed that the distance between orthogonal channels is 5 MHz. Thus, given a 2.4 GHz band and 20 MHz width, the distance between two channels should be at least 5. The interval of length n should have $n-1$ channels in between, which is the reason for the *add* factor.

ALGORITHM 3: ChannelInterference

1 **Function** ChannelInterference(ch_1, ch_2, $width$)**:**

2   add $\leftarrow 0$

3   **if** $ch_1 < 36$ **then**

4   add $\leftarrow 1$

5   **if** $|ch_1 - ch_2| \geq \dfrac{width}{5} + add$ **then**

6   **return** 0

7   **else**

8   **return** 1

ALGORITHM 4: OnIfaceInterference

Input: *groupState*: table containing state of **G**

interface: interface to calculate interference for *channel*: channel to calculate interference on

Output: Estimated cumulative interference score experienced by *interface*

1   *ifaceScanData* ← *scandata*[*interface*]

2   *cumOuterInterf* ← 0

   // Calculate interference from outer APs

3   **foreach** *otherChannel, otherChannelSignals in ifaceScanData.OuterBSS* **do**

4   *channelInterfScore* ←
      ChannelInterference(*otherChannel, interface.Channel, interface.ChannelWidth*)

5   *channelOuterInterf* ← 0

6   **foreach** *signal in otherChannelSignals* **do**

7   *sigRating* ← signal.RxPower (in dBm) normalized to [0, 1] scale

8   *channelOuterInterf* ← *channelOuterInter + finterferenceScore × sigRating*

9   *cumOuterInterf* ← *cumOuterInterf + channelOuterInterf*

10   *cumInnerInterf* ← 0

   // Calculate interference from inner APs

11   **foreach** *otherInterface, signalsFromOtherInterface in ifaceScanData.InnerBSS* **do**

12   *channelInterfScore* ←
      ChannelInterference(*oth.Settings.Central, interfaceChannel, interfaceChannelWidth*)

13   **if** *channelInterfScore == 0* **then**

14   continue

15   *channelOuterInterference* ← 0

16   **foreach** *signal in signalsFromOtherInterface* **do**

17   *adjustedSourceSignal* ← *signal + otherInterface.TxDiff*

18   *sigRating* ← *adjustedSourceSignal* power (in dBm) normalized to [0, 1] scale

19   *channelOuterInterference* ← *channelInterfScore × sigRating*

20   *cumInnerInterf* ← *cumInnerInterf + channelOuterInterf*

21   **return** *cumOuterInterf + cumInnerInterf*

Function *FromIfaceInterference( )* (Algorithm 5) calculates the amount of interference originating from w_i and experienced by other interfaces in the group $\boldsymbol{G}$. The function returns two values: cumulative interference quantity estimating how much w_i affects the RRM group, and maximum RSSI of w_i on some other interface w_j. Using *FromIfaceInterference( )*, RRMGreedy finds the *worst interference* and gradually reduces its transmit power to improve the group interference score.

ALGORITHM 5: FromIfaceInterference()

Input: *groupState*: table containing state of **G**

      interface: interface to calculate interference from

Output: *cumInterfFromInterface*: cumulative interference quantity estimating how much

      interface affects the RRM group

      maxSignal: maximum signal strength from *interface* experienced by other AP

1 *cumInterfFromInterface* ← 0

2 *maxSignal* ← *MinSignal*

3 **foreach** *otherInterface in scandata* **do**

4   **if** *otherInterface == interface* **then**

5   continue

6   **if** *otherInterface has not received signals from interface* **then**

7   continue

8   *interfScore* ←

      ChannelInterference(*interface.Channel, otherInterface.channel, otherInterface.channelWidth*)

9   **if** *interfScore == 0* **then**

10   continue

11   *interfaceInterfSum* ← 0

12   **foreach** *sig in thisIface* **do**

13   *adjustedSignalPower* ← *interface.Signal* + *interface.TxDiff*

14   *maxsignal* ← max(*signal, maxsignal*)

15   *sigRating* ← adjustedSignalPower normalized to [0, 1]

16   *interfaceInterfSum* ← *interfaceInterfSum* + (*interfScore* × *sigRating*)

17   *cumInterfFromInterface* ← *cumInterfFromInterface* + *interfaceInterfSum*

18 **return** *cumInterfFromInterface, maxsignal*

B. Evaluating asymptotic complexity of RRMGreedy

In this section, I evaluate asymptotic complexity of *RRMGreedy* algorithm. First, I consider the complexity of used auxiliary functions:

1. *ChannelInterference*$(ch_1, ch_2, width)$ has constant complexity $O(1)$;

2. *OnIfaceInterference*(G, w_i) has complexity linear w.r.t. the number of signals sensed by w_i, i.e., $O(|S_i|) = O(|S_i^{outer}| + |S_i^{inner}|)$;

3. *FromIfaceInterference*(G, w_i) has complexity proportional to the product of number of interfaces in G and amount of signal samples from w_i heard by other interfaces, which is limited by some constant C, reasonably small in practice $(C < 10)$, i.e., $O(|G| \cdot C) = O(|G|)$.

Next, I analyze time complexity of the main phases of the algorithm:

1. **ProcessScanDataForGroup** has complexity $O(|G| \cdot |S|)$, where $|S|$ is the total number of signals in all scans from all interfaces in G;

2. **Set Maximum Transmission Power** has complexity $O(|G|)$;

3. **Optimize Channel Selection** is an iterative algorithm, going for each $w_i \in G$ and for each channel $c \in w_i.channels$ running *OnIfaceInterference*(G, w_i) until there is no improvement, so its worst-case time complexity can be estimated as $O(|G| \cdot C \cdot K \cdot |S_i|)$, where C is the number of channels and K is the number of iterations;

4. **Adjust Transmission Power** is an iterative algorithm, going for each $w_i \in G$ until there is no improvement yielded by *FromIfaceInterference*$()$, so its

worst-case time complexity can be estimated as $O(|G| \cdot K \cdot |G|) = O(|G|^2 \cdot K)$, where K is the number of iterations;

5. **Final Data Assembly and Event Generation** has complexity $O(|G|)$.

C.   RRMGreedy: limitations and conclusions

Based on the analysis carried out above in this chapter, we can identify the following issues with the *RRMGreedy* algorithm:

1. *ChannelInterference()*, calculating how much two channels interfere, only yields binary values 1 or 0. Due to the presence of cross-channel interference, the result of *ChannelInterference()* should be continuous in range [0; 1], not discrete;

2. Noise floor should be included into the interference metric, since it can drastically affect the performance on the channel;

3. Channel utilization is not taken into account. Consider the case when 3 APs with no associated stations operate on channel *A*, and 1 AP with 10 associated stations operates on channel *B*. The algorithm does not keep track of the number of stations or other channel utilization data, and will favor *B*, even though *A* is less utilized, since the only traffic APs on *A* are likely to produce is rare and periodic beacon frames;

4. Assuming maximum transmit power for each in-group AP in the Step 1 of Algorithm 1 is not reasonable, since it tends to exaggerate the interference scores and thus lead to unnecessary channel switches;

5. When calculating the interference score, if there are several signals from the same APs, they will be summed as they were different APs, again leading to exaggerated interference scores;

6. Even if next iteration yields worse interference score, there is no rollback to the previous, better state;

7. Change with no improvements will be staged anyway, so disruptive channel switch will occur even if it is not necessary;

8. Finally, selecting channel that minimizes the interference for an interface in a greedy way could not lead to the global optimum.

III RRMGreedy++ design

Founding on limitations identified in Section 3.2.3, I introduce the following improvements:

1. **Continuous interference estimation**: I propose a slightly updated *ChannelInterference()* implementation that differs based on how much channels overlap (Algorithm 6);

2. **Updated cumulative metrics approach**:

   - **Noise floor** for *FromIfaceInterference()* cumulative estimation. This affects channel selection decision, discouraging channels with high noise floor level;

   - **Channel utilization**: cumulative interference metric now takes into account channel utilization in its simplest form: number of clients of

the access point. This will discourage APs from using channels with many active clients that can be heard.

3. **RRMGreedy Flow Improvements**:

- The "*TxDiff* → max" phase of RRMGreedy (Step 1 of Algorithm 1) is moved after the Channel Selection part and before the Transmission Power Adjustment part;

- The *GroupState* structure, containing the current state of **G**, is saved for the previous iteration of the algorithm. If the interference score increases, *RRMGreedy++* aborts and returns to the previous *GroupState*;

- Iterations that yield a new group configuration with *GroupInterference* $< \epsilon$, where ϵ is a configurable threshold pre-defined to $\epsilon = 0.005$, are discarded.

ALGORITHM 6: ChannelInterference v2

1 **Function** ChannelInterference(ch_1, ch_2, *width* = 20):

2   **if** $ch_1 < 36$ *and* $ch_2 < 36$ **then**

3   *freqDiff* $\leftarrow |ch_1 - ch_2| \times 5$

4   *maxDistance* $\leftarrow$ *width* + 5

5   **else if** $ch_1 \geq 36$ *and* $ch_2 \geq 36$ **then**

6   *freqDiff* $\leftarrow |ch_1 - ch_2| \times 20$

7   *maxDistance* $\leftarrow$ *width*

8   **if** *freqDiff* > *maxDistance* **then**

9   **return** 0.0

10   *overlap* $\leftarrow 1 - \dfrac{freqDiff}{maxDistance}$

11   **return** *overlap*

IV Conclusion

In this chapter, I have carried out analysis of a production algorithm called *RRMGreedy*, used by a leading Russian telecommunications company. By analyzing source code of *RRMGreedy*, I have estimated asymptotic complexity and found several issues potentially affecting the efficiency of the algorithm. To address those issues, I propose a set of enhancements for *RRMGreedy*, together called *RRMGreedy++*. In Chapter 5, I proceed with experimental evaluation of the RRM algorithms, including *RRMGreedy++*, by introducing a new framework for ns-3 network simulator I have created to facilitate RRM assessment.

Chapter 4

Implementation and evaluation

This chapter is dedicated to implementing the *RRMGreedy++* algorithm described in Section 3.3, as well as other algorithms described in Section 2 and comparing their performance. In this study, I use the ns-3 discrete-event simulation framework for implementing and evaluating different RRM algorithms.

The chapter is organized as follows:

- Section 4.1 describes the simulation methodology used to evaluate the performance of RRM algorithms, and the process of adjusting ns-3 to needs of running and evaluating RRM algorithms;

- Section 4.2 describes my implementation of *RRMGreedy++*, *RRMGreedy*, and *Least Congested Channel Search* (LCCS) algorithms in ns-3;

- Section 4.3 presents the results of the simulation study, comparing RRM algorithms: RRMv2, RRMGreedy, and Least Congested Channel Search (LCCS), in terms of network throughput and interference;

- Finally, Section 4.4 concludes the chapter with a summary of the results and a discussion of the implications of the study.

I   Simulation methodology

In this study, I implement LCCS, RRMGreedy and RRMGreedy++ algorithms in ns-3 simulator to have a scalable, flexible (from dozens to hundreds of APs), and reproducible (unlike real-world testbeds) way to evaluate performance of different RRM algorithms under different conditions.

A.   *RRM support in ns-3*

The ns-3 simulator is a comprehensive tool for research in computer networks, modeling each layer of the OSI model and providing simulations for wired and wireless communication technologies and protocols, including LTE, Wi-Fi, and Ethernet. Nevertheless, despite its extensive capabilities, there is a limited support for simulating RRM algorithms.

In addressing this gap, [32] introduced an RRM framework that employs an additional radio interface, termed a "scan interface." Unlike the standard "data interface" used for communication, the scan interface periodically surveys a specified set of channels. It operates in *promiscuous mode*, allowing it to capture all frames and physical layer headers intercepted by the data interface. Thus, scan interface is able to record Received Signal Strength Indicator (RSSI) values from transmissions between devices that the data interface is engaged with. This resembles scanning radios found in high-end Cisco access points. Unlike the Cisco solution, this scanning interface, however, can be used both with stations and access points.

However, this model presents several deviations from real-world scenarios:

- Most hardware, apart from high-end Cisco models, lacks a dedicated scanning interface, leading to scanning activities disrupting normal data exchanges. This interruption, referred to as "dead time," denotes periods when scanning supersedes the standard operation of an Access Point (AP).

- For mobile devices, where battery efficiency is paramount, scanning capabilities are significantly restricted.

- The `RriModule` code and its associated examples, designed for ns-3 version 3.20, are incompatible with the current ns-3 version 3.40 API and the legacy Waf build system. As a part of my contribution, I have refactored the original code to align with the latest ns-3 version 3.41 API, specifically updating the `RriModule` to integrate with the latest `WifiMac` ns-3 API and the CMake build system. This adaptation ensures the compatibility and execution on the newest ns-3 version. The modified source code is publicly available on GitHub.

B.   *Providing scanning capabilities in ns-3*

In real-world scenarios, wireless access points implement scanning by setting the wireless adapter to monitor mode and sequentially switching to each channel that requires scanning. Although direct support for monitor mode is absent in ns-3, a comparable functionality can be simulated using the `MonitorSnifferRx` event source. Subscribing to such events allows for the interception of all frames that the physical layer of the access point can receive and demodulate, even if those frames would be normally discarded as not intended for the current AP

[33]. However, this method does not exclude standard operations of an access point; the AP still continues to receive and transmit frames. This is not the case in actual access points, where operating monitor mode is mutually exclusive to AP's normal operation. To achieve nearly the same result, my approach prevents the AP from transmitting frames during the scanning period.

Based on the consideration above, I designed an external `Scanner` class that encloses `WifiNetDev` of access point nodes. Scanner can be used on arbitrary AP node to give it background scanning capabilities. The `LCCScanner` class is invoked on `MonitorSnifferRx` event, simulating receiving frames in monitor mode. For each received frame, if this frame comes from an AP (either a beacon frame or has `FromDs` flag set to 1), it is added to `knownAps` table. If scheduled, Scanner periodically switches to other channels for specified *channel dwell period*, listening for frames. To reduce disruptions in AP's regular BSS operation, the AP returns to its operating channel after performing scanning on one channel, proceeding to scan next channel after a specified *scan interval*. Such behavior is also implemented in Cisco access points [1]. Based on this frame capture, Scanner builds a table of known APs, containing information such as BSSID, operating channel, RSSI and SNR levels, and the number of client stations associated with the AP. This table is then used by RRM algorithms to make decisions on channel and power adjustments.

Another limitation I have encountered during the implementation is the lack of channel switching logic for unassociated stations performing scans to discover an access point for association. In practical scenarios, stations scan all available channels within the operating bands to discover access points, however, as of the latest ns-3 version 3.41, released in February 2024, such functionality is absent in `ns3::StaWifiMac` class, which implements the logic for STA (stations) op-

eration. Consequently, ns-3 Wi-Fi stations can only scan for APs present on the current operating channel. For instance, if both AP and stations are on channel 1, and the AP shifts to channel 6, the stations remain unable to scan beyond channel 1 to ascertain the AP's new location on channel 6. To address this problem, I have implemented a custom scanning logic, allowing stations to scan all available channels in its operating band. This amendment is available in the modified source code on GitHub.

II   Implementation

This section provides an implementation of various Radio Resource Management (RRM) strategies, utilizing the ns-3 discrete-event simulation framework.

A.   *Least Congested Channel Search (LCCS)*

LCCS is a patented local per-cell algorithm described in [14]. Essentially, it breaks down to estimating the interference level on each channel and selecting the least congested one. The algorithm is implemented in practically every home Wi-Fi router. The algorithm is simple, but it does not coordinate channel selection across multiple APs, which is crucial for performance in large-scale deployments. In my ns-3 LCCS implementation, I introduce `LCCSAlgo` class. Method `LCCSAlgo::Decide` perform LCCS channel switching decision and is invoked as a callback from `Scanner` class, described in Section 4.1.2. Algorithm 7 describes my LCCS implementation.

ALGORITHM 7: Least Congested Channel Search

Input: *scanningAP*: AP that invoked the LCCS call

Output: *newChannel*: new channel for AP to switch to

1 Function `LCCSAlgo::Decide(`*scanningAP*`)`:

2   metrics ← table that maps each channel to its metric, initially 0 for each channel

3   **foreach** *apEntry in scanningAP.knownAps* **do**

4   metrics[apEntry.channel] ←
         metric[apEntry.channel] + 1 + 10 × size(apEntry.clients)

5   **end**

6   newChannel ← argmin(metrics)

7   return newChannel

Note that I intended to keep the algorithm simple to conform closer with the original flow described in [14]. Real-world LCCS implementation tend to use more sophisticated heuristics and metrics to estimate channel congestion.

B. RRMGreedy

RRMGreedy is a centralized super-cell algorithm used by a leading Russian telecommunications vendor, described and analyzed in Chapter 3.2. Implementing RRMGreedy in ns-3 requires a centralized entity to gather data from all APs and perform computations. I introduce `RRMGreedyAlgo` class, which is responsible for collecting data from all APs and making decisions on channel and power adjustments. Unlike LCCS, RRMGreedy is not invoked by APs due to its centralized nature. Since the original RRMGreedy does not requests scanning data directly from APs, but rather queries the database to which APs send periodical reports, I aimed to achieve similar behavior. Thus, `RRMGreedyAlgo::AddApScandata` is set as a callback for each `Scanner` instance in an RRM group to submit the individual scanning results to RRMGreedy after every full scanning cycle. Each `scandata` item has a timestamp, and RRMGreedy

does not accept scanning data older than *scandataStaleTime* parameter (in seconds). Callback method `RRMGreedyAlgo::Decide` is invoked by the simulation scheduler at regular intervals to perform RRM decisions. Channel Planning is implemented using the same approach found in LCCS. However, implementing TPC was a considerable challenge, since ns-3 does not support setting AP's transmit power directly. To address this issue, I manipulated starting and ending Tx power levels of phy layer of a WifiNetDevice of an AP node.

C.   RRMGreedy++

TBD

III   Evaluation

To evaluate the performance of RRM algorithms, I have conducted a series of test cases for simulations in ns-3. The simulation setup is as follows.

A.   Simulation settings

TABLE I
Simulation parameters (shared among all test cases)

Parameter	Value
Frequency band	2.4 GHz
IEEE 802.11 amendment	802.11n
Data rate	54 Mbit/s
Channel width	20 MHz

- **Test Case 1:**

- Number of APs: 3;

- Number of STAs: 7;

- STA allocations by AP: $\{3, 2, 2\}$;

- Initial channel allocations by AP: $\{1, 1, 1\}$;

- Initial channel allocations on stations: all start on channel 1 and until given SSID is found;

- Traffic model: each AP has at least one pair of STAs, where first STA in the pair generates UDP generating traffic as UDP client, and second echoing it back to client as UDP server;

- Simulation time: 10s.

TABLE II
Test Case Configurations

Test Case 1	
Parameter	**Value**
Number of APs	3
Number of STAs	7
Number of STA allocated for each AP	$\{3, 2, 2\}$
Initial channel allocations for APs	$\{1, 1, 1\}$
Initial channel allocations for STAs	$\{1, 1, 1, 1, 1, 1, 1\}$

Traffic model	each AP has at least one pair of STAs, where first STA in the pair generates UDP traffic as client, and second echoing it back to client as UDP server
Simulation time	10s

B. Metrics

I have evaluated the performance of RRM algorithms in terms of following metrics:

- Total throughput of a wireless LAN;

- Average delay;

- Total SNR (Signal-to-Noise Ratio) of a wireless LAN;

- Tx busy time;

- Rx idle time.

C. Results

The results for Test Case 1 are presented below. The scenario had shown that TBD

IV   Conclusion

Chapter 5

Results and Discussion

This chapter discusses the results of the study. Section 5.1 presents the results of the evaluation. Section 5.2 discusses the limitations of the study. Section 5.3 outlines the further work that can be done to improve the study. Finally, Section 5.4 concludes the chapter.

I Simulation Results

RRMGreedy++ is shown to both RRMGreedy and LCCS in terms of convergence rate, SNR, and throughput.

II Limitations

Using the simulated environment provides a controlled and reproducible environment for evaluation, however, networking and physical signal propagation models used pose their constraints. For example, `YansWifiPhy` propagation model does not consider cross-channel interference between adjacent channel,

which can significantly impact channel planning, especially on 2.4 GHz band.

III   Further Work

As comes from the limitations (Section 5.2), I can identify two directions for the next research:

1. Improving the simulation model: the next steps would be to extend the simulation environment to consider more realistic scenarios. This includes employing cross-channel interference model SpectrumWifiPhy, considering the impact of hidden nodes, and introducing a wireless LAN controller (WLC) as a part of the simulated network, since current implementation works "above" the nodes, being a part of the script logic;

2. Implementing the RRMv2 algorithm in a real-world testbed: since the algorithm is designed with respect to constraints described in Section 2.5, metrics required for its operation can be obtained from the majority of OpenWRT-based Wi-Fi access points available on the market.

IV   Conclusion

In this thesis, I have derived a new super-cell Radio Resource Management (RRM) algorithm, RRMv2, and implemented it using ns-3 network simulator. To be able to complete this, I performed analysis of another super-cell RRM algorithm, RRMGreedy, and identified its flaws. I have also implemented a local per-cell RRM algorithm, Least Congested Channel Search (LCCS), as a baseline for comparison. Moreover, since ns-3 does not provide facilities for implementing

RRM algorithms out of the box, and since Wi-Fi station channel switching logic was not implemented as of current ns-3 release, I have introduced a completely new RRM module for ns-3 and a set of amendments into the existing codebase. Finally, the algorithms were using diverse simulation settings with various network topologies and traffic patterns. The results of the simulation show that RRMv2 outperforms both RRMGreedy and LCCS in terms of convergence rate, SINR, and throughput.

Bibliography cited

[1] Cisco. "Radio Resource Management White Paper," Cisco. (), [Online]. Available: `https://www.cisco.com/c/en/us/td/docs/wireless/controller/technotes/8-3/b_RRM_White_Paper.html` (visited on 09/23/2023).

[2] "Understanding ARM." (), [Online]. Available: `https : / / www . arubanetworks . com/techdocs/ArubaOS_62_Web_Help/Content/ArubaFrameStyles/ARM/ARM_Overview.htm` (visited on 01/16/2024).

[3] "Cisco Unified Wireless Network Solution Overview [Design Zone for Mobility]," Cisco. (), [Online]. Available: `http : / / www . cisco . com / en / US / docs / solutions / Enterprise / Mobility / emob30dg / Overview.html` (visited on 03/19/2024).

[4] D. D. Coleman and D. A. Westcott, *CWNA Certified Wireless Network Administrator Study Guide: Exam CWNA-108*, 6th edition. Sybex, Mar. 9, 2021, 1088 pp., ISBN: 978-1-119-73450-5.

[5] A. S. Tanenbaum, *Computer Networks*, 6th edition. Pearson, Apr. 24, 2020, 0 pp., ISBN: 978-0-13-676405-2.

[6] J. Zander, "Radio resource management in future wireless networks: Requirements and limitations," *IEEE Communications Magazine*, vol. 35,

no. 8, pp. 30–36, Aug. 1997, ISSN: 1558-1896. DOI: `10 . 1109 / 35 . 606024`.

[7] "IEEE Standard for Information technology Local and metropolitan area networks Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Spectrum and Transmit Power Management Extensions in the 5 GHz Band in Europe," *IEEE Std 802.11h-2003 (Amendment to IEEE Std 802.11, 1999 Edn. (Reaff 2003))*, pp. 1–75, Oct. 2003. DOI: `10.1109/IEEESTD.2003.94393`.

[8] A. Könsgen, "Spectrum Management Algorithms," in *Design and Simulation of Spectrum Management Methods for Wireless Local Area Networks*, A. Könsgen, Ed., Wiesbaden: Vieweg+Teubner Verlag, 2010, pp. 103–132, ISBN: 978-3-8348-9738-1. DOI: `10.1007/978-3-8348-9738-1_4`.

[9] "Cisco Catalyst 9800 Series Wireless Controller Software Configuration Guide, Cisco IOS XE Bengaluru 17.4.x - BSS Coloring [Cisco Catalyst 9800 Series Wireless Controllers]," Cisco. (), [Online]. Available: `https://www.cisco.com/c/en/us/td/docs/wireless/controller/9800/17-4/config-guide/b_wl_17_4_cg/m_bss_coloring_ewlc.html` (visited on 03/20/2024).

[10] K. Sui, Y. Zhao, D. Pei, and L. Zimu, "How bad are the rogues' impact on enterprise 802.11 network performance?" *2015 IEEE Conference on Computer Communications (INFOCOM)*, pp. 361–369, Apr. 2015. DOI: `10.1109/INFOCOM.2015.7218401`.

[11] A. Levanti, F. Giordano, and I. Tinnirello, "A CAPWAP-Compliant Solution for Radio Resource Management in Large-Scale 802.11 WLAN," in *IEEE GLOBECOM 2007-2007 IEEE Global Telecommunications Confer-*

ence, Washington, DC, USA: IEEE, Nov. 2007, pp. 3645–3650, ISBN: 978-1-4244-1042-2 978-1-4244-1043-9. DOI: `10.1109/GLOCOM.2007.692`.

[12] F. Bouhafs, M. Seyedebrahimi, A. Raschella, M. Mackay, and Q. Shi, "Per-Flow Radio Resource Management to Mitigate Interference in Dense IEEE 802.11 Wireless LANs," *IEEE Trans. on Mobile Comput.*, vol. 19, no. 5, pp. 1170–1183, May 1, 2020, ISSN: 1536-1233, 1558-0660, 2161-9875. DOI: `10.1109/TMC.2019.2903465`.

[13] W. Lee and J.-B. Seo, "Deep Learning-Aided Channel Allocation Scheme for WLAN," *IEEE Wireless Commun. Lett.*, vol. 12, no. 6, pp. 1007–1011, Jun. 2023, ISSN: 2162-2337, 2162-2345. DOI: `10.1109/LWC.2023.3257128`.

[14] M. Achanta, "Method and apparatus for least congested channel scan for wireless access points," U.S. Patent 20060072602A1, Apr. 6, 2006. [Online]. Available: `https://patents.google.com/patent/US20060072602A1/en` (visited on 11/16/2023).

[15] Arunesh Mishra, A. Mishra, Suman Banerjee, *et al.*, "Weighted coloring based channel assignment for WLANs," *Mobile Computing and Communications Review*, vol. 9, no. 3, pp. 19–31, Jul. 1, 2005. DOI: `10.1145/1094549.1094554`.

[16] M. Michalski and K. Staniec, "A simple performance-boosting algorithm for transmit power control in WLAN access points," *2016 21st International Conference on Microwave, Radar and Wireless Communications (MIKON)*, pp. 1–4, May 9, 2016. DOI: `10.1109/mikon.2016.7492075`.

[17] .. and.., ",", vol. 7, no. 3, pp. 25–37, 2021, ISSN: 1813-324X, 2712-8830. DOI: `10.31854/1813-324X-2021-7-3-25-37`.

[18] K. Ramachandran, R. Kokku, H. Zhang, and M. Gruteser, "Symphony: Synchronous two-phase rate and power control in 802.11 wlans," *IEEE/ACM Transactions on Networking*, pp. 132–145, Jun. 17, 2008. DOI: `10.1145/1378600.1378616`.

[19] Y. Lee, K. Kim, and Y. Choi, "Optimization of AP placement and channel assignment in wireless LANs," in *27th Annual IEEE Conference on Local Computer Networks, 2002. Proceedings. LCN 2002.*, Nov. 2002, pp. 831–836. DOI: `10.1109/LCN.2002.1181869`.

[20] R. Rodrigues, G. Mateus, and A. Loureiro, "On the design and capacity planning of a wireless local area network," in *NOMS 2000. 2000 IEEE/IFIP Network Operations and Management Symposium 'The Networked Planet: Management Beyond 2000' (Cat. No.00CB37074)*, Apr. 2000, pp. 335–348. DOI: `10.1109/NOMS.2000.830394`.

[21] A. Raschellà, M. Mackay, F. Bouhafs, and B. I. Teigen, "Evaluation of Channel Assignment Algorithms in a Dense Real World WLAN," in *2019 4th International Conference on Computing, Communications and Security (ICCCS)*, Oct. 2019, pp. 1–5. DOI: `10.1109/CCCS.2019.8888082`.

[22] "WiFi Market - Size, Share & Trends." (), [Online]. Available: `https://www.mordorintelligence.com/industry-reports/wi-fi-market` (visited on 11/16/2023).

[23] S. Arena, F. S. Crippa, N. Darchis, and S. Katgeri, *Understanding and Troubleshooting Cisco Catalyst 9800 Series Wireless Controllers*, 1st edition. Cisco Press, Jul. 25, 2022, 656 pp., ISBN: 978-0-13-749232-9.

[24] "Cisco CleanAir Technology At-a-Glance," 2014.

[25] JuniperNetworks, director, *Understanding Radio Resource Management RRM*, Jul. 19, 2023. [Online]. Available: `https://www.youtube.com/watch?v=BUmhVytJfXI` (visited on 10/31/2023).

[26] "Radio Management (technology) | Mist | Juniper Networks." (), [Online]. Available: `https://www.juniper.net/documentation/us/en/software/mist/mist-wireless/topics/concept/wireless-rrm.html` (visited on 11/16/2023).

[27] "Ruckus ChannelFly Feature Sheet." (Jan. 24, 2023), [Online]. Available: `https://web.archive.org/web/20230124150501/https://webresources.ruckuswireless.com/pdf/feature-sheets/fs-channelfly.pdf` (visited on 11/16/2023).

[28] "ArubaOS 8.6.0.0 User Guide," PSNow. (), [Online]. Available: `https://www.hpe.com/psnow/doc/a00092459en_us` (visited on 01/16/2024).

[29] "ARM Overview." (), [Online]. Available: `https://www.arubanetworks.com/techdocs/ArubaOS_60/UserGuide/ARM.php` (visited on 01/16/2024).

[30] "Taming Aruba ARM | Wifizoo." (Nov. 9, 2012), [Online]. Available: `https://wifizoo.org/2018/01/11/taming-aruba-arm/` (visited on 01/16/2024).

[31] "ARM Metrics." (), [Online]. Available: `https://www.arubanetworks.com / techdocs / ArubaOS _ 62 _ Web _ Help / Content / ArubaFrameStyles/ARM/ARM_Metrics.htm` (visited on 11/01/2023).

[32] K. Bharadwaj, S. Kalpalatha, and V. Ramaiyan, "A simulation framework for radio resource management in WiFi networks," in *2017 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, Bhubaneswar: IEEE, Dec. 2017, pp. 1–6, ISBN: 978-1-5386-2347-3. DOI: `10.1109/ANTS.2017.8384106`.

[33] "Ns-3: All TraceSources." (), [Online]. Available: `https://www.nsnam.org/docs/release/3.41/doxygen/de/de5/_trace_source_list.html` (visited on 05/06/2024).

[34] "The Authoritative Dictionary of IEEE Standards Terms, Seventh Edition," *IEEE Std 100-2000*, pp. 1–1362, Dec. 2000. DOI: `10.1109/IEEESTD.2000.322230`.

Term	Definition
Signal	Airborne RF energy
Channel	A band of frequencies that 802.11 devices can use for communications [34]
Inteference	Destructive influence of another signal leading to degradation of signal quality and loss of frames
Noise	Signal that cannot be demodulated as 802.11 signal
SNR (Signal-to-Noise Ratio)	Signal quality metric, defined as ratio of signal power to the noise
SINR (Signal-to-Interference-plus-Noise Ratio)	Signal quality metric, defined as ratio of signal power to the sum of noise power and power of interefering signals
RSSI (Received Signal Strength Indicator)	A metric of a wireless signal quality defined in 802.11, varying from 0 to 255, where exact mappings to Rx signal power are vendor-specific.
Radio cell	Geographical area covered by a radio transmitter [5]
Basic Service Set (BSS)	Set of stations belonging to the same radio cell and exchanging information [8]
Distribution System (DS)	Network that interconnects multiple BSSs and provides connectivity to a wired network [8]
Extended Service Set (ESS)	Set of Basic Service Sets connected by a DS [8]
Station (STA)	Client device that can connect to a wireless LAN
Access Point (AP)	Device providing stations wireless access to a distribution system
Infrastructure mode	Centralized mode of 802.11 WLAN operation using a star topology, where AP serves as a central entity for managing WLAN and switching traffic
Network capacity	Maximum transmission rate that any station can achieve in a given WLAN

TABLE III
Used terms and definitions