SDN-aware Framework for the Management of Cooperative WLANs/WMNs

by

Seyed Dawood Sajjadi Torshizi
B.Sc., Shahid Bahonar University of Kerman, Iran, 2004
M.Sc., Universiti Putra Malaysia, 2014

A Dissertation Submitted in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in the Department of Computer Science

SDN-aware Framework for the Management of Cooperative WLANs/WMNs

by

Seyed Dawood Sajjadi Torshizi
B.Sc., Shahid Bahonar University of Kerman, Iran, 2004
M.Sc., Universiti Putra Malaysia, 2014

Supervisory Committee

Dr. Jianping Pan, Supervisor
(Department of Computer Science)

Dr. Yvonne Coady, Departmental Member
(Department of Computer Science)

Dr. Xiaodai Dong, Outside Member
(Department of Electrical and Computer Engineering)

ABSTRACT

Drastic growth and chaotic deployment of Wireless Local Area Networks (WLANs) in dense urban areas are some of the common issues of many Internet Service Providers (ISPs) and Wi-Fi users. These issues result in a substantial reduction of the throughput and impede the balanced distribution of bandwidth among the users. Most of these networks are using unmanaged consumer-grade Access Points (APs) and there is no cooperation among them. Moreover, the conventional association mechanism that selects APs with the strongest Received Signal Strength Indicator (RSSI) aggravates this situation.

In spite of all these challenges, there is a great opportunity to build cooperative overlay networks among the APs that are owned by different ISPs, companies or individuals in dense urban areas. In fact, ISPs can distribute the resources among their customers in a cooperative fashion using a shared overlay platform which is constructed on top of the existing infrastructures. This approach helps the ISPs with efficient utilization of their resources and promoting the Quality of their Services (QoS). For instance, cooperative association control among the APs of different ISPs enables them to alleviate the drastic impact of interference in populated areas and improves the network throughput. Indeed, all Wi-Fi customers can associate to the APs from different ISPs and it leads to the construction of a large unified WLAN that expands the network coverage, significantly. Moreover, it results in a notable reduction of deployment costs and enhancement of customer satisfaction. Hence, as one of the key contributions of this dissertation, a cooperative framework for fine-grained AP association in dense WLANs is presented. On top of this framework, a thorough formulation and a heuristic solution to solve the aforementioned problems are introduced. The key enabler of the proposed solution is Software Defined Networking (SDN) which not only gives us an exceptional level of granularity but also empowers us to utilize high-performance computing resources and more sophisticated algorithms.

Also, over the past few years, some of the largest cellular operators restricted their unlimited data plans and proposed tiered charging plans enforced by either strict throttling or large overage fees. While cellular operators are trying to guarantee the QoS of their services in a cost-effective and profitable manner, WLANs and Wi-Fi Mesh Networks (WMNs) as viable complements can be used to form a multi-hop backhaul connection between the access and the core networks. Indeed, the

utilization of WMNs provides an opportunity to achieve a high network capacity and wide coverage by the employment of inexpensive commercial off-the-shelf products. Moreover, by bridging the WMNs and cellular networks, and the fine-grained traffic engineering of network flows, it is possible to provide a cost-effective Internet access solution for people who cannot afford the high cost of data plans. However, there are certain requirements in terms of QoS for different services over multi-hop backhaul networks. In addition, the process of service provisioning in WMNs incorporates tightly correlated steps, including AP association, gateway selection and backhaul routing. In most of the prior studies, these steps were investigated as independent NP-hard problems and no unified formulation that considers all these steps (at different tiers of WMNs) has been presented. Hence, as another contribution of this dissertation, a structured and thorough scheme to address the demands of end-users over SDN-aware WMNs is introduced. In contrast to most of the former work, this scheme takes the key characteristics of wireless networks into account, especially for Multi-Channel Multi-Radio WMNs. The proposed solution can be applied to the large-scale scenarios and finds a near-optimal solution in polynomial time. Furthermore, since the presented solution may split the packets of a single flow among multiple paths for routing and there are non-trivial drawbacks for its implementation, a randomized single-path flow routing for SDN-aware WMNs is introduced. The randomized nature of the introduced solution avoids the complexities of implementing a multi-path flow routing and it presents a viable routing scheme that guarantees certain performance bounds.

The functionality and performance of all the presented solutions have been assessed through extensive numerical results and real testbed experimentations as a proof of concept. It is important to note that the solutions presented in this dissertation can be utilized to provide a large variety of services for Wi-Fi users, while they guarantee different QoS metrics.

Contents

List of Tables

List of Figures

List of Abbreviations

ACG	Association using Conflict Graph
ACO	Ant Colony Optimization
AP	Access Point
APC	Access Point Coordinate
BW	Bandwidth
CCA	Clear Channel Assessment
CFLP	Capacitated Facility Location Problem
CHM	CHannel monitoring Module
CR	Communication Range
CSMA/CA	Carrier Sensing Multiple Access/Collision Avoidance
DCF	Distributed Coordination Function
ET	Exposed Terminal
EL	Egress Link
GAP	Generalized Assignment Problem
GC	Gateway Candidate
HT	Hidden Terminal
IL	Ingress Link
IS	Independent Set
ISP	Internet Service Provider
JFI	Jain's Fairness Index
LHS	Left-Hand-Side
LLDP	Link Layer Discovery Protocol
LLF	Least Loaded First
LP	Linear Programming
LRC	Link-Radio-Channel
MC-MR	Multi-Channel Multi-Radio
MCFP	Multi-Commodity Flow Problem

MDCG	Multi-Dimensional Conflict Graph
MILP	Mixed Integer Linear Programming
MINLP	Mixed Integer Non-Linear Programming
MIMO	Multiple-Input Multiple-Output
MIS	Maximal Independent Set
MR	Mesh Router
MRAG	Mesh Router, Access Point, and Gateway
NG	Network Graph
NLAO	Non-Linear Approximation Optimization
NOM	Network Optimizer Module
OFDP	OpenFlow Discovery Protocol
OGM	Originator Message
OOB	Out-Of-Band
OVS	Open vSwitch
PCS	Physical Sensing Range
PGC	Potential Gateway Candidate
PREQ	Path Request
QoS	Quality of Service
RHS	Right-Hand-Side
RSSI	Received Signal Strength Indicator
RSR	Randomized Single-path flow Routing
RTS/CTS	Request To Send/Clear To Send
SC-SR	Single-Channel Single-Radio
SDN	Software Defined Networking
SD-WLAN	Software Defined Wireless Local Area Network
SISO	Single-Input Single-Output
SP	Simple Path
SSF	Strongest Signal First
SSR	Simple Single-path flow Routing
STC	Station Coordinate
VAP	Virtual Access Point
VR	Violation Rate
WLAN	Wireless Local Area Network
WMN	Wi-Fi Mesh Network

ACKNOWLEDGEMENTS

First, I would like to thank my wife, Maryam, as a true colleague, companion and friend whose support and encouragement during my PhD program helped me to have many achievements. Then, I would like to express my gratitude to Prof. Jianping Pan for his guidance and advice during my studies. I appreciate having his support within the past 4 years and I am thankful for his comments that helped me to enhance the quality of my publications and this dissertation.

Also, I am very grateful to Prof. Yvonne Coady and Prof. Xiaodai Dong for spending their precious time to serve as the members of my supervisory committee. Furthermore, I would like to appreciate Dr. Rukhsana Ruby due to her great help and cooperation during my research. Finally, my gratitude goes also to my teachers, friends and fellow labmates that not only helped me to broaden my knowledge and develop my research skills but also create many beautiful and memorable moments during my academic life at UVic.

Dawood Sajjadi

DEDICATION

This dissertation is dedicated to my dearest wife, Maryam, for her wholehearted support and my caring and devoted parents for their unconditional love.

Chapter 1

Introduction

1.1 Overview

Nowadays, around 6 billion mobile phone connections exist world-wide and the impacts of wireless technologies permeate most aspects of our societies including health care, finance, education and community empowerment. During recent years, the emergence of diverse online applications for almost every daily task has led to the tremendous growth of wireless traffic over the cellular and Wi-Fi networks.

By the deployment of cutting-edge cellular technologies such as LTE/4G networks, some people inferred that Wireless Local Area Networks (WLANs) will be diminished gradually in the wireless communications market. Nevertheless, due to the presence of high-demand applications and devices, it is predicted that a significant amount of 4G traffic will be offloaded to the Wi-Fi networks. For instance, according to Cisco visual networking index [1], the amount of cellular traffic that will be offloaded to WLANs is expected to increase from 3.9 exabytes/month in 2015 to 38.1 exabytes/month by 2020. Fig. 1.1a and Fig. 1.1b illustrate a clear picture on the offloaded traffic growth in the span of 5 years and the share of this traffic over different cellular technologies, respectively. Moreover, multiple mobile carriers have recently presented Voice over Wi-Fi services to their customers. This service is a great help to address the challenges related to the Quality of Service (QoS) for the indoor mobile users as well as cutting down the roaming charges. According to Fig. 1.1c, Voice over Wi-Fi overtook Voice over LTE at the end of 2016 and it will have 53% of the total mobile IP voice traffic by 2020. In addition, the expansion of community hotspots as a key element of public Wi-Fi plays a pivotal role in this process. In fact, community hotspots allow a portion

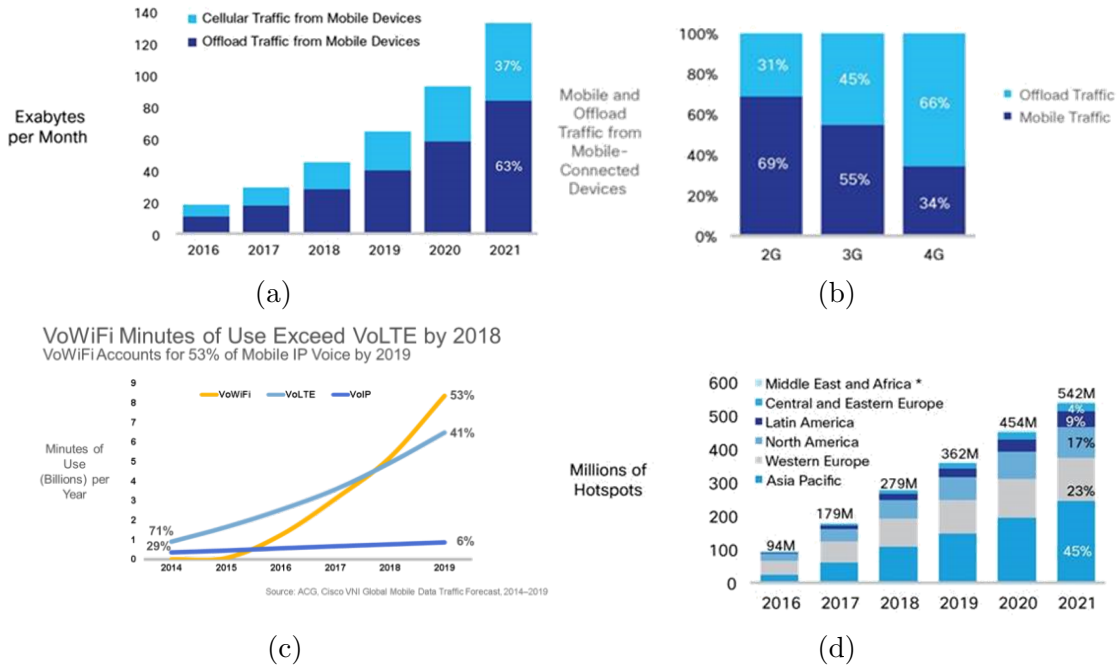
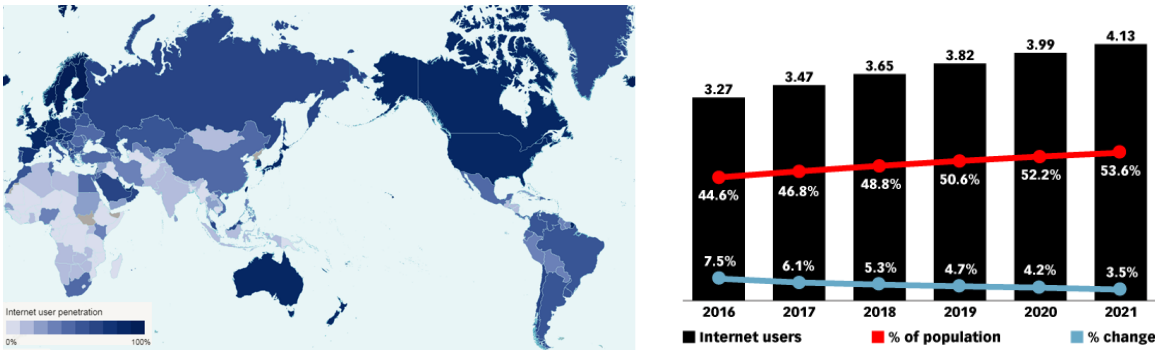


Figure 1.1: Fast growth of Wi-Fi traffic according to Cisco VNI [1].

of their capacities to be open for the public that facilitates the offloading of cellular traffic. The expected growth of Wi-Fi hotspots around the world by 2021 is shown in Fig. 1.1d.

All these facts not only show the critical role of WLANs on developing the next-generation of wireless networks but also substantiate that they are becoming drastically dense and chaotic, which is one of the consequences of bandwidth provisioning for highly growing user demands through adding more Access Points (APs). By increasing the number of APs, theoretically we can promote the Quality of Experience (QoE) for the users through reducing the number of associated users to each AP. However, since the number of non-overlapping channels is limited, neighboring APs within the dense areas must operate on the same channel. This fact exacerbates the design complexity of WLANs. Furthermore, due to the contention based nature of MAC protocols in Wi-Fi networks and the backoff procedure, there is a considerable throughput degradation in such areas. Also, it is important to note that AP selection based on the strongest RSSI has remained as the most common approach for associating the Wi-Fi users to WLANs and it cannot reach the maximum network throughput. In spite of all these challenges for the management of dense WLANs, there is an opportunity for Wi-Fi stations to associate with the APs which



(a) (b)

Figure 1.2: Map of Internet penetration and the growth of Internet users [2].

provide them with the highest end-to-end throughput. Hence, dynamic association of Wi-Fi clients (to the APs) can be used as an efficient strategy to maintain network throughput while it does not need any modification of the existing Wi-Fi standards and protocols.

Also, according to [3], around 3.7 billion people (half of the Earth's population) still do not have any kind of Internet access and the lack of this connectivity deprives them from social and economic benefits of the Internet. Fig. 1.2a shows a world map of the Internet user penetration that substantiates the fact that a large number of people are still deprived of using the Internet services. Fig. 1.2b illustrates an explicit representation of this fact. It is important to note that although more than 90% of the world's population is living within the range of cellular networks [4], the current prices of mobile data plans are not universally affordable and many people cannot use them to access Internet [5]. In addition, over the past few years, some of the largest cellular operators restricted their unlimited data plans and proposed tiered charging plans to reduce their expenses [6]. While cellular operators attempt to guarantee the quality of their services in a profitable manner, Wi-Fi networks can be considered as a viable complement. The massive growth of inexpensive Wi-Fi-enabled devices as well as the utilization of unlicensed spectrum makes Wi-Fi an attractive data offloading technology for the operators to avoid substantial deployment costs of cellular networks.

The new generations of wireless solutions promise high throughput and economic connectivity to everyone (especially for disconnected communities), which can be provided through existing and emerging technologies. One of the existing technologies is Wi-Fi Mesh Networks (WMNs), which not only offer a promising and affordable

solution but also provide diverse types of services where wired infrastructure is either not ready or too costly to be deployed. In general, WMN supports long distance communications by the establishment of multi-hop paths. Since WMNs can be constructed using inexpensive off-the-shelf products and unlicensed spectrum, it is a valuable solution.

Among the emerging technologies, Software Defined Networking (SDN), as a centralized network paradigm, has made significant headways in communication networks. One of the key advantages of SDN is implementing the control functions of network equipment at logically centralized modules. This feature allows us to utilize simple networking devices only for packet switching operations that reduces the deployment cost and energy consumption [7, 8]. Moreover, the centralized placement of control functions empowers us to utilize high-performance computing resources and more sophisticated algorithms for resource management. Indeed, SDN presents more flexibility and fine-granularity in terms of traffic engineering that can be applied to many communication platforms, including WMNs.

Regarding the stated problems and potential solutions, in the next section, we elaborate the key motivations and applications discussed in this dissertation.

1.2 Motivations

In the following subsections, we discuss the key motivations to conduct my research for finding efficient and feasible solutions for the aforementioned problems. These motivations have been categorized based on their applications and benefits that provide to the Wi-Fi clients.

1.2.1 Cooperative AP association in dense Software Define WLANs (SD-WLANs)

In one of the recent large-scale measurement conducted by Meraki [9], it is shown that the average number of interfering APs in dense urban areas can be more than 50 (in 2.4 GHz bands). It is important to note that the primary channel access scheme in enterprise WLANs is still the Distributed Coordination Function (DCF), which uses a random access mechanism to address the contention among Wi-Fi transmitters. Random back-off process and uncoordinated medium access result in the waste of airtime resources. To improve the performance of dense WLANs, a well-known

solution is to maximize the number of successful concurrent transmissions of existing links. However, this goal is too challenging in WLANs due to the contention-based nature of the DCF protocol [10]. Moreover, the growing demands on WLANs and mobile Internet services (using Wi-Fi hotspots) necessitate the need for coordination among neighbor WLANs. Cisco [11], Meraki [12] and Huawei [13], as some of the key players of Wi-Fi market have realized that centralized frameworks are more productive for resource allocation over enterprise WLANs. The main advantage of a centralized design is to have a holistic view of the network that enables us to find a fair estimation of the time to transmit a data packet for an interference-free reception.

If we extend this vision to multiple ISPs, they can share their resources to control client usages in a cooperative manner for maximizing the customer satisfaction and network performance. Indeed, sharing a common WLAN infrastructure among several ISPs extends the network coverage which facilitates the user connectivity as well as reducing the operational costs. In this case, Software Defined WLANs (SD-WLANs) can be utilized as a promising technology to build such a platform for providing different services. Among diverse strategies to maximize the throughput of WLANs, AP association, as the initial step in the establishment of any Wi-Fi connection can be utilized to increase the number of successful transmissions. Hence, dynamic AP association through a centralized approach over a cooperative SD-WLANs empowers us to propose a feasible and effective solution for alleviating the undesired impact of interference.

1.2.2 Fine-grained traffic engineering on SDN-aware WMNs (SD-WMNs)

As stated earlier, due to the high price of data plans, many people with low income cannot afford the cost of cellular data plans for the Internet access. However, by using an intermediate approach, we can bridge data traffic from cellular to Wi-Fi networks and share it among users through a WMN with an affordable cost. Applying SDN technology to a WMN to form an SD-WMN allows us to employ more sophisticated algorithms for resource management. Indeed, SDN presents more flexibility and fine-granularity in terms of traffic engineering that can be applied to different communication platforms, including WMNs. Also, for serving every user of a WMN, three key steps, including AP association, gateway selection and backhaul flow routing should be taken into account. Despite the tight correlation among these

steps, most of the former studies either investigated them as independent NP-hard problems or did not present a unified and all-inclusive solution. Indeed, finding a unified and thorough solution allows us to ensure that all the flows get a minimum guaranteed bandwidth w.r.t. the predefined QoS constraints over the mesh backhaul network as well as to maximize the aggregated network throughput.

In addition, it should be noted that minimizing the number of required (cellular) gateways in such a platform not only is in line with the tiered charging strategy of large cellular operators but also has a significant impact on the deployment process. Hence, it is important to control the number of Mesh Routers (MRs), as the building blocks of WMNs, that function as cellular gateways to reduce the deployment cost. Otherwise, the users that run bandwidth-greedy applications impose a significant cost on the service providers by downloading a large volume of data from cellular networks. Thus, by activating the gateway functionality just for a limited number of MRs and steering the traffic flows, it is possible to reduce the cost of service provisioning and provide an affordable solution for the Internet access.

Furthermore, the dense deployment of WLANs in urban areas can impose significant costs on service providers in terms of the wired connectivity and in such a scenario, WMNs can be used as a cost-effective solution to form a multi-hop backhaul connection between the access and the core networks. The majority of existing WMN solutions are characterized by their distributed nature and they suffer from inflexible traffic routing mechanisms and long recovery of link failures. This approach not only leads to a poor performance but also is not able to utilize the total backhaul capacity as well as reacting swiftly to the unexpected topology changes [14]. Hence, by building a logically centralized solution using SDN technology, we can have a holistic view of the network dynamics and propose an efficient single-path flow routing solution that can be applied to SD-WMNs. It is important to note that the implementation of multi-path flow routing over Multi-Channel Multi-Radio (MC-MR) WMNs has extra complexities and to avoid them, a randomized single-path flow routing can be applied on top of the SDN-aware WMNs.

1.3 Contributions

A summary of the key contributions presented in this dissertation is as follows. It should be noted that one of the key features of the presented solutions is their minimum requirement for system/protocol modifications as well as their backward-

compatibility with the existing Wi-Fi standards.

1. A cooperative and modular framework based on dynamic AP association to share resources among different ISPs is presented. By using a logically centralized management approach, the introduced framework considers all the APs belonging to different ISPs as a single entity and shares the network resources among their customers. The main goal of this framework is fairness provisioning among Wi-Fi stations through dynamic AP association. We presented a thorough mechanism for AP association, channel assignment, and bandwidth control of network flows in dense WLANs. The key enabler of this solution is SDN technology that gives us fine-grained control for both downlink and uplink traffic. An optimization problem to find an approximated optimal solution for AP association is introduced. One of the key features of our scheme is finding a bounded solution in polynomial time that makes it a suitable choice for the large-scale SD-WLANs. Also, since finding the Maximal Independent Sets (MISs) of a given network graph is an NP-hard problem, We introduce a heuristic algorithm that not only finds the best candidates as the inputs of the presented problem, but also reduces the overall complexity of our solution significantly. Moreover, a practical mechanism for quantifying the load of each Wi-Fi channel to use it as an input of the optimization problem is presented. The performance of our work is evaluated through simulations and testbed experimentations as a proof of concept.

2. An optimization framework for joint traffic engineering including gateway selection and flow routing for SDN-aware WMNs is introduced. The introduced framework can bridge the cellular and mesh networks to provide an affordable Internet access solution. A Mixed Integer Non-Linear Programming (MINLP) problem is presented to solve the joint problem. The presented solution finds the least number of required cellular gateways in an adaptive manner and it guarantees the predetermined QoS metrics for the network flows. Also, due to the NP-hardness of the MINLP problem, a heuristic algorithm to find a near-optimal solution in polynomial time is proposed. The performance of the presented solution has been evaluated through extensive simulations and testbed experimentation. Moreover, since the implementation of multi-path flow routing over MC-MR WMNs brings some complexities including packet re-ordering and the modification of the SDN protocols, a randomized single-

path flow routing solution is presented. Through numerical results, it is shown that the presented routing solution follows the theoretical, tighter and more general performance bounds. Besides, a testbed-based performance comparison between the introduced scheme and the most popular WMN routing protocols in terms of network throughput and control overhead is carried out.

Note that the key novelty of this dissertation is twofold. First, in Section 2.3.2 and Section 2.3.3 of Chapter 2, a traffic engineering technique for AP association in SDN-aware WLANs is presented. By calculating a subset of the MISs for the conflict graph of a given network topology, we can find a 2-approximation of the optimal association map (for both uplink and downlink traffic) that outperforms the state-of-the-art solution. Second, in contrast to all the prior related work, a polynomial time algorithm to find a near-optimal solution for one of the major joint traffic engineering problems on MC-MR WMNs is introduced in Chapter 3. The proposed algorithm in this chapter considers all the key steps of the service provisioning process on MC-MR WMNs.

1.4 Outline of the Dissertation

The outline of this dissertation is as follows.

Chapter 1 contains a brief introduction of the research potentials and the motivations on the selected topic. Also, a statement of the proposed contributions and the organization of the dissertation are presented in this chapter.

Chapter 2 contains the problem statement and illustration of the proposed cooperative framework for dynamic AP association in WLANs. We introduce a solution that can be used to find an approximation of the optimal association map. In addition, a thorough performance evaluation of the introduced solution are presented and discussed in this chapter.

Chapter 3 contains the corresponding problem formulation for access provisioning via joint flow routing and gateway selection in mesh networks. Also, it presents a problem formulation for randomized single-path flow routing for SDN-aware WMNs. The introduced solution avoids the complexities of a multi-path routing

and introduces certain performance bounds. The functionality of the introduced solutions is evaluated through simulations and a real testbed implementation.

Chapter 4 contains a restatement of the claims and the results presented in this dissertations. Moreover, a list of the potential applications and future work on top of the presented solutions are outlined in this chapter.

Chapter 2

Cooperative AP Association in Dense Software Defined WLANs (SD-WLANs)

2.1 Background

Drastic growth of data traffic in cellular networks forces mobile network operators to use Wi-Fi offloading as a cost-effective and feasible alternative for maintaining the quality of their services. Hence, there would be a tremendous contention among neighbor APs in dense WLANs to access the spectrum. According to a study [9], the average number of interfering APs in dense urban areas can be more than 50 and there is no direct relationship between the number of nearby APs and the utilization of Wi-Fi channels, necessarily.

In WLANs, the link quality is subject to several parameters including spectral diversity, MAC dynamics and the degree of channel load. The primary channel access scheme in enterprise WLANs is still the Distributed Coordination Function (DCF), which uses a random access mechanism to address the contention among Wi-Fi transmitters. Random back-off process and uncoordinated medium access result in waste of airtime resources. The 802.11 DCF protocol adopts carrier sensing mechanism in order to avoid collisions among devices, and it is well-known to result in the suboptimal scheduling of transmission opportunities. If the sensing range is too high, the links within the active sensing range are not able to transmit simultaneously, which is called the Exposed Terminal (ET) problem [10]. On the

other hand, if the sensing range is too low, nodes might not be able to sense conflicting transmissions that produce collisions, which is known as the Hidden Terminal (HT) problem. Existing solutions including DCF and RTS/CTS (due to the significant airtime overhead) are not able to mitigate this issue completely [15]. Also, it should be noted that the existing 802.11 MAC protocol attempts to give the same chance to all the stations that are associated to the same WLAN and it reduces the overall network throughput in multi-rate WLANs, significantly. The main reason behind this phenomenon is the unbalanced channel occupancy among the stations with different data rates, which is known as rate anomaly (due to the low data rates of legacy stations) [16]. Moreover, the stations may suffer from the interference caused by non-Wi-Fi devices, such as cordless phones and microwave ovens that do not follow 802.11 standard and cause significant link performance degradation [17].

Network over-provisioning can be used as a short-term solution in dense WLANs by adding more APs to maintain user satisfaction. Although this strategy can be justified by the increasingly inexpensive off-the-shelf 802.11 equipment, it leads to a high level of contention among nearby APs, and poor user experience in the long run. To improve the performance of dense WLANs, a well-known solution is to maximize the number of successful concurrent transmissions of existing links. However, this goal is too challenging in WLANs due to the contention-based nature of the DCF protocol [10]. The growing demands on residential WLANs and mobile Internet services aggravate the need for coordination among neighbor WLANs. Cisco [11], Meraki [12] and Huawei [13], as some of the key players of Wi-Fi market have realized that centralized frameworks are more productive for resource allocation over enterprise WLANs. The main advantage of a centralized design is to have a holistic view of the network that enables us to find a fair estimation of the time to transmit a data packet for an interference-free reception. Since the dominant fraction of traffic in enterprise WLANs is on downlink, a centralized solution can considerably reduce potential interference among neighbor APs [15].

In spite of all the challenges for the management of dense WLANs, there is an opportunity for Wi-Fi stations to associate with the APs that provide them with the highest end-to-end throughput. In addition, in the populated urban areas which contain many APs, ISPs can share their resources to control client usages through a cooperative manner for maximizing the customer satisfaction. In fact, cooperative association control enables the ISPs to alleviate the impact of drastic interference and throughput degradation in such environments. Furthermore, a common WLAN

infrastructure among several ISPs extends the network coverage and reduces the deployment costs, significantly. In this situation, since the network capacity is shared within a WLAN infrastructure, using an efficient resource sharing scheme through a centralized controller is a necessity for the management of cooperative WLAN infrastructures. In contrast to vendor-specific services for (centralized) enterprise WLANs which are based on proprietary products, SDN-based solutions not only can be applied to the scenarios with a diverse set of equipment, but also present an exceptional opportunity for provisioning and development of efficient services. Indeed, SDN enables network programmability as well as the separation of control and data planes. Using SDN, the control functions of network devices are accomplished by logically centralized software, and hence networking equipment becomes simple forwarding engines. Note that this approach brings significant enhancements to all the services placed at the SDN controllers in terms of performance and security.

In this chapter, we introduce a Software Defined framework for WLANs (SD-WLANs) to provide a set of services for the stations that share a common Wi-Fi infrastructure. The presented solution considers the key characteristics of WLANs to increase the number of concurrent transmissions w.r.t. the aforementioned issues. Moreover, we focus on one of the most essential problems in WLANs, which is AP association. AP association, as the initial step in the establishment of any Wi-Fi connection, plays a crucial role in the operation of WLANs. Finding the best association of APs to the stations in a given WLAN is NP-hard and can be mapped to the classic computer science Generalized Assignment Problem (GAP). By taking the relaxation and rounding techniques, it is possible to find an approximation of the optimal solution for the AP association in polynomial time.

2.2 Related Work

In the past decade, a large body of research work was conducted on the development of centralized or SDN-based frameworks for Wi-Fi infrastructures. Each of these solutions provides a different set of features to improve the user experience and network performance. In this section, we first introduce some of the most recent work on centralized solutions for resource management of WLANs. Then, the most related work about dynamic AP association in Wi-Fi networks is explained.

2.2.1 Centralized Frameworks for WLANs

CloudMAC [18], Odin [19], OpenRoads [20], OpenRadio [21], CENTAUR [15], CoAP [22], meSDN [23] and UniFlex [24] are some of the most recent and popular centralized solutions that extend the SDN technology to Wi-Fi infrastructures. Indeed, they intend to transform the distributed nature of conventional WLAN services to a centralized basis which empowers us to develop more advanced and efficient solutions.

The development of managed services for residential WLANs attracted tremendous attentions in recent years. For instance, CoAP [22] proposed a vendor-neutral centralized framework for the management of residential 802.11 networks using an open API. In other related work [25, 26], the scholars developed a framework for secure and fine-grained radio resource management of residential WLANs. The proposed solutions were implemented and evaluated as prototypes that can be applied to the legacy systems. This centralized strategy can be extended to the mobile devices as well. In [23], the authors presented an extension of SDN for mobile devices which is called meSDN. meSDN presents a pseudo-TDMA resource allocation scheme in transport layer that controls uplink/downlink traffic and operates on top of the Wi-Fi driver. In spite of this feature, it carries out coarse-grained airtime scheduling which is not on a per-packet basis and does not guarantee interference-free airtime.

In addition to the introduced work, there is a large body of research to minimize the impact of interference in WLANs using centralized solutions. One of the most efficient ways to address this issue is to increase the number of simultaneous interference-free transmissions in WLANs. In [27], the authors proposed a solution (CMAP) to determine conflicting links during concurrent transmissions. By reactively and empirically learning of transmission conflicts, they presented a solution to resolve the ET/HT problem. In another centralized hybrid scheme [15], a network conflict graph was constructed periodically and leverages centralized scheduling to address downlink interference. Note that these solutions either assume a uniform bit rate among APs and stations, or adopt existing bandwidth allocation schemes for interference-free scenarios. However, other work [28] shows that there is a significant impact of bit rate on the ET problem, which affects the interfering links of a given network. In another related work [29], signal strength values are utilized to build a channel quality map of downlink traffic for the interference management in SD-WLANs. Also, in [30], the scholars presented a centralized solution for building and

management of Virtual APs (VAPs) to achieve finer channel assignment and better load control over WLANs. Considering high density scenarios, similar framework was introduced in [31] for the aggregation of multiple VAPs into a single physical AP. TDM was utilized as the channel access mechanism to avoid collision among the aggregated VAPs and all VAPs had to be placed within the same broadcast domain.

It is important to note that the last hop of WLANs is a shared half-duplex medium and DCF does not allow APs to control uplink transmissions of Wi-Fi stations. Extension of SDN to the Wi-Fi stations can be utilized as a solution to resolve this issue. Most of the SDN-aware solutions [20, 32] only limit the downlink traffic and do not have any control on uplink transmissions. In contrast to most of this work, we propose a hybrid mechanism to control downlink and uplink traffic in our SDN-aware framework. Due to the large volume of downlink traffic in WLANs [15], a fine-grained bandwidth allocation is applied to the downlink traffic. On the other hand, similar to the approach presented in [23], we use a traffic control strategy for uplink flows generated by the stations. The stepping stone of these processes is the construction of a multi-dimensional conflict graph that considers the impact of ET/HT to find the interference-free links. Moreover, we use the load of Wi-Fi channels in the link scheduling process to have a better estimation of the optimal results.

2.2.2 AP Association in Wi-Fi Infrastructures

Rapid growth of Wi-Fi traffic and the limited number of non-overlapping channels in unlicensed bands exacerbate the complexity of resource allocation in dense WLANs. In such a condition, finding near-optimal solutions for AP association can play a pivotal role in increasing the network throughput and fair distribution of network resources among greedy stations.

In several studies, the scholars developed centralized association schemes for WLANs to improve the overall network performance. In one of the most recent work [33], researchers proposed a mathematical model to find an association map that maximizes a logarithmic utility function to achieve the best trade-off between the aggregate throughput and fairness. A local search algorithm was presented to approximate the optimal result and the functionality of the algorithm was evaluated via extensive simulations. In another work [34], energy efficiency and migration constraint were considered as the main criteria to find the best user association map. The scholars formulated the problem as an integer linear programming model and by

using simulation and numerical analysis investigated the performance of their work.

In some work, the scholars focused on application-specific scenarios to apply different association control mechanisms. For instance, in [35], a multi-step centralized solution to improve the quality of video streams for Wi-Fi clients was proposed. The authors utilized selective roaming and re-association for load balancing of Wi-Fi clients in WLANs and they validated the performance of their scheme through different simulation scenarios. Moreover, in [36, 37], the researchers proposed centralized and tiered solutions for the association control problem in heterogeneous networks including Wi-Fi and LTE users. Similar to most of the prior work, aggregate throughput and fairness index were chosen as the key metrics to evaluate the performance of the proposed solution.

In [38], the authors used a Markovian model to formulate a flow-level delay-minimized client association problem. Although the authors developed a suboptimal heuristic solution to address the problem, the presented work was only evaluated for downlink traffic and the assessed scenario did not represent a dense WLAN with a large number of stations. In another work [39], a meta-heuristic mechanism for dynamic AP association in shared WLANs on a common infrastructure was introduced. The presented scheme showed convincing results in terms of fairness provisioning and was evaluated through extensive numerical analysis and real testbed experimentations. However, the assigned channels to the APs were assumed as a given input.

In one of the most relevant work [40], a non-linear approximation optimization algorithm was proposed to find the optimal association matrix for multi-rate Wi-Fi stations. To solve the optimization problem, a compensation function was introduced to fill the integrality gap caused by the relaxation. Also, an online algorithm for optimal association of newly joined users was presented and examined for multi-rate WLANs. However, the functionality of the introduced schemes is evaluated only for downlink traffic flows in single-channel WLANs. In another similar work [41], a centralized collaborative association scheme to provide proportional fairness among the customers of ISPs with the same upstream link is presented.

Also, some scholars used monomial approximation techniques to solve the problem of optimal association and airtime control over multi-cell WLANs. For instance, in [42, 43] it was shown that by solving the geometric-based optimization problem, network throughput and fairness can be guaranteed among ISPs regardless of the number of ISP users. The main downside of this work is disregarding the impact of

interference in dense environments. It was assumed that all APs were functioning in orthogonal channels, which is not reasonable in populated areas. Also, the authors assumed that each user can associate to multiple APs, simultaneously.

In [44], another centralized approximation mechanism is proposed to solve the joint association and bandwidth allocation problem in mesh networks. Moreover, in a similar work [45] the authors considered the impact of user migration cost to find a better approximation of the AP association for max-min fairness provisioning. It is important to note that although all these schemes have been evaluated through extensive simulations, they are only applicable to downlink scenarios without presenting any channel assignment solution. In addition, the impact of HT was either ignored or not considered accurately in the presented outcomes and most of them did not evaluate the functionality of their work via a testbed experimentation. It should be noted that although the majority of related work has focused on centralized strategies, there is limited work that used decentralized approaches to solve the association problem [46].

In this chapter, we present a thorough centralized association scheme that not only performs the station association in a fair manner, but also carries out channel assignment, bandwidth allocation and link scheduling through our introduced SDN-aware framework. In the next section, the different components of our proposed solution are introduced in detail.

2.3 Proposed Framework

Nowadays, we can easily find a large number of hotspots in public areas such as city centers that belong to different service providers for their own customers. Due to the large presence of these APs especially in dense environments and the lack of a centralized resource coordinator, the Wi-Fi clients suffer from low download rate and severe interference caused by nearby APs. Now, if we assume that the service providers agreed to share their APs (hotspots) such that each client is able to associate with any AP in its neighborhood, then we have a large common WLAN infrastructure that not only extends the service coverage of the customers, but also improves the user experience. However, we need a central controller to utilize the network resources in an optimal fashion by maximizing the number of concurrent successful transmissions. In addition, since the network capacity is shared within such an infrastructure that serves the customers of different service providers, the

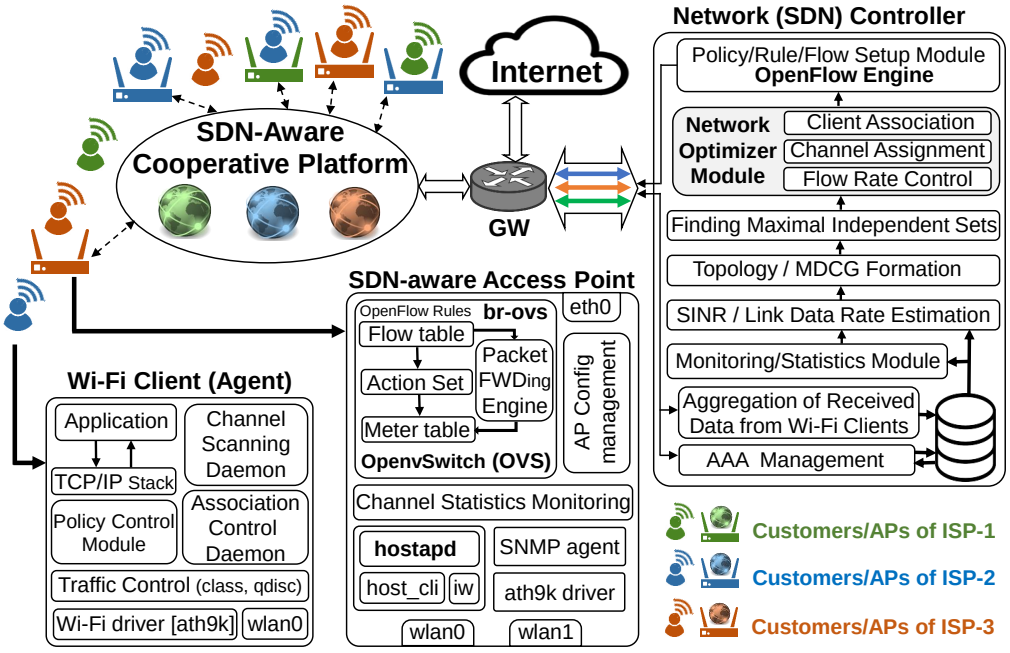


Figure 2.1: The big picture of the presented framework.

growth of network traffic generated by customers of one provider can lead to the traffic decline in another one. Thus, we have to distribute the airtime among the clients based on the share assigned to their service providers.

For the formation of such a common platform, SDN can be considered as the best candidate. To build this platform, three major elements are required. The first one is the network controller which monitors and manages services including AP association, bandwidth allocation, channel assignment, link scheduling and flow setup. Note that this component observes the hotspots that belong to different service providers as a unified WLAN. The second component of the introduced solution consists of Software-Defined APs which are in charge of fine-grained airtime allocation to the downlink traffic. Moreover, the APs monitor the spectrum activity to find the best channel candidates to operate in. The last part is the software agent installed on the Wi-Fi station to provide dynamic AP association and traffic control for the end-users. It is important to note that recently, many companies with large-scale services encourage their customers to install verified software agents on desktop/mobile devices for improving the performance and security of their services [47]. Hence, this part of our proposed framework can be deployed as a module of these agents on the end-

devices. In addition, this component enables us to manage uplink transmissions over SD-WLANs to avoid the destructive impact of greedy uplink flows.

Fig. 2.1 illustrates a big picture of the introduced framework. In this example, we have three ISPs that share their APs among the Wi-Fi clients. Every AP broadcasts the SSIDs of three different ISPs. So, the customers of each ISP can be associated to any AP that belongs to the illustrated infrastructure and they will be served based on their service agreements and the total airtime share of their ISPs. The resource allocation including AP association will be handled by a north-bound application on top of the controller which is shown as the `Network Optimizer Module (NOM)` in Fig. 2.1. As illustrated, several preprocessing steps at the SDN controller have to be taken before passing the required inputs to the north-bound application. Note that the modularity and fine-granularity of the proposed SDN-aware framework empower us to develop scalable and hierarchical resource allocation schemes. Hence, for the large-scale scenarios, it is possible to apply tiered clustering approaches to the given network topology to reduce the complexity and the convergence time of the original problem. In the next subsection, further details of the proposed AP association scheme are presented.

2.3.1 AP Association in Dense WLANs

As mentioned in Section 2.1, finding the best AP association map in WLANs has an outstanding impact on the network performance. In addition, it is possible to involve other resource allocation processes such as channel assignment and link scheduling into this problem. It should be noted that at the first look, it might seem possible to pre-configure the APs in crowded areas such that they do not interfere with each other. However, due to the chaotic deployment of neighbor APs (especially in residential and commercial areas) and the existence of non-Wi-Fi devices operating in ISM bands, having co-channel interference is inevitable. Furthermore, since the number of non-overlapping channels in ISM bands is limited and by increasing the number of neighbor APs (especially in dense WLANs), it is inevitable to assign the same channels to the nearby APs which causes co-channel interference. Hence, the pre-configuration of AP channels alone is not enough to address this issue and we need to change the AP channels in an adaptive way to minimize the impact of interference and maintain a minimum level of service. In this section, we introduce a systematic approach for AP association in high-density environments using a Software-Defined

Wi-Fi infrastructure for service provisioning. Our solution needs to collect topology information continuously with the installed software agent and use them to find a good approximation of the optimal association map. After collecting the required information, we have to take the following steps.

Building Network Graph

First, we build the topology graph to represent possible links that can be established between APs and stations. To build such a graph, in addition to the coordinates of APs and stations, we need to have the Communication Range (CR) of each node. By using the installed agents at Wi-Fi clients, we are able to collect channel information similar to the mechanism explained in [39] to build the network graph. In this approach, every station sends a list of its nearby APs and their respective Received Signal Strength Indicators (RSSIs) to the SDN controller. Then, we can calculate the SINR and consequently the data rate of wireless links between every station and its surrounding APs. These phases are shown at the SDN controller in Fig. 2.1. The outcome of this step is a directed graph that illustrates the traffic flows either from the APs to the stations (downlink) or vice versa (uplink). It should be noted that the frequency of building the network graph is a function of the network conditions and the predefined thresholds to detect the users' movements. Moreover, all the stations perform their first AP association through the Strongest Signal First (SSF) mechanism. After getting connected to the network, they access the controller for sending the data collected by their software agents. Then, the SDN controller makes a decision and sends them further instructions to associate with the selected AP(s).

Building Conflict Graph

To model the impact of interference caused by ET/HT, we utilize the introduced approach in [48, 49]. In fact, we build a Multi-Dimensional Conflict Graph (MDCG) based on the links in the constructed topology graph and the number of AP radios/channels. Each vertex of the MDCG represents a resource entity and it is shown as a Link-Radio-Channel (LRC) tuple. Once the MDCG is constructed, we can compute its Maximal Independent Sets (MISs) as the sets of its non-adjacent vertices that can cover the entire topology graph. Thus, the vertices (tuples) of MISs determine the links, channels and interfaces that can transmit simultaneously without interfering with each other. Note that we use the protocol interference model,

a well-known technique, which is utilized in the recent work [48, 50, 51] to model the link interference in WMNs, and according to [52], it can be converted to the physical interference model. Note that the conversion process is out of the scope of this dissertation and we invite the interested readers to study [52]. For wireless transmissions, if any of the following conditions is satisfied, we place a conflict edge between the tuples of the MDCG.

- Simultaneous data transmissions to multiple stations through a single interface of any AP.
- Simultaneous data receptions from multiple APs through single interface of any station.
- Two APs are placed in the interference range of each other when operating in the same channel.
- Simultaneous data receptions and transmissions from/to the transmitter and receiver of two tuples that are placed within the interference range of each other.

Finding Maximal Independent Sets (MISs)

Since finding all the MISs of a given graph is an NP-hard problem, similar to [48] we present a heuristic algorithm to find a subset of MISs with more tuples. The heuristic used in [48] exploits the network topology information to choose the links that might be scheduled with high possibility. In our work, after building the MDCG, we first sort its tuples based on their degrees and then use our algorithm to find a subset of MISs that cover all the tuples of MDCG. The number of selected elements in MISs depends on several parameters such as the topology graph and the number of available channels and wireless interfaces at APs and stations. For instance, if a tuple represents a link on the boundary of the topology, it has a smaller degree in comparison to the tuples for center APs. So, the tuples with smaller degrees are our main interest.

As it is mentioned, we intend to maximize the aggregate network throughput by finding the MISs that contain more tuples. To achieve this goal, we have to increase the number of concurrent transmissions by considering the MISs with more tuples. Therefore, our algorithm starts the search process from the tuples with the

Algorithm 1 Finding Maximal Independent Sets (MISs)

```
1: Initialize set P to the anchor tuples in increasing order.
2: Initialize S_p = φ, ∀p ∈ P and T = P.
3: while T is not empty do
4:     Pick an anchor tuple p ∈ P.
5:     if N_p is empty then
6:         Add {p} into set S_p.
7:         if p ∈ T then
8:             T ← T \ p
9:         end if
10:    else
11:        for every p' ∈ N_p do
12:            for every independent set s ∈ S_p do
13:                if p' can be added into set s then
14:                    Add p' into set s.
15:                    Break the loop.
16:                end if
17:            end for
18:            if p' cannot be added into set s then
19:                Create a new independent set {p, p'}.
20:                Add {p, p'} into S_p.
21:            end if
22:            if p' ∈ T then
23:                T ← T \ p'
24:            end if
25:        end for
26:    end if
27: end while
28: for every found independent set s do
29:    if set s is not Maximal then
30:        Add tuples from the MDCG to make it Maximal.
31:    end if
32: end for
```

smallest degrees (with fewer neighbors) in the MDCG. To do this, we first initialize the algorithm by setting all the ordered tuples represented by set $\mathbf{P}$ as anchor tuples. Each anchor tuple is equipped with a set of Independent Sets (ISs), called $\mathbf{S_p}$, and is initialized to empty. We define another set, called $\mathbf{T}$ as the to-be-covered set that contains the tuples of the MDCG that have not been covered by any IS yet. This set is initialized by $\mathbf{P}$ and when a new tuple in $\mathbf{T}$ is covered by certain ISs, this tuple will be removed from $\mathbf{T}$.

The algorithm starts by selecting a tuple $p \in \mathbf{P}$ that leads to a round of the search process for finding ISs that cover tuple p and the set of its non-neighbor tuples denoted by $\mathbf{N_p}$. Note that the found ISs are stored in $\mathbf{S_p}$. The algorithm runs iteratively along the anchor tuples in increasing order until all the tuples are covered by certain ISs. The search process stops when $\mathbf{T}$ becomes empty. It means all the tuples of the MDCG are covered. Finally, all the non-maximal sets of found ISs are extended to MISs. It is imperative to note that the algorithm tends to form MISs that have more tuples due to two reasons. First, the tuples with smaller degrees have more chance to operate as anchor tuples. Second, in each round of the search process, the anchor tuples are covered in all ISs stored in $\mathbf{S_p}$, but this is not necessary for the other tuples. In a nutshell, the proposed algorithm constitutes two phases. In Phase 1, we search for ISs that cover all the tuples of the MDCG, and we extend the ISs (if it is necessary) in Phase 2. The entire process is summarized in *Algorithm 1* and the time complexity of the algorithm is discussed in Section 2.3.4.

2.3.2 Optimization Problem

After finding the MISs of the created conflict graph, we employ a two-step optimization problem to find an approximation of the optimal association map. We carry out this process through two steps. The first step is solving the relaxed optimization problem, and the second one is rounding the relaxed outputs to find the final solution. Before going through these phases, we present our formulated optimization. The problem of finding an association map for greedy Wi-Fi stations of different service providers over a shared infrastructure to maximize the network throughput with max-min fairness can be formulated in two steps.

Step 1:

$$\text{Maximize} \quad b_{\min} \tag{2.1}$$

$$\text{s.t.,} \quad b_{\min} \leq b_s, \qquad \forall p \in \mathbf{P'}, \forall \mathbf{s} \in \mathbf{S'_p} \tag{2.2}$$

$$\sum_{a\in\mathbf{A}} x_{as} = 1, \qquad \forall p \in \mathbf{P'}, \forall \mathbf{s} \in \mathbf{S'_p} \tag{2.3}$$

$$\sum_{i\in\mathbf{I}} \alpha_i \leq 1, \qquad 0 \leq \alpha_i \leq 1 \tag{2.4}$$

$$b_{l_s^a} = \sum_{i:l_s^a\in i} \alpha_i \, r^c_{il_s^a}, \qquad \forall l_s^a \in \mathbf{L}, \forall \mathbf{c} \in \mathbf{C} \tag{2.5}$$

$$b_s = \sum_{a\in\mathbf{A}} x_{as} \, b_{l_s^a}, \qquad \forall p \in \mathbf{P'}, \forall \mathbf{s} \in \mathbf{S'_p} \tag{2.6}$$

$$\sum_{s\in\mathbf{S'_p}} \sum_{a\in\mathbf{A}} b_s \leq \sum_{p'\in\mathbf{P'}} \sum_{s\in\mathbf{S'_{p'}}} \sum_{a\in\mathbf{A}} \beta_p \, b_s, \forall p \in \mathbf{P'} \tag{2.7}$$

$$x_{as} \in \{0,1\}, b_s \in [0, r^c_{il_s^a}], b_{\min} \geq 0 \tag{2.8}$$

Step 2:

$$\text{Maximize} \qquad \sum_{p\in\mathbf{P'}} \sum_{s\in\mathbf{S'_p}} \sum_{a\in\mathbf{A}} b_s \tag{2.9}$$

$$\text{s.t.,} \qquad \text{constraints} \quad (2.3) - (2.7), \text{and}$$

$$b'_{\min} \leq b_s, \qquad \forall p \in \mathbf{P'}, \forall \mathbf{s} \in \mathbf{S'_p} \tag{2.10}$$

$$x_{as} \in \{0,1\}, b_s \in [0, r^c_{il_s^a}]. \tag{2.11}$$

At the first step, the goal is finding the largest possible value that can be assigned to all traffic flows as their minimum bandwidth. Then, the output of this step ($b'_{\min}$) will be used in Step 2, which finds the maximum aggregate network throughput. In fact, this process attempts to provide max-min fairness by maximizing the minimum assigned bandwidth as well as maximizing the aggregate throughput. Note that the same technique is utilized in [44] to find the optimal outcome for max-min fairness provisioning among wireless stations.

The term in (2.1) shows the objective function that maximizes $b_{\min}$ as the minimum guaranteed bandwidth. The first constraint in (2.2) ensures the allocated

rate to every station s from service provider $\mathbf{S'_p}$ is greater than or equal to $b_{\min}$. In this equation, b_s represents the allocated bandwidth to station s. The second constraint in (2.3) guarantees that every station is associated to only one AP. In this constraint, binary variable x_{as} is used to show either station s is associated to AP a or not.

One of the key features of the presented formulation is to use the calculated MISs of the MDCG to estimate the allocated bandwidth to the flows (stations). Suppose I shows the set of calculated MISs of the MDCG that is derived from *Algorithm 1*. We use α_i to represent the assigned proportion of time to MIS i which is a value in [0, 1]. The sum of these fractions over I should be less than or equal to 1. This condition is shown in the constraint in (2.4). Also, the introduced scheduling scheme, which is based on the derived MISs from the MDCG, has two attributes. First, it enables the link scheduling, channel assignment, and radio assignment at the same time. Indeed, only one MIS can be active at a time and the LRC tuples of the active MIS can operate simultaneously without having interference. In addition, it considers the quality of channels through the scheduling process by involving their link data rates. This fact is shown in the constraint in (2.5) where α_i is used to determine the average allocated bandwidth to station s from AP a under traffic scheduling w.r.t. the impact of interference. In this relation, $r^c_{il^a_s}$ shows the data rate of link l^a_s over channel c in MIS i. It means, after finding the MISs of the constructed MDCG, we can show the relationship between the allocated bandwidth to each link and the physical link data rate as given in the constraint in (2.5). Hence, the links associated with the LRC tuples (vertices) of the MDCG determine the transmission scheduling decision. By considering a set of available channels (either overlapping or non-overlapping) in the formation of MDCG (as a part in the LRC tuples of its vertices) and finding corresponding MISs, it would be possible to schedule the non-overlapping channels to operate in the same scheduling cycle. However, regarding our design, it is guaranteed not to enable two links which function in overlapping channels and are within the interference range of each other at the same cycle. Note that in the presented model, each link (or LRC tuple) can operate on different channels over time and its physical data rate is a function of the selected channel for the link (tuple) based on the achieved SINR values and available modulation and coding schemes [48].

The next constraint in (2.6) determines the allocated bandwidth to station s as the sum of link data rates acquired by station s from its nearby APs. Note that regarding the former condition, b_s is equal to the data rate of the link between station s and its assigned AP a ($b_{l^a_s}$) calculated through (2.5). The constraint in (2.7) ensures

that the total allocated airtime to the stations of service provider p is no more than its predefined share β_p. The left-hand-side (LHS) of this constraint represents the obtained bandwidth by the stations of service provider $\mathbf{S'_p}$, and the right-hand-side (RHS) shows the upper bound of the assigned portion to the same service provider ($\mathbf{S'_p}$). The last constraint of Step 1 ensures the integrality of x_{as} as well as showing the bounds for b_s.

In Step 2, the objective is to maximize the total allocated bandwidth to the stations of all the service providers which is shown in (2.9). In this step, in addition to the constraints (2.3)–(2.7) presented in Step 1, we need to consider one more condition which is shown in (2.10). According to this constraint, the obtained bandwidth by all the stations should be more than or equal to the outcome of Step 1 ($b'_{\min}$).

2.3.3 Approximation of the Optimal Solution

Although by solving the presented optimization problem, we are able to provide max-min fairness among the stations, it should be noted that this problem is NP-hard. In fact, the relation in (2.6) as a quadratic constraint with a binary variable does not allow us to solve the problem for large-scale scenarios in polynomial time. Hence, we need to find an approximation of the optimal solution. To do this, we apply the introduced approach in [44] to solve our problem through relaxation and rounding techniques. As the first step of this process, the binary variable x_{as} is set to 1. Then, the relaxed optimization problem is linear and can be solved in polynomial time. The outcome of the relaxed optimization in Step 1 is called $b^*_{\min}$, which is used as the input of Step 2. By solving the relaxed version of Step 2, we can find the fractional association solution which is calculated as follows.

$$x'_{as} = \frac{b'_{l_s^a}}{b'_s}, \qquad \forall p \in \mathbf{P'}, \forall \mathbf{s} \in \mathbf{S'_p}, \forall \mathbf{a} \in \mathbf{A} \tag{2.12}$$

In (2.12), $b'_{l_s^a}$ and b'_s are the outcomes of Step 2 after relaxation and they represent the upper bound of the bandwidth that can be obtained by each link and station, respectively. Now, to find an approximated solution, we apply the technique introduced in [53] to the relaxed outcomes. One of the most popular approaches to find a 2-approximation of the optimal result for the problems that can be converted to GAP is presented in [53]. In GAP, we need to distribute tasks among the machines such that each task is assigned to only one machine and the total weights of the

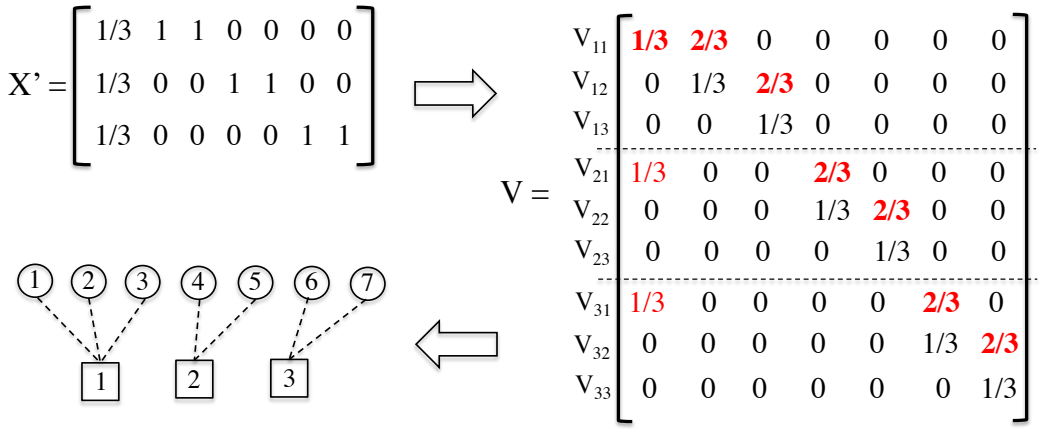


Figure 2.2: Rounding example for a scenario including 7 stations and 3 APs.

assigned tasks to each machine should be no more than its capacity. Regarding the presented objective and constraints, the machines and tasks can be replaced with APs and stations, respectively. The essence of the rounding process is based on the formation of a bipartite graph where two sets of its disjoint vertices are the Wi-Fi stations and resource slots of the APs.

To have a better understanding about this technique, we present an example in Fig. 2.2. Assume that the fractional association solution acquired from the relaxed optimization problem for a scenario including 3 APs and 7 stations is represented by matrix $X^{'}$. By following the instructions provided in [53], we have to map each row of $X^{'}$ to multiple rows in matrix V. The number of corresponding rows (in matrix V) for each row of $X^{'}$ is the ceiling value for the sum of shown numbers in each row. This value for each row of matrix $X^{'}$ is 3. Thus, we have 3 rows for every AP in matrix V (in total 9 rows). Now, by picking the largest number of each column (that represents a station) of V, we can find the assigned AP to every station after the rounding process. Picking the largest number for all the stations (except the first one) is a straightforward job. For the first station (first column in V), since the same value is shown for all the APs, we can pick each of them as the assigned AP. So, here we picked the first AP and the final association map is illustrated in the last part of Fig. 2.2. we refer the readers to [53] as well as the example presented in [40, 44] to find further details on the rounding process. The summary of the used symbols in the formulation is shown in Table I.

It is important to note that the presented formulation can be applied to both

Table 2.1: Utilized parameters in the problem formulation

Symbol	Description
$\mathbf{P'}$	The set of service providers
$\mathbf{S'_p}$	The set of Stations of service provider $p \in \mathbf{P'}$
$\mathbf{A}$	The set of access points
$\mathbf{C}$	The set of access Channels
$\mathbf{L}$	The set of wireless Links
$\mathbf{I}$	The set of Maximal Independent Sets
x_{as}	1, if station s is associated to AP a, otherwise 0
x'_{as}	Fractional AP association variable
$r^c_{il^a_s}$	Physical data rate of link l on channel c in MIS i
$b_{l^a_s}$	Allocated bandwidth to link l between station s and AP a
b_s	Allocated bandwidth to station s
$b'_{l^a_s}$	Allocated bandwidth to link l after relaxation of Step 2
b'_s	Allocated bandwidth to station s after relaxation of Step 2
$b_{\min}$	Minimum guaranteed bandwidth to all the stations
$b'_{\min}$	Minimum guaranteed bandwidth found in Step 1
$b^*_{\min}$	Minimum guaranteed bandwidth found in *Relaxed* Step 1
α_i	Allocated share to independent set $i \in I$
β_p	Predefined share of service provide p

greedy downlink and uplink stations. This change can be represented by the direction of the edges in the network graph. Also, note that in this chapter, our main goal is presenting a fine-grained and flow-level traffic engineering solution using dynamic association for Wi-Fi stations. However, the presented formulation can be extended to a flow-based association problem. For instance, we can model each station with multiple flows by multiple dummy stations with a single flow such that each one is associated to one AP.

The required steps of the introduced solution are shown in *Algorithm 2*. Note that the proposed scheme has to be performed periodically. Also, in *Algorithm 2*, APC, STC, and PCS represent AP/Station coordinates and the Physical Carrier Sensing range, respectively. To present a better picture of our solution, a simple example is illustrated in Fig. 2.3. As the first step of the demonstrated process in this figure, the topology graph including four APs and five stations is constructed based on the given CR. Then, using the coordinates of the nodes, chosen channel model and given PCS range, the conflict graph is created. Note that for the sake of simplicity, we assume a single channel scenario in Fig. 2.3. In the next step, the

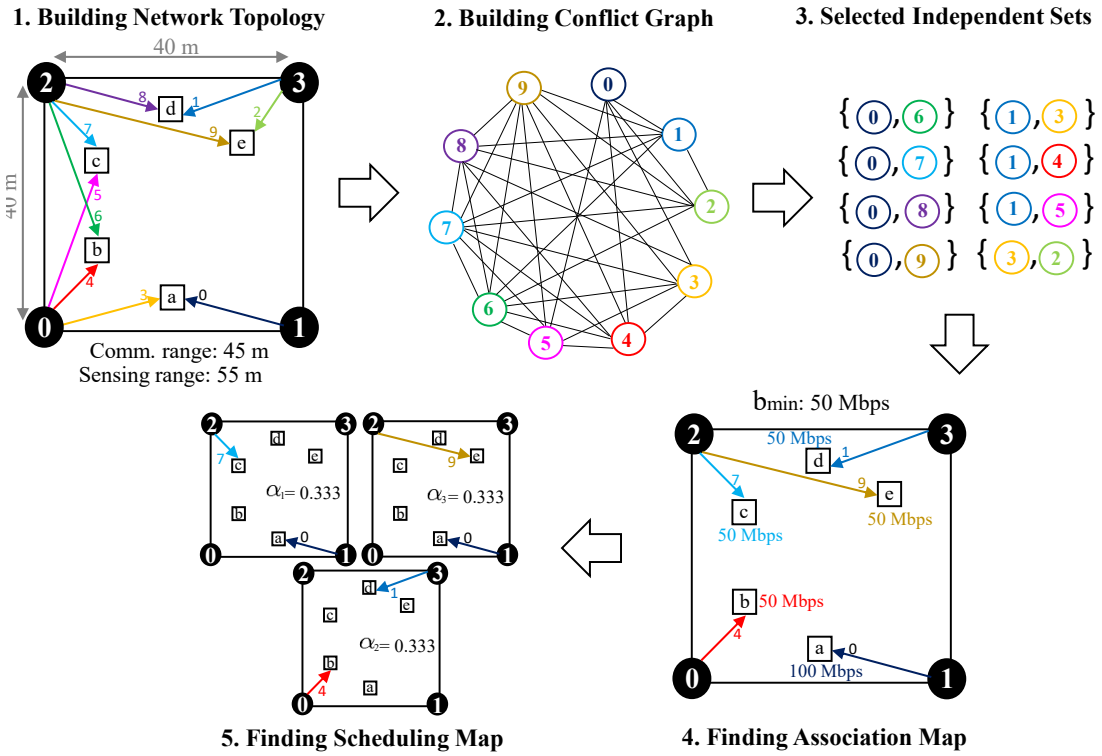


Figure 2.3: The big picture of the association solution.

Algorithm 2 Proposed solution for AP association problem

1: **for** every time interval t **do**
2: Update Network Statistics.
3: NG ← BuildNetworkTopology(APC, STC, CR).
4: MDCG ← BuildConflictGraph(NG, PCS).
5: MIS ← FindIndependentSets(MDCG).
6: b'_s ← Solve the Relaxed Optimization using MIS.
7: Utilize BG-Rounding to find a near-optimal result.
8: Apply the obtained results to SD-WLANs.
9: **end for**

MISs that cover the entire conflict graph are found. By using these MISs, we solve the optimization problem (through relaxation and rounding) and find the association and scheduling map of the given topology. In this example, it is assumed that the maximum downlink capacity of each AP is 150 Mbps and at most two APs may have concurrent transmissions based on the defined PCS range. Thus, as displayed in Step 4 of Fig. 2.3, the aggregate throughput of the stations is 300 Mbps. The scheduling map of the active links and the corresponding α_i are shown in Step 5 of Fig. 2.3.

2.3.4 Time Complexity Analysis

To simplify the time complexity analysis of the proposed solution, a basic operation as an interaction event between two tuples is defined. Note that the upper bound for the number of tuples in an MDCG is LR^2C such that L, R, C represent the number of links in the network graph, number of radio interfaces at each AP/station and number of available channels for APs, respectively. If we assume $N = LR^2C$, then the formation of MDCG is $O(N)$. By considering the presented steps in *Algorithm 1*, including the neighbor discovery and finding the independent sets, it can be seen that the overall complexity of the algorithm is polynomial ($O(N^4)$). Then, the outcome of the algorithm is used as the input of the presented optimization problem in Section 2.3.2. Finally, by applying the introduced relaxation and rounding technique (using bipartite graph [53]), we can find a 2-approximation of the optimal result in polynomial time. To have a better picture about the running time of our solution, the average running time for some of the scenarios is measured in Section 2.5.1.

2.4 Performance Evaluation

Since our main interest is the performance assessment of the service provisioning for greedy traffic flows and we do not need to explore the dynamics of the protocol stack, we have not used a packet-level simulator. Hence, we carry out the experiments using our own simulator with extensive numerical results. In fact, by focusing on greedy Wi-Fi stations (for bandwidth-greedy applications) and regarding the features of the Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) protocol, we ensure that the bandwidth allocated to the stations is mainly a function of their data rates and priorities. It means that for the greedy flows, the air time will be distributed among the stations associated to the same AP proportional to their physical data rate (between the stations and the AP) and their given priorities. This approach to evaluate the functionality and the performance of our work is in par with the experiments carried out in [38, 39, 41, 44, 54] that did not use packet-level simulators as well. Moreover, to promote the credibility of our solution, we developed a testbed setup as a proof of concept to validate its functionality. Further explanations about the simulator are presented in Section 2.4.2. In this section, we delineate more details on the key assumptions and experiment setup.

2.4.1 Key Assumptions

For all the scenarios, it is assumed that one flow is assigned to every station and all the flows have the same priority. Our proposed scheme is evaluated under a stable network condition, i.e., no new station joins or leaves our network which is consistent with our scheme when the scale of join/leave is larger than the period of service provisioning. We have utilized multi-rate APs that serve stations with diverse data rates. In addition, it is supposed that all APs use omni-directional Single-Input Single-Output (SISO) antennas. Moreover, we use ISM frequency bands, and the impact of HTs has been taken into account during the creation of the MDCG. Note that in most of the Wi-Fi commodity equipment, the impact of HT can be avoided by enabling the restart-mode in Wi-Fi chipsets [55, 56]. Also, similar to the presented work in [48, 49, 52], the protocol interference model is used to capture the impact of internal (inter-flow) interference. We assume that the network flows are greedy through the conducted scenarios. It means, either the stations are running bandwidth-greedy applications or the transmission queues of the APs are always full. This assumption is in par with the rapid growth of bandwidth-greedy applications such as HD streaming and online gaming [57]. Hence, we focus on the worst-case scenario in which all the Wi-Fi stations are greedy with homogeneous demands and the utilized APs have the same capacity. In addition, since almost two thirds of the top activities on smart-phones are primarily nomadic rather than mobile [58], we use nomadic scenarios with stationary nodes. Note that the presented scheme can be applied directly to the scenarios where a large number of users placed in the large areas such as stadiums and conference halls.

2.4.2 Experiment Setup

We have conducted a set of experiments over a 5×5 grid topology deployed within a 140×140 m^2 area. This topology contains 25 APs and various numbers of stations are distributed in uniform and hotspot fashions. The stations are equipped with Wi-Fi interfaces that support 802.11n standard and we applied the wireless channel used in [39] to our scenarios which is an ITU-R channel model. For running the experiments, we have developed our own network simulator in Python/Gurobi [59] (version 6.5.2) and a revised version of the code developed in [48] is used to find the MISs of the generated MDCG. We carry out the performance assessment for different numbers of stations including {50, 100, 200} in uniform and hotspot distributions.

In hotspot scenarios, we have considered a 100×100 m^2 square-shaped subarea at the center of the grid topology.

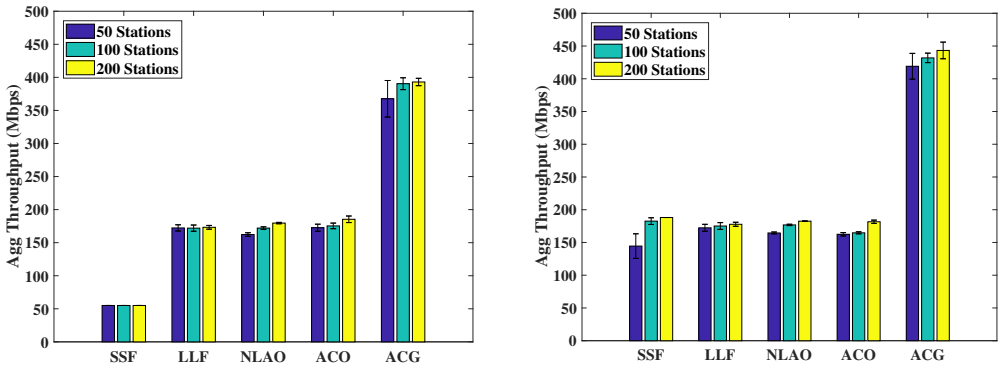
The presented results for different station densities are the average of the outcomes obtained from five scenarios for a random distribution of stations. It is assumed that APs are using single-stream 802.11n that can reach up to 150 Mbps data rate. Moreover, we have considered the impact of varying sensing ranges of {139.9, 180}m on the performance of our scheme. Regarding the network topology and the key assumptions, we choose these numbers (ranges) such that they can reflect different aspects of the functionality and performance of our solution. Further details about these ranges are provided in the next section. Per-station throughput, aggregate throughput and fairness index [60] are used as the key performance metrics. In addition, to get a better insight into our work, we compare the performance of our solution which is called **A**ssociation using **C**onflict **G**raph (**ACG**) with some of the known schemes including Strongest Signal First (SSF), Least Loaded First (LLF) [61], association using Non-Linear Approximation Optimization (NLAO) [40] and association using Ant Colony Optimization (ACO) [39]. It should be noted that for having a fair comparison, all the selected schemes (except SSF) are centralized association solutions that have a holistic view of the network topology. These schemes utilize non-linear optimization (NLAO), heuristic (LLF) and meta-heuristic (ACO) approaches to find the association map. In addition to the centralized schemes, we compared the performance of ACG with SSF which is the most dominant association mechanism and its outcomes provide a baseline for us.

2.5 Results and Discussion

In order to investigate the impact of different parameters on the performance of the introduced solution, we carry out the following experiments.

2.5.1 Impact of Sensing Range on Downlink Performance

There are several studies such as [62, 63] that showed it is possible to improve the network performance significantly through the proper adjustment of Clear Channel Assessment (CCA) threshold (PCS range). In addition, many wireless products in the market allow us to set the transmit power, receiver sensitivity thresholds, and CCA threshold to enhance the user experience. These parameters can be utilized to



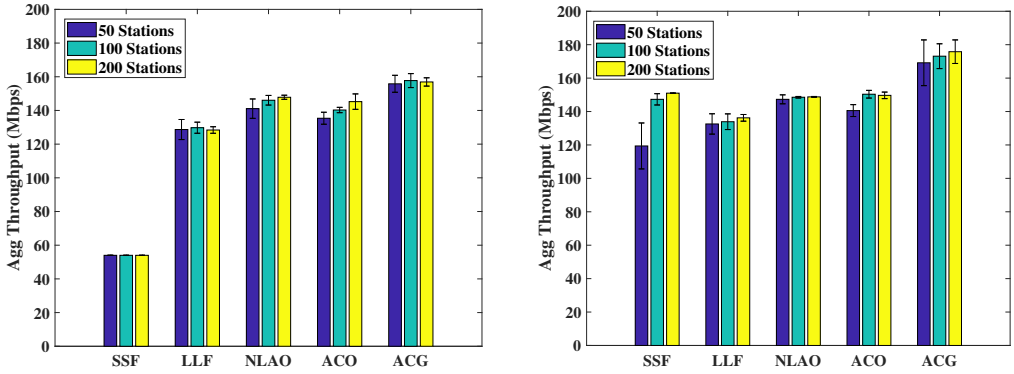
(a) Hotspot (b) Uniform

Figure 2.4: Measured downlink throughput, PCS range: 139.9 m.

shrink the effective cell size of an AP by adjusting the received signal strength before an AP and station consider the channel is accessible for the data transmission. Thus, by the proper adjustment of these parameters, we can control the inter-AP distances and use it as a feature to control the impact of interference, consequently [64].

In this section, we have conducted a set of experiments to see the impact of PCS range on the downlink performance of ACG and the selected schemes for different number and distribution of stations within the grid topology. Fig. 2.4a and Fig. 2.4b illustrate the quantified aggregate throughput of the selected association schemes for different number and distribution of stations when CR and PCS ranges are set to 98 m and 139.9 m, respectively. Note that by the adjustment of the introduced parameters, we are able to set the CR and PCS ranges. So, regarding the specifications of the network topology and to show a better illustration on the features of our introduced solution, we chose the mentioned numbers for CR and PCS ranges.

Note that to have a fair comparison among the chosen schemes, we use co-channel APs to perform this experiment. There is a difference between the displayed results in Fig. 2.4a and Fig. 2.4b that comes from a diverse distribution of the stations. Placing the stations inside the hotspot area has a drastic impact on the performance of SSF and leads to a small reduction of the ACG aggregate throughput as well. This is the main ramification of concentrating the stations in a restricted area which is far from the boundary APs. In fact, it impedes the stations from the establishment of high data rate links and consequently degrades the aggregate throughput. Although the other selected schemes have been less affected, they could not exploit the available network capacity efficiently.

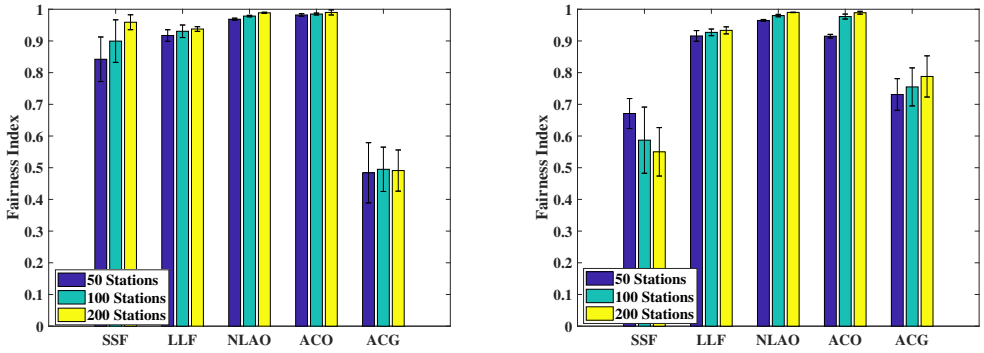


(a) Hotspot (b) Uniform

Figure 2.5: Measured downlink throughput, PCS range: 180 m.

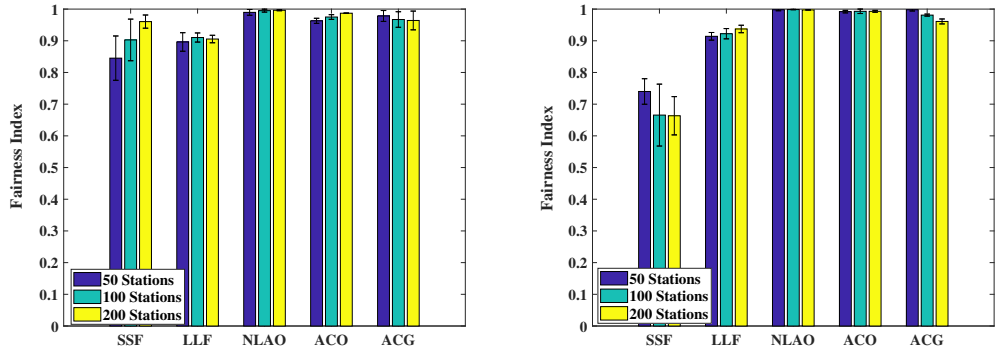
In the next experiment, we increased the PCS range to 180 m which has a severe impact on the performance of all the schemes. Only for ACG, it has reduced the aggregate throughput to less than half of the prior case. By comparing the presented results in Fig. 2.4 with Fig. 2.5, it can be seen that the adjustment of CCA threshold (to have a shorter PCS range) allows us to achieve a notable performance improvement using ACG. This improvement is almost two times of the obtained results from the other schemes in terms of the aggregate throughput. This outcome is demonstrated in Fig. 2.5 and the main reason behind it is the aggravation of the induced interference from the co-channel APs due to the expansion of the PCS range. It should be noted that enlarging the PCS range decreases the number of possible concurrent transmissions and it leads to the significant decline of the aggregate throughput.

In addition to the aggregate throughput, we calculate the fairness index w.r.t. different densities for the conducted experiments and the outcomes for PCS ranges of 139.9 m and 180 m are shown in Fig. 2.6 and Fig. 2.7, respectively. Regarding the presented results in Fig. 2.6a and 2.6b, although ACG outperforms the selected association schemes in terms of aggregate throughput, it does not provide a perfect fairness among all the stations. This is due to the fact that it is not feasible to achieve the maximum network utilization and fairness at the same time [65]. However, it is clear that still ACG attains an acceptable level of fairness (more than 0.75) especially in dense uniform scenarios. Furthermore, we can see that the impact of concentrating stations in the hotspot area lessens the fairness index of ACG which is consistent with the results shown in Fig. 2.4. On the other hand, by increasing the PCS range to 180 m, ACG provides the maximum fairness in the cost of reducing the aggregate



(a) Hotspot (b) Uniform

Figure 2.6: Calculated Fairness Index, PCS range: 139.9 m.



(a) Hotspot (b) Uniform

Figure 2.7: Calculated Fairness Index, PCS range: 180 m.

throughput. The results presented in Fig. 2.7a and 2.7b substantiate this fact.

We have measured the average running time of ACG for the results presented in Fig. 2.4 and Fig. 2.5. The simulations were carried out on a Dell server (PowerEdge R820, E5-4640 product family) with Intel Xeon processor (2.4 GHz) and 64 GB RAM. By increasing the number of stations from 50 to 100 in the described scenario, the average running time is increased from 2.933±0.49 sec to 11.69±1.76 sec. For 200 stations, this value is increased to more than one minute (i.e., 61.6±5.37 sec). Note that since there is not much change in the network topology and channel conditions during the experiments (due to being in stationary position), the period of service provisioning even can be extended to a minute. Also, this result substantiates that in spite of the exponential growth of the search space by increasing the number of APs and stations (7.888×10^{69} combinations for 25 APs and 50 stations), our scheme

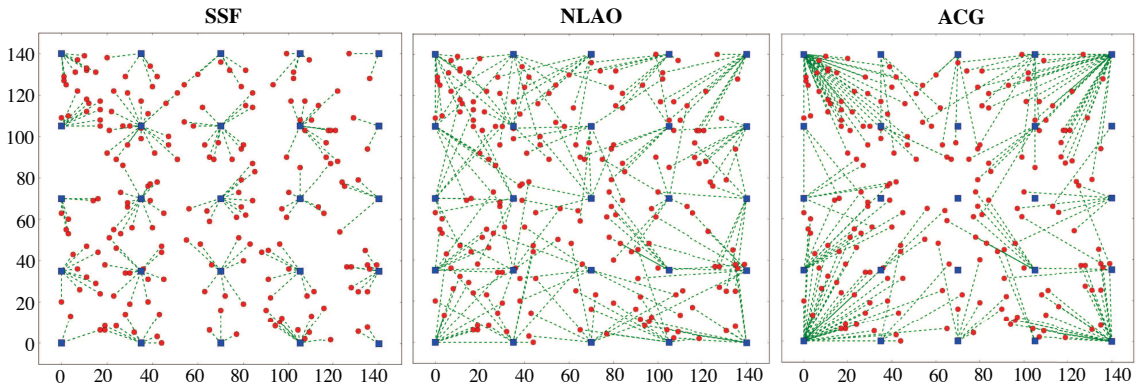
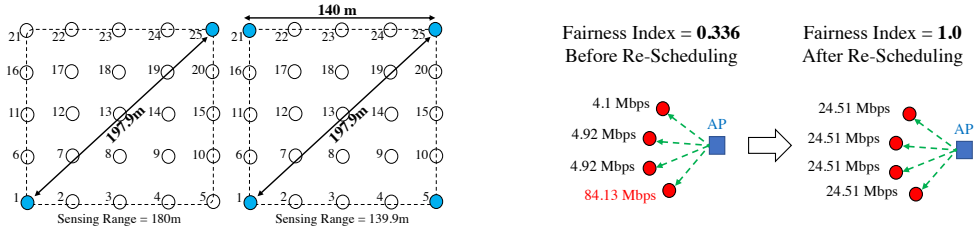


Figure 2.8: Association maps of 3 schemes for 200 stations.

as a part of an SDN-based centralized solution can converge with a reasonable speed for the defined scenarios (with stationary greedy stations). Moreover, it should be noted that for the large topologies, this running time can be reduced significantly by grouping nearby or similar stations and enabling the multi-threading feature of computation-rich resources that run our solution. In our implementation, due to the limitation of Gurobi academic license, each process can utilize up to 12 CPU cores.

Note that for having a fair comparison between ACG and the prior work (that ignored the impact of HTs), we do not consider the impact of HTs in the shown results. To have a better understanding about the functionality of ACG, we have illustrated the association map of ACG versus SSF and NLAO schemes for 200 uniformly distributed stations in Fig. 2.8. In this figure, blue squares and red circles represent the APs and Wi-Fi stations, respectively. The dashed green lines show each station is associated to which AP for the selected scheme. In this scenario, the APs are co-channel and the PCS range is set to 139.9 m. As it can be seen, in contrast to SSF that clusters the stations around the APs, NLAO distributes the stations among different APs to achieve a better fairness. On the other hand, ACG attempts to achieve a good trade-off between maximizing the fairness index and aggregate throughput. To do this, ACG pushes the stations to the APs placed on the boundary of the grid network and at the same time it avoids associating the stations to the APs in the center region of the topology. Hence, we can see five APs without any associated stations in the ACG association map, shown in Fig. 2.8. This observation shows the capability of our introduced scheme to decrease the number of active co-channel APs in dense environments, which leads to saving power and a substantial reduction of deployment cost as well.



(a) Sensing ranges. (b) Re-scheduling at APs.

Figure 2.9: The impact of sensing range on the maximum number of concurrent transmissions in the 5x5 grid topology.

Furthermore, it is clear that among the boundary APs, ACG favors the ones that are located at the corners of the grid topology. In fact, in the given grid topology, these APs are the best candidates to increase the number of concurrent transmissions by avoiding the impact of the interference caused by the ETs using the introduced scheduling mechanism. Fig. 2.9a illustrates an instance about the impact of the PCS range on the maximum number of concurrent transmissions. According to the topology map shown in this figure that represents indexed APs, when the PCS range is 180 m, only the APs placed at the endpoints of the diagonal of the grid topology are able to have simultaneous transmissions (without collision). It means that the maximum size of the ISs found by *Algorithm 1* is 2. This number can be increased to 4, by reducing the PCS range to 139.9 m, which leads to the significant growth of the aggregate throughput as shown in Fig. 2.4. The candidate APs are shown with blue color (at the corners) in Fig. 2.9a. Note that since ACG attempts to maximize the aggregate throughput by increasing the number of simultaneous transmissions through finding the largest ISs (as the inputs of the optimization problem), it may cause an unbalanced bandwidth distribution among the stations that are associated to the same AP. This result stems from the fact that we only consider a subset of the possible ISs to find an input of the optimization problem in polynomial time. However, due to the centralized nature of our scheme, our scheduling module (at the SDN controller) is able to reschedule the links for a fair distribution of channel capacity among the stations. This process is displayed through an example in Fig. 2.9b that has led to the remarkable improvement of the fairness index (from 0.336 to 1).

Finally, as the last graph of this subsection, we illustrate the sorted bandwidth (throughput) obtained by 100 uniformly distributed stations in Fig. 2.10. The most important impression of the presented result is the superiority of ACG in terms of providing a higher bandwidth for all the (downlink greedy) stations. This can be seen

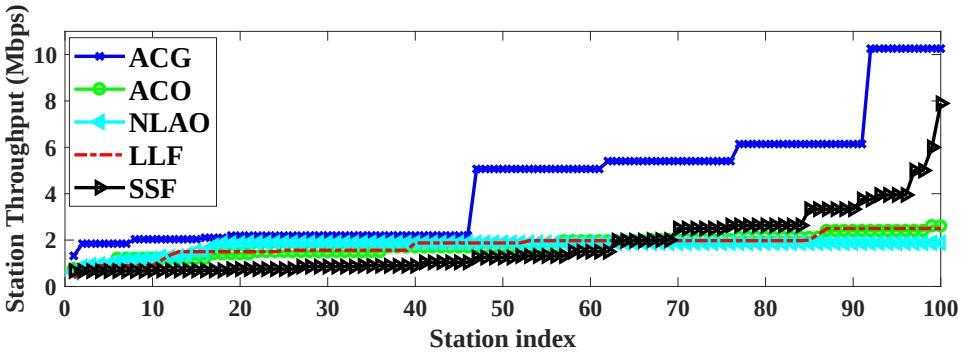


Figure 2.10: Sorted per-station throughput for different schemes.

by checking the minimum assigned bandwidth to the first station index in the shown graph. Although some of the other schemes such as ACO and NLAO have more balanced outcomes which leads to have a better fairness, they could not utilize the network capacity efficiently. Overall, ACG has achieved a better balance of network capacity and fairness index in the carried out experiments.

2.5.2 Impact of HTs on the Performance of ACG

As it is pointed out in Section 2.1, in dense topologies, HTs may cause a significant performance degradation and it is inevitable to consider their influence on the performance of our proposed scheme. Therefore, we conduct an experiment to measure the (downlink) performance of ACG in the presence and absence of HTs in a dense scenario including 200 stations. The stations uniformly deployed inside the grid topology with co-channel APs and the PCS range is set to 139.9 m. The drastic impact of HTs on the aggregate throughput and fairness index (43% and 62% drop, respectively) is shown in Fig. 2.11. Moreover, regarding the explained association strategy of ACG in Section 2.5.1 and the shown results in Fig. 2.12, we see that more than 80% of the allocated bandwidth to the stations is supplied by the APs located at the corners of the grid topology (including APs 1, 5, 21, and 25).

This situation happens regarding the impact of HTs in the arranged topology. In addition, it should be noted that in both cases, there are APs with light load (or even none) to alleviate the adverse effect of APs that act as ETs for their neighbors. Indeed, it is the key consequence of using our scheduling algorithm that attempts to maximize the number of active links and thwarting inter-AP interference at the same time. Note that reducing the PCS range may result in the exacerbation of the results

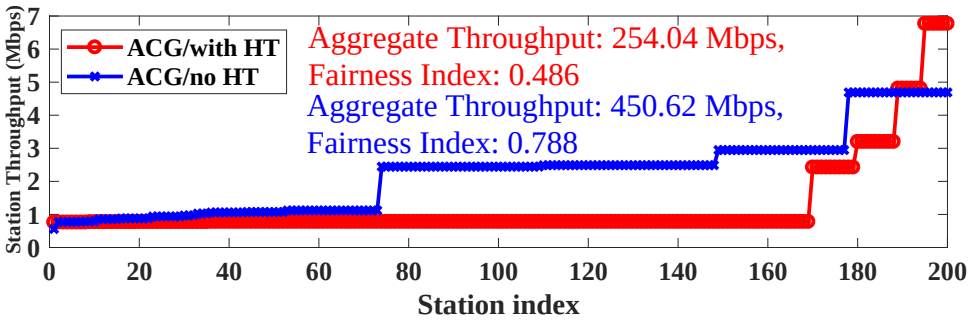


Figure 2.11: Sorted per-station throughput with or without HT scenarios.

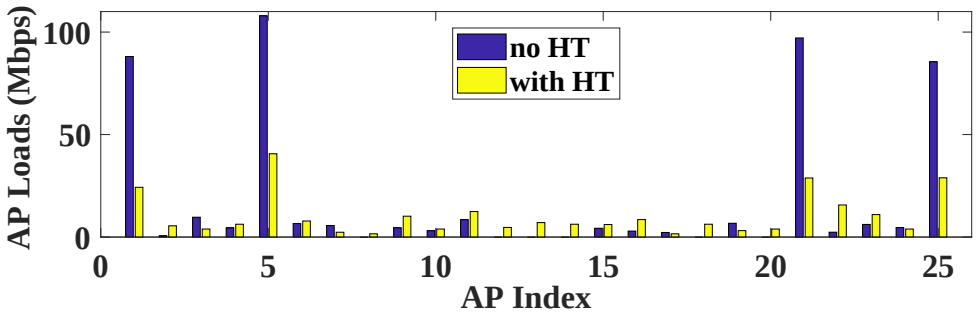


Figure 2.12: Load of APs in the 5×5 grid topology.

for the case where the impact of HTs has been considered. In addition to the shown graphs, Fig. 2.13 presents the calculated CDF of the average allocated bandwidth to the stations for three different schemes. According to the illustrated graph, by using ACG, the probability that a station receives a bandwidth more than 1 Mbps is around 80%. However, the same probability for SSF and NLOA is around 20% and 0%, respectively. In fact, it can be observed that ACG finds a better balance between fairness provisioning and maximizing the network throughput.

2.5.3 ACG Performance for Uplink Flows

Fig. 2.14 illustrates the station/aggregate throughput and the calculated fairness index for 200 greedy uplink stations with/without considering the effect of HTs in uniform/hotspot scenarios. If we compare these (uplink) results with the outcome shown in Fig. 2.11 for downlink scenarios (displayed with blue and red colors in both graphs), we only see a negligible difference between the quantified metrics for uniform scenarios. However, by checking the results presented in [39] (Fig. 7 and Fig. 9), we

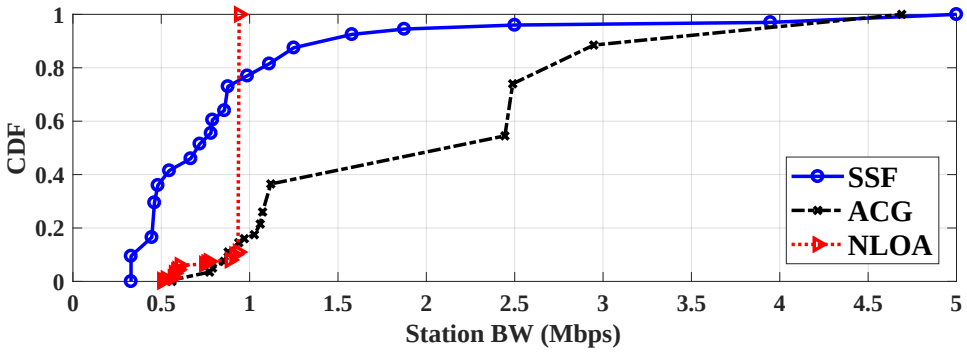


Figure 2.13: Calculated CDF for the bandwidth of 200 stations.

can see a notable gap (up to 70% reduction of the aggregate throughput) between the outcomes of downlink and uplink traffic for different association schemes. The root cause of such an upshot is the existence of severe contention among the greedy uplink traffic flows (generated by the stations) to access the Wi-Fi channel. We can see the same trend for the other association schemes (SSF, LLF, NLAO and ACO) discussed in [39]. It is important to note that there is no significant difference between the results of these schemes for the greedy uplink scenario and none of them can achieve a high performance (throughput) for uplink traffic flows. On the other hand, ACG is able to maintain the network performance for both uplink and downlink scenarios. This interesting result is the consequence of building sets of non-interfering links through the ACG scheduler that can transmit concurrently. However, the clock synchronization of the Wi-Fi transmitters to apply the ACG scheduling to the uplink traffic generated by the stations is a challenging task. For the downlink traffic, the clock synchronization of the APs can be done through the wired backbone. On the contrary, using the cellular network to schedule the uplink traffic can be considered as an option. Note that the unpleasant effect of HTs and hotspot distribution of the stations on the selected metrics are demonstrated in Fig. 2.14 as well.

2.5.4 Impact of Using Non-overlapping Channels

So far, we analyzed the performance of our presented scheme with single channel APs to not only have a clear understanding about its functionality, but also present a fair comparison with other work. In Fig. 2.3, we have shown how multiple links can be scheduled to have simultaneous transmissions with co-channel APs. Now, if we assume that each AP is able to operate in multiple non-overlapping channels, it would

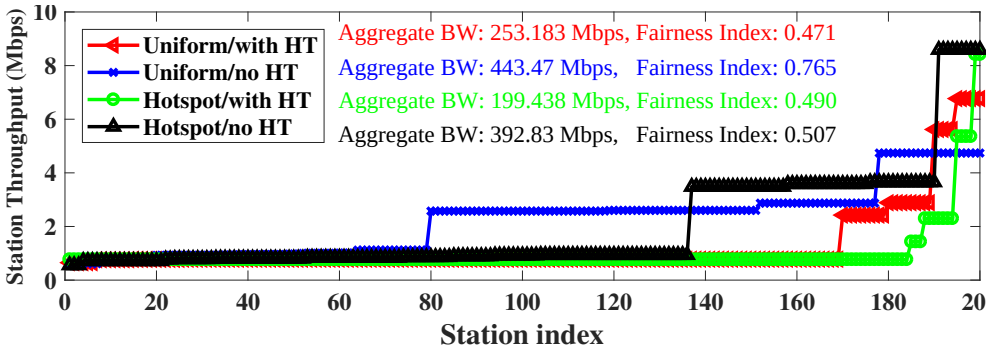


Figure 2.14: Station throughput for uplink scenarios (ACG).

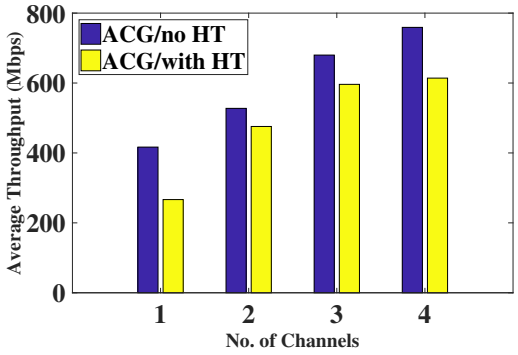


Figure 2.15: Impact of using multiple channels on aggregate throughput.

be possible to increase the size of the selected ISs that leads to the enhancement of the network throughput. To have a better picture, we estimate the aggregate throughput of a network topology including 50 uniformly distributed stations for different numbers of AP channels and the results are shown in Fig. 2.15. As it is displayed, increasing the number of non-overlapping channels improves the network throughput significantly for both scenarios with/without HTs.

Furthermore, according to Section 2.3.4, increasing the number of links, channels, and radio interfaces has a direct impact on the number of LRC tuples that is shown in Fig. 2.16. In fact, this figure represents the existing correlation between the number of calculated LRC tuples/ISs and the number of non-overlapping channels. Moreover, we can see negligible variations on the number of estimated ISs for different numbers of channels. This outcome is derived from the proposed algorithm (*Algorithm 1*) to find a subset of the ISs as the input of the optimization problem shown in Section 2.3.2 that maximizes the objective function. Note that the algorithm maintains its polynomial time complexity by selecting this set of ISs. In the next section, we delineate our

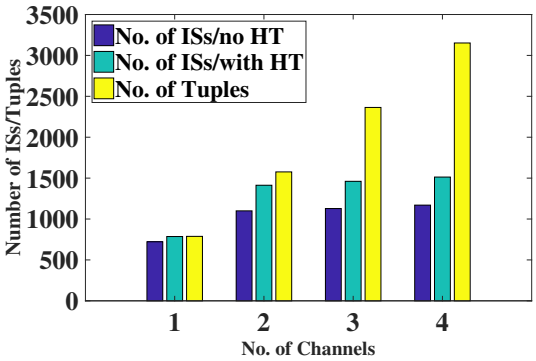


Figure 2.16: Impact of channel numbers on the number of found ISs/tuples.

testbed setup and the carried out experiments for its performance assessment.

2.6 Testbed Implementation

In this section, we delineate our testbed setup and the experiments carried out to assess the functionality of our work.

2.6.1 Testbed Setup

Fig. 2.1 shows a detailed picture on the utilized components to build the major elements of the proposed framework. Our prototype is mainly developed in Python, which makes it a cross-platform solution to run on different machines and expedites the prototyping process as well. Thus, we choose `Ryu` [66] as a well-known Python-based SDN controller to be in charge of the resource allocation process through the conducted experiments. In fact, our developed solution is a north-bound application that utilizes the collected information as the inputs of the `Network Optimizer Module (NOM)` shown in Fig. 2.1. After finding the association map, the controller signals the installed agents on the stations to associate with the determined APs. Moreover, it announces to the APs their new operating channels as well as the allocated bandwidth to the stations associated to them. At the APs, we used a modified version of `Open vSwitch (OVS)` [67] and `Hostapd` [68] for communicating with the controller and applying the determined channels to the radio interface, respectively. Note that OpenFlow [69] as the engine of the SDN controller is in charge of traffic forwarding at APs through the access radio interface (to serve Wi-Fi stations) and the Ethernet port (to access the controller and the Internet).

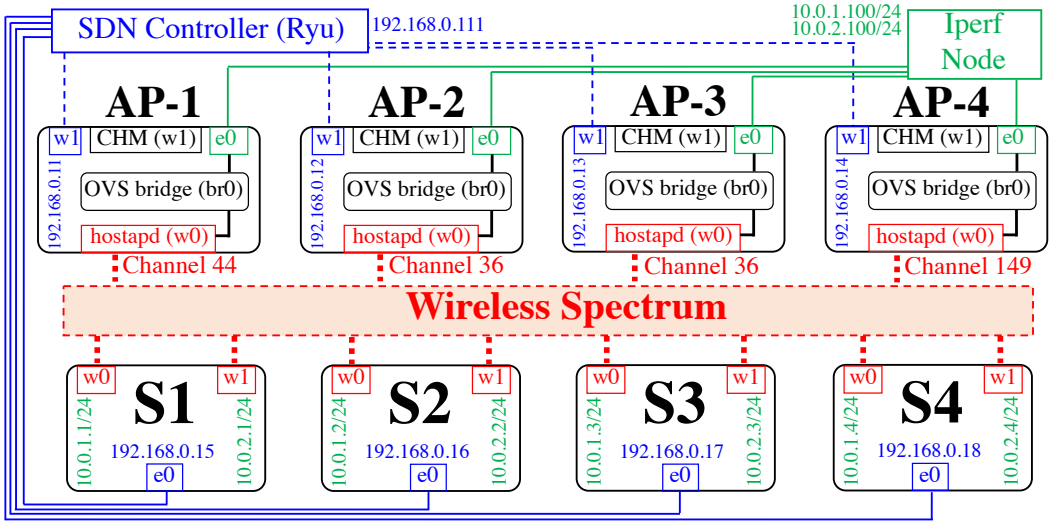


Figure 2.17: Testbed setup for the conducted experiments.

In our testbed setup, we use eight Alix3d2 [70] boards where four of them are used as APs and the rest are utilized as Wi-Fi stations. Fig. 2.17 shows the overall picture of our testbed setup. As illustrated, all the APs are connected to the SDN controller through an out-of-band WLAN (shown by dotted blue lines). The Ethernet interfaces of the APs are connected to the `Iperf` node through a wired network (shown by solid green lines). At each AP, the Ethernet (`e0`) and wireless (`w0`) interfaces are bridged together using an `OVS` interface (`br0`). We use `Hostapd` on interface `w0` to provide the AP association service. In Fig. 2.17, it can be seen that AP-1 and AP-4 operate on non-overlapping channels (44 and 149), while AP-2 and AP-3 are co-channel (36). In addition to provide an out-of-band access to the SDN controller, the second wireless interface (`w1`) of each AP is used in `Channel Monitoring Module (CHM)` as well. Four Alix3d2 boards (each one is equipped with two wireless radios) are used to represent eight Wi-Fi stations in total. As illustrated, we use different ranges of IP addresses for the wireless interfaces of the stations. Thus, the interfaces of every board (station) can associate to different APs and perform download/upload operations independently from each other. Moreover, to access the Wi-Fi stations for running the `Iperf` agents (to generate uplink traffic), an out-of-band wired network (shown by solid blue lines) is used.

As it is shown in Section 2.3.4, the number of available channels plays a crucial role on the complexity of the proposed solution. If each AP determines the number of non-congested channels as the input of the optimization problem, it helps to reduce

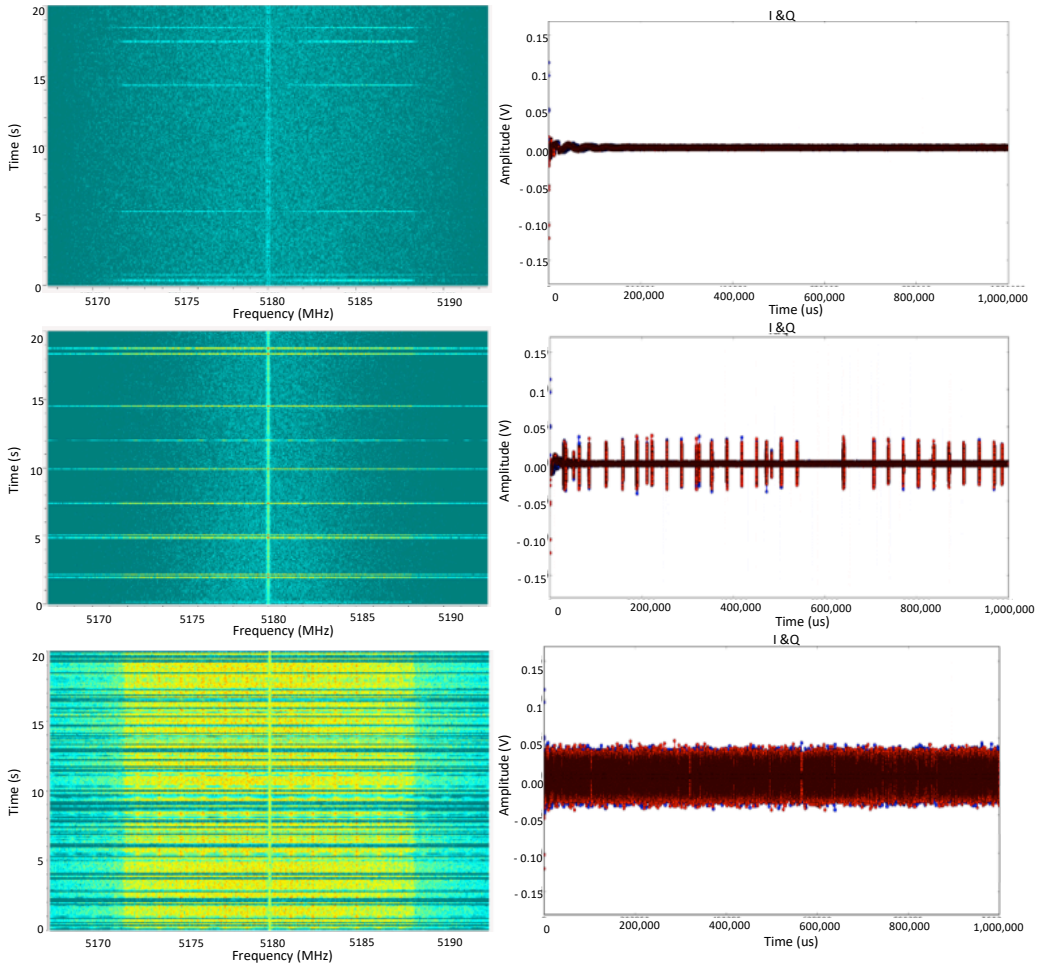


Figure 2.18: Estimation of the channel occupancy using USRP.

the total processing time. Therefore, as mentioned earlier, the wireless interface `w1` of each AP is used to collect channel load information and the other one (`w0`) is utilized for association. To collect channel information, we develop a packet sniffing module using `tshark` [71] at the APs to get an estimation of the current traffic load on Wi-Fi channels. To verify the functionality of our `CHM`, we use a `USRP2` [72] device to quantify the channel load using the number of captured I/Q samples. In signal processing, I/Q samples illustrate the changes in amplitude and phase of a sine wave. If amplitude and phase changes occur in a predetermined fashion, we can use these changes to encode information upon a sine wave that is known as modulation. By monitoring the amplitude of I/Q samples for a certain time period over different channels, it is feasible to obtain a good estimation of the channel activities. To do this, we need to count the number of I/Q samples that are higher than the noise threshold.

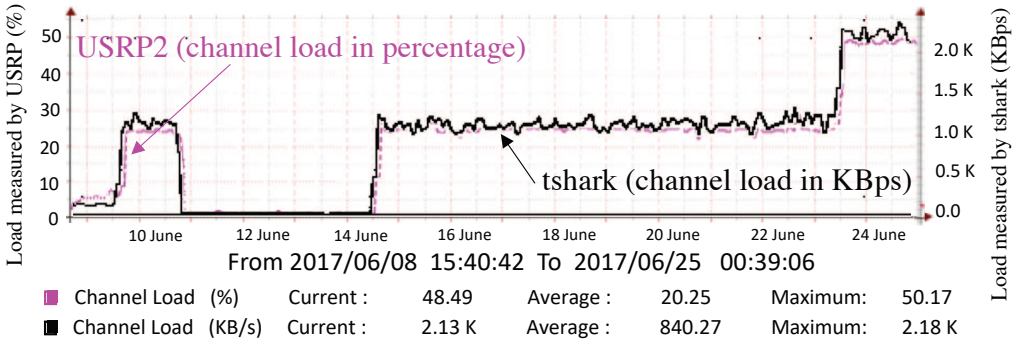


Figure 2.19: Comparison of the obtained results for channel load.

Fig. 2.18 illustrates the captured samples by a `USRP2` device for a channel under three different (low, medium and high) load conditions. Through capturing the I/Q samples of different Wi-Fi frequency bands over a long period, we can predict the best channels for the APs to maximize the network performance. Note that the left-side graphs display the time of channel occupancy versus different frequencies. On the other hand, the right-hand side ones illustrate the amplitude of the sampled signals over time. Also, Fig. 2.19 displays the obtained measurement results using `CHM` (based on `tshark`) and `USRP2` for a Wi-Fi channel during a period of two weeks. We have used `SNMP/Cacti` [73] to store and plot the channel load information calculated by the both schemes and according to the illustrated graphs, the outcomes presented for both approaches are consistent. Note that in the graphs shown in Fig. 2.19, the measured loads by `USRP2` and `CHM` are displayed by pink and black colors, respectively.

Moreover, similar to the approach introduced in [39], we use software agents developed in Python at Wi-Fi stations to send the Received Signal Strength Indicator (RSSI) information from different nearby APs to the SDN controller. These data are aggregated at the controller for the estimation of SINR and consequently calculating the data rates of wireless links that can be established between the stations and their nearby APs. Note that these link data rates are used in the constraint (2.5) of Section 2.3.2. In addition to the data collection, the agents can extract the application information and utilize it for QoS control. Thus, we can use them to control the downlink and uplink rate of Wi-Fi stations in case of having greedy application flows that may starve non-greedy flows.

To achieve a fine-grained airtime allocation, the best approach is using resource allocation at MAC layer to define small time slots for the alleviation of channel

contention among the Wi-Fi stations. Recently, scholars have introduced some mechanisms similar to `hMAC` [74] that do not need to change any part of the standard 802.11 MAC protocol and usually provide an API to define the duration of the assigned time slots to each Wi-Fi station. However, since these solutions still are under active development, there are serious concerns in terms of their stability and performance. Thus, to control the greedy downlink flows, we have used the flow rate limiting feature of `OVS` at the APs which is a resilient choice. This feature not only allows us to have a fine-grained control (using IP/MAC address, port number and protocol ID) over the flows but also assists in applying the defined policies by the controller to the stations. For the uplink traffic, most of the existing techniques cannot cap greedy uplink UDP flows that monopolize the channel capacity. Similar to [23], we employ Linux `tc` [75] at the software agents to control the generated uplink traffic by the stations. It should be noted that since it is hard to maintain the custom changes throughout the release period of hardware driver, manufacturers of end-devices prefer not to make custom changes on the drivers provided by the chipset vendors. Hence, it is recommended to keep the Wi-Fi stack of end-devices intact to have more freedom for the development of new software-based solutions. We considered this fact as a principle to select the components of our proposed design. Table 2.2 summarizes a list of the components utilized in our testbed setup.

Before discussing the outcomes obtained from the testbed setup, it is important to note that the experiments are carried out in an environment with external interference caused by the neighbour APs on which we have no control and their operating channels are variable over time. Moreover, as stated earlier, for the simulations, we utilize the protocol interference model to find a near-optimal link scheduling solution using a subset of the MISs calculated from the MDCG of the network topology. This approach empowers us to model the inter-flow interference among the wireless links using the calculated MISs. However, this model cannot reflect all the characteristics of the wireless channel which may impact the physical data rates of the wireless links established between the APs and stations. During the testbed experiments, we have considered such an impact to calculate the effective data rates between the APs and stations that represent the real channel conditions. In addition, in the simulations, we have assumed an ideal channel condition with no external interference (caused by the nearby WLANs) and only the impact of inter-flow interference is taken into account. Thus, there is a gap between the results acquired from the testbed and simulation for the same scenarios. Further details are presented in the next section.

Table 2.2: Components of the testbed setup

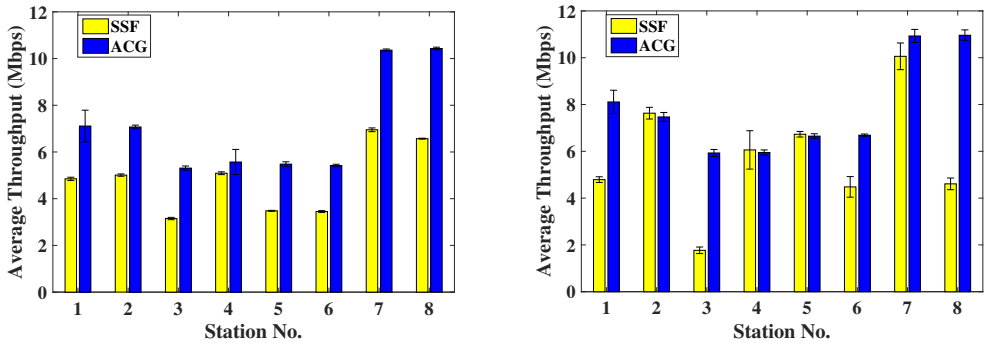
Components	Description
Controller hardware	Dell PC (Optiplex 745)
AP/Station hardware	PC Engine Alix3d2
AP software	Hostapd v2.1-devel
Wi-Fi standard	IEEE 802.11n (MCS-5) / 5 GHz
Wireless radio adapters	Compex WLM200NX (Atheros 9k driver)
Operating system	Ubuntu 14.04 LTS (Server Edition)/ kernel 3.13.0-32-generic
Monitoring tools	Nagios 4.1.1 / Cacti 0.8.8f / USRP2
SDN components	Ryu Controller 4.15 / OpenFlow 1.3 / OpenVSwitch 2.8.1
Measurement tools	Iperf (2.0.5) traffic generator

2.6.2 Experiments and Results

The presented results in Section 2.5.1 substantiate the superiority of our proposed solution versus the other association schemes by simulation. The results obtained from the simulations show the efficacy of the presented solution for large scale scenarios. However, since we are not able to deploy a large scale testbed setup, we have carried out specific experiments using a limited number of devices to assess the functionality and performance of our work as a proof of concept. In this section, we validate the functionality of our scheme (ACG) through a real testbed setup. We define four separate scenarios to investigate different aspects of our solution in terms of bandwidth allocation, traffic forwarding, traffic overhead and adaptability. The performance of ACG for UDP and TCP traffic flows in uplink, downlink and mixed scenarios is evaluated. In the following subsections, we explain the conducted experiments and discuss the obtained results.

The performance of bandwidth allocation

We have investigated the performance of ACG versus SSF (as a practical baseline that almost is used everywhere as the default association scheme) in terms of the allocated bandwidth and aggregate throughput for different combinations of downlink and uplink traffic flows. All the APs use the same SSID and function on 5 GHz frequency bands. We use eight Wi-Fi stations with greedy traffic flows that are either generated from or destined to the `Iperf` node shown in Fig. 2.17. As mentioned earlier, in the conducted scenarios, each Alix board represents two stations (using two Wi-Fi interfaces) and a single traffic flow is assigned to every station. We measure the obtained bandwidth (throughput) by the greedy flows in uplink/downlink scenarios for UDP and TCP traffic.



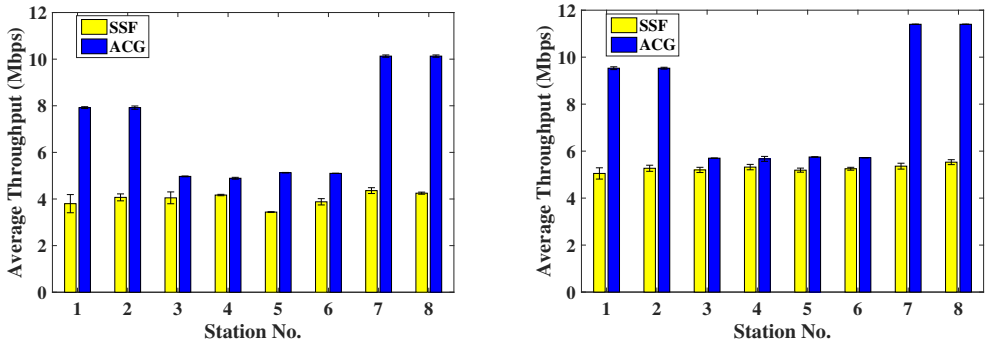
(a) TCP uplink traffic. (b) UDP uplink traffic.

Figure 2.20: Obtained results for greedy uplink traffic.

In the first scenario, the stations generate eight greedy uplink TCP and UDP flows that are destined to the `Iperf` node and we quantify the uplink throughput of the network flows (stations) for the default association scheme (SSF). Then, we carry out the same process for our proposed solution (ACG) and the average outcomes of both schemes for greedy uplink TCP and UDP traffic are shown in Fig. 2.20a and Fig. 2.20b, respectively. According to the shown graphs, the average aggregate throughput for both types of traffic is improved significantly (TCP: +47%, and UDP: +35.8%) by using ACG. Note that for all the experiments, when we use SSF, the stations regarding the received signal strength from the APs are associated to AP-1, AP-2 and AP-3, and AP-4 is not used by the SSF. In this case, just two non-overlapping channels (36 and 44) are utilized and the capacity of the third one is totally ignored by SSF. On the contrary, ACG achieves better results by an efficient utilization of the available network capacity.

In the second scenario, we carry out a similar process for the greedy downlink TCP and UDP traffic flows. In this case, we run `Iperf` at the Wi-Fi stations in server mode to receive the generated downlink traffic by the `Iperf` node. The outcomes of downlink measurements for TCP and UDP are illustrated in Fig. 2.21a and Fig. 2.21b, respectively. Note that in comparison to the uplink results, we see a better distribution of the allocated bandwidth for SSF. However, since SSF does not use the capacity of channel 149, there is a notable gap (TCP: 24.17 Mbps, and UDP: 22.53 Mbps) between the aggregate throughput of SSF and ACG.

As stated earlier, due to using the protocol interference model and the impact of external interference, we expect to see a gap between the testbed and simulation results for each scenario. For instance, according to the association map calculated



(a) TCP downlink traffic. (b) UDP downlink traffic.

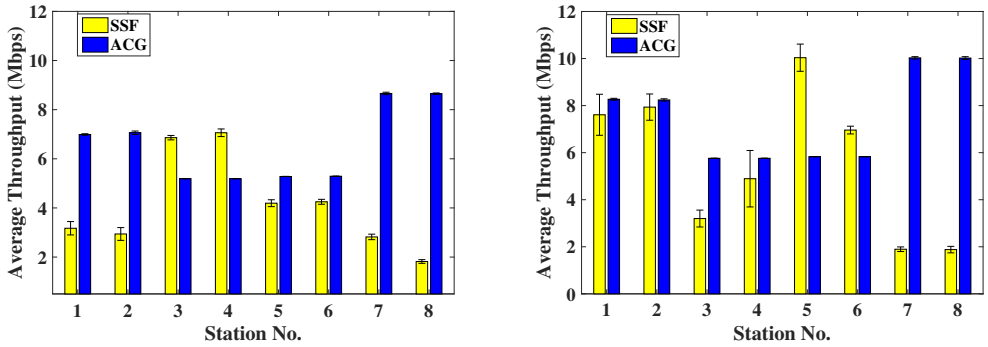
Figure 2.21: Obtained results for greedy downlink traffic.

Table 2.3: Allocated bandwidth to the stations in downlink scenario (simulation)

Station Index	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Throughput (Mbps)	15.8	15.8	8.95	8.95	8.95	8.95	17.9	17.9

by ACG for the downlink greedy scenario (shown in Fig. 2.21), stations {1, 2} and {7, 8} should be associated to AP-1 and AP-4, respectively both of which are working on non-overlapping channels. The remaining stations are associated with AP-2 and AP-3, which are co-channel. Since the traffic flows are greedy, ACG shares the channel capacity in a fair fashion among the Wi-Fi stations. Hence, as illustrated in Fig. 2.21a and Fig. 2.21b, the channel capacity of each of AP-1 and AP-4 (on channels 44 and 149) is shared equally among stations {1, 2} and {7, 8}, respectively. The same trend can be seen for the other stations associated with AP-3 and AP-4 which are operating on channel 36. The aggregate UDP bandwidth of the Wi-Fi stations measured from the testbed setup for channels 44, 36, and 149 are $\approx$ 20 Mbps, $\approx$ 24 Mbps, and $\approx$ 23 Mbps, respectively. The simulation results for the same scenario without considering the accurate impact of inter-flow interference and the external interference caused by the nearby APs are $\approx$ 32 Mbps, $\approx$ 36 Mbps, and $\approx$ 36 Mbps, respectively. The allocated bandwidth to each Wi-Fi station is shown in Table 2.3. It should be noted that in spite of such a performance difference, the main goal of ACG, which is the fair distribution of the (existing) channel capacity among the Wi-Fi stations, has been achieved and it is shown via the presented results. The same trend happens for the other testbed and simulation scenarios.

Finally, we quantify the obtained bandwidth by the stations when there is a mixture of greedy uplink and downlink traffic. In this scenario, four greedy uplink



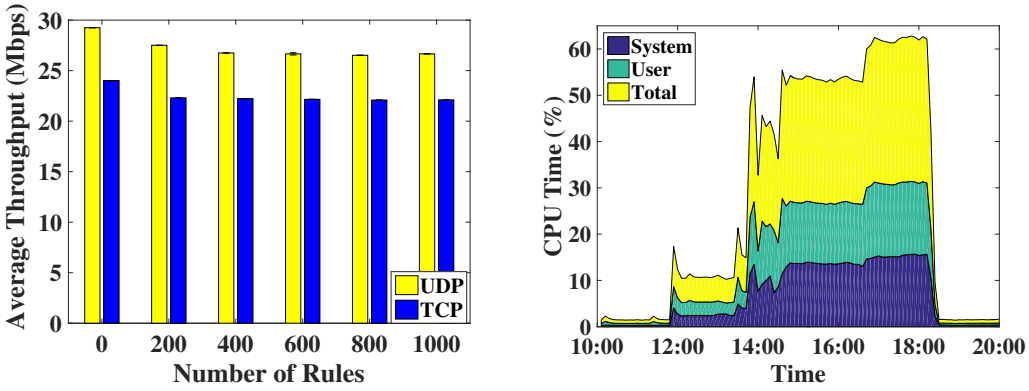
(a) TCP mixed traffic. (b) UDP mixed traffic.

Figure 2.22: Obtained results for greedy uplink and downlink traffic.

flows generated by S2 and S3 are destined to the `Iperf` node and at the same time, four greedy downlink flows are generated by the `Iperf` node are sent to S1 and S4. The results of this experiment for TCP and UDP traffic are shown in Fig. 2.22a and Fig. 2.22b. One of the key outcomes of this experiment is the considerable improvement of the Jain's Fairness Index (JFI) for ACG versus SSF (TCP: +13%, and UDP: +20%). In addition, the aggregate throughputs of TCP and UDP traffic flows are increased by 60% and 35%, respectively. It is important to note that through all the experiments, applying ACG does not always increase the allocated bandwidth to all the stations. This case can be seen for Stations 3 and 4 in the results presented in Fig. 2.22a. However, ACG not only improves the aggregate throughput but also distributes the existing capacity among the flows in a fair manner. The duration of each measurement is 5 minutes and the results are the averages of 5 consecutive runs.

The performance of SDN traffic forwarding

In this section, we explore the impact that the number of defined rules (in the `OVS` flow tables of the APs) has on the traffic forwarding performance. To do this, we create different numbers of dummy complex rules at the `OVS` of AP-1. Then, a single greedy uplink UDP flow (generated by S1) is used to measure the forwarding throughput for different numbers of complex rules. Note that the complex rules are comprised of several fields including IP/MAC addresses and UDP port numbers, and they should be processed before matching the desired rule to forward the UDP traffic. The average throughput versus the number of `OVS` rules is shown in Fig. 2.23a. Moreover, we use `SNMP/Cacti` to measure the CPU utilization of AP-1 for different numbers of rules



(a) Throughput vs. # of Rules. (b) CPU Load (AP-1).

Figure 2.23: The performance of SDN traffic forwarding.

over time as displayed in Fig. 2.23b.

Regarding the results shown in Fig. 2.23a, it can be seen that increasing the number of complex rules up to 1000 has a trivial impact on the average forwarding throughput of UDP and TCP traffic. This feature substantiates the scalability of our SDN-based solution (in the data plane) to process a large number of traffic flows without affecting the network performance. In a similar experiment shown in [76], the performance of the introduced solution was degraded conspicuously by increasing the number of rules from 20 to 100. Moreover, the CPU utilization versus time (which is correlated with the number of rules) is illustrated in Fig 2.23b. Note that the number of SDN rules is increased over time and the impact of this action on the CPU utilization is shown in this figure. It can be seen that it has a drastic impact on the CPU utilization of AP-1 and it has consumed up to 60% of the total CPU time. Also, in the shown graph, there is a small improvement ($\approx 10\%$) in the network throughput when no rule is defined at `OVS` (first bar shown in Fig. 2.23a). This result comes from the fact that regardless of the number of rules, the throughput of `OVS` data path is lower than the throughput of a regular Linux IP stack.

The overhead of control traffic

In this section, we measure the volume of control traffic generated by different components of our system. We use `Wireshark` to capture the control traffic at AP-1. The captured traffic over one minute is shown in Fig. 2.24. According to the illustrated graph, 12.1% and 2.9% of the total captured traffic are `packet_in` and `packet_out` requests respectively, while only 0.2% of this traffic is generated by the agents (ACG)

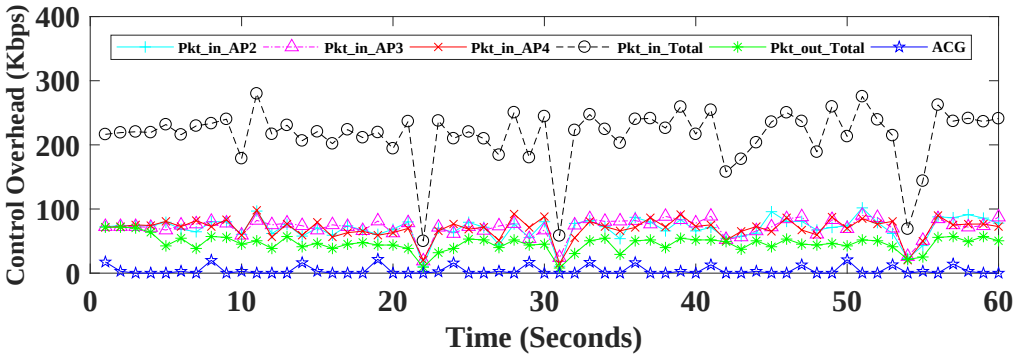


Figure 2.24: The volume of control traffic at the controller.

installed in Wi-Fi stations (for association control).

It is important to note that all the SDN controllers perform topology discovery using OpenFlow Discovery Protocol (OFDP). To discover a link between two nodes, the controller encapsulates a Link Layer Discovery Protocol (LLDP) packet in a `packet_out` message and sends it to the first node. The `packet_out` contains the instruction for this node to forward the LLDP packet to the second node. By receiving the LLDP packet, the second node encapsulates it in a `packet_in` message and sends it back to the controller. Then, the controller discovers the existence of a link between two specific nodes. Regarding this explanation, AP-1 receives `packet_in` messages from all the other APs which are connected to the controller through the same network. In Fig. 2.24, it can be seen that the volume of `packet_out` traffic destined to AP-1 (shown by green color in the figure) is in the range of the generated `packet_in` messages by each AP. In addition, it is clear that the volume of the control association traffic (generated by the agents installed in the stations), which is shown in blue color (ACG), is quite negligible in comparison to the average rate of aggregate SDN traffic ($\approx$ 250 Kbps).

The adaptability of ACG

In the next experiment, we take a closer look at the dynamics and the response time of our solution to the network changes. We set the running period of our developed solution at the controller `(NOM)` to 15 sec. At the beginning of the experiment, six Wi-Fi stations (represented by S1, S2, and S3) join the network using the SSF association scheme. After association, they generate six greedy UDP uplink flows destined to the `Iperf` node. The first 15 seconds in Fig. 2.25 portray this interval

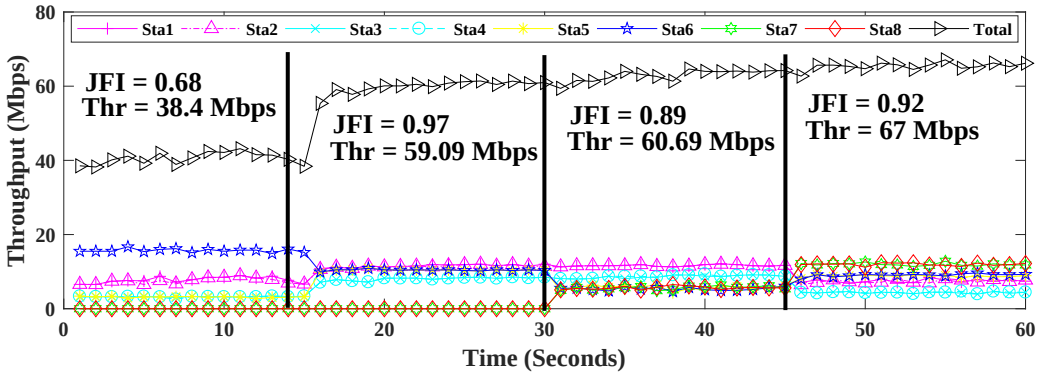


Figure 2.25: The measured throughput over time.

and the measured aggregate throughput and JFI values in this time period are 38.4 Mbps and 0.68, respectively. At $t = 15$ sec, our scheme (ACG) calculates and applies a new association map for the Wi-Fi stations. This leads to a swift and sharp increase of the aggregate throughput (+53%) and JFI (+42%). As mentioned in prior sections, regarding the placement of Wi-Fi stations, SSF forces them to associate with the first three APs that use two non-overlapping channels.

On the other hand, ACG exploits the extra capacity provided by AP-4 (channel 149). In addition, it regulates the allocated bandwidth to the stations in a fair fashion. This fact can be seen one second after applying our solution ($t = 16$ sec). Then, at $t = 31$ sec, two new stations (Sta7 and Sta8) join the network using SSF. Although this change makes a trivial increase on the aggregate throughput, this metric has a noticeable improvement ($\approx +10\%$) after the next run of ACG at $t = 45$ sec. Note that the black vertical lines shown in Fig. 2.25 represent the moments that the `NOM` recalculates the association map. Regarding the explained result, it is clear that ACG quickly reacts after each cycle to improve the network performance. It is possible to decrease the response time of our solution by reducing the duration of this cycle. However, note that it may impose extra overhead on the control and data planes.

2.7 Conclusion

The considerable degradation of network performance in dense WLANs is one of the main consequences of using the existing distributed resource allocation mechanism. Software-Defined APs, as the building blocks of SD-WLANs, empower us to develop more efficient and sophisticated solutions to enhance the performance of network

services. In this chapter, we introduced an SDN-aware framework for the traffic engineering of WLANs. Our modular solution builds a cooperative platform to provide fine-grained resource allocation for the stations of different service providers. In addition, we proposed an approximated optimal solution for AP association (as the first step to establish a Wi-Fi connection) in large-scale dense WLANs that runs in polynomial time. In contrast to most of the prior work, our solution is able to provide a set of key services including association control, channel assignment and link scheduling for both downlink and uplink traffic. Moreover, the performance of the presented scheme is evaluated through extensive numerical results and its functionality is validated through a testbed setup that substantiates the practicality of our scheme. Context-aware network configuration and device profiling to predict the network activities can be considered as the potential topics for the future work.

Chapter 3

Fine-grained Traffic Engineering on SDN-aware Wi-Fi Mesh Networks

3.1 Background

According to [3], around 3.7 billion people (half of the Earth's population) still do not have any kind of Internet access and lack of this connectivity deprives them from social and economic benefits of the Internet. In another study [5], it is mentioned that the current prices of mobile data plans have to be reduced to 10% of the average prices in 2016 to make them universally affordable. Moreover, in a report [4] published by Facebook in 2015, it is stated that about 96% of the world's population is living within the range of cellular networks that can be utilized for provisioning the Internet access to the disconnected population. On the other hand, the dramatic growth of data traffic has left the mobile operators no choice but to use Wi-Fi networks as an economic solution to maintain the quality of their services.

In fact, Cisco has forecasted that 63% of the total IP traffic will be offloaded to Wi-Fi networks by 2021 [77]. Over the past few years, some of the largest cellular operators restricted their unlimited data plans and proposed tiered charging plans to reduce their expenses [6]. While cellular operators attempt to guarantee the QoS for their services in a profitable manner, Wi-Fi can be considered as a viable complement. The massive growth of inexpensive Wi-Fi-enabled devices and public hotspots as well as the utilization of unlicensed spectrum makes Wi-Fi an attractive data offloading technology for the operators to avoid the substantial deployment costs of cellular networks.

The new generations of wireless solutions promise high throughput and economic connectivity to everyone (especially for disconnected communities), which can be provided through existing and emerging technologies. One of the existing technologies is WMNs, which as a promising and affordable solution, provides diverse types of services where wired infrastructure is either not ready or too costly to be deployed. In general, mesh architecture supports long distance communications by the establishment of multi-hop paths. Since WMNs can be constructed using inexpensive off-the-shelf products and unlicensed spectrum, it is a valuable solution for most of the service providers. However, the majority of existing WMN solutions are characterized by their distributed nature and they suffer from inflexible traffic engineering mechanisms and long recovery of link failures.

It is important to note that in the current implementation of conventional mesh routing protocols, each mesh router as the building block of WMN has to act independently and make routing decisions based on its local information. This approach not only leads to a poor performance but also is not able to utilize the total backhaul capacity as well as reacting swiftly to unexpected topology changes [14]. In fact, the distributed nature of these routing solutions imposes noticeable convergence latency in large networks and it might miss the near-optimal configuration due to the lack of a holistic view of the system. Furthermore, the frequent exchange of control messages among mesh routers to keep an updated map of the network topology incurs a non-trivial overhead that may reduce the long-term throughput [78].

As stated in the previous chapter, SDN as an emerging centralized network paradigm has made significant headways in communication networks. Although the usage of decentralized systems may seem more pragmatic, the growth of centralized schemes enables us to implement more efficient and cost-effective solutions. Moreover, the centralized placement of control functions empowers us to utilize high-performance computing resources at data centers and more sophisticated algorithms for resource management. Indeed, SDN presents more flexibility and fine-granularity in terms of traffic engineering that can be applied to many communication platforms, including WMNs. The growth of centralized schemes not only enables us to implement more efficient solutions but also allows us to utilize simple networking devices for packet forwarding, which reduces the deployment cost, energy consumption and control overhead [7, 8, 78]. Thus, by enabling SDN over mesh routers, we can exploit a set of salient features that not only enhance the QoS but also decrease the deployment costs. Therefore, adopting SDN-aware WMNs can be considered as a

resilient approach for the Internet access provisioning. In addition, the placement of control functions in data centers fortifies the security measures by running advanced spectrum monitoring algorithms for detection of suspicious traffic and malicious attacks as well as taking the respective countermeasures [79].

For serving every user of a WMN, three key steps, including AP association, gateway selection and backhaul flow routing should be taken into account. By taking these steps, traffic flows stream from/to the selected gateways over the WMN topologies. One of the interesting facts about these steps is the feasibility of mapping them to the classic computer science problems. Generalized Assignment Problem (GAP), Capacitated Facility Location Problem (CFLP) and Multi-Commodity Flow Problem (MCFP) are three well-known examples that can be applied to AP association, gateway selection and backhaul flow routing problems, respectively. Despite the tight correlation among these steps, most of the former studies either investigated them as independent NP-hard problems or did not present a unified and all-inclusive solution. In this chapter, by enabling SDN over WMNs, we propose a fine-grained framework to find suboptimal solutions in polynomial time which have acceptable gaps with the optimal outcomes. Actually, we intend to ensure that all the flows get a minimum guaranteed bandwidth w.r.t. the predefined QoS constraints over the mesh backhaul network as well as to maximize the aggregated network throughput. Furthermore, one of the key features of our scheme is minimizing the number of required gateways that not only is in line with the tiered charging strategy of large cellular operators but also has a significant impact on the deployment process. Hence, our work facilitates the agile formation of WMNs for the Internet access.

To have a better picture about the problem, suppose that we intend to provide low-cost or free Internet access to participants of an event in a non-urban area of a developing country where the access to the wired backbone is either not possible or too costly. On the other hand, as mentioned earlier, cellular networks are accessible almost everywhere, but due to the high price of data plans, many people with low income cannot afford the cost of cellular data plans for the Internet access. However, by using an intermediate approach, we can bridge data traffic from cellular to Wi-Fi networks and share it among users through a WMN in an affordable way. In the explained scenario, it is important to control the number of mesh routers that function as cellular gateways to reduce the deployment cost. Otherwise, the users that run bandwidth-greedy applications impose a significant cost on the service providers by downloading a large volume of data from cellular networks. Hence, we have to cap the

greedy flows and guarantee a minimum level of QoS for the users. So, we can construct a WMN by mesh routers equipped with cellular interfaces that can function as a Router, an AP and a Gateway (MRAG). Throughout this chapter, we use MRAG to represent a device that supports all these three functions. By activating the gateway functionality just for a limited number of MRAGs and steering the traffic flows, we are able to reduce the cost of service provisioning significantly. Furthermore, using SDN helps us to formulate the explained scenario as a set of optimization problems to find the optimal configuration for gateway selection and flow routing problems.

In this chapter, we present a structured scheme to address the demands of end-users over SDN-aware WMNs. We first introduce an optimization problem to maximize the proportional fairness for AP association in the access layer. Then, we propose a Mixed Integer Non-Linear Programming (MINLP) problem to solve a joint optimization problem including gateway selection and flow routing on SDN-aware WMNs. The presented formulation considers the AP association through involving the bandwidth obtained by every flow in the access layer as an upper bound for the final allocated rate to the same flow. Moreover, it finds the least number of required gateways adaptively. Due to the NP-hardness of the MINLP problem, also we introduce a heuristic algorithm to find a suboptimal solution in polynomial time which is designed based on the outcomes of the relaxed MINLP problem. In addition, since our presented scheme gives us a multi-path flow routing solution that brings extra complexities in terms of implementation over MC-MR WMNs, we propose a complementary routing solution. Indeed, the presented solution is a randomized single-path flow routing algorithm for SDN-aware WMNs that converts a multi-path flow routing problem to a single-path one. The performance of the presented solutions has been evaluated through extensive numerical results and testbed implementation. It should be noted that our work is protocol-independent and it does not rely on the underlying physical layer protocols. Also, it can be applied to the mesh topologies that are constructed using MC-MR MRAGs.

3.2 Related Work

In this section, we first review the most recent work on SDN-aware WMNs. Then, we introduce some of the related publications on the joint traffic engineering over WMNs. Finally, we discuss the challenges of flow routing over MC-MR WMNs.

3.2.1 SDN-aware Wireless Mesh Networks

In the span of the past few years, several studies have been carried out for applying SDN to WMNs [76, 80, 81, 82]. In one of the recent ones [83], an SDN architecture for traffic orchestration over WMNs is presented. The authors proposed a modular solution for traffic scheduling and spectrum allocation using software-defined mesh routers. The main focus of this work was on spectrum sharing and Dijkstra algorithm was used to implement a simple traffic routing scheme for single-radio mesh networks. In another work [84], the authors proposed a crowd-shared WMN platform to interconnect home routers that provides multiple Internet gateways and utilizes an SDN-based control plane. Although the efficiency of the presented work is evaluated through extensive simulations, the acquired results have not been compared with other similar schemes. Moreover, the proposed solution is not applicable to MC-MR scenarios in WMNs. One of the missing aspects of the prior work on SDN-aware WMNs is the optimal traffic engineering such that not only the network throughput is maximized, but also the number of the required gateways to guarantee the defined QoS is minimized. In the next section, we introduce some of the related work on joint gateway selection and flow routing.

3.2.2 Joint Gateway Selection and Flow Routing

As stated before, there is a strong relation among AP association, gateway selection and flow routing processes. Due to the NP-hardness of these problems, we have to use approximation or heuristic schemes to find suboptimal solutions. For the first problem, several studies were carried out on WLANs that can be extended to WMNs. In one of the most recent work [39], a meta-heuristic solution to find an approximation of the optimal result is presented and evaluated through testbed setup. In another work [44], the authors formulated a joint station association and rate allocation in WMNs. Despite the efficacy of the presented solution, it can be applied to single-radio single-gateway WMNs with unlimited backhaul capacity.

To solve the flow routing problem in a centralized way, MCFP is used as one of the most popular approaches. This is an optimization problem with multiple constraints that illustrates the characteristics of a given network. Since finding the optimal solution for single-path MCFP in large-scale networks is NP-hard, many studies are conducted to find suboptimal solutions. For example, in [85], an energy-efficient flow-based routing is introduced and the authors used the ACO to find a

suboptimal solution. In another work [49], an optimal capacity planning scheme for MC-MR WMNs based on MCFP is presented. By using a multi-dimensional conflict graph and a relaxed MILP problem, a structured approach for finding the optimal capacity is introduced. In a similar work [86], the performance of a 2-hop routing for mobile ad-hoc networks is investigated and a suboptimal solution to maximize the aggregated throughput via MCFP is presented. In another work [87], the scholars presented centralized and distributed MCFP-based solutions to find an approximation of the optimal routing solution over WMNs. It should be taken into account that in the majority of the prior work, the gateway selection is ignored.

Although by increasing the number of active gateways we can improve the network throughput, it increases the deployment cost significantly and we have to find a trade-off between the number of gateways and the network capacity. To address this problem in [88], an Internet gateway deployment strategy for MC-MR WMNs is presented. The authors formulated a multi-objective optimization problem and utilized a hierarchical clustering for gateway placement. There is another work [89] where authors introduced a solution for grid-based topologies to maximize the network throughput. Also, the scholars presented heuristic solutions for the throughput maximization and single/multi-gateway assignment [90] in MC-MR WMNs. In addition, using CFLP is considered as one of the efficient techniques to find suboptimal gateway placements [91, 92]. Building spanning trees and clustering the topologies are the key steps of the gateway selection problem in most of the former studies. In another recent work [82], the scholars proposed a joint flow-based routing and gateway selection optimization problem that can be developed on top of an SDN-aware WMN. Although they presented a fine-grained solution, it cannot solve the large-scale problems in polynomial time.

It should be noted that none of the prior work proposed a unified joint solution (including AP association, flow routing and gateway selection) that can be solved in polynomial time. Note that finding the optimal results for the joint problems in MC-MR WMN scenarios necessitates to deal with NP-hard problems [49].

3.2.3 Single-Path Flow Routing over MC-MR WMNs

Although there are several studies [93, 94] on the development of multi-path routing schemes for wired networks, these solutions cannot be applied directly to the WMNs due to the dynamic and half-duplex nature of wireless networks. In contrast to the

wired networks where each link has a reserved capacity, there is no dedicated capacity for the wireless links of a shared spectrum. Such a behaviour is mostly caused due to the existence of co-channel interference and HTs. However, by using an MCFP-based flow routing and employing efficient link scheduling mechanisms, we can have an LP formulation for multi-path flow routing that can be solved in polynomial time, but it should be noted that the implementation of multi-path flow routing over MC-MR WMNs brings extra complexities including packet re-ordering and the modification of the SDN protocols (at network equipment). To avoid this complexities, using a single-path flow routing is an alternative, however, the LP formulation should be converted to an MILP which is NP-hard. Hence, we need to find a right trade-off for solving the flow routing problem over MC-MR WMNs.

There is a limited work on the development of efficient SDN-aware routing protocols for WMNs. The key challenges of implementing OpenFlow-based routing over multi-hop wireless networks are discussed in [14, 95, 96]. In one of the earliest studies [76], the authors applied OpenFlow to WMNs to improve the performance of handover due to client mobility. The presented outcomes were compared with the OLSR protocol and evaluated through a real testbed. In addition, some applications using an SDN-based routing in WMNs were discussed in [97, 98] and the scholars did a limited performance comparison to evaluate the functionality and efficiency of their schemes. In another work [80], an OpenFlow-based framework for gateway load balancing and traffic routing was introduced. The authors used OLSR as an alternative routing mechanism in case of failure of the SDN controller as well as routing the SDN control traffic through an Out-Of-Band (OOB) mesh network.

Also, to address the limitations of conventional WMN routing protocols, some hybrid routing solutions introduced in [99, 100] that were evaluated through simulations. Although the functionality of the introduced schemes was assessed for different scenarios, it cannot be applied to the MC-MR WMNs. In another recent work [101], a system model for the joint routing and resource allocation in OFDMA-based software-defined wireless backhaul networks was proposed. However, some aspects of the introduced routing scheme such as the protocol overhead were not investigated and just a preliminary performance analysis through simulation was carried out. It is important to note that almost none of the related work considered the impact of using MC-MR in their presented solutions and they did not propose a feasible single-path flow routing mechanism for SDN-aware WMNs. In the next section, our proposed architecture and solution components are delineated.

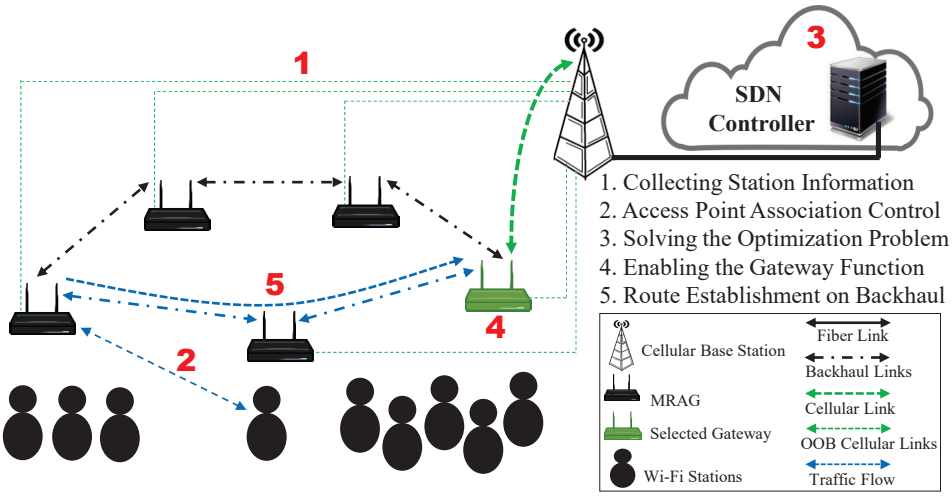


Figure 3.1: The big picture of the presented architecture.

3.3 System Model and Architecture

In this section, we introduce the architecture, its different components and the structure of our proposed solution.

3.3.1 Solution Architecture

As mentioned in Section 3.1, to guarantee a minimum service level for the Wi-Fi clients over the access and backhaul networks, we need to design a systematic approach that consists of the three aforementioned key steps. One of the crucial requirements of this process is collecting the network statistics. In fact, by relying on the global topology information, an SDN controller can determine which MRAGs have to be selected as the cellular gateways. To collect the global network information, we can use software agents installed on the Wi-Fi clients. It is important to note that recently, many companies with large-scale services encourage their customers to install verified software agents on their devices for improving the performance and security of their services. Hence, this part of our solution can be deployed as a module of these agents to collect network information. In addition, we use a monitoring module at the MRAGs to collect different sort of information about the Wi-Fi clients and the conditions of the wireless channel. Moreover, the SDN controller estimates the association map and the allocated bandwidth to the end-users. Fig. 3.1 presents the big picture of the introduced scenario in Section 3.1. In this figure, the selected gateway and the cellular links are shown in green color. As illustrated, the formed

mesh network among the MRAGs (shown in dashed black lines) is used to route the flows from the selected gateway to the access tier and vice versa. In fact, this figure demonstrates the entire process to establish an end-to-end connection between a Wi-Fi client and its selected gateway based on our introduced design.

Note that our proposed scheme for SDN-aware WMNs periodically communicates with MRAGs and utilizes the network information as the input of the presented optimization problem in the next section. The corresponding outcomes will be used by the SDN controller to adjust the forwarding rules at MRAG nodes. Also, one of the challenges of SDN-aware WMNs is the reliable delivery of control traffic between the MRAGs and the controller. This task can be done through either an isolated control network (out-band) or using the same backbone to transmit the control and data traffic (in-band). In our presented solution, the cellular network and the mesh backhaul are the possible choices for the out-band and in-band scenarios, respectively. However, for the scenarios with multiple gateways that are located in a network separated from the SDN controller, using out-band network is the best choice. For the cases with a limited number of MRAGs with mainly user-to-user communications and predetermined gateways (located in the same network), in-band network can be considered as a viable option. In such a scenario, only the gateway needs to have an active cellular connection and the other MRAGs do not require to use the cellular network to communicate with the controller that reduces the operational cost. Note that for in-band solution, we also need to prioritize the control traffic over the backhaul network to avoid severe contention between the control and data traffic.

It is important to note that one of the main scenarios to which the introduced solution can be applied is when the mobile operators charge their customers based on the tiered data plans and the volume of data traffic that is received from or transmitted to the cellular networks. In this case, controlling the amount of this traffic (and its overhead) through minimizing and dynamic adjustment of the cellular gateways plays a pivotal role in the service provisioning process. Hence, in the countries with the flat rate service plans, using the introduced solution does not reduce the cost of Internet service provisioning, significantly. However, for the scenarios in which it is not possible to provide cellular access for all the MRAGs (due to having a limited number of cellular interfaces or SIM cards), our solution is able to find the minimum number of required cellular gateways and their best placement to guarantee the defined QoS constraints. This feature is independent from the type of service plans provided by the mobile operators.

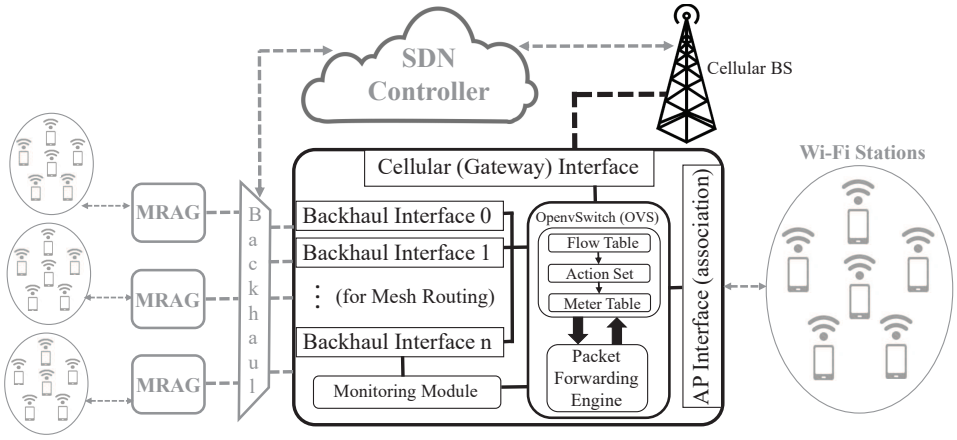


Figure 3.2: The proposed MRAG architecture.

3.3.2 Solution Components

The key elements of the presented solution are SDN-aware MRAG nodes that form the mesh backhaul and serve the end-users. Fig. 3.2 illustrates the components of an MRAG node and its correlation with other network elements. As shown, each MRAG has three types of interfaces including AP, backhaul and cellular gateway interfaces. The first type is utilized for the association of end-users to join the access network. After association, the access traffic should be routed through the backhaul interfaces. Either the source (for download) or the destination (for upload) of this traffic is one of the selected gateways. Every MRAG can have multiple radio interfaces to build an MC-MR mesh backhaul with other MRAGs. In addition to the mesh backhaul interfaces, we assume that each MRAG is equipped with a cellular interface that can enable its gateway functionality, if needed. Note that as illustrated in Fig. 3.2, for the AP association and cellular connectivity, we utilize a single-radio interface, and the multi-radio feature is applied to the backhaul routing process.

In our design, the integrator of the introduced interfaces is `Open vSwitch (OVS)` to switch the traffic flows among them. The traffic flows can be managed either via the Controller, whose engine is the `OpenFlow` protocol, or `OVS` flow table. In both cases, the defined rules are composed of `<match, action>` pairs. In fact, the flow table is made of a group of flows which are processed in a predefined order. The `match` part of a flow determines the field(s) of a packet to be matched to hit the flow. Once a match happens, the `action` part defines what should be done (from the `Action` set) when the flow is hit. For monitoring the flow rates and collecting network statistics, we utilize the `Meter` table and monitoring module, respectively. Also, we have assumed

that all MRAGs are deployed in an area with cellular coverage. Thus, the initial configuration of the MRAGs can be carried out through a secure communication link over the cellular network. Then, the control traffic from the controller to the MRAGs and vice versa can be delivered through the cellular or backhaul networks.

3.3.3 The Structure of the Proposed Solution

As stated before, enabling the gateway functionality of all MRAG nodes imposes extra costs on the service providers. Hence, we need to minimize the number of gateway-enabled MRAGs and maintain the defined service level as well. To achieve this, the traffic flows in both access and backhaul tiers have to be served in an optimal fashion. By using the features of SDN over WMN topologies, this goal can be formulated as a collection of correlated optimization problems. Fig. 3.3 represents a detailed picture of the required steps. The illustrated process consists of three key steps as follows. At the first step, we deal with the AP association by building the rate matrix of the stations (Step 1.1) and solving an optimization problem that is delineated in the next subsection (Step 1.2). In Step 2, three phases take place before solving a joint gateway selection and flow routing problem in Step 3. Note that for a scenario with downlink greedy traffic, the results of Step 1 and Step 3 determine the destination and the source of the given network flows, respectively. The acquired results from Step 3 can be used for a multi-path flow routing over mesh backhaul networks as well. Further details on these steps are provided in the next section.

3.4 Problem Formulation and Proposed Solution

We delineate our MINLP problem formulation and the presented heuristic algorithm in this section.

3.4.1 AP Association Problem

As the first step in Fig. 3.3, we determine MRAGs that Wi-Fi stations have to be associated with. Indeed, we intend to solve an optimization problem by mapping every station to an MRAG such that it maximizes the objective function. The presented scheme in the previous chapter for AP association is based on the max-min fairness which maximizes the aggregate throughput and guarantees a minimum bandwidth

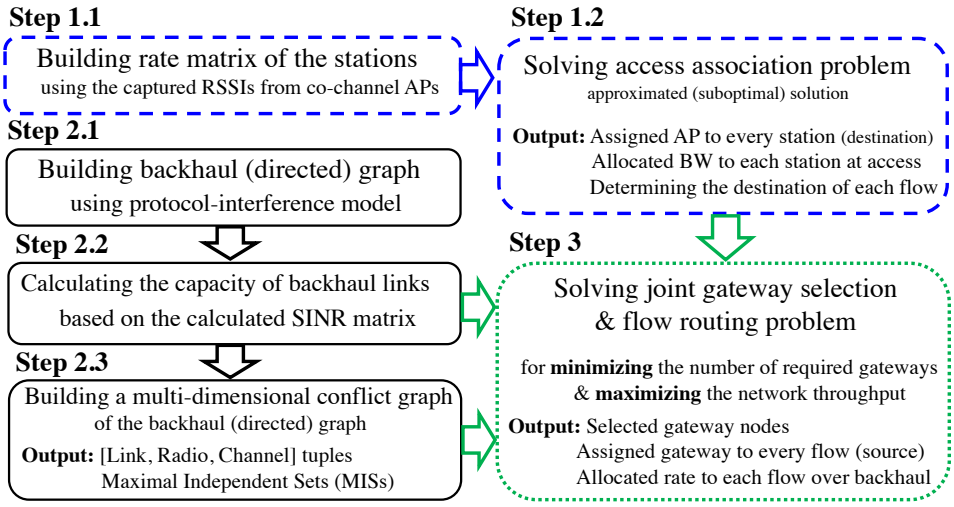


Figure 3.3: The structure of the proposed fine-grained solution.

for every station. On the other hand, in this chapter, we use proportional fairness to assign bandwidth proportional to the link data rates of the stations that achieves better performance in terms of fairness index. Hence, it ensures that the access layer is not being a bottleneck in the bandwidth allocation process over the backhaul layer. Also, it should be noted that in special cases, both schemes may achieve the same performance. For instance, in [102], it is shown that with a saturated AP, throughput-based fairness and max-min fairness are equivalent and if all the stations have the same priority, time-based fairness and proportional fairness are also equivalent [40]

By finding the corresponding MRAG for every station, we can calculate the throughput of each station in the access layer. These values are the inputs (flow demand) of the next step which solves the joint optimization problem to maximize the aggregate backhaul throughput. As it is explained in the previous chapter, AP association is formulated as a two-step optimization problem. It is NP-hard and can be mapped to the GAP. In this section, we present an optimization problem to maximize the proportional fairness among the Wi-Fi stations. Due to the popularity of download traffic and the growth of downlink greedy applications [57, 103], we first introduce the problem formulation for a downlink greedy scenario. Then we obtain an approximation of the optimal result for the large-scale situations. The access association problem of the Wi-Fi stations is formulated as follows.

$$\text{Maximize} \qquad \sum_{s \in \boldsymbol{S}} \sum_{f \in \boldsymbol{F_s}} \log(b_{fs}) \tag{3.1}$$

$$\text{s.t.} \quad b_{fs} = \sum_{m \in \boldsymbol{M}} \sum_{c \in \boldsymbol{C}} x^c_{sm}\, r^c_{sm}\, t^c_{fsm}, \forall s \in \boldsymbol{S}, \forall f \in \boldsymbol{F_s} \tag{3.2}$$

$$\sum_{m \in \boldsymbol{M}} \sum_{f \in \boldsymbol{F_s}} \sum_{c \in \boldsymbol{C}} x^c_{sm}\, t^c_{fsm} \leq 1, \qquad \forall s \in \boldsymbol{S} \tag{3.3}$$

$$\sum_{s \in \boldsymbol{S}} \sum_{f \in \boldsymbol{F_s}} \sum_{c \in \boldsymbol{C}} x^c_{sm}\, t^c_{fsm} \leq 1, \qquad \forall m \in \boldsymbol{M} \tag{3.4}$$

$$\sum_{m \in \boldsymbol{M}} \sum_{c \in \boldsymbol{C}} x^c_{sm} = 1, \qquad \forall s \in \boldsymbol{S} \tag{3.5}$$

$$t^c_{fsm} = \frac{v_{sm}\, p_{fs}\, x^c_{sm}}{(\sum\limits_{s' \in \boldsymbol{S}} \sum\limits_{f' \in \boldsymbol{F_{s'}}} x^c_{s'm}\, p_{s'f'})(\sum\limits_{m' \in \boldsymbol{M}} q^c_{mm'})}, \quad \forall m \in \boldsymbol{M}, \forall c \in \boldsymbol{C}, \forall s \in \boldsymbol{S}, \forall f \in \boldsymbol{F_s} \tag{3.6}$$

$$t^c_{fsm} \in [0,1], \qquad x^c_{sm}, v_{sm}, q^c_{mm'} \in \{0,1\} \tag{3.7}$$

In this formulation, the main objective in (3.1) is maximizing the throughput of the access network in a proportionally fair manner. In the objective function, $\boldsymbol{S}$ and $\boldsymbol{F_s}$ represent the set of stations and the set of flows that belong to station s, respectively. It is assumed that the flows are greedy and each station (and its respective flows) is associated to only one MRAG. The first constraint in (3.2) defines the bandwidth allocated to flow f of station s (b_{fs}). In this constraint, x^c_{sm} is a binary association variable that is used to check the association of station s to MRAG m over channel c. The allocated airtime to flow f of station s over the wireless link to MRAG m on channel c is shown as t^c_{fsm}. Indeed, it can be defined as the effective transmission time between a given station and an AP. For the greedy traffic flows, we use this variable to determine the share of each traffic flow from the (AP) downlink transmissions. Also, the data rate of the wireless link between m and s over channel c is represented by r^c_{sm}. Note that for each pair of `<station, MRAG>` over different access channels, we can calculate r^c_{sm} values and represent them through a 3-dimensional matrix (Step 1.1). The second constraint in (3.3) ensures the total allocated airtime to the flows of

every station over the set of all MRAGs ($\boldsymbol{M}$) and existing channels ($\boldsymbol{C}$) is not more than 1. Similarly, the total allocated airtime to each MRAG should not be more than 1, as shown in (3.4). The next constraint in (3.5) guarantees that every station s is associated to only one MRAG. As it is mentioned before in Chapter 2, we can model each station with multiple flows by multiple dummy stations with a single flow such that each one is associated to one AP. This can be guaranteed by applying constraint (3.5) to every dummy station that represents a single flow.

Due to the seamless coverage of MRAGs in the access layer and using homogeneous nodes, protocol interference model [104] is utilized as an efficient choice to model the co-channel interference. Thus, we used binary variable v_{sm} to denote that station s either is located or not inside the communication range of MRAG m, and likewise $q^c_{mm'}$ shows the presence of node m' within the carrier sensing range of MRAG m over channel c. Also, since it is assumed that all stations are greedy, the allocated airtime to the flows of every station is a function of the total number of associated flows (stations) to MRAG m and their priorities. This is shown in the first term of the denominator in (3.6). Note that we used p_{fs} to denote the priority of flow f from station s. For the greedy downlink traffic, we need to do an equalized airtime allocation to all the flows at each MRAG as shown in (3.6).

In addition, according to the CSMA/CA mechanism, the existence of co-channel transmitters within the carrier sensing range of each other while they are on the attempt to transmit data, causes interference. In a greedy downlink scenario, since the stations receive and the MRAGs transmit data, the number of co-channel MRAGs with the MRAG that station s is associated with (m) determines the second term of the denominator in (3.6). Hence, this term of the denominator counts the number of co-channel MRAGs within the carrier sensing range of m that function on channel c. In fact, (3.6) guarantees to have collision-free communications by defining t^c_{fsm} as a function of the number of co-channel MRAGs placed within the carrier-sensing range of each other as well as the equal distribution of airtime among the stations associated with the same MRAG. Finally, the last constraint represents the range of airtime variables and the integrality constraints. Now, the controller has to find x^c_{sm} for all the stations such that the objective function is maximized.

It can be seen that the some of the introduced constraints are similar to the ones presented formulation in Section 2.3.2. However, in this formulation the objective function is based on providing the proportional fairness and we did not use any link scheduling mechanism. In fact, it is a simplified representation of the presented

solution in Chapter 2 with a different objective function. Also, due to the NP-hardness of the problem, by taking the relaxation and rounding techniques using bipartite graphs, we can use the same technique used in Chapter 2 to find a 2-approximation of the optimal solution in polynomial time. Further details can be found in [39, 40]. Note that the obtained results at this step (b_{fs}) are used as the inputs of Step 3 in Fig. 3.3. In addition, before solving the optimization problem, the values of v_{sm} and $q^c_{mm'}$ can be obtained based on the information of the given topology.

3.4.2 Joint Gateway Selection and Flow Routing

According to Fig. 3.3, prior to solving the joint problem at Step 3, three preliminary phases take place. As the first phase (Step 2.1), based on the position of MRAGs and the communication range of their backhaul radios, we build a directed graph. Note that similar to the access layer, the protocol interference model is used here as well. In Step 2.2, we calculate the data rate of wireless links in the constructed graph by using the nodal distance and the selected path loss model. Finally, in Step 2.3, we form an MDCG based on the backhaul links and the number of backhaul radios/channels. Note that similar to the discussed terms and definitions in Section 2.3.2, each vertex of the MDCG represents a resource entity (of the MC-MR WMN) and it can be noted as a *Link-Radio-Channel (LRC)* tuple. We utilize the introduced technique in [48, 49] to consider the impact of interference over the backhaul links. By having the MDCG, we can compute its MISs that represent the non-adjacent vertices of the MDCG. Thus, the vertices (tuples) placed in these MISs can transmit simultaneously without interfering with each other. In the last step (Step 3), we solve the joint optimization problem. As stated before, the final goals are minimizing the number of required gateways and maximizing the network throughput w.r.t. the defined QoS constraints. Since achieving these goals through a single objective function is not feasible, the problem is formulated to find the maximum throughput for a given number of gateways. Thus, we first present the joint gateway selection and flow routing optimization problem for a given number of gateways. Then, we propose an algorithm to find the optimal number of gateways (through an iterative process) in the next section. The original joint optimization problem can be formulated as an MINLP problem as follows.

$$\text{Maximize} \qquad \sum_{s \in \boldsymbol{S}} \sum_{f \in \boldsymbol{F_s}} b'_{fs} \tag{3.8}$$

$$\text{s.t.} \quad b_{\min} \leq b'_{fs} \leq b_{fs}, \qquad \forall s \in \boldsymbol{S}, \forall f \in \boldsymbol{F_s} \tag{3.9}$$

$$\sum_{m \in \boldsymbol{M}} \sum_{\substack{l \in \boldsymbol{L} \\ out(m)}} r_{fsl}\, \gamma_{fsl}\, z_{fsm}\, e_m \leq b_{fs}, \forall s \in \boldsymbol{S}, \forall f \in \boldsymbol{F_s} \tag{3.10}$$

$$\sum_{\substack{l \in \boldsymbol{L} \\ out(m)}} \sum_{s \in \boldsymbol{S''_m}} \sum_{f \in \boldsymbol{F_s}} r_{fsl}\, \gamma_{fsl}\, z_{fsm}\, e_m + \sum_{s \in \boldsymbol{S'_m}} \sum_{f \in \boldsymbol{F_s}} b'_{fs}\, z_{fsm}\, e_m \leq G_m\, e_m + \sum_{\substack{l \in \boldsymbol{L} \\ in(m)}} \sum_{s \in \boldsymbol{S''_m}} \sum_{f \in \boldsymbol{F_s}} r_{fsl}\, \gamma_{fsl}\, z_{fsm}\, e_m, \forall m \in \boldsymbol{M} \tag{3.11}$$

$$\sum_{\substack{l \in \boldsymbol{L} \\ in(m)}} \sum_{f \in \boldsymbol{F_s}} r_{fsl}\, \gamma_{fsl}\, (1 - z_{fsm}) = \sum_{\substack{l \in \boldsymbol{L} \\ out(m)}} \sum_{f \in \boldsymbol{F_s}} r_{fsl}\, \gamma_{fsl}\, (1 - z_{fsm}), \forall m \in \boldsymbol{M}, \forall s \in \boldsymbol{S''_m} \tag{3.12}$$

$$\sum_{\substack{l \in \boldsymbol{L} \\ in(m)}} \sum_{f \in \boldsymbol{F_s}} r_{fsl}\, \gamma_{fsl}\, (1 - z_{fsm}) = \sum_{f \in \boldsymbol{F_s}} b'_{fs}(1 - z_{fsm}), \quad \forall m \in \boldsymbol{M}, \forall s \in \boldsymbol{S'_m} \tag{3.13}$$

$$\sum_{s \in \boldsymbol{S}} \sum_{f \in \boldsymbol{F_s}} b'_{fs}\, z_{fsm} \leq G_m\, e_m, \qquad \forall m \in \boldsymbol{M} \tag{3.14}$$

$$\sum_{s \in \boldsymbol{S''_m}} \sum_{f \in \boldsymbol{F_s}} b'_{fs}\, z_{fsm} \leq \sum_{\substack{l \in \boldsymbol{L} \\ out(m)}} C_l, \qquad \forall m \in \boldsymbol{M} \tag{3.15}$$

$$\sum_{\substack{l \in \boldsymbol{L} \\ in(m)}} \gamma_{fsl}(1 - z_{fsm}) \leq 1, \forall m \in \boldsymbol{M}, \forall s \in \boldsymbol{S}, \forall f \in \boldsymbol{F_s} \tag{3.16}$$

$$\sum_{\substack{l\in \boldsymbol{L}\\ out(m)}} \gamma_{fsl}(1-z_{fsm}) \leq 1, \forall m \in \boldsymbol{M}, \forall s \in \boldsymbol{S}, \forall f \in \boldsymbol{F_s} \tag{3.17}$$

$$\sum_{\substack{l\in \boldsymbol{L}\\ out(m)}} \gamma_{fsl} = 0, \quad \forall m \in \boldsymbol{M}, \forall s \in \boldsymbol{S'_m}, \forall f \in \boldsymbol{F_s} \tag{3.18}$$

$$\sum_{l\in \boldsymbol{L}} \gamma_{fsl} \leq d_{fs}, \qquad , \forall s \in \boldsymbol{S}, \forall f \in \boldsymbol{F_s} \tag{3.19}$$

$$\sum_{s\in \boldsymbol{S}} \sum_{f\in \boldsymbol{F_s}} r_{fsl} \leq C_l, \qquad \forall l \in \boldsymbol{L} \tag{3.20}$$

$$\sum_{i\in \boldsymbol{I}} \alpha_i \leq 1, \qquad 0 \leq \alpha_i \leq 1 \tag{3.21}$$

$$\sum_{i:l\in i} \alpha_i w_{li}^c = C_l, \qquad \forall l \in \boldsymbol{L} \tag{3.22}$$

$$\sum_{m\in \boldsymbol{M}} z_{fsm} = 1, \qquad \forall s \in \boldsymbol{S}, \forall f \in \boldsymbol{F_s} \tag{3.23}$$

$$z_{fsm} \leq e_m, \qquad \forall s \in \boldsymbol{S}, \forall f \in \boldsymbol{F_s}, \forall m \in \boldsymbol{M} \tag{3.24}$$

$$\sum_{m\in \boldsymbol{M}} e_m = N_g \tag{3.25}$$

The term in (3.8) shows the objective function that maximizes the aggregated bandwidth (throughput) allocated to the traffic flows of all the stations over the backhaul network. So, b'_{fs} shows the acquired bandwidth by flow f from station s over the backhaul links. The first constraint in (3.9) assures that the bandwidth of each flow (b'_{fs}) is no less than the guaranteed bandwidth ($b_{\min}$) and no more than its demand, b_{fs} (the bandwidth acquired in the access layer). The second constraint in (3.10) ensures that the allocated bandwidth to flow f from station s (r_{fsl}) over the egress (out) links of its assigned gateway m is no more than its demand. Three binary variables are introduced in this constraint. The first one (e_m) determines whether MRAG m is functioning as a gateway or not. The second one (z_{fsm}) is set to 1 when flow f from station s is assigned to MRAG m as its selected gateway,

otherwise it is 0. The third binary variable (γ_{fsl}) determines whether flow f from station s has traversed link l or not. The next inequality in (3.11) expresses that the sum of the outgoing traffic from the selected gateways is no more than the gateway capacity (G_m) and the load of the incoming traffic. Note that the outgoing traffic contains the egress traffic to the associated stations with MRAG m ($\boldsymbol{S'_m}$) and the egress traffic that belongs to the reachable stations from MRAG m through other MRAGs over the backhaul network ($\boldsymbol{S''_m}$). The next two equations ensure the flow conservation rule over the mesh backhaul.

Fig. 3.4 illustrates three different cases of the flow conservation rule for gateway (G) and non-gateway (NG) MRAGs and the equations in (3.12) and (3.13) capture these cases. The relation in (3.12) guarantees the total ingress (in) and egress (out) traffic of the flows that traverse a given MRAG m are equal if m is not the selected gateway of the network flows (i.e., $z_{fsm} = 0$). The relation in (3.13) assures a similar situation for the flows that are destined to the associated stations with MRAG m. So, the sum of the incoming traffic (for the flows that are not assigned to m) over the ingress links of m is equal to the total allocated bandwidth to the associated flows with MRAG m. The cases presented in Fig. 3.4 demonstrate a clear picture of these constraints. The next constraint in (3.14) ensures that the total load of all assigned flows to their selected gateway m is no more than its capacity. The relation in (3.15) shows a similar constraint for the egress links of the selected gateways. In fact, it assures that the aggregated load of the assigned flows to gateway m is no more than the total capacity of its egress links. Note that the condition presented in (3.15) is defined for the downlink traffic. For the uplink traffic, the outgoing backhaul links have to be replaced with the incoming links. The next three constraints including relations (3.16), (3.17), and (3.18) ensure that we have a loop-free network by enforcing the flows to traverse each MRAG just once. To do this, we use γ_{fsl} binary variable to ensure when a flow enters/leaves a non-gateway node, i.e., $z_{fsm} = 0$, it is not going to be seen on the ingress/egress links of the same node more than once. Also, for delay-sensitive applications, we have to limit the number of the links traversed by each flow. Thus, we define delay bound d_{fs} for every flow f to station s and its respective constraint is shown in relation (3.19). The next constraint in (3.20) guarantees that the total load of the flows passing through link l is no more than the link capacity (C_l).

As stated before, in MC-MR WMNs, each link can operate on different channels over time. Therefore, the link capacity is a function of its operating channel. Also,

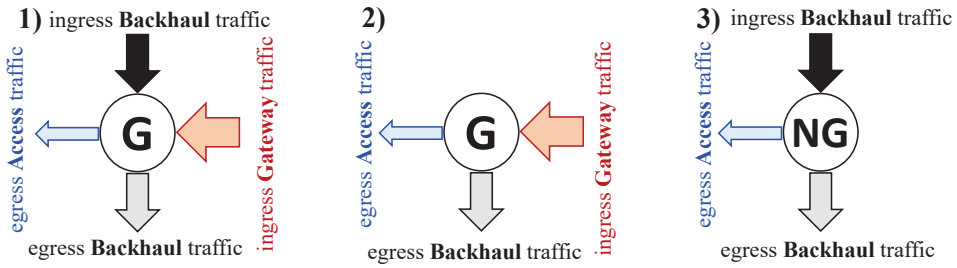


Figure 3.4: Different combinations of the flow conservation rule.

the interference over the backhaul network has to be taken into account by building the MDCG of the network graph. Then, by finding MISs of the created MDCG, we are able to show the relationship between the link capacity and the physical link data rate as given in (3.21) and (3.22). Indeed, these equations solve the optimal capacity planning problem of an MC-MR WMN, and ensure that just one MIS can be active at any time. Suppose that I represents the set of the calculated MISs of the MDCG which is derived from the backhaul topology. So, the first relation in (3.21) expresses that the assigned fraction of time to MIS i ($\alpha_i, i \in \boldsymbol{I}$) is a value in [0, 1], and the sum of these fractions over I is no more than 1. In fact, by the estimation of α_i for every MIS i, we determine its scheduling decision and consequently the capacity of backhaul links. The next constraint in (3.22) utilizes this factor to determine the average capacity of link l under the scheduling decision. In this equation, w_{li}^{c} shows the physical data rate of link l (in bit per second) over channel c within MIS i. So, we can find the capacity of link l (C_l) w.r.t. the impact of co-channel interference. Note that by using the MIS-based scheduling, we find the radio/channel assignment as well as the maximal number of interference-free transmissions. Indeed, the links associated with the LRC tuples (vertices) of the MDCG determine the transmission scheduling decision [49]. Moreover, it conveniently exploits the channel diversity to optimally utilize channels with higher data rates to achieve the maximum link capacity. Note that although finding MISs for a given graph is NP-hard and out of the scope of this work, there are heuristic solutions to find an approximation of MISs [48]. It should be noted that the equations introduced in (3.21) and (3.22) are discussed in Section 2.3.2 as well to implement the similar scheduling strategy for AP association problem presented in Chapter 2.

The next two equations show the gateway selection constraints which are based on CFLP. The constraints in (3.23) and (3.24) assure that each network flow is assigned to only one MRAG as their designated gateway. The next constraint in (3.25) defines

the total number of selected gateways. If this number (N_g) is not selected properly, the optimization problem becomes infeasible and we have to pick a new N_g. Since the illustrated formulation is an MINLP problem, we propose a systematic approach to find a suboptimal solution with an acceptable gap with the optimal outcome in polynomial time. Thus, first to reduce the complexity of the original formulation, we remove variable γ_{fsl} and handle the loop avoidance/delay bound by finding the simple paths between the nodes of the network graph. Then, by applying McCormick envelopes [105] to the equations in (3.10–3.15), we linearize the illustrated formulation similar to the presented technique in [106]. To do this, we used five auxiliary variables to avoid having cubic or quadratic constraints in the optimization problem. These variables including y_{fsm}, h'_{fsm}, h''_{fsm}, k'_{fsml} and k''_{fsml} are shown in Table 3.1. After replacing the variables in the original formulation including the relations in (3.8–3.25), the new MILP problem known as problem $\boldsymbol{A}$, is defined as follows.

$$\text{Maximize} \qquad \sum_{s \in \boldsymbol{S}} \sum_{f \in \boldsymbol{F_s}} b'_{fs} \tag{3.26}$$

$$\sum_{m \in \boldsymbol{M}} \sum_{\substack{l \in \boldsymbol{L} \\ out(m)}} k''_{fsml} \leq b_{fs}, \forall s \in \boldsymbol{S}, \forall f \in \boldsymbol{F_s} \tag{3.27}$$

$$\sum_{\substack{l \in \boldsymbol{L} \\ out(m)}} \sum_{s \in \boldsymbol{S''_m}} \sum_{f \in \boldsymbol{F_s}} k''_{fsml} + \sum_{s \in \boldsymbol{S'_m}} \sum_{f \in \boldsymbol{F_s}} h''_{fsm} \leq G_m \, e_m, \forall m \in \boldsymbol{M} \tag{3.28}$$

$$\sum_{\substack{l \in \boldsymbol{L} \\ in(m)}} \sum_{f \in \boldsymbol{F_s}} (r_{fl} - k'_{fsml}) = \sum_{\substack{l \in \boldsymbol{L} \\ out(m)}} \sum_{f \in \boldsymbol{F_s}} (r_{fl} - k'_{fsml}), \quad \forall m \in \boldsymbol{M}, \forall s \in \boldsymbol{S''_m} \tag{3.29}$$

$$\sum_{\substack{l \in \boldsymbol{L} \\ in(m)}} \sum_{f \in \boldsymbol{F_s}} (r_{fl} - k'_{fsml}) = \sum_{f \in \boldsymbol{F_s}} (b'_{fs} - h'_{fsm}), \quad \forall m \in \boldsymbol{M}, \forall s \in \boldsymbol{S'_m} \tag{3.30}$$

$$\sum_{s \in \boldsymbol{S}} \sum_{f \in \boldsymbol{F_s}} h'_{fsm} \leq G_m \, e_m, \qquad \forall m \in \boldsymbol{M} \tag{3.31}$$

$$\sum_{s \in \boldsymbol{S''_m}} \sum_{f \in \boldsymbol{F_s}} h'_{fsm} \leq \sum_{\substack{l \in \boldsymbol{L} \\ out(m)}} C_l, \qquad \forall m \in \boldsymbol{M} \tag{3.32}$$

$$(9), (20), (21), (22), (23), (24), (25)$$

As illustrated, the relations in (3.27–3.32) are the replacements for the relations in (3.10–3.15) of the original problem. The remaining constraints are the same for both formulations. Also, it should be noted that according to the McCormick technique, after replacing the non-linear constraints using the introduced variables, we have to add several linear constraints for each auxiliary variable to the presented formulation as well. For instance, the following constraints should be applied to the new formulation for the new defined variable, h'_{fsm}. Note that in the constraints shown in (3.33–3.36), U_b and L_b represent the upper and lower bounds of variable b'_{fs}, respectively.

$$0 \leq h'_{fsm} \leq U_b \, z_{fsm}, \quad \forall s \in \boldsymbol{S}, \forall f \in \boldsymbol{F_s}, \forall m \in \boldsymbol{M} \tag{3.33}$$

$$h'_{fsm} \leq b'_{fs} - L_b(1 - z_{fsm}), \forall s \in \boldsymbol{S}, \forall f \in \boldsymbol{F_s}, \forall m \in \boldsymbol{M} \tag{3.34}$$

$$b'_{fs} - U_b(1 - z_{fsm}) \leq h'_{fsm}, \forall s \in \boldsymbol{S}, \forall f \in \boldsymbol{F_s}, \forall m \in \boldsymbol{M} \tag{3.35}$$

$$L_b \, z_{fsm} \leq h'_{fsm}, \forall s \in \boldsymbol{S}, \forall f \in \boldsymbol{F_s}, \forall m \in \boldsymbol{M} \tag{3.36}$$

Similar constraints should be defined for the rest of the introduced auxiliary variables. Then, by relaxing the binary variables to the real numbers, we have a fully linearized problem formulation that can be solved in polynomial time. By using one of the outcomes of the linear formulation, i.e., $\hat{e}_m$ which represents the relaxed e_m, we can pick some of the MRAGs as the potential gateway candidates. Next, we attempt to solve the problem for the selected gateways. Now, by having $\hat{e}_m$, we need to solve the original problem with quadratic constraints including integer (z_{fsm}) and real (b'_{fs}, r_{fl}) variables. Since this problem still is NP-hard, we utilize the introduced technique in [53, 40] to find a 2-approximation of the optimal solution.

According to this approach, we first set variable z_{fsm} to 1 and then we solve the relaxed optimization problem. The relaxed problem ($z_{fsm} = 1$) for the given selected gateways ($\hat{e}_m$), which is called problem $\boldsymbol{B}$ is as follows.

$$\text{Maximize} \qquad \sum_{m \in \boldsymbol{M}} \sum_{s \in \boldsymbol{S}} \sum_{f \in \boldsymbol{F_s}} \hat{b'}_{fsm}\, \hat{e}_m \tag{3.37}$$

$$\text{s.t.} \quad b_{\min} \leq \sum_{m \in \boldsymbol{M}} \hat{b'}_{fsm} \hat{e}_m \leq b_{fs}, \forall s \in \boldsymbol{S}, \forall f \in \boldsymbol{F_s} \tag{3.38}$$

$$\sum_{m \in \boldsymbol{M}} \sum_{\substack{l \in \boldsymbol{L} \\ out(m)}} r_{fl}\, \hat{e}_m \leq b_{fs}, \forall s \in \boldsymbol{S}, \forall f \in \boldsymbol{F_s} \tag{3.39}$$

$$\sum_{\substack{l \in \boldsymbol{L} \\ out(m)}} \sum_{s \in \boldsymbol{S''_m}} \sum_{f \in \boldsymbol{F_s}} r_{fl}\, \hat{e}_m + \sum_{s \in \boldsymbol{S'_m}} \sum_{f \in \boldsymbol{F_s}} \hat{b'}_{fsm}\, \hat{e}_m \leq G_m\, e_m, \forall m \in \boldsymbol{M} \tag{3.40}$$

$$\sum_{s \in \boldsymbol{S}} \sum_{f \in \boldsymbol{F_s}} \hat{b'}_{fsm} \leq G_m\, \hat{e}_m, \qquad \forall m \in \boldsymbol{M} \tag{3.41}$$

$$\sum_{s \in \boldsymbol{S''_m}} \sum_{f \in \boldsymbol{F_s}} \hat{b'}_{fsm} \leq \sum_{\substack{l \in \boldsymbol{L} \\ out(m)}} C_l, \qquad \forall m \in \boldsymbol{M} \tag{3.42}$$

$$(3.20), (3.21), (3.22)$$

Note that at this stage, by relaxing variable z_{fsm}, each flow can be assigned to more than one gateway. However, after finding the final solution, each flow will be assigned to one gateway. Thus, to find the total allocated bandwidth to each flow, we have to replace b'_{fs} with $\hat{b'}_{fsm}$ that represents the partial allocated bandwidth to flow f of station s received from the selected gateway m. Due to this change, the objective function of the relaxed formulation, shown in (3.37), is updated as well. By solving problem $\boldsymbol{B}$, we get $\hat{b'}_{fsm}$ values that can be utilized as the inputs of the original problem to estimate $\hat{z}_{fsm}$ values. Hence, the updated formulation, known as problem $\boldsymbol{C}$ is shown as follows.

$$\text{Maximize} \qquad \sum_{m\in\boldsymbol{M}}\sum_{s\in\boldsymbol{S}}\sum_{f\in\boldsymbol{F_s}} \hat{b'}_{fsm}\,\hat{e}_m\,\hat{z}_{fsm} \tag{3.43}$$

$$\sum_{m\in\boldsymbol{M}}\sum_{\substack{l\in\boldsymbol{L}\\ out(m)}} r_{fl}\,\hat{e}_m\,\hat{z}_{fsm} \le b_{fs}, \forall s\in\boldsymbol{S}, \forall f\in\boldsymbol{F_s} \tag{3.44}$$

$$\sum_{\substack{l\in\boldsymbol{L}\\ out(m)}}\sum_{s\in\boldsymbol{S''_m}}\sum_{f\in\boldsymbol{F_s}} r_{fl}\,\hat{e}_m\,\hat{z}_{fsm} +$$
$$\sum_{s\in\boldsymbol{S'_m}}\sum_{f\in\boldsymbol{F_s}} \hat{b'}_{fsm}\,\hat{e}_m\,\hat{z}_{fsm} \le G_m\,\hat{e}_m, \forall m\in\boldsymbol{M} \tag{3.45}$$

$$\sum_{\substack{l\in\boldsymbol{L}\\ in(m)}}\sum_{f\in\boldsymbol{F_s}} r_{fl}\,(1-\hat{z}_{fsm}) = \sum_{\substack{l\in\boldsymbol{L}\\ out(m)}}\sum_{f\in\boldsymbol{F_s}} r_{fl}\,(1-\hat{z}_{fsm}),$$
$$\forall m\in\boldsymbol{M}, \forall s\in\boldsymbol{S''_m} \tag{3.46}$$

$$\sum_{\substack{l\in\boldsymbol{L}\\ in(m)}}\sum_{f\in\boldsymbol{F_s}} r_{fl}\,(1-\hat{z}_{fsm}) = \sum_{f\in\boldsymbol{F_s}}\sum_{m\in\boldsymbol{M}} \hat{b'}_{fsm}\,\hat{e}_m(1-\hat{z}_{fsm}),$$
$$\forall m\in\boldsymbol{M}, \forall s\in\boldsymbol{S'_m} \tag{3.47}$$

$$\sum_{s\in\boldsymbol{S}}\sum_{f\in\boldsymbol{F_s}} \hat{b'}_{fsm}\,\hat{z}_{fsm} \le G_m\,\hat{e}_m, \qquad \forall m\in\boldsymbol{M} \tag{3.48}$$

$$\sum_{s\in\boldsymbol{S''_m}}\sum_{f\in\boldsymbol{F_s}} \hat{b'}_{fsm}\,\hat{z}_{fsm} \le \sum_{\substack{l\in\boldsymbol{L}\\ out(m)}} C_l, \qquad \forall m\in\boldsymbol{M} \tag{3.49}$$

$$\sum_{m\in\boldsymbol{M}} \hat{z}_{fsm} = 1, \qquad \forall s\in\boldsymbol{S}, \forall f\in\boldsymbol{F_s} \tag{3.50}$$

$$\hat{z}_{fsm} \le \hat{e}_m, \qquad \forall s\in\boldsymbol{S}, \forall f\in\boldsymbol{F_s}, \forall m\in\boldsymbol{M} \tag{3.51}$$

$$(3.20), (3.21), (3.22)$$

By using the acquired results ($\hat{z}_{fsm}$), and building a bipartite graph as illustrated in Fig. 2.2 and explained in [53], we can find a 2-approximation of the optimal mapping

Algorithm 3 The suboptimal solution for the joint problem

1: $lb \leftarrow \max\{\frac{b_{\min}|S|}{G_m}$, Min No. of MRAGs w.r.t. defined $d_{fs}\}$.
2: $ub \leftarrow |M|$.
3: **while** $ub > (1+\delta)lb$ **do**
4: $N_g \leftarrow \frac{lb+ub}{2}$.
5: $\hat{e}_m \leftarrow$ Solve the joint problem $\boldsymbol{A}$ for $\lceil N_g \rceil$.
6: $\boldsymbol{PGC} \leftarrow$ Prune and Sort the acquired $\hat{e}_m$.
7: **for** T times run the following steps **do**
8: $\boldsymbol{GC} \leftarrow$ Select N_g elements from $\boldsymbol{PGC}$, randomly.
9: **if** the selected $\boldsymbol{GC}$ satisfies the delay constraint **then**
10: $\hat{z}_{fsm} \leftarrow$ Solve the problems $\boldsymbol{B}$ and $\boldsymbol{C}$ for $\boldsymbol{GC}$.
11: $\bar{z}_{fsm} \leftarrow GatewayAssignment(\hat{z}_{fsm})$.
12: $\boldsymbol{SP} \leftarrow$ Find all *simple paths* with length $\leq d_{fs}$.
13: $\boldsymbol{IL} \leftarrow$ Build set of *ingress* links using $\boldsymbol{SP}$.
14: $\boldsymbol{EL} \leftarrow$ Build set of *egress* links using $\boldsymbol{SP}$.
15: Solve the *original* problem for $\hat{e}_m$, $\bar{z}_{fsm}$, $\boldsymbol{IL}$, $\boldsymbol{EL}$.
16: **if** the *original* problem is $feasible$ **then**
17: $Result \leftarrow$ Estimated *aggregated throughput*.
18: **if** $|Result - MaxCapacity| \leq \epsilon$ **then**
19: $ub \leftarrow N_g$.
20: *go to* line 3.
21: **else**
22: *go to* line 7.
23: **end if**
24: **end if**
25: **end if**
26: **end for**
27: $lb \leftarrow N_g$.
28: **end while**
29: Consider N_g as the number of the required gateways ($N_g = \lceil N_g \rceil$).

solution from the set of flows ($\boldsymbol{F}$) to the set of our selected gateways. Now, by having the selected gateways ($\hat{e}_m$) and the map of the flow assignments ($\hat{z}_{fsm}$), we can solve the original MINLP problem and find the suboptimal allocated bandwidth to the flows over the backhaul. In the next step, we present an algorithm to solve the delineated problem through a systematic approach. A summary of the used symbols in the formulation is shown in Table 3.1.

Table 3.1: Utilized parameters in the problem formulation

Symbol	Description
$\boldsymbol{M}$	The set of all Mesh Router/AP/Gateway nodes (MRAG)
$\boldsymbol{S}$	The set of all Stations
$\boldsymbol{F_s}$	The set of the Flows that belong to station $s \in S$
$\boldsymbol{S'_m}$	The set of the associated stations to $m \in M$
$\boldsymbol{S''_m}$	The set of the reachable stations from $m \in M$
$\boldsymbol{C}$	The set of access Channels
$\boldsymbol{L}$	The set of Links of the given network graph
$\boldsymbol{I}$	The set of calculated MISs of a given MDCG
x^c_{sm}	1, if station s is associated to m on channel c
r^c_{sm}	Link data rate between station s and MRAG m on channel c
t^c_{fsm}	Airtime of flow f (station s) associated to m on channel c
p_{fs}	The priority (weight) of flow f from station s
v_{sm}	1, if m is placed within the communication range of s
q^c_{sm}	1, if m is placed within the sensing range of s on channel c
$b_{\min}$	Min guaranteed bandwidth for all the flows on the backhaul
b_{fsm}	Throughput of flow f from station s in the access layer
b'_{fs}	Throughput of flow f from station s in the the backhaul
d_{fs}	Delay bound of flow f from station s in the backhaul
d'_{fs}	Demand of flow f from station s
r_{fsl}	Allocated rate to flow f from station s on backhaul link l
γ_{fsl}	1, if flow f from station s has traversed link l
z_{fsm}	1, if flow f from station s assigned to MRAG m (gateway)
e_m	1, if m is selected as a gateway
C_l	Capacity of backhaul link l
w^c_{li}	Data rate of link l on channel c in independent set $i \in I$
α_i	Fraction of time that is allocated to independent set $i \in I$
G_m	Capacity of node m as a gateway
N_g	Number of the selected gateways
y_{fsm}	Auxiliary variable for linearization ($y_{fsm} = e_m\, z_{fsm}$)
h'_{fsm}	Auxiliary variable for linearization ($h'_{fsm} = b'_{fs}\, z_{fsm}$)
h''_{fsm}	Auxiliary variable for linearization ($h''_{fsm} = b'_{fs}\, y_{fsm}$)
k'_{fsml}	Auxiliary variable for linearization ($k'_{fsml} = r_{fsl}\, z_{fsm}$)
k''_{fsml}	Auxiliary variable for linearization ($k''_{fsml} = r_{fsl}\, y_{fsm}$)
$\hat{z}_{fsm}$	Relaxed binary variable z_{fsm} ($0 \leq \hat{z}_{fsm} \leq 1$)
$\hat{e}_m$	Relaxed binary variable e_m ($0 \leq \hat{e}_m \leq 1$)
U_b	The upper bound of b'_{fs} used in h'_{fsm} and h''_{fsm}
U_r	The upper bound of r_{fsl} used in k'_{fsml} and k''_{fsml}
L_b	The lower bound of b'_{fs} used in h'_{fsm} and h''_{fsm}
L_r	The lower bound of r_{fsl} used in k'_{fsml} and k''_{fsml}

3.4.3 The Algorithm for the Joint Optimization Problem

The proposed algorithm to find the least number of required gateways is demonstrated in *Algorithm 3*. This algorithm finds a suboptimal value by searching over an interval $[lb, ub]$ with variable bounds. In fact, it carries out an iterative process that checks the feasibility of the optimization problem and adjusts the bounds of the search interval. At the beginning, we make an initial guess ($\frac{lb+ub}{2}$) for the optimal N_g, and we shrink the search space in each iteration. Note that the initial value of lb is calculated based on the predetermined values including the minimum guaranteed bandwidth ($b_{\min}$), number of stations ($|\boldsymbol{S}|$), gateway capacity (G_m) and delay bound (d_{fs}). In fact, we first calculate the number of required gateways with the assumption of having unlimited capacity in the backhaul ($\frac{b_{\min}|\boldsymbol{S}|}{G_m}$). Then, we find the minimum number of required gateways to cover all the nodes of the graph w.r.t. the given d_{fs}.

Finally, lb is set to the maximum of these values and ub is initialized to $|\boldsymbol{M}|$ as its maximum value. Inside the `while` loop, we first solve the joint problem $\boldsymbol{A}$ for the given N_g. Then, the sorted results obtained for $\hat{e}_m$ are used as the inputs of the function which is in charge of selecting the Potential Gateway Candidates (***PGC***). Now, we have to keep looking for a set of Gateway Candidates (***GC***) out of ***PGC*** that leads to a feasible solution with the minimum number of required gateways. Thus, we use the second loop to check different random combinations of ***GC***. In this case, since the time complexity of finding all the possible combinations is $O(n!)$, we limit the random selection of distinct ***GC*** sets (out of ***PGC***) to T times. It should be noted that for the random selection of ***GC*** elements from ***PGC***, we consider a higher priority for the MRAGs that are located on the boundary of the given topology. This heuristic is utilized according to the presented results in [82]. In each iteration of the `for` loop, we first initialize ***GC*** to N_g randomly selected elements from ***PGC***. In the next step, by using a feasible ***GC***, we solve problem $\boldsymbol{B}$ and problem $\boldsymbol{C}$ to find an estimation of $\hat{z}_{fsm}$. The outcome is going to be used by `GatewayAssignment` function to find a feasible assignment of the network flows to the ***GC***.

For the gateway assignment process, we first start with the flows that have only one gateway choice and then expand the process to the remaining flows w.r.t. the loads of their reachable ***GC***s. It means, if a flow has multiple reachable gateway candidates, it picks the one with the minimum number of assigned flows. Then, for the given flows over the network graph, we find the set of simple paths (***SP***), i.e., the paths with no loop between their assigned gateways (as source) and the APs that

are associated with (as destination). Now, we can build the sets of ingress ($\boldsymbol{IL}$) and egress ($\boldsymbol{EL}$) links for every node of the graph using $\boldsymbol{SP}$. These sets have been used in the majority of the constraints in the original formulation. By using the outcomes of the prior steps, now we are able to solve the original problem. If the problem is feasible, we then measure the utilization of the selected/assigned gateways by the estimation of the aggregated throughput (*Result*). By checking the gap (ϵ) between the aggregated throughput and the maximum capacity of the selected gateways, we determine whether to change the number of selected gateways (N_g) or try another set of $\boldsymbol{GC}$ out of the current $\boldsymbol{PGC}$. This process continues until the lower bound reaches the upper bound. Finally, $\lceil N_g \rceil$ is used as the number of the required gateways.

In fact, by solving the introduced problems, i.e., $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$, and checking the feasibility of the original joint optimization problem for the obtained results over several iterations, we have an estimation of the optimal required gateways (N_g). In addition, we acquire the allocated rate to each flow f over the links of the given topology (r_{fl}) to maximize the objective function. Note that variable δ used in the first `while` loop is a small number to control the time complexity of the algorithm.

As mentioned in Section 3.3.1, after running the presented solution (*Algorithm 3*), the new network configurations should be applied to the MRAGs through either an in-band or out-band control network. Note that the frequency of running the proposed algorithm is a function of the traffic volume and the number of newly generated (or joined) traffic flows. To reduce the overhead of control traffic in dynamic scenarios (when stations join/leave frequently), we can use clustering/grouping techniques for the joint gateway selection and flow routing problem to ensure the scalability of our solution as well. In fact, this strategy alleviates the possibility of encountering long convergence intervals after each run of the algorithm. Also, since the introduced scheme can limit the rate of uplink and downlink traffic flows, the functionality and stability of the system will not be affected by the greedy Wi-Fi stations.

3.4.4 Time Complexity Analysis of the Algorithm

Since the introduced optimization problems ($\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$) are LP problems, we can solve them using the Simplex method in polynomial time [107, 108]. This feature enables us to find a suboptimal solution using *Algorithm 3* in polynomial time. To prove this claim, we assess the time complexity of each part of the algorithm and calculate its overall time complexity. First, w.r.t. the presented details in Section 3.4.2,

the optimization problems can be solved in $O(N^k)$ time, where N depends on the number of input variables and the number of constraints, and k is a positive constant. For the original problem formulation presented in Section 3.4.2, the number of variables and the number of constraints are $|\boldsymbol{S}||\boldsymbol{F_s}|(2|\boldsymbol{M}|^2 - |\boldsymbol{M}| + 1) + |\boldsymbol{M}|^2 + |\boldsymbol{I}|$ and $4|\boldsymbol{S}||\boldsymbol{F_s}|(|\boldsymbol{M}| + 1) + |\boldsymbol{M}|(2|\boldsymbol{S}| + 2|\boldsymbol{M}| + 1) + |\boldsymbol{I}| + 1$, respectively. It can be seen that N is a function of the number of stations, network flows, MRAGs and the MISs.

Now, we walk through the different steps of the algorithm. The main `while` loop iterates $\log_2(\frac{ub-lb}{\delta.lb})$ times and the first three lines inside it can be done in $O(1)$, $O(N^k)$, and $O(N_g \log N_g)$ time, respectively. The second loop iterates for T times and it has to pick N_g random elements from $\boldsymbol{PGC}$. By using the Fisher-Yates shuffling algorithm [109], this task can be done in $O(N_g)$ time. Then, if the selected $\boldsymbol{GC}$ satisfies the delay constraint (d_{fs}) for all the flows, we have to run a sequence of operations. As it mentioned, the optimization and the gateway assignment problems can be solved in $O(N^k)$ time. Next, the set of all simple paths with the length no more than d_{fs} for the given network flows/graph can be constructed in $O(|\boldsymbol{F}|(b\ d_{fs})^{d_{fs}})$ time. Note that $|\boldsymbol{F}|$, b and d_{fs} represent the total number of network flows, branching factor (the average number of neighbors of a vertex), and the maximum depth of the search process (using the DFS algorithm), respectively. It can be seen that for the large network graphs, increasing d_{fs} has a drastic impact on the overall time complexity of our algorithm.

By using the set of the calculated simple paths ($\boldsymbol{SP}$), we find the set of ingress ($\boldsymbol{IL}$) and egress ($\boldsymbol{EL}$) links in $O(|\boldsymbol{M}||\boldsymbol{L}|)$ time. $|\boldsymbol{M}|$ and $|\boldsymbol{L}|$ denote the number of MRAGs and the number of links in the network graph, respectively. Now, we can check the feasibility of the original problem using the obtained inputs from the prior steps in polynomial time. If the problem is feasible, then we calculate the aggregated throughput and find its gap with the $MaxCapacity$ (defined as $N_g G_m$). Since all the operations of the inner loop are done in $O(TN^k)$ time, the overall time complexity of *Algorithm 3* is $O(TN^k \log_2(\frac{ub-lb}{\delta.lb}))$, which is an order of polynomial series.

3.4.5 Multi-Path vs. Single-Path Flow Routing

Our formulation in Section 3.4.2 presents a multi-path flow routing over the backhaul network regarding the defined constraints. So, the packets belong to the same flow can take different routes to reach their destination. However, as it is discussed in Section 3.2.3, the implementation of multi-path flow routing over MC-MR WMNs

brings extra complexities and to avoid them, we may use a single-path flow routing as an alternative. In this case, we need to convert the general form of MCFP (as an LP problem) to an MILP problem (which is NP-hard) to ensure all the packets belong to the same flow take the same path. To deal with this complexity of single-path flow routing to find a solution that guarantees certain performance bounds, we introduce a randomized single-path flow routing algorithm. Before applying this algorithm to a given scenario, we need to solve the multi-path flow routing problem. To have a clear picture about the proposed algorithm, without loss of generality, we focus on a simplified case of the presented formulation in Section 3.4.2, where the gateway nodes and the flow assignments (to the given gateways) are predetermined. Therefore, we can formulate the multi-path flow routing and link scheduling on the mesh backhaul as an LP problem shown as follows.

$$\text{Maximize} \qquad \sum_{s \in \boldsymbol{S}} \sum_{f \in \boldsymbol{F_s}} b'_{fs} \tag{3.52}$$

$$\text{s.t.} \quad b_{\min} \leq b'_{fs} \leq d'_{fs}, \qquad \forall s \in \boldsymbol{S}, \forall f \in \boldsymbol{F_s} \tag{3.53}$$

$$\sum_{\substack{l \in \boldsymbol{L} \\ out(m)}} r_{fsl} \leq d'_{fs}, \qquad \forall s \in \boldsymbol{S}, \forall f \in \boldsymbol{F_s}, \forall m \in \boldsymbol{M} \tag{3.54}$$

$$\sum_{\substack{l \in \boldsymbol{L} \\ out(g)}} \sum_{s \in \boldsymbol{S}} \sum_{f \in \boldsymbol{F_s}} r_{fsl} \leq G_g, \qquad \forall g \in \boldsymbol{G} \tag{3.55}$$

$$\sum_{\substack{l \in \boldsymbol{L} \\ in(m)}} \sum_{s \in \boldsymbol{S}} \sum_{f \in \boldsymbol{F_s}} r_{fsl} = \sum_{\substack{l \in \boldsymbol{L} \\ out(m)}} \sum_{s \in \boldsymbol{S}} \sum_{f \in \boldsymbol{F_s}} r_{fsl}, \quad \forall m \in \boldsymbol{M} \tag{3.56}$$

$$\sum_{\substack{l \in \boldsymbol{L} \\ in(m)}} \sum_{f \in \boldsymbol{F_s}} r_{fsl} = \sum_{f \in \boldsymbol{F_s}} b'_{fs}, \qquad \forall s \in \boldsymbol{S} \tag{3.57}$$

$$(3.20), (3.21), (3.22)$$

The notations of this formulation are based on the ones illustrated in Table 3.1. So, similar to the term in (3.8), the objective function in (3.52) maximizes the total allocated bandwidth to the flows over a given mesh network. The first constraint

in (3.53) assures that the bandwidth of each flow (b'_{fs}) is no less than a guaranteed bandwidth ($b_{\min}$) and no more than its demand (d'_{fs}). The second constraint in (3.54) ensures that the allocated bandwidth to flow f belongs to station s (r_{fsl}) over the outgoing links of its assigned gateway g is no more than its demand. The next relation in (3.55) expresses that the sum of the outgoing traffic from the selected gateways is no more than the gateway capacity (G_g). The next two equations ensures the flow conservation. The relation in (3.56) guarantees the aggregated ingress and egress traffic flows over the links of any given MRAG m are equal. The relation in (3.57) assures the similar situation for the flows that are destined to the stations connected to MRAG m. The next three constraints consider the link capacity and traffic scheduling as explained in Section 3.4.2. As it is discussed in Section 3.4.4, the shown LP problem can be solved in polynomial time. In the next section, we introduce our randomized single-path flow routing algorithm that utilizes the outcomes of multi-path flow routing to find a bounded solution in polynomial time. It is important to note that the proposed algorithm relies on SDN technology to be employed over MC-MR WMNs.

3.4.6 Randomized Single-Path Flow Routing Algorithm

As it is mentioned before, a multi-path flow routing can be converted to a single-path routing problem by adding an extra binary variable to the shown LP formulation. However, it makes an NP-hard problem that cannot be solved in polynomial time. Hence, to avoid the complexities of implementing a multi-path flow routing as well as the NP-hardness of single-path routing formulation, we present a randomized single-path flow routing that delivers all packets belonging to the same flow over a single path. The proposed algorithm that stems from one of Motwani's work presented in Chapter 11 of [110] guarantees a bounded solution with less complexity. Note that many heuristic solutions for this problem are possible, but not all of them can guarantee a solution with a certain probability. So, by having the traffic information and the calculation of the optimal routing during periodic time intervals (regarding the channel conditions), it is possible to dynamically adjust the forwarding rules at MRAG nodes. Fig. 3.5 illustrates the big picture of the proposed algorithm.

It can be seen that two main components of the randomized single-path flow routing solution are the flow selection and the flow routing functions. The former does the random flow selection based on flipping a biased coin and the latter picks

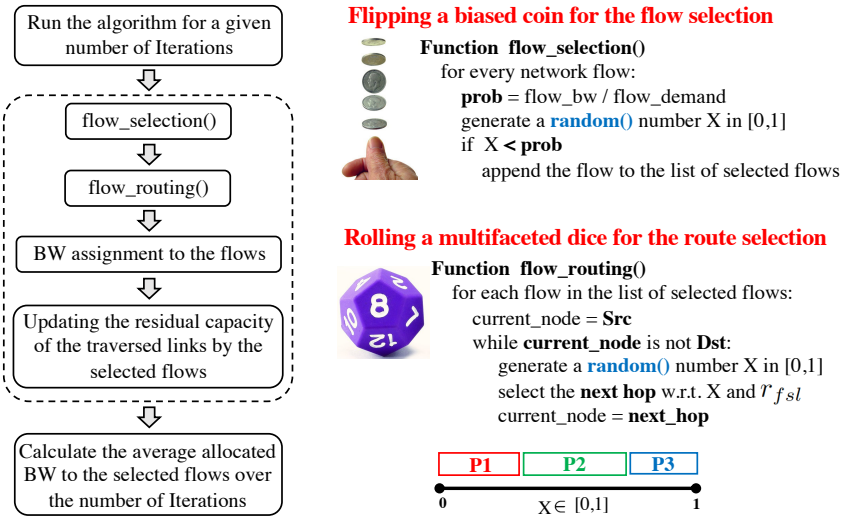


Figure 3.5: The main components of randomized single-path flow routing solution.

the next hop (at each node of a given topology) through rolling a multifaceted dice. In both functions, we use the results obtained from multi-path flow routing problem including b'_{fs} and r_{fsl}. Indeed, in flow selection function, the flow bandwidth (b'_{fs}) and its demand (d'_{fs}) have been utilized in the process of random flow selection. A similar approach can be taken for a random walk over a given topology graph using the assigned bandwidth to the network flows over different edges (r_{fsl}) of the topology. As it is shown in Fig. 3.5, these values should be converted to a set of probabilities, which are in the range [0, 1]. Also, due to the randomized nature of our algorithm, we need to run it for a large number of iterations. This process is displayed as a part of the flowchart shown in Fig. 3.5.

Now, we elaborate our randomized single-path flow routing algorithm in detail. *Algorithm 4* shows the required steps for routing a set of flows over the backhaul network. As the first step of the algorithm (*Line 1*), the capacity of all the links is divided by $(1 + \epsilon)$, where is an approximation ratio. Since every time we choose a link for one of the flows and the capacity of the respective link will be reduced, we need to keep track of the Residual Capacity of each link (RC_l). So, we initialize RC_l to the capacity of the respective links (*Line 2*). Now, by having the acquired results from the LP problem shown in Section 3.4.5 and using an iterative process, we are able to solve the single-path flow routing problem.

The first step of this iterative process is flow selection through flipping a biased coin (*Line 4*). The probability of selecting a given flow f is proportional to its

Algorithm 4 Randomized single-path flow routing (RSR)

1: $C_l \leftarrow (\frac{1}{1+\epsilon})\ C_l, \forall l \in L$.
2: Residual Capacity $(RC_l) \leftarrow (C_l), \forall l \in L$.
3: **for** $i = 1 : Iterations$ **do**
4: $SelectedFlows \leftarrow FlowSelection(F, b'_{fs}, d_{fs})$.
5: $RoutedFlows \leftarrow FlowRouting(SelectedFlows, b'_{fs})$.
6: **for** every flow $f \in SelectedFlows$ **do**
7: **for** every $SelectedRoute_f$ in $RoutedFlows$ list **do**
8: $minRate = b'_{fs}$.
9: **for** every link $l \in SelectedRoute_f$ **do**
10: **if** $minRate > \min(b'_{fs}, RC_l)$ **then**
11: $minRate \leftarrow \min(b'_{fs}, RC_l)$.
12: **end if**
13: **end for**
14: **end for**
15: Add $minRate$ as the rate of flow f into $FlowRates$.
16: **for** every link $l \in SelectedRoute_f$ **do**
17: $RC_l \leftarrow RC_l - minRate$.
18: **end for**
19: **end for**
20: $FlowRatesPerIteration_i \leftarrow FlowRates$.
21: **end for**
22: **for** every flow $f \in FlowRatesPerIteration$ **do**
23: $AFRate \leftarrow AvgRate_f$ over the $Iterations$.
24: Add $AFRate$ to the $AvgAFRates$ list.
25: **end for**

backhaul bandwidth (b'_{fs}) over its demand (d'_{fs}). This means if we assume all the flows have the same demand, the ones with larger b'_{fs} have a higher chance of being selected. This fact can be seen in *Algorithm 5* that shows the flow selection process. To simulate the coin flipping process for every flow in the *FlowSelection* function, we first generate a random number called Z in [0, 1]. For flow f from station s, if $\frac{b'_{fs}}{d'_{fs}}$ is greater than Z, then we add the flow into the *SelectedFlows* list.

Next, we need to route the selected flows over the backhaul network. This task is carried out by the *FlowRouting* function shown in *Algorithm 6*. In this algorithm, we have to fully serve the selected flows. If there is not enough bandwidth to serve the demand of a flow, we use the residual link capacity. For every selected flow f, we start by traversing the network graph from its assigned gateway node which is predetermined. This process continues until the flow reaches the MRAG whose

Algorithm 5 Flow selection function (FlowSelection)

1: **for** $f = 1 : |F|$ **do**
2: $Z \leftarrow$ Generate a random number in range [0, 1].
3: $Prob \leftarrow \frac{b'_{fs}}{d'_{fs}}$.
4: **if** $Prob > Z$ **then**
5: Add flow f into *SelectedFlows* list.
6: **end if**
7: **end for**
8: **return** *SelectedFlows*.

Algorithm 6 Flow routing function (FlowRouting)

1: $CurrentNode \leftarrow Gateway$
2: **for** $f = 1 : |SelectedFlows|$ **do**
3: **while** $CurrentNode$ is not *Destination* node **do**
4: **for** $l = 1 : |L|$ **do**
5: **if** $l.src == CurrentNode$ **and** $r_{fl} > 0$ **then**
6: Add link l into $CLinks$ list.
7: Add r_{fl} into $CRates$ list.
8: **end if**
9: **end for**
10: **if** $Sum(CRates)$!= 0 **then**
11: **for** $r = 1 : |CRates|$ **do**
12: $r_{fl} \leftarrow CRates[r]$
13: $Prob \leftarrow r_{fl}$ / $Sum(CRates)$
14: Add $Prob$ into $LinkSP$ list.
15: **end for**
16: $NextNode = RRounding(LinkSP, CLinks)$
17: CurrentNode = NextNode
18: Add $CurrentNode$ into *SelectedRoute* of f.
19: **end if**
20: **end while**
21: Add *SelectedRoute* for flow f into *RoutedFlows* list.
22: **end for**
23: **return** *RoutedFlows*

corresponding station is associated with (*Destination* node). Note that at every MRAG, flow f should make a decision to pick its next hop. At the first step of the decision making process, a list of the candidate links for flow f at its current node is created. The sender (source) of all the candidate links (shown by $l.src$ in *Algorithm 6*) is the current node. By using the estimated r_{fsl} values for the egress

Algorithm 7 Randomized routing function by rolling a multifaceted dice

```
1: X ← Generate a random number in range [0, 1]
2: LowerBound ← 0
3: UpperBound ← LinkSelectionProb[0].
4: for i = 1 : |LinkSelectionProb| do
5:     if LowerBound ≤ X ≤ UpperBound then
6:         NextNode ← Ending node of the link i from CandidateLinks list.
7:         return NextNode
8:     end if
9:     LowerBound ← UpperBound
10:    if i != |LinkSelectionProb| - 1  then
11:        UpperBound ← UpperBound + LinkSelectionProb[i + 1]
12:    end if
13: end fornextnode = L[CandidateLinks[i]][1]
```

links of the current node, we can find the probability of selecting a link l from the candidate list.

As mentioned earlier, the randomized process is very similar to rolling a multifaceted dice where each facet represents one of the links of the candidate list. In fact, it is just like traversing the vertices of a directed graph such that at each vertex you have to roll a dice to choose the next vertex to visit. At each vertex, the number of facets is equal to the number of egress edges of the vertex and it might vary from one vertex to another one. In addition, the probability of selecting any facet (or egress edge) is proportional to its weight, which is the allocated bandwidth to the link that represents the selected facet. This process is represented by *RRounding* function in *Algorithm 6*. In fact, the probability of picking a link is proportional to the data rate of flow f on the selected link l, from the multi-path routing result. Thus, this process decides the next MRAG that flow f should take to reach its final destination. The entire process of randomized route selection is shown in *Algorithm 7*. Now, we can initiate the next step which is finding the allocated bandwidth to the selected flows. In this step, we pick the selected flows one by one and traverse the selected route for them ($SelectedRoute_f$). For each selected flow f at link l, we keep track of its allocated bandwidth, which is the minimum of b_{fs} and RC_l (shown as *minRate*).

By finding the *minRate* value for all the links that flow f has traversed, we get the allocated bandwidth to this flow and the corresponding value is stored in the *FlowRates* list. It is important to note that the utilized bandwidth by flow f should be deducted from the residual capacity of all the traversed links. The record of

all the estimated *FlowRates* at each iteration is kept in the *FlowRatesPerIteration* list. Finally, after running the single-path flow routing for the predefined number of *Iterations*, it is possible to calculate the average allocated flow rate of every selected flow f over the total runs of the algorithm. The final results are stored in the Average Allocated Flow Rates (*AvgAFRates*) list.

It should be noted that the input parameters of the original problem formulation introduced in Section 3.4.2 can affect the outcomes of the presented algorithm in this section. It is based on the fact that the LP problem presented in Section 3.4.5 stems from the original MINLP problem and its outputs are used as the inputs of *Algorithm 4*. Therefore, the selected values for the input parameters such as the minimum guaranteed bandwidth, gateway capacity and physical data rate of backhaul links may impact b'_{fs}, r_{fsl} and C_l, which play key roles in the randomized flow routing.

The overall time complexity of *Algorithm 4* is a function of several key parameters including the number of iterations (T), the number of network flows ($|\boldsymbol{F}|$), and the number of edges in the network graph ($|\boldsymbol{L}|$). Regarding the illustrated steps in the algorithm, the overall time complexity of our randomized single-path flow routing is $O(T(|\boldsymbol{F}|+|\boldsymbol{F}||\boldsymbol{L}|+|\boldsymbol{F}||\boldsymbol{L}|)+|\boldsymbol{F}|)$ that can be shortened as $O(|\boldsymbol{F}|(T+2|\boldsymbol{L}|T+1))$. By considering the fact that single-path MCFP is a known NP-hard problem that cannot be solved in polynomial time, it can be seen that the randomized process of flow/link selection has a significant impact on improving the performance of the algorithm in terms of the running time [110]. In the next section, we discuss the bound analysis of the algorithm.

3.4.7 Bound Analysis of the Randomized Algorithm

As it is shown in *Algorithm 4*, at the first step, we decrease the capacity of wireless links by $\frac{1}{1+\epsilon}$ factor (i.e., $\widetilde{C}_l = \frac{C_l}{1+\epsilon}$), where $0 \leq \epsilon \leq 1$. The main reason behind this step is reducing the chance of violating the link capacity constraint. Also, in [111], the researchers calculated the probability of violating the capacity constraint as $\Pr[X \geq C_l] = \Pr[X \geq (1+\epsilon)\widetilde{C}_l]$, where X is the aggregated bandwidth of the flows that traversed link l, so that a Chernoff bound can be derived. However, it was under a strong assumption which is $(\frac{c_{\min}}{\Delta_{\max}.d'_{\max}}) \geq (\frac{4.2+\epsilon}{\epsilon^2})(1+\epsilon)\ln|\boldsymbol{L}|$. In this assumption, $c_{\min}$, $d'_{\max}$ and $\Delta_{\max}$ represent the minimum capacity, maximum demand, and graph diameter, respectively. According to this formula, $c_{\min}$ must be a large number (at least hundred times of $d'_{\max}$) to meet the assumption and it cannot be applied to many

scenarios with a limited link capacity. Therefore, to avoid such a strong assumption, we consider a general form of the Chernoff bound [112]. It should be noted that we can apply Chernoff bound to our randomized solution since the flow selection and routing events are independent. Moreover, in the Chernoff bound formulation, the probability of being far away from the average value decreases exponentially. So, we define a random variable X as the sum of independent Poisson trials that represents the total benefit (i.e., allocated bandwidth) of an event and $\mu = E[X]$. So, for $\delta > 0$, we can write the following

$$\Pr[X \geq (1+\delta)\mu] \leq (\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}})^{\mu}, \tag{3.58}$$

and for the specific cases, where $0 < \delta < 1$, we have

$$\Pr[X \leq (1-\delta)\mu] \leq (\frac{e^{-\delta}}{(1-\delta)^{(1-\delta)}})^{\mu}. \tag{3.59}$$

Note that since we are applying the general form of the Chernoff bound, we can set ϵ to 0 (at the first step of the algorithm), especially when achieving the optimal benefit (aggregated bandwidth) is our first priority. After obtaining the optimal (fractional) solution in Section 3.4.5, we can take two separate strategies for the single-path flow routing. The first one is Simple Single-path flow Routing (SSR) and the second one is called Randomized Single-path flow Routing (RSR), as introduced in *Algorithm 4*.

In SSR, we intend to avoid any violation of the link capacity constraint. Thus, for a flow with a multi-path routing choice, we select the (single) path that provides the highest bandwidth from the optimal solution and serve whatever is already assigned to the flow on the selected route. This algorithm could be useful when there is only a small number of multi-path flows in the fractional solution. Note that although it is possible to lose a portion of the bandwidth assigned to the multi-path flows, this loss is insignificant due to the small number of these flows in such scenarios. In fact, according to the obtained results for different scenarios, we have noticed that most of the flows use a single-path to traverse the network graph and only a small number of them were routed through multiple paths.

On the other hand, to achieve the maximum benefit, we use the second approach (RSR), which may violate the link capacity constraint, so such a probability should be bounded. There is a certain chance that RSR achieves a higher bandwidth than

SSR. Although RSR might also reach a less aggregated bandwidth than SSR, RSR serves the demands of the flows in a fair fashion. As explained in *Algorithm 4*, RSR randomly selects the network flows to serve them in each scheduling period, so the violation of link capacity constraint can be evened out. Also, it should be noted that to alleviate the overhead of flow switching during different iterations of the algorithm, we can use a traffic (flow) serialization technique similar to the one that is introduced in [113]. In fact, it can be done by grouping the flows during the scheduling process to reduce the number of switching operations.

As it is shown, the probability of flow selection is equal to $\frac{b'_{fs}}{d'_{fs}}$ and after the flow selection process in RSR, the entire b'_{fs} should be served through the single-path routing. For RSR, we can calculate two Chernoff bounds. The first bound is used to have an estimation of the total benefit that can be defined as the aggregated bandwidth. To calculate it, we define X as the total benefit obtained from the RSR algorithm and (3.60) shows the probability of the event that the total benefit is less than $(1-\delta)\mu$, where $0 < \delta < 1$.

$$\begin{aligned} \Pr[X \leq (1-\delta)\mu] &= \Pr[X \leq (1-\delta)E[X]] \\ &= \Pr[X \leq (1-\delta)\sum_{s\in S}\sum_{f\in F_s} b'_{fs}(\frac{b'_{fs}}{d'_{fs}})] \\ &\leq (\frac{e^{-\delta}}{(1-\delta)^{(1-\delta)}})^{\mu}. \end{aligned} \tag{3.60}$$

In (3.60), b'_{fs} refers to the (optimal) bandwidth assigned to flow f from station s. The term inside the summation represents the expectation of the assigned bandwidth to flow f from station s, which is served through either the single-path or multi-path routing. Hence, $\sum_{s\in S}\sum_{f\in F_s} b'_{fs}(\frac{b'_{fs}}{d'_{fs}})$ shows the expectation of the total assigned bandwidth to all the flows. In addition to the first bound, to determine the probability of violating the link capacity constraint, we need to calculate a second bound. So, we define another random variable (Y) as the aggregated bandwidth of the flows that pass through a specific link of the given topology. Thus, the probability of the aggregated bandwidth greater than the capacity of link l (C_l) can be shown as follows.

$$\begin{aligned}\Pr[Y > C_l] &= \Pr[Y > (1+\delta)\mu] \\ &= \Pr[Y > (1+\delta)E[X]] \\ &= \Pr[Y > (1+\delta)\sum_{s\in S}\sum_{f\in F_s} b'_{fs}(\frac{b'_{fs}}{d'_{fs}})y_{fl}] \\ &\leq (\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}})^{\mu},\ \forall l \in L,\end{aligned} \tag{3.61}$$

where δ is a positive number that is calculated by solving the following relation.

$$(1+\delta)\ E[X] = C_l. \tag{3.62}$$

As shown above, we added a new binary variable (y_{fl}) to (3.61) that determines either flow f has traversed link l or not. It should be noted that the Chernoff bound is tighter than the bound calculated using the Markov's inequality [112]. Indeed, if we calculate the second bound using Markov's inequality, for many links, the results would be greater than 1, which means that the bound is too loose.

In the next section, we evaluate the performance of our proposed solutions and discuss the obtained results from several scenarios in detail.

3.5 Framework Evaluation

In this section, we intend to evaluate the performance of the proposed solutions for different scenarios. As it is explained in Section 2.4, since our main interest is the performance assessment of the service provisioning process, we have conducted the experiments using our developed simulator with extensive results based on the following assumptions.

3.5.1 Key Assumptions

For all the scenarios, one flow is assigned to every station and it is assumed that all the flows have the same priority. Our proposed scheme is evaluated under a stable network condition, i.e., no new station joins or leaves our topology. In fact, we have focused on a scenario in which network flows should be routed on a WMN constructed with multiple stationary MRAGs. In this case, the recurrence of updating channel conditions is insignificant and the MRAGs are able to keep track of the traffic

demands and link conditions, continuously. It should be noted that there are many applications of such a scenario, including building mesh networks among traffic lights and different buildings especially in urban areas. We have utilized multi-rate MRAGs (in access and backhaul) that serve stations with diverse data rates. In addition, it is supposed that all MRAG nodes use omni-directional Single-Input Single-Output (SISO) antennas in access layer and Multi-Input Multi-Output (MIMO) antennas in backhaul links. Moreover, different frequency bands are utilized for access (2.4 GHz) and backhaul (5 GHz) layers, and the impact of Hidden Terminals (HTs) can be taken into account during the creation of the MDCG. Note that the impact of HTs can also be avoided by enabling the restart mode in Wi-Fi chipsets. This feature enables the radio to capture the stronger signal when multiple packets are transmitted (capture effect). The same technique for the similar scenarios is adopted in [55, 56]. Moreover, similar to [48, 50, 51], we use the protocol interference model in our simulations. For the conducted scenarios, the communication and carrier sensing ranges of this model are set to 100 m and 150 m, respectively.

The backhaul topology is represented using an undirected graph and due to the drastic growth of downlink greedy applications [57, 103], we assume that all the flows are downlink greedy in the access layer. However, as it is shown in section 2.6.2, the introduced solution in the access layer can be applied to both uplink and downlink traffic. Also, it should be noted that the presented flow routing and traffic scheduling solution can be used for traffic flows (in both directions) over the backhaul network. We use uniform demands and capacities for the flows and the selected gateways, respectively. Moreover, according to [58], since two thirds of the top activities on smart-phones are nomadic and the mobility of users during web browsing is limited, we assume that the users are stationary. It means, our solution can be applied directly to the cases that a large number of users are placed in conference halls or stadiums. The majority of the assumptions are based on the scenarios in [40, 82].

3.5.2 Simulation Setup

We have conducted a series of simulations over grid and random topologies that are deployed within a 200×200 m^2 area. These topologies are represented by undirected graphs, which contain 25 and 16 MRAGs, respectively with uniformly distributed stations. In our setup, MRAGs use the IEEE 802.11n standard to form the mesh backhaul and the Wi-Fi stations support the IEEE 802.11 g/n standards. In

addition, we apply the channel model used in [39] to our scenarios. Other simulation parameters are delineated in the following sections. For running the simulations, we have developed our own network simulator in Python/Gurobi [59] and we have used *igraph* library to find the MISs of the created MDCG from the network graph.

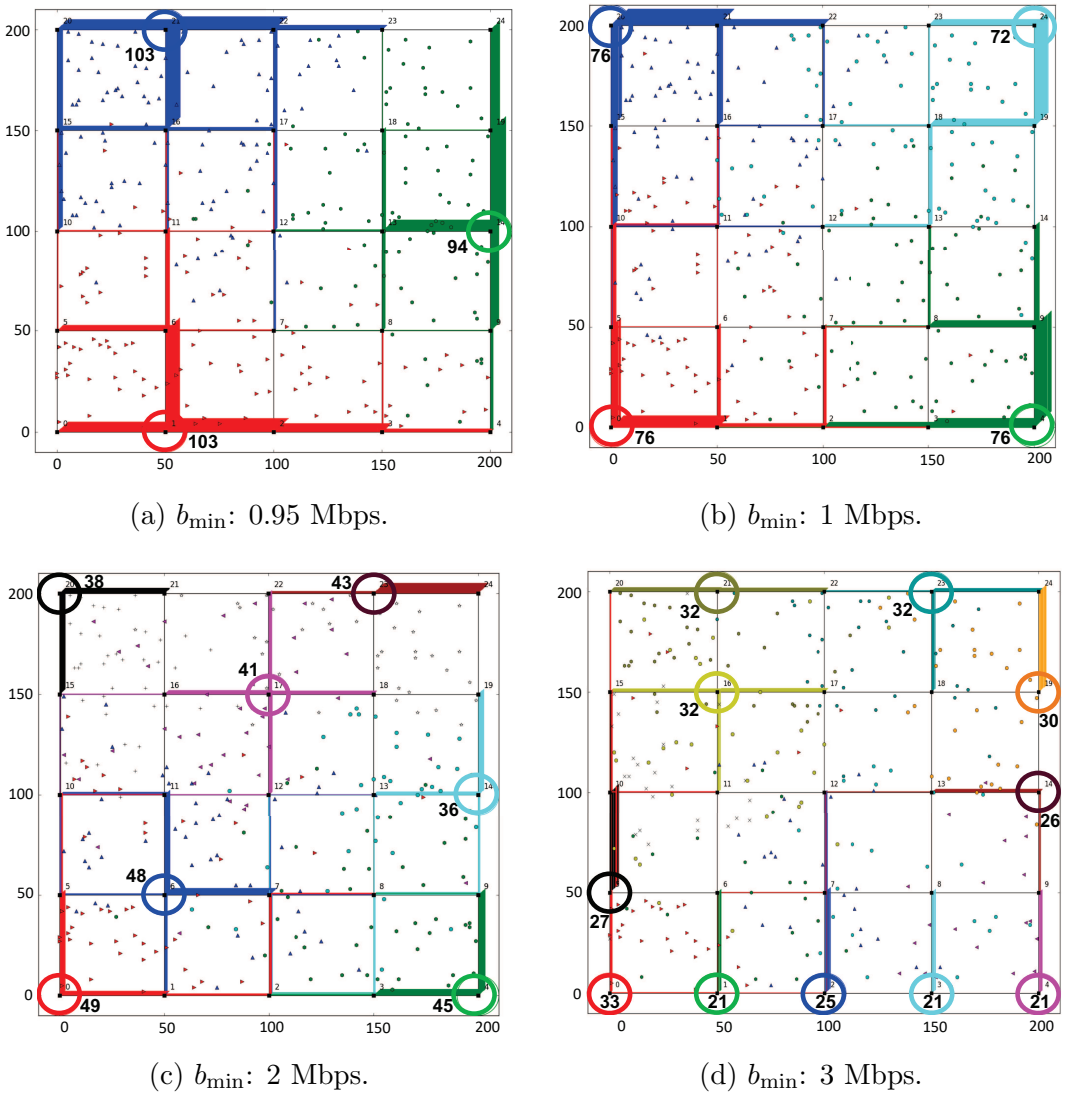
In addition, to capture the severe impact of interference, we consider the grid as the dominant topology to construct the mesh backhaul network. Moreover, except the results presented for the Multi-Channel (MC) backhaul scenarios, we have used a Single-Channel Single-Radio (SC-SR) setup in the backhaul network of all the simulations to demonstrate a clear picture on the performance of our solution. It is important to note that the running time of the simulations may vary from a couple of minutes to more than an hour, which is a function of the given parameters and the topology. This running time can be reduced significantly by using grouping and partitioning techniques. In this case, the main scenario can be divided into multiple scenarios with smaller inputs. Then, by solving the new scenarios and integration of the obtained results, we would be able to reduce the complexity of the original problem. Also, enabling the multi-threading feature of computationally rich resources that run our solution (as a part of the control plane at the SDN controller) can be very helpful to deal with the complex scenarios. Moreover, by solving the problem for specific scenarios in advance and storing the results (as profiles) and applying them directly to the given topologies, there is not much concern on the running time and it meets the requirements in terms of latency.

3.6 Simulation Results

In the next four sections, we analyze the functionality and performance of the introduced solution in Section 3.4.3 (the algorithm for the joint optimization problem) by conducting a set of simulations. Then, as the last part of the simulation results, the performance of Section 3.4.6 (randomized single-path flow routing) is evaluated.

3.6.1 Gateway Placement and Flow Routing Map

In the first simulation, we investigate the impact of $b_{\min}$ on the number of selected gateways and their placements as well as the routing map of network flows (from their assigned gateways to their associated MRAGs). Fig. 3.6 illustrates the routing maps of 300 network flows for different $b_{\min}$ values. The shown figures are the outcomes of



(a) $b_{\min}$: 0.95 Mbps.

(b) $b_{\min}$: 1 Mbps.

(c) $b_{\min}$: 2 Mbps.

(d) $b_{\min}$: 3 Mbps.

Figure 3.6: The impact of $b_{\min}$ on the selected gateways and routing map.

a scenario where gateway capacity (G_m) and delay bound (d_{fs}) are set to 100 Mbps and 4 hops, respectively. In each sub-figure, the selected gateways are displayed by the colored circles and the number of the associated flows with each one is shown close to them. It can be seen that increasing $b_{\min}$ has a significant impact on the number of the required gateway nodes. For instance, by changing $b_{\min}$ value from 1 Mbps to 3 Mbps, the number of required gateways is increased from 4 to 11.

In addition, regarding the number of the selected gateways for each scenario, we are able to find an estimation of the existing gap between the best possible optimal number of required gateways and the outcomes of our heuristic scheme. Since one

of the objectives of the proposed formulation is minimizing the number of required gateways, the lower bound of this value ($\frac{b_{\min}|\boldsymbol{S}|}{G_m}$) in each scenario can be considered as the best possible optimal result. Thus, by assuming an unlimited capacity for the backhaul links and then the calculation of this value for each scenario, we can find the worst possible gap between the heuristic and optimal outcomes. For instance, the lower bound values for the optimal number of the required gateways w.r.t. the shown $b_{\min}$ values in Fig. 3.6 are 3, 3, 6 and 9, respectively. Also, the calculated numbers of required gateways using our solution for the same $b_{\min}$ values are 3, 4, 7 and 11. Hence, it can be seen that the worst possible gap among the conducted simulations is no more than 2. Note that this gap is the consequence of considering the impact of interference on the capacity of backhaul links in our solution. Also, as it is shown in the first line of *Algorithm 3*, the other utilized parameters in the problem formulation such as $b_{\min}$ and d_{fs} may affect the minimum number of required gateways. For instance, increasing $b_{\min}$ decreases the difference between lb and ub variables that consequently shrinks the search space and reduces the gap between the heuristic and optimal results.

According to the presented graphs, it is clear that the intensity of the network flows at the egress links of the selected gateways is high and regarding the defined delay bound, the flows are not allowed to traverse more than 4 hops (links) to reach their gateways. Moreover, as demonstrated in Fig. 3.6, the preference of our solution in the gateway selection process is placing the gateways on the boundaries of the grid topology to maximize the aggregated throughput by minimizing the drastic impact of the inter-flow interference. In the situation when the assignment of $b_{\min}$ to all the stations cannot be guaranteed just by using the gateways placed on the boundary, the inner MRAGs are selected as gateways too. This case can be seen in the results shown in Fig. 3.6c and Fig. 3.6d.

For each of the conducted scenarios, we also measured several metrics including the number of the traversed hops by the flows, allocated bandwidth to the flows, and the traffic load (throughput) imposed by the associated stations on the MRAGs. The obtained results for the three $b_{\min}$ values, which are illustrated in Fig. 3.7 to Fig. 3.9, are the counterpart of those shown in Fig. 3.6. The first graph of each sub-figure shows the number of the traversed hops by the flows in ascending order. By comparing the first graphs shown in Fig. 3.7 to Fig. 3.9, we see that increasing $b_{\min}$ has led to reducing the number of the traversed hops by the majority of the flows. For instance, the number of flows directly connected to the selected gateways (0 hop)

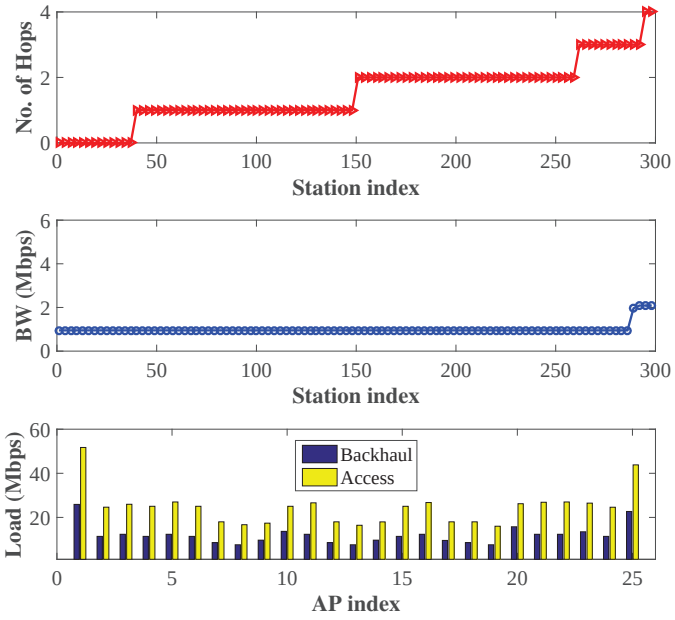


Figure 3.7: Calculated metrics for $b_{\min}$: 0.95 Mbps (3 gateways).

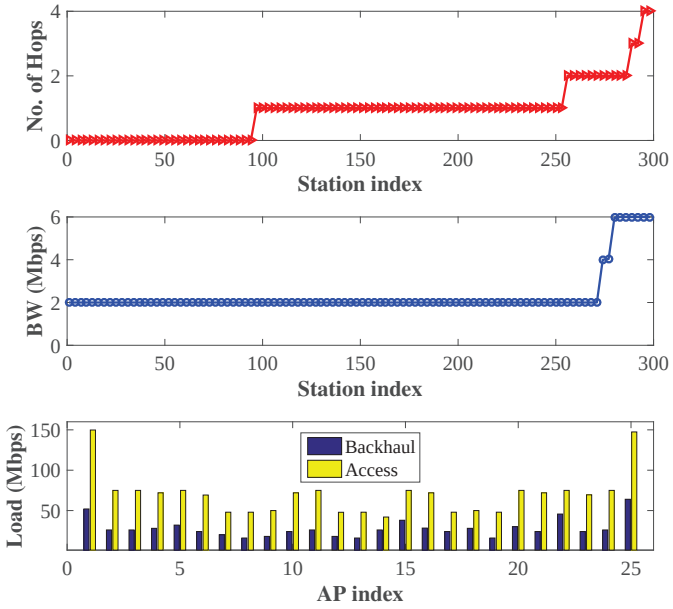


Figure 3.8: Calculated metrics for $b_{\min}$: 2 Mbps (7 gateways).

for $b_{\min}$: 3 Mbps is 4 times of the case for $b_{\min}$: 0.95 Mbps. This is the consequence of increasing the number of selected gateways. Moreover, increasing $b_{\min}$ results in having more flows with an allocated bandwidth higher than the defined $b_{\min}$.

This fact is displayed in the second graphs of the sub-figures presented in Fig. 3.7 to Fig. 3.9. In the last graph of each sub-figure, the load of every MRAG is illustrated

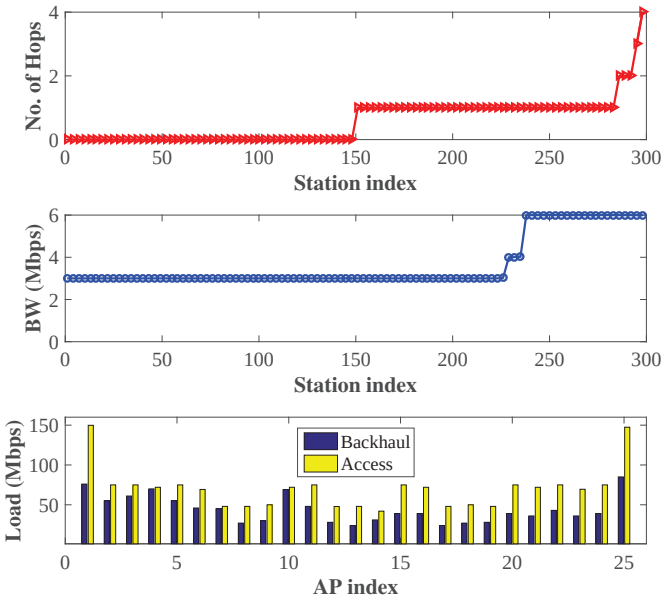


Figure 3.9: Calculated metrics for $b_{\min}$: 3 Mbps (11 gateways).

in terms of the downlink traffic load of the associated stations (in Mbps). By following the index and the location of the MRAGs shown in the sub-figures of Fig. 3.6 and checking the corresponding results in the shown sub-figures, it can be seen that less load is imposed on the inner MRAGs. Note that since all the Wi-Fi stations are receiving downlink greedy traffic, we can find a correlation between the number of associated stations and the traffic load in each MRAG as well. Moreover, in the last graph of each sub-graph, we make a comparison between the acquired bandwidth by the stations based on the formulation presented in Section 3.4.1 (Access) and *Algorithm 3* (Backhaul). Note that due to using SC-SR setup in the backhaul network and applying non-overlapping channels to the access radios [39] of MRAGs, we see a gap between the bandwidth acquired by the flows in the access (yellow bars) and the backhaul (blue bars) layers.

3.6.2 Per-flow Throughput and Flow Load Balancing

In the second simulation, due to the complexity of finding the optimal result based on [82] for 300 flows, we conduct this scenario for 100 flows and make a comparison between the outcomes of the optimal and heuristic solutions. Fig. 3.10 illustrates the obtained bandwidth by 100 stations that are uniformly distributed over a 5×5 grid topology. The results are presented for both the heuristic and optimal schemes w.r.t.

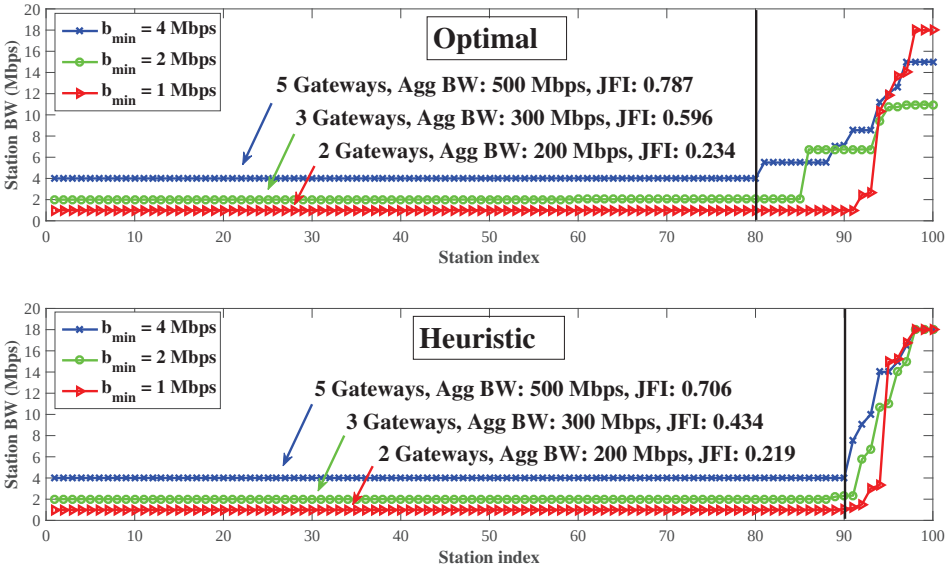
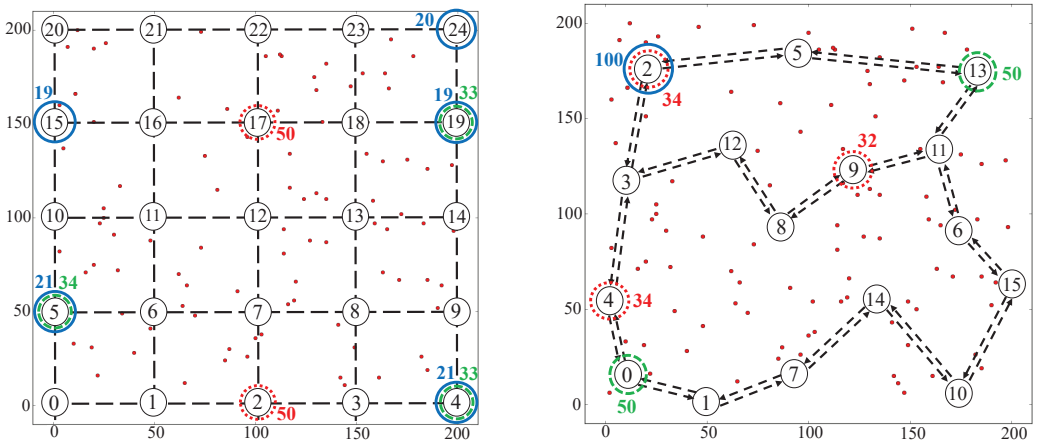


Figure 3.10: Station bandwidth (BW) for different $b_{\min}$ values.

different $b_{\min}$ values. According to the shown outcomes, our heuristic scheme achieves the same aggregated throughput and the same number of gateways as the optimal solution. By comparing the graphs shown in Fig. 3.10, we see there is a negligible degradation in the calculated fairness index (JFI) for the heuristic outcomes versus the optimal ones. This upshot is the consequence of having fewer stations with the bandwidth higher than $b_{\min}$ in the results of the heuristic scheme. This value is 10 stations for the heuristic and 20 stations for the optimal schemes and it can be seen by the comparison of the shown vertical black lines in the graphs illustrated in Fig. 3.10. Hence, the latter has a small improvement in terms of fairness.

The map of the selected gateways by the heuristic scheme for different $b_{\min}$ values is displayed in Fig. 3.11a. In this figure, the colors of the selected gateways and the shown numbers next to them are consistent with the same colors that are used in Fig. 3.10 for representing various $b_{\min}$ values. It can be seen that there is a balanced distribution of network flows among the selected gateways for all $b_{\min}$ values. Moreover, except the $b_{\min}$: 1 Mbps case, the gateways are selected from the MRAGs placed on the boundary of the grid topology. Also, it is important to note that our presented formulation supports multi-path flow routing over the backhaul network regarding the defined constraints. So, it is possible that the packets belong to the same flow take different routes to reach their destination. For instance, in one of the scenarios, the packets of a multi-path flow have taken two separate routes to reach their destination (MRAG 2) from their assigned gateway (MRAG 14) in the topology



(a) Grid topology. (b) Random topology.

Figure 3.11: The map of selected gateways for different topologies.

shown in Fig. 3.6c. If we use the traversed MRAGs by the packets of this flow, the selected routes can be shown as {14, 9, 8, 3, 2} and {14, 13, 12, 7, 2}. This flow has acquired different amount of bandwidth through each route which is 1.005 Mbps from the first route and 0.995 Mbps from the second one. Although the obtained bandwidth through each route is less than the defined $b_{\min}$ (2 Mbps), the aggregate bandwidth assigned to this flow at the destination is 2 Mbps. Moreover, the selected routes have satisfied the defined delay bound (d_{fs}) constraint which is set to 4 hops.

3.6.3 Impact of MC-MR on the Number of Gateways

To explore the impact of using MC-MR, we carry out a simulation on the random topology illustrated in Fig 3.11b. In this simulation, we set G_m and $b_{\min}$ to 170 Mbps and 1.7 Mbps, respectively. Also, the maximum data rate of backhaul links is set to 150 Mbps (a single MIMO stream of 802.11n). Then, we calculate the obtained bandwidth by 100 stations that are uniformly distributed over the random topology for different number of backhaul channels. This process helps us to observe the significant impact of using non-overlapping channels on the number of gateways. The plots presented in Fig. 3.12 illustrate the per-flow throughput for the given number of channels. The first plot shows the sorted bandwidth acquired by the flows in the access layer, which is the upper bound of the flow bandwidth in the backhaul network. The next three plots in Fig. 3.12 show the bandwidth for different numbers of non-overlapping backhaul channels. According to these plots, by increasing the number

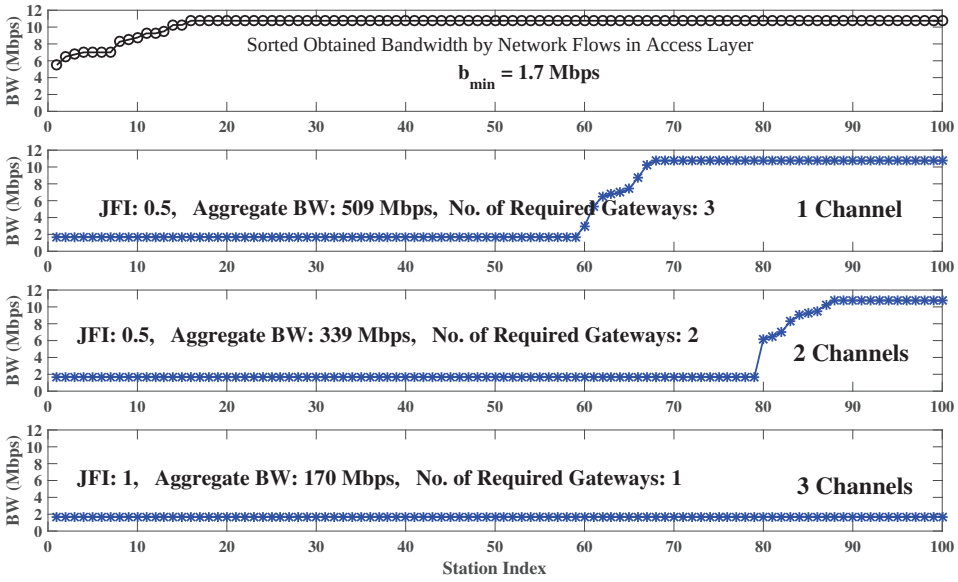


Figure 3.12: The results of MC in the random topology.

of backhaul channels from 1 to 2, the number of the required gateways has reduced from 3 to 2. The same outcome can be seen when the number of channels is increased from 2 to 3. As illustrated, the usage of 3 channels allows us to achieve a perfect fairness (JFI = 1) and the maximum network capacity (170 Mbps) achieved by one gateway. As shown in Fig. 3.12, for the other scenarios, the aggregated throughput of the heuristic scheme is very close to the maximum values.

The map of the selected gateways for this simulation is shown in Fig. 3.11b. Similar to Fig. 3.11a, different colors are used to show the selected gateways. For the first scenario (1 channel), nodes {2, 4, 9} are chosen as the gateways and are highlighted using red circles. For the second (2 channels) and the third (3 channels) scenarios, green and blue circles are used to represent nodes {0, 13} and {2} as the gateways, respectively. The colored numbers show the number of the assigned flows to each gateway. In this simulation, no limitation on the number of traversed hops is set. Hence, by using a single gateway and 3 channels in the backhaul, the defined QoS constraint ($b_{\min}$: 1.7 Mbps) can be guaranteed. In addition, by reducing the number of required gateways (through increasing the number of channels), the number of flows with the bandwidth more than $b_{\min}$, is reduced as well. The same trend of results can be seen by increasing the number of the backhaul radios.

Table 3.2: Aggregated throughput of the selected schemes

Optimal	Heuristic	ACO	Random	Shortest Path
90 Mbps	90 Mbps	81.6±2.93 Mbps	71.83±8.8 Mbps	79 Mbps

3.6.4 Comparative Analysis of Flow Routing

As the last part of this section, we did a comparative analysis of flow routing performance among some of the known centralized WMN routing schemes, including the Optimal fine-grained routing [82], our heuristic scheme, Ant Colony based routing (ACO) [85], random routing strategy and shortest path routing. In this simulation (based on the topology shown in Fig. 3.11a), we considered 10 greedy flows generated from one side of the mesh topology and destined to the other side. In fact, the selected sources and destinations for the first 5 flows are nodes {0, 5, 10, 15, 20} and {4, 9, 14, 19, 24}, respectively. For the second 5 flows, these sets are {20, 21, 22, 23, 24} and {0, 1, 2, 3, 4}. The flows have to traverse the grid topology to reach their destinations and at the same time, they try to maximize their aggregated throughput after finding the largest possible minimum bandwidth. We calculated the aggregated throughput of the given flows for different schemes as shown in Table 3.2.

By having a holistic view of the topology (provided by SDN) our heuristic scheme is able to find the best outcome in terms of the aggregated throughput which is 90 Mbps (equal to the optimal result). Note that this is the maximum achievable value for the selected metric w.r.t. the configuration of SC-SR backhaul links, the gateway capacity and the interference range. For the random and ACO schemes, since we may have different results at each run of the simulation, we present the average outcomes over 10 runs. It can be seen that the random selection of the next hop reduces the aggregated throughput and it leads to having a large standard deviation (≈12% of the mean for the random scheme). On the other hand, ACO achieves a better performance by exploring the alternate routes to alleviate the impact of interference, however, it could not reach the maximum aggregated throughput. Finally, for the shortest path scheme, there is a 14% performance gap between its outcome and the results obtained from the heuristic scheme. Note that although the average aggregated throughput for the random scheme is less than the shortest path routing, in some runs the acquired results by the random scheme are more than the ones from the shortest path scheme (shown by the large standard deviations in Table 3.2).

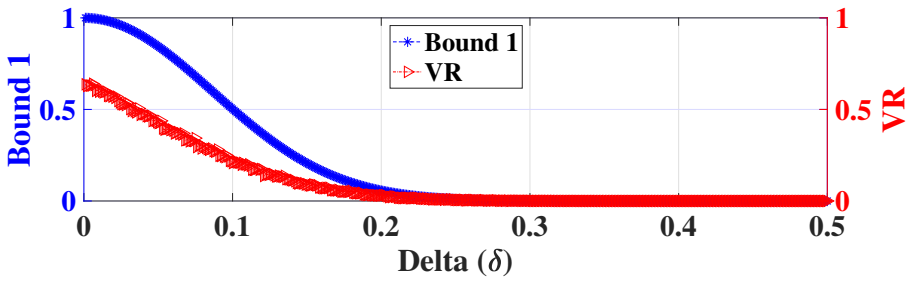


Figure 3.13: Results for the first bound and its violation rate.

3.6.5 Randomized Single-Path Flow Routing

As the last part of the simulation results, we validate the functionality of the randomized single-path flow routing solution through a set of experiments. In fact, we mainly focused on the calculation of the shown bounds in (3.60) and (3.61) for a selected scenario. The simulation setup of this section is mostly based on the details provided in Section 3.5.2. However, as it has been mentioned in Section 3.4.5, we use a single gateway scenario, which is placed at the top-right corner of the grid topology shown in Fig. 3.11a. Moreover, the number of flows, minimum guaranteed bandwidth ($b_{\min}$) and the gateway capacity (G_g) are set to 100, 1 and 300 Mbps, respectively.

To verify the first bound shown in (3.60), we set δ to 0.001 and increase it by step 0.001. Note that δ is a factor that indicates the deviation of random variable X from its expectation ($E[X]$). Thus, it can be interpreted as a tolerance factor to ensure the total benefit is not being less than a proportion of the average results for multi-path routing. For each value of δ, we run the experiment 10,000 times and then we count the number of times when the total benefit of RSR is less than the expectation of the product of the total benefit of the multi-path routing and $(1-\delta)$. We use the term of *Violation Rate (VR)* to refer to the occurrence ratio of this event.

Eventually, the calculated *VR* should be less than the bound that we have obtained through (3.60). The results for the first bound and its corresponding *VR* are shown in Fig. 3.13. As illustrated, for all the δ values, the *VR* is always less than the calculated bound for the explained scenario and there is almost no violation when $\delta > 0.2$. Indeed, the illustrated graph in Fig. 3.13 proves that we can assure the total benefit (bandwidth) of RSR is greater than a certain value with a deterministic probability. In other words, we prove that the probability of the event for the aggregated bandwidth of RSR less than a certain value is bounded according to (3.60). Also, it can be seen that by increasing δ to the values larger than 0.2, we reduce the gap between X and

Table 3.3: Calculated $Bound1$ and VR for key δ values

δ	**0.001**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**
$Bound1$	0.99995	0.65954	0.17767	0.01747	0.00054	4.38×10^{-6}
VR	0.4582	0.021	0	0	0	0

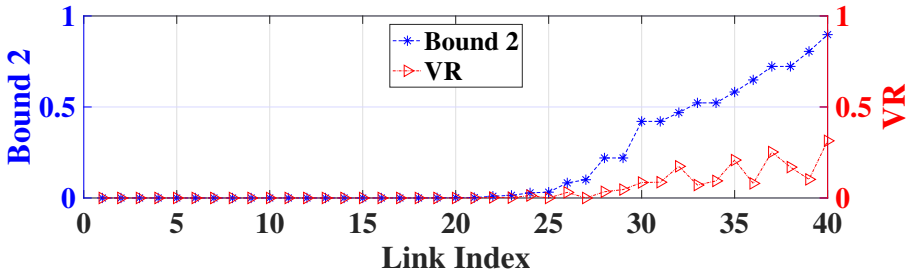


Figure 3.14: Results for the second bound and its violation rate.

its expectation that leads to having no violation. On the contrary, by shrinking δ to the values smaller than 0.2, this gap is increased that intensifies the VR. To have a better picture, we use Table 3.3 to show the first bound and VR for the key δ values displayed in Fig. 3.13. It can be seen that the growth of δ has a significant impact on shrinking the gap between the first bound and VR.

For the second bound, we define VR as the occurrence ratio for the event when the aggregated bandwidth on a specific link exceeds its capacity. We can calculate the VR for each link of the given network graph. Thus, we need to find which flows have traversed each single link of the graph and then regarding the obtained link capacity under the introduced scheduling, the VR can be quantified for the respective link. The acquired results for the second bound and its corresponding VR are illustrated in Fig. 3.14. This graph illustrates the link capacity violation probability and the Chernoff bound for the edges (links) of the 5×5 grid topology. The presented results are sorted based on the calculated bound of the links and it can be seen that the calculated VR values for all the links are less than the estimated Chernoff bounds. In fact, Fig. 3.14 shows that we can guarantee that there is a certain probability for any link where its capacity constraint has not been violated. In other words, for any link, the capacity violation probability is bounded to what we have calculated in (3.61).

Furthermore, it is important to note that we can use a reserved bandwidth or a buffering mechanism to absorb the violation rate for the links with positive VR values shown in Fig. 3.14. The presented results for both bounds validate the correctness of the presented algorithm in Section 3.4.6 for the randomized single-path flow routing.

Table 3.4: Selected flows and their acquired bandwidth over one iteration

Flow Index	**Bandwidth (Mbps)**	**Flow Index**	**Bandwidth (Mbps)**
2	7.9131	65	7.9131
4	7.9131	69	7.9131
13	1.0	70	1.0
14	7.6729	74	4.3120
19	5.3919	80	7.9131
20	7.9131	81	1.0
24	7.9131	84	4.4109
28	4.4414	85	1.0
30	7.9131	87	1.0973
43	4.0622	89	7.9131
46	7.9131	90	1.0
47	7.9131	91	1.0
48	7.9131	93	1.0
49	7.9131	94	7.9131
52	1.0	99	7.9131
58	7.9131		

By combining these two bounds, we assure that the total bandwidth has a certain probability to be no less than a given value, while the *VR* for any link is bounded. Note due to the random-walk nature of the flows in *Algorithm 6*, flow traversing events over link l are i.i.d. and so the Chernoff bounds shown in (3.60) and (3.61) can be calculated for the explained scenario. Furthermore, for the described scenario, we have calculated the total assigned bandwidth to the flows for the RSR and multi-path routing schemes. The multi-path results represent the optimal throughputs for every flow as the outcome of the optimization problem. For RSR, after running *Algorithm 4* for 10,000 iterations, we have calculated the average and the aggregated bandwidth allocated to the flows. There is an around 4% difference between the total benefit (bandwidth) of the multi-path (133 Mbps) and the single-path (128 Mbps) routing schemes, which is due to the randomized nature of RSR.

Also, as we discussed already, the flow selection and flow routing are the key elements of the randomized algorithm and these functions are being called multiple times (based on defined number of iterations). It means that in each iteration, a subset of the network flows is selected (as shown in Fig. 3.5 and *Algorithm 5*) to be routed over the grid topology under the scheduling process. Table 3.4 shows a subset of the selected flows and their acquired bandwidth in one iteration. It can be

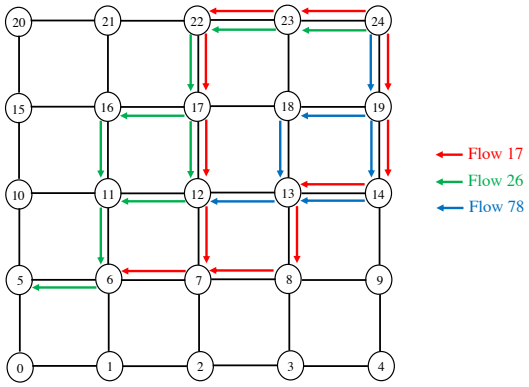


Figure 3.15: Multi-path routing map of three network flows.

seen that in the shown iteration, 31 flows are randomly selected and the bandwidth obtained by them varies between 1 Mbps (as the defined $b_{\min}$) and 7.9131 Mbps. As it is explained in *Algorithm 4*, by having the results of all the iterations, the average allocated bandwidth to the network flows can be calculated.

Another interesting insight from the conducted scenarios is the limited number of multi-path flows (in comparison to the single-path ones) obtained from the formulation shown in Section 3.4.5. For instance, through the conducted experiments in this section, only 3% of the network flows are routed through multiple paths and the rest used a single path to deliver the traffic from the source to the destination. The key reason behind such a behaviour of traffic flows is the drastic impact of inter-flow interference within the SC-SR grid topology. Since the main objective is maximizing the aggregate throughput, the scheduling mechanism attempts to minimize the influence of interference by adjusting the number of active links within the interference range of the transmitters. Fig. 3.15 illustrates the links traversed by these flows over the grid topology. All these flows have the same source (node 24), which is their gateway and are destined to three different destinations (flow 17: node 6, flow 26: node 5, and flow 78: node 12). Note that the allocated bandwidth to these flows is divided between the traversed routes. For example, at nodes 17 and 19, the assigned bandwidth to flows 26 and 78 is divided equally between the (two separate) traversed routes. This assignment for flow 17 is 25% (for the route from node 24 to node 23) and 75% (for the route from node 24 to node 19). These percentages play a key role in the route selection process as explained in *Algorithm 6*.

3.7 Testbed Implementation

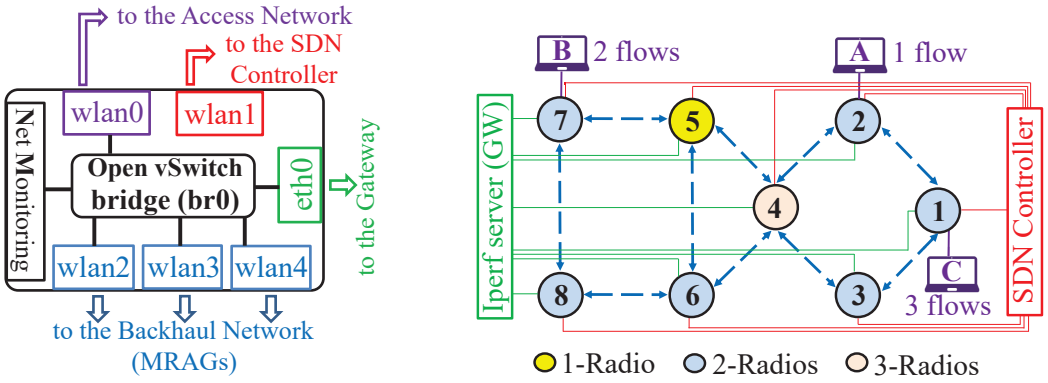
In this section, we delineate our testbed setup and the experiments carried out to assess the functionality of the presented solutions. The section is divided into two subsections. In the first one, we discuss the testbed setup and the obtained results of the joint traffic engineering solution. In the second one, the testbed configuration and the experimentation results of the randomized single-path flow routing are explained.

3.7.1 Joint Traffic Engineering on MC-MR WMNs

Testbed Setup

Fig. 3.16a illustrates the MRAG design that is utilized through the experiments of this section. In each MRAG, we have multiple wireless interfaces and one Ethernet interface that play specific roles. The first two wireless interfaces (`wlan0` and `wlan1`) are used to serve the access stations (to function as an AP) and provide out-band network access to the SDN controller, respectively. In addition, we have used one or more wireless interfaces to form a mesh backhaul network among the MRAGs as displayed in Fig. 3.16b. Also, the Ethernet interface (`eth0`) acts as a cellular gateway interface. To emulate the attributes of a cellular interface on `eth0`, `tc/netem` [114] is used. The software components of the testbed are mainly developed in Python and therefore we have chosen `Ryu` [66] as a well-known Python-based SDN controller to be in charge of the traffic engineering and flow routing process. It should be noted that our scheme is implemented as a north-bound application of the SDN controller that utilizes the collected network/topology information by the `Network Monitoring (NM)` module at the MRAGs (shown in Fig. 3.16a) as its inputs parameters. Then, it is able to find a suboptimal solution for the joint gateway selection and flow routing.

At the MRAGs, all the interfaces (except `wlan1` for the out-band communication) are bridged through an `OVS` interface (`br0`) that forwards the flows based on the defined rules by the SDN controller. Note that `OpenFlow` as the engine of the controller can be used for traffic engineering at MRAGs through the access, backhaul and the gateway interfaces. In fact, after finding a suboptimal solution, our north-bound application signals the `OVS` bridge (shown in Fig. 3.16a) to enable the gateway functionality or flow routing process. Also, to control the downlink flows, we have used the flow rate limiting feature of `OVS` at the MRAGs to have a fine-grained control over the network flows. We have used eight Alix3d2 [70] boards to form a heterogeneous



(a) MRAG. (b) Testbed setup.

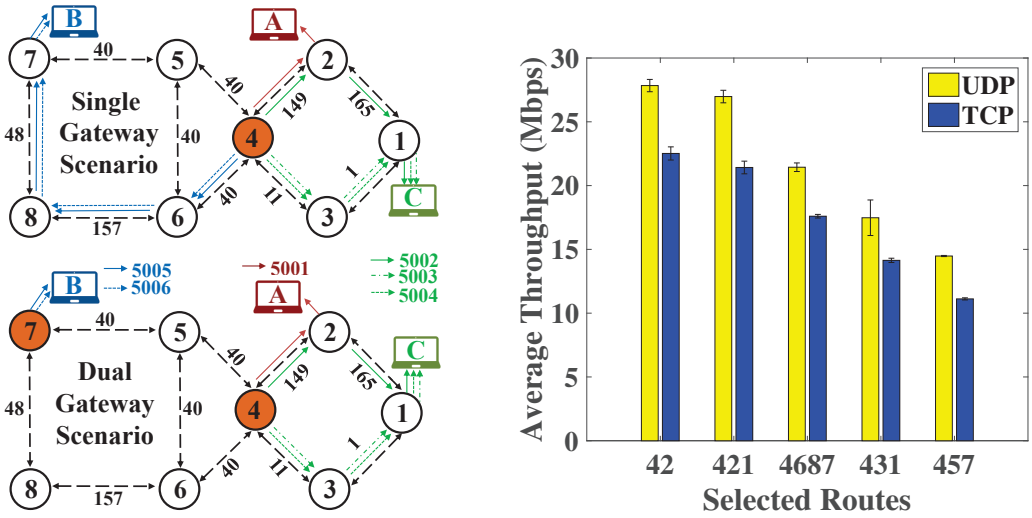
Figure 3.16: Testbed setup and MRAG components.

mesh backhaul network illustrated in Fig. 3.16b, which consists of MRAGs with a different number of backhaul interfaces. In the displayed topology, all the MRAGs are connected to the SDN controller through an out-band WLAN (shown by the solid red lines). The Ethernet interfaces are connected to an `Iperf` server through a wired network (shown by the solid green lines) that generates the greedy downlink traffic to the stations. Also, the backhaul links are shown by the dashed blue lines.

In the topology shown in Fig. 3.16b, except MRAGs 4 and 5 (with three and one radio interface, respectively), all the MRAGs have two backhaul interfaces. Moreover, since the AP association is a preprocessing phase of our proposed solution (as discussed in Chapter 2 and Section 3.4.1), we have used a predefined association map for three stations (with a different number of flows) as shown in Fig. 3.16b. At the stations, we run `Iperf` in the server mode to receive the downlink traffic generated by the selected gateways. `Hostapd` is used on interface `wlan0` of MRAGs to provide the access connectivity. The key elements of our testbed setup are based on the components mentioned in Table 2.2. For further details on the testbed configuration, we encourage the readers to read our prior work in [39, 81].

Experimentation Results

To explore the functionality of our scheme presented in Section 3.4.3, we have conducted a set of experiments. As shown in Fig. 3.16b, we have six greedy downlink flows that are associated with MRAGs 1, 2 and 7. The values of $b_{\min}$, d_{fs} and G_m are set to 5 Mbps, 3 hops and 60 Mbps, respectively. Our north-bound application as an SDN module receives the collected information about the stations and wireless

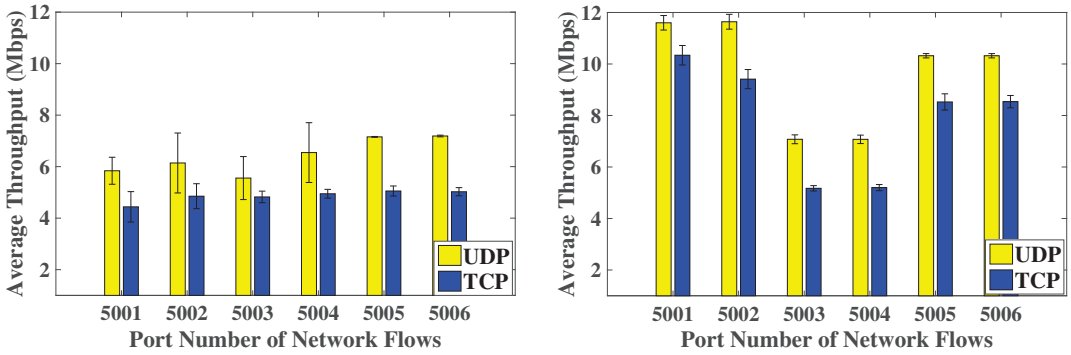


(a) Flow routing map. (b) Flow throughput.

Figure 3.17: The flow routing maps and obtained flow throughputs.

channel. Then, it determines the number of the required gateways and the selected links to route the traffic flows. Regarding the testbed configuration and the given values for the key parameters of the problem, MRAG 4 is selected as the gateway of our MC-MR WMN. This assignment is shown in the upper side of Fig. 3.17a. Note that since MRAG 4 has three backhaul interfaces, this makes it the best candidate to function as the mesh gateway.

Moreover, it satisfies the defined delay bound (d_{fs}: 3 hops) for all the traffic flows. In addition to the selected gateway, Fig. 3.17a shows a flow routing map and the assigned channels to the backhaul links. As illustrated, three interfaces of MRAG 4 operate on separate orthogonal channels (from the set of available ones). By checking Fig. 3.17a, it can be seen that in contrast to the other routing schemes, our solution routes the flows over different paths to maximize the aggregated throughput. For instance, one flow of Station C is routed via MRAG 2 (port 5002) and the rest (port numbers 5003 and 5004) are routed through MRAG 3. Note that we have differentiated the flows destined to the same station using different port numbers. In addition, our scheme does not necessarily select the shortest path from the gateway to the stations. This fact can be seen for the traffic flows (with port numbers 5005 and 5006) destined to host B. The main reason behind this decision is to avoid passing through MRAG 5 (as a single-radio intermediate node) which reduces the aggregated throughput. Note that in this scenario the backhaul capacity is the main bottleneck.



(a) Shortest path routing. (b) Heuristic routing.

Figure 3.18: The comparison of obtained flow throughput.

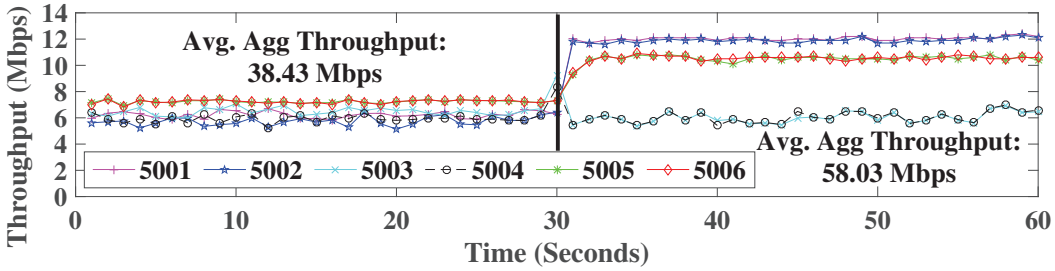


Figure 3.19: The measured flow throughputs over one minute.

Thus, if we use the same $b_{\min}$ and reduce G_m to 20 Mbps, then we need an additional gateway to guarantee the defined $b_{\min}$. In this case, the second gateway is MRAG 7 as shown in the lower side of Fig. 3.17a.

Fig. 3.17b shows the measured end-to-end throughput for a single greedy downlink UDP/TCP flow over different routes. The x-axis represents different routes from the selected gateway (MRAG 4) to the stations. For instance, 421 denotes the route that starts from MRAG 4, passed through MRAG 2 and destined to MRAG 1. By checking the quantified throughput for the first two routes, we see the impact of MC-MR on maintaining the throughput over multiple hops. For the third route (4687), although there is a degradation of the throughput, it still could be kept within a reasonable range over three hops. Also, for the last route (457), since MRAG 5 has only one radio, there is a significant drop in the throughput. Furthermore, we have measured the throughputs of UDP/TCP flows for the shortest path routing and our heuristic scheme that are illustrated in Fig. 3.18.

According to these results, our scheme improves the aggregated throughput for UDP and TCP traffic flows by +51% and +60%, respectively, which is a remarkable

outcome. Moreover, the variation of the UDP traffic flows (over time) for the explained scenario is shown in Fig. 3.19. As illustrated, 30 sec after the beginning of the experiment, the SDN controller has applied our scheme (instead of the shortest path) to the MRAGs, so that by defining new flow routing policies, the aggregated throughput (of UDP flows) is increased significantly. Note that due to the severe interference caused by the other nearby APs on channels 1 and 11 (2.4 GHz), after a small transition at $t = 30$ sec, although the flows pass through route 431 could not increase their throughput, they have kept their performance. The duration of each measurement is 5 minutes and the results are the averages of 5 consecutive runs. It is important to note that the presented results in this section complement the results presented in Section 3.6. Similar to the most of prior studies, we utilized the simulations to verify the performance of our work in large-scale setup and the testbed implementation was carried out to validate our scheme as a proof of concept.

3.7.2 Single-Path Flow Routing Solution

Testbed Setup

In this section, we conduct a set of experiments to compare the performance of our SDN-based single-path flow routing scheme with some of the conventional WMN routing protocols, including Open80211s, BATMAN-adv and OLSR, in terms of the network throughput and protocol overhead. The experiments are carried out for a specific scenario in which three disjoint paths can be established between the determined source and destination nodes over an MC-MR WMN topology, which is illustrated in Fig. 3.20. Similar to the topology shown in Fig. 3.16b, the illustrated topology consists of 8 MRAGs. Note that in contrast to the prior section, the gateway functionality of MRAGs is disabled in this section and we focus on the flow routing process. The same hardware and software components which were introduced in Section 3.7.1 are utilized in the scenarios of this section. All the MRAGs are equipped with at least two wireless radios (tagged with w in Fig. 3.20). The established wireless links between the radio pairs operate in different ISM frequency bands and are shown with different colored lines.

We used two laptops connected to the Ethernet ports (tagged with *eth0* in Fig. 3.20) of MRAG1 and MRAG8 as the source and destination of the generated flows that should be routed over the topology. In fact, we generated three network flows at Host-A (attached to MRAG1) that are destined to Host-B (attached to MRAG8).

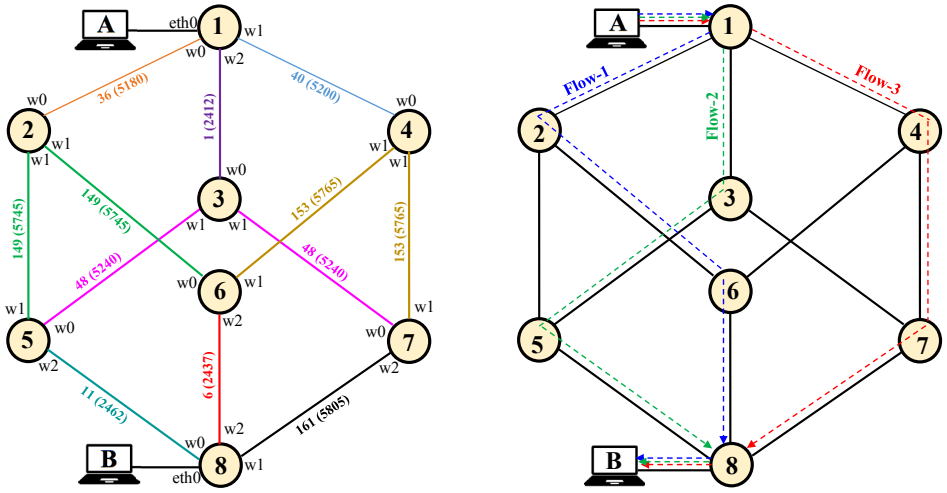


Figure 3.20: The MC-MR topology used for testbed setup.

In our experiments, to ensure that the wireless links of the mesh backhaul are the main bottleneck, we set the wireless radios to function on 802.11n (MCS 5, HT/SGI disabled) that provides the maximum data rate of 52 Mbps while the capacity limit of the Ethernet ports is 100 Mbps. Also, we setup an OOB management network for the monitoring and management of the MRAGs.

Except MRAG1 and MRAG8, all the MRAGs are connected to the SDN controller through a wired OOB network. Since the Ethernet ports of these two nodes are used for the measurement process, we utilize a WLAN as the OOB network of these MRAGs, which is in line with the existing work [115]. Note that there are several disjoint paths over the wireless backhaul from Host-A to Host-B, which contributes towards routing flexibility. The throughput measurements and packet capturing are conducted using `Iperf` and `Wireshark`, respectively. Also, to quantify the overhead of the selected routing protocols, we used `Cacti` and `Wireshark`. The components of the testbed configuration can be found in Table 2.2.

Experimentation Results

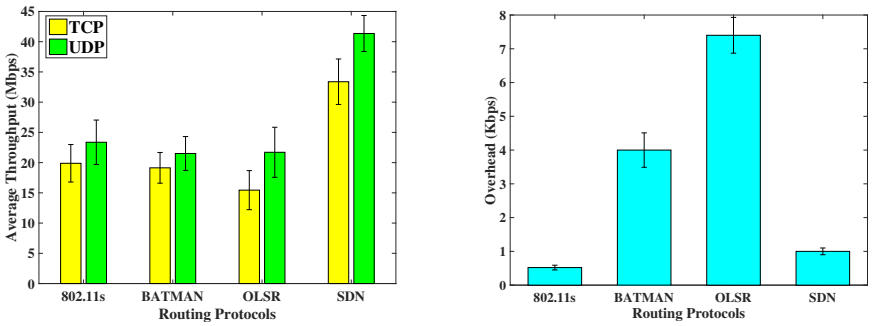
In the first set of the experiments, we have measured the aggregated throughput of three network flows that originated from MRAG1 and destined to MRAG8. The obtained results are shown in Fig. 3.21a. It should be noted that the conventional mesh routing protocols with a distributed nature, route all the network flows through the same path. For instance, in the illustrated topology, OLSR as a Layer-3 mesh routing protocol has chosen MRAGs {1, 3, 7, 8} as the relay nodes to route the

(three) network flows from MRAG1 to MRAG8. On the other hand, using SDN over WMN enables us to achieve a finer granularity for flow routing. Thus, by applying our routing scheme to the illustrated MC-MR WMN topology, it is possible to route the network flows through different (disjoint) routes.

This approach has a significant impact on the network performance and as displayed in Fig. 3.21a, it has almost doubled the aggregated throughput (benefit) in comparison to the conventional mesh routing protocols. Indeed, our SDN-based routing scheme takes advantage of the MC-MR feature on the mesh topology and as illustrated in Fig. 3.20, it routes the flows through separate non-overlapping routes (with different frequency channels) to improve the end-to-end throughput. Note that the average UDP throughput of each single flow over its 3-hop route (for SDN) is about 21.32±0.94 Mbps, but due to the power leakage of adjacent channels, the impact of inter-flow interference over the backhaul links, and the external interference caused by the neighbour WLANs (that we have no control of), the aggregated throughput is less than 3 times of individual ones. In addition, as shown in Fig. 3.21a, although there is not a significant difference between the performance of the selected conventional mesh routing protocols in terms of the total average throughput, Layer-2 routing protocols including 802.11s and BATMAN-adv could achieve better results. This is due to the fact that Layer-2 routing protocols impose a less overhead in comparison to Layer-3 routing protocols such as OLSR. Also, it can be seen that UDP flows (due to their connectionless nature and lack of having a congestion control mechanism) achieve a higher throughput in comparison to TCP flows.

In addition to the network throughput, we have measured the protocol overhead of the selected routing schemes under their default configuration. During the measurement, no network traffic was generated over the backhaul links and only the control traffic was captured. According to the shown results in Fig. 3.21b, 802.11s and OLSR have the lowest and highest amount of overhead, respectively. The main portion of the protocol overhead is due to the topology discovery messages generated by the conventional routing protocols. For instance, Originator Messages (OGM) in BATMAN-adv and Path Request (PREQ) messages in 802.11s constitute the illustrated overhead. In our SDN-based routing solution, Link Layer Discovery Protocol (LLDP) messages are in charge of neighbor discovery and it has the lowest overhead after 802.11s among the selected schemes, which is ≈1 Kbps.

It is important to note that although it is possible to reduce the protocol overhead of the selected mesh routing schemes by increasing their message exchange interval,

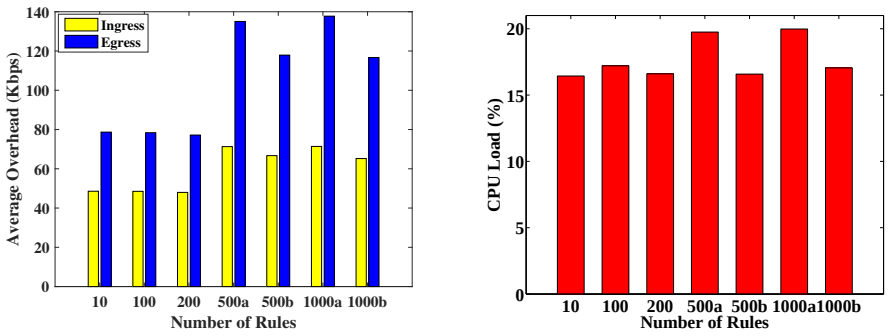


(a) Throughput (b) Overhead

Figure 3.21: Performance comparison of the selected schemes.

it may have a drastic impact on the link recovery process. It is clear that in dynamic network scenarios with mobility, the exchange of discovery messages plays a crucial role in maintaining the end-to-end connectivity. However, since in the carried out experiments, we focused on the stationary scenarios, the main function of these messages is the link recovery due to the channel switching at the backhaul radio interfaces. In fact, the introduced link scheduling mechanism may require to change (switch) the assigned channels to the MRAG radios over time. If this process imposes non-trivial switching latency, then the network performance and service provisioning might be affected. Hence, we need to have an insight about the duration of the recovery process. In Fig. 4b of [96], we have shown that this interval is 55±4 ms for SDN-based routing which is by far lower than the measured values for the conventional mesh routing protocols in the same scenario. For instance, the recovery intervals of BATMAN-adv and OLSR are 5.08±1.7 sec and 16±3.5 sec, respectively.

Note that reducing the message exchange interval of WMN routing protocols may improve their recovery time, however, it increases their protocol overhead due to increasing the frequency of the message exchange. In short, for the conventional routing protocols, there is a trade-off between the protocol overhead and the recovery time such that reducing one leads to an increase in the other one. On the other, by using the introduced SDN-based routing, we can achieve a low overhead and fast recovery time, simultaneously. Also, we have to mention that in this section, we only measured the overhead of underlying topology discovery messages and the overhead of the control traffic in our proposed solution is excluded from the results shown in Fig. 3.21b. This approach can be justified since we have considered using out-band cellular network (instead of mesh backhaul) to guarantee the reliable delivery



(a) SDN overhead (b) CPU

Figure 3.22: The overhead and CPU utilization vs. number of rules.

of control traffic in real-case scenarios. Moreover, we have to consider the fact that in case of any failure for the SDN controller, conventional WMN routing protocols still are the best alternatives to provide the network and service connectivity

Note that in the above scenario, to investigate the overhead of LLDP, no additional SDN rule is defined at the controller and MRAG nodes. Thus, in another experiment, we have measured the control overhead and CPU utilization of MRAGs for a different number of SDN rules. Note that the rules are defined in the `Open vSwitch` flow tables of the MRAGs and in case of using multi-path flow routing, more SDN rules should be defined, which increases the size of flow table. For this experiment, we have generated different numbers of dummy complex rules at MRAG1. Then, a greedy UDP flow is generated from Host-A and destined to Host-B. The complex rules are comprised of several fields including IP/MAC addresses and UDP port numbers, and they should be processed before matching the desired rule to forward the UDP traffic.

The average overhead and CPU utilization for a different number of rules are shown in Fig. 3.22a and Fig. 3.22b, respectively. We measured these metrics for a set of numbers including {10, 100, 200, 500, 1000} rules. For the cases with a large number of rules (specifically 500 and 1000), we considered two types of scenarios. In the first one (shown with postfix a), we generated a greedy UDP flow between Host-A and Host-B during the measurement process. In the second one (shown with postfix b), there was no traffic between the hosts over the mesh backhaul. However, in both cases, MRAG1 was connected to the SDN controller. According to the results shown in Fig. 3.22a, the existing gap between the overhead of the first three numbers of rules including 10, 100 and 200 is negligible. By increasing the number of SDN rules to more than 200 (i.e., 500 and 1000), the overhead increases around 55%. However, by

checking Fig. 3.22b, we see less than 20% changes in the CPU utilization. Also, there is a non-trivial difference between the illustrated outcomes of the defined cases (a and b) in Fig. 3.22a. For instance, the average overhead of the case $500a$ (for the egress traffic) is around 20 Kbps more than the case $500b$. Similar gap between the CPU utilization of these cases can be seen in Fig. 3.22b. The existence of a greedy UDP flow over the mesh backhaul causes such a gap between these cases. The same trend can be seen for 1000 SDN rules. Note that there is a clear correlation between the displayed graphs in Fig. 3.22. The presented outcomes may vary by changing the parameters of the defined SDN rules and the frequency of communications between the MRAGs and the controller. In fact, it is important to note that as far as the SDN controller and the MRAGs are communicating with each other, there is a control overhead that is a function of several factors including the number of MRAGs operating in the same frequency bands, the protocol configuration (i.e., the inter-arrival of control packets) and the age of defined SDN rules at the flow tables.

3.8 Conclusion

SDN-aware solutions provide flexibility and effectiveness for the implementation of advanced and structured traffic engineering strategies. By applying SDN to WMNs as a practical and well-known networking paradigm, we can introduce an agile and cost-effective Internet provisioning solution especially for the places where the access to the wired backbone is either too expensive or impossible.

In this chapter, we proposed a thorough fine-grained solution that considers three major steps of service provisioning over SDN-aware WMNs including AP association, gateway selection and flow routing. We have already introduced a cooperative association scheme for Software Defined WLANs in Chapter 2 that can be used in the first step of our solution. However, there is a tight correlation among these steps and most of the former studies either investigated them as independent NP-hard problems or did not present a unified and all-inclusive solution. To address this issue, we have introduced a unified solution to maximize the aggregated throughput of network flows in a WMN while satisfying the QoS constraints and reducing the deployment cost by minimizing the number of the required gateways. Due to the complexity of the presented formulation, we introduced a heuristic algorithm to solve the delineated problem in polynomial time. In contrast to many existing work, our scheme considers the key restrictive factors within the different tiers of WMN

(access, backhaul and gateway) to present a realistic model. Moreover, due to the challenges of implementing a multi-path flow routing on MC-MR WMNs, we have introduced a randomized single-path flow routing solution for these networks that can be implemented using the SDN technology. The performance of the presented solutions is evaluated through extensive simulations and testbed.

Chapter 4

Conclusions and Future Work

4.1 Conclusions

In this dissertation, we have investigated several fundamental traffic engineering problems in Software Defined WLANs and WMNs. We have formulated these challenges as optimization problems and due to their high complexity and NP-hardness, we have presented approximation algorithms to find bounded solutions in polynomial time. It is important to note that we have introduced a systematic and unified approach to solve the discussed problems. Moreover, the introduced solutions require minimum protocol-level modifications and are backward-compatible with the existing Wi-Fi standards. More specifically, the main contributions of this dissertation can be summarized as follows:

1. **Design of a cooperative AP association scheme for dense SD-WLANs.** We have introduced a modular SDN-aware framework for resource allocation in Software Defined WLANs. Our solution builds a cooperative and fine-grained platform for Wi-Fi stations of different service providers. Indeed, it provides a set of key services including association control, channel assignment and link scheduling for both downlink and uplink traffic. Moreover, we have formulated the AP association (which is the first step to establish a Wi-Fi connection) as a two-step optimization problem and we have proposed an approximated optimal solution that runs in polynomial time and can be applied to large-scale dense WLANs. Finally, the performance of the presented scheme is evaluated through extensive simulation scenarios and its functionality is validated through a testbed setup that substantiates the practicality of our work.

2. **Design of a fine grained traffic engineering solution for SD-WMNs.** By applying SDN to WMNs as a practical and well-known networking paradigm, we have introduced an agile and cost-effective Internet provisioning solution especially for the places where the access to the wired backbone or cellular networks is either too expensive or impossible. Our fine-grained solution considers major steps of service provisioning over SDN-aware WMNs including AP association, gateway selection and flow routing. There is a tight correlation among these steps and most of the former studies either investigated them as independent NP-hard problems or did not present a unified and all-inclusive solution. To address this concern, we have presented a unified solution to maximize the aggregated throughput of network flows (in a WMN) while satisfying the QoS constraints and reducing the deployment cost by minimizing the number of the required gateways. Due to the complexity of the presented formulation, we proposed an approximation algorithm to solve the introduced MINLP problem in polynomial time. In contrast to many related work, our scheme considers the key restrictive factors within the different tiers of WMN (access, backhaul and gateway) to present a realistic model. The performance of the presented scheme is evaluated through the simulations and testbed setup.

   Moreover, we have presented a randomized single-path flow routing solution for MC-MR WMNs that can be implemented using the SDN technology. The flow routing problem is formulated as an MCFP for WMNs and then to avoid the complexities of implementing multi-path routing protocols, we introduced a randomized mechanism to route the flows over WMNs. We have validated the correctness of our solution through the bound analysis. To have a better picture, we have compared the features of our scheme with the most popular WMN routing protocols in terms of the network throughput and control overhead through testbed setup. The obtained results substantiate that the proposed single-path flow routing solution can be used to provide a seamless connectivity among small cells over multi-hop wireless links with QoS support.

4.2 Future Work

For the future work that plans beyond this dissertation, different aspects of the presented solutions can be investigated and improved. This section provides some

directions and ideas that can be considered as potential research topics.

1. Not much work has been carried out to explore different aspects of applying SDN technology to the WMNs in terms of performance assessment. Dynamic adjustment of protocol parameters in Wi-Fi equipment and finding the best settings to maximize the network performance w.r.t. the configuration of SDN switches, e.g., *OVS*, and network protocols are some of the topics of interest that have practical applications.

2. There are not much choices to simulate the functionality of an SDN-based solution over wireless networks, accurately. One of the most recent tools is *Mininet-WiFi* [116] that adds Wi-Fi features to the *Mininet* as a well-known SDN simulator. There is some work that used this platform to evaluate the performance of several algorithms in SD-WLANs. Implementation of the proposed algorithms in this dissertation as the modules of *Mininet-WiFi* for association control, gateway selection and flow routing in SD-WMN scenarios can be considered as a subject for the future studies.

3. Applying Q-learning and deep reinforcement learning algorithms to the networking problems has attracted a lot of attentions among scholars. Although many papers have been published based on these technique for the wired networks, not much research has been conducted to utilize them for the joint resource allocation problems in WMNs. Hence, taking this approach and solving the discussed problems from this point of view may result in efficient and fast solutions that can be compared with the presented results in this work.

4. Although we have focused on the stationary scenarios and did not discussed the impact of mobility on the performance of the proposed solutions, there are many applications that involve mobility. Hence, the investigation of this topic and improvement of the introduced formulation to incorporate mobility can be considered as an interesting research subject for the future work. Moreover, the impact of applying different types of traffic to the model and formulation can be analyzed as an extension of our work.

Appendix A

Selected Publications

The following lists the outcome of my research and collaboration with my thesis advisor and labmates in terms of publication during my PhD program at UVic. The first paper represents the study performed in Chapter 2. Also, the outcome of the fifth paper is used in Chapter 2 partially. The second, third and fourth papers present the research carried out in Chapter 3. Finally, the outcomes of the sixth and seventh papers are utilized in the conducted experiments in the first three papers.

1. **D. Sajjadi**, B. Hu, M. Tanha, R. Ruby and J. Pan, "An approximated optimal solution for AP association in dense software defined WLANs," submitted to Comput. Commun., September 2018.

2. **D. Sajjadi**, Z. Zheng, R. Ruby, and J. Pan, "Randomized single-path flow routing on SDN-aware Wi-Fi mesh networks", in Proc. of IEEE MASS, Chengdu, October 2018.

3. **D. Sajjadi**, R. Ruby, M. Tanha and J. Pan, "Fine-grained traffic engineering on SDN-aware Wi-Fi mesh networks," IEEE Trans. Veh. Technol., vol. 67, no. 8, pp. 7593–7607, August 2018.

4. **D. Sajjadi**, R. Ruby, M. Tanha and J. Pan, "Fine-grained access provisioning via joint gateway selection and flow routing on SDN-aware Wi-Fi mesh networks," in Proc. of IEEE WiMob, Rome, October 2017.

5. **D. Sajjadi**, M. Tanha, and J. Pan, "Meta-heuristic solution for dynamic association control in virtualized multi-rate WLANs," in Proc. of IEEE LCN, Dubai, November 2016.

6. **D. Sajjadi**, M. Tanha, and J. Pan, "A Comparative study of channel switching latency for conventional and SDN-based routing in multi-hop multi-radio wireless mesh networks," in Proc. of IEEE CCNC, Las Vegas, January 2016.

7. M. Tanha, **D. Sajjadi**, and J. Pan, "Demystifying failure recovery for software-defined wireless mesh networks," in Proc. of IEEE NetSoft, PVE-SDN Workshop, Montreal, June 2018.

Bibliography

[1] Cisco, "Cisco Visual Networking Index: Forecast and Methodology, 2015–2020," White Paper, 2016. [Online]. Available: http://goo.gl/ylTuVx

[2] eMarketer, "Internet Users and Penetration Worldwide, 2016–2021," White Paper, 2017. [Online]. Available: https://goo.gl/kmtgFD

[3] S. Kemp, "The Global State of the Internet in April 2017," White Paper, 2017. [Online]. Available: https://goo.gl/uVwXP4

[4] Facebook, "State of Connectivity 2015: A Report on Global Internet Access," White Paper, 2016. [Online]. Available: http://bit.ly/1T68UJQ

[5] PwC, "Connecting the World: Ten Mechanisms for Global Inclusion, May 2016," White Paper, 2016. [Online]. Available: https://goo.gl/6SUa1T

[6] J. Swearingen, "True Unlimited Phone Data Plans Are Dead," New York Magazine, 2017. [Online]. Available: https://goo.gl/dUQhKq

[7] A. Fernandez-Fernandez, C. Cervello-Pastor, and L. Ochoa-Aday, "Achieving energy efficiency: An energy-aware approach in SDN," in *Proc. of IEEE GLOBECOM*, 2016, pp. 1–7.

[8] R. Bolla, R. Bruschi, F. Davoli, and C. Lombardo, "Fine-grained energy-efficient consolidation in SDN networks and devices," *IEEE Trans. Network Serv. Manage.*, vol. 12, no. 2, pp. 132–145, 2015.

[9] S. Biswas, J. Bicket, E. Wong, R. Musaloiu, A. Bhartia, and D. Aguayo, "Large-scale measurements of wireless network behavior," in *Proc. of ACM SIGCOMM*, 2015, pp. 153–165.

[10] M. Kim, S. Han, and M. Lee, "Demand-aware centralized traffic scheduling in wireless LANs," in *Proc. of IFIP Networking Conference and Workshops*, May 2016, pp. 144–152.

[11] Cisco, "Cisco Digital Network Architecture Vision, An Overview," White Paper, 2017. [Online]. Available: http://goo.gl/g6nz9r

[12] Meraki, "MR Cloud Managed Wireless Access Points," White Paper, 2017. [Online]. Available: http://goo.gl/WBLjBt.

[13] Huawei, "Build a Robust Wi-Fi Network in Dense Office," White Paper, 2017. [Online]. Available: http://goo.gl/qhCq6A.

[14] H. C. Yu, G. Quer, and R. R. Rao, "Wireless SDN mobile ad hoc network: from theory to practice," in *Proc. of IEEE ICC*, May 2017, pp. 1–7.

[15] V. Shrivastava, N. Ahmed, S. Rayanchu, S. Banerjee, S. Keshav, K. Papagiannaki, and A. Mishra, "CENTAUR: realizing the full potential of centralized WLANs through a hybrid data path," in *Proc. of ACM MobiCom*, 2009, pp. 297–308.

[16] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, "Performance anomaly of 802.11b," in *Proc. of IEEE INFOCOM*, March 2003, pp. 836–843.

[17] A. Patro, S. Govindan, and S. Banerjee, "Observing home wireless experience through WiFi APs," in *Proc. of ACM MobiCom*, 2013, pp. 339–350.

[18] J. Vestin, P. Dely, A. Kassler, N. Bayer, H. Einsiedler, and C. Peylo, "CloudMAC: Towards software defined WLANs," *ACM SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 16, no. 4, pp. 42–45, 2013.

[19] L. Suresh, J. Schulz-Zander, R. Merz, A. Feldmann, and T. Vazao, "Towards programmable enterprise WLANs with Odin," in *Proc. of ACM HotSDN*, 2012, pp. 115–120.

[20] K.-K. Yap, M. Kobayashi, R. Sherwood, T.-Y. Huang, M. Chan, N. Handigol, and N. McKeown, "OpenRoads: Empowering research in mobile networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 1, pp. 125–126, 2010.

[21] M. Bansal, J. Mehlman, S. Katti, and P. Levis, "OpenRadio: A programmable wireless dataplane," in *Proc. of ACM HotSDN*, 2012, pp. 109–114.

[22] A. Patro and S. Banerjee, "COAP: A software-defined approach for home WLAN management through an open API," in *Proc. of ACM MobiArch*, 2014, pp. 31–36.

[23] J. Lee, M. Uddin, J. Tourrilhes, S. Sen, S. Banerjee, M. Arndt, K.-H. Kim, and T. Nadeem, "meSDN: Mobile extension of SDN," in *Proc. of ACM MCS*, 2014, pp. 7–14.

[24] P. Gawłowicz, A. Zubow, M. Chwalisz, and A. Wolisz, "UniFlex: A framework for simplifying wireless network control," in *Proc. of IEEE ICC*, 2017, pp. 1–7.

[25] S. Zehl, A. Zubow, M. Döring, and A. Wolisz, "ResFi: A secure framework for self organized radio resource management in residential WiFi networks," in *Proc. of IEEE WoWMoM*, 2016, pp. 1–11.

[26] S. Zehl, A. Zubow, and A. Wolisz, "Hotspot slicer: Slicing virtualized home Wi-Fi networks for air-time guarantee and traffic isolation," in *Proc. of IEEE WoWMoM (Demo Section)*, 2017.

[27] M. Vutukuru, K. Jamieson, and H. Balakrishnan, "Harnessing exposed terminals in wireless networks," in *Proc. USENIX Symposium on Networked Systems Design and Implementation*, 2008, pp. 59–72.

[28] J. Huang, G. Xing, and G. Zhou, "Unleashing exposed terminals in enterprise WLANs: A rate adaptation approach," in *Proc. of IEEE INFOCOM*, April 2014, pp. 2481–2489.

[29] R. Riggio, M. K. Marina, and T. Rasheed, "Interference management in software-defined mobile networks," in *Proc. of IFIP/IEEE International Symposium on Integrated Network Management*, 2015, pp. 626–632.

[30] Y. E. Lin and T. M. Tsai, "Creation, management and migration of virtual access points in software defined WLAN," in *Proc. of CCBD*, 2015, pp. 313–320.

[31] T. Nagai and H. Shigeno, "A framework of AP aggregation using virtualization for high density WLANs," in *Proc. of INCoS*, 2011, pp. 350–355.

[32] A. Gudipati, D. Perry, L. E. Li, and S. Katti, "SoftRAN: Software defined radio access network," in *Proc. of ACM HotSDN*, 2013, pp. 25–30.

[33] M. Amer, A. Busson, and I. G. Lassous, "Association optimization based on access fairness for Wi-Fi networks," *Comput. Networks*, vol. 137, pp. 173–188, 2018.

[34] W. Wu, J. Luo, K. Dong, M. Yang, and Z. Ling, "Energy-efficient user association with congestion avoidance and migration constraint in green WLANs," *Wireless Commun. and Mobile Comput.*, 2018.

[35] A. Suresh, S. Mena, D. C. Tomozei, L. Granai, X. Zhu, and S. Ferrari, "Load balancing video clients in enterprise wireless networks," in *Proc. of IEEE WCNC*, April 2018, pp. 1–6.

[36] G. Arvanitakis, T. Spyropoulos, and F. Kaltenberger, "An analytical model for flow-level performance in heterogeneous wireless networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 1488–1501, March 2018.

[37] W. Huang, D. Meng, J. N. Hwang, J. Park, Y. Xu, and W. Zhang, "QoE based SDN heterogeneous LTE and WLAN multi-radio networks for multi-user access," in *Proc. of IEEE WCNC*, April 2018, pp. 1–6.

[38] J. Chen, B. Liu, H. Zhou, Q. Yu, G. Lin, and X. Shen, "QoS-Driven efficient client association in high-density software defined WLAN," *IEEE Trans. Veh. Technol.*, vol. 66, no. 8, pp. 7372–7383, 2017.

[39] D. Sajjadi, M. Tanha, and J. Pan, "Meta-heuristic solution for dynamic association control in virtualized multi-rate WLANs," in *Proc. of IEEE LCN*, 2016, pp. 253–261.

[40] W. Li, S. Wang, Y. Cui, X. Cheng, R. Xin, M. Al-Rodhaan, and A. Al-Dhelaan, "AP association for proportional fairness in multirate WLANs," *IEEE/ACM Trans. Networking*, vol. 22, no. 1, pp. 191–202, 2014.

[41] O. Karimi, J. Liu, and J. Rexford, "Optimal collaborative access point association in wireless networks," in *Proc. of IEEE INFOCOM*, April 2014, pp. 1141–1149.

[42] M. Derakhshani, X. Wang, T. Le-Ngoc, and A. Leon-Garcia, "Airtime usage control in virtualized multi-cell 802.11 networks," in *Proc. of IEEE GLOBECOM*, Dec 2015, pp. 1–6.

[43] M. Derakhshani, X. Wang, T. Le-Ngoc, and A. Leon-Garcia, "Virtualization of multi-cell 802.11 networks: Association and airtime control," *arXiv preprint arXiv:1508.03554*, 2015.

[44] J. Yu and W. C. Wong, "Optimal association in wireless mesh networks," *IEEE Trans. Veh. Technol.*, vol. 64, no. 5, pp. 2084–2096, May 2015.

[45] W. Wong, A. Thakur, and S. H. G. Chan, "An approximation algorithm for AP association under user migration cost constraint," in *Proc. of IEEE INFOCOM*, April 2016, pp. 1–9.

[46] M. Touati, R. El-Azouzi, M. Coupechoux, E. Altman, and J. M. Kelif, "A controlled matching game for WLANs," *IEEE J. Sel. Areas Commun*, vol. 35, no. 3, pp. 707–720, March 2017.

[47] Riverbed, "Steelhead Mobile Optimization," White Paper, 2017. [Online]. Available: http://goo.gl/uqLmmo.

[48] H. Li, Y. Cheng, C. Zhou, and P. Wan, "Multi-dimensional conflict graph based computing for optimal capacity in MR-MC wireless networks," in *Proc. of IEEE ICDCS*, June 2010, pp. 774–783.

[49] Y. Cheng and H. Li, "Optimal capacity planning in multi-radio multi-channel wireless networks," in *Proc. of IEEE ICC*, May 2010, pp. 1–5.

[50] G. Kuperman and E. Modiano, "Providing guaranteed protection in multi-hop wireless networks with interference constraints," *IEEE Trans. Mob. Comput.*, vol. 16, no. 12, pp. 3502–3512, 2017.

[51] S. M. Kala, M. P. K. Reddy, R. Musham, and B. R. Tamma, "Interference mitigation in wireless mesh networks through radio co-location aware conflict graphs," *Wireless Networks*, vol. 22, no. 2, pp. 679–702, 2016.

[52] Y. Shi, Y. T. Hou, J. Liu, and S. Kompella, "How to correctly use the protocol interference model for multi-hop wireless networks," in *Proc. of ACM MobiHoc*, 2009, pp. 239–248.

[53] D. B. Shmoys and É. Tardos, "An approximation algorithm for the generalized assignment problem," *Mathematical programming*, vol. 62, no. 1-3, pp. 461–474, 1993.

[54] L. Li, M. Pal, and Y. R. Yang, "Proportional fairness in multi-rate wireless LANs," in *Proc. of IEEE INFOCOM*, April 2008, pp. 1004–1012.

[55] P. C. Ng and S. C. Liew, "Throughput analysis of IEEE 802.11 multi-hop ad hoc networks," *IEEE/ACM Trans. Networking*, vol. 15, no. 2, pp. 309–322, 2007.

[56] R. Laufer and L. Kleinrock, "The capacity of wireless CSMA/CA networks," *IEEE/ACM Trans. Networking*, vol. 24, no. 3, pp. 1518–1532, 2016.

[57] ITU-R, "IMT traffic estimates for the years 2020 to 2030," White Paper, 2017. [Online]. Available: https://goo.gl/Bkof6G

[58] Cisco, "Cisco Internet Business Solutions, the Future of Mobile Networks," White Paper, 2013. [Online]. Available: https://goo.gl/DLnzNq

[59] Gurobi Optimizer, http://www.gurobi.com.

[60] R. Jain, A. Durresi, and G. Babic, "Throughput fairness index: An explanation," Department of CIS, The Ohio State University, Tech. Rep., 1999.

[61] O. Ekici and A. Yongacoglu, "A novel association algorithm for congestion relief in IEEE 802.11 WLANs," in *Proc. of ACM IWCMC*, 2006, pp. 725–730.

[62] I. Jamil, L. Cariou, and J. F. Helard, "Improving the capacity of future IEEE 802.11 high efficiency WLANs," in *Proc. of ICT*, May 2014, pp. 303–307.

[63] I. Jamil, L. Cariou, and J. F. Hélard, "Efficient MAC protocols optimization for future high density WLANs," in *Proc. of IEEE WCNC*, March 2015, pp. 1054–1059.

[64] Cisco, "Cisco Wireless Control System Configuration Guide," White Paper, 2018. [Online]. Available: https://goo.gl/G7jBSG

[65] D. Bertsimas, V. F. Farias, and N. Trichakis, "The price of fairness," *Operations research*, vol. 59, no. 1, pp. 17–31, 2011.

[66] Ryu framework, https://osrg.github.io/ryu/.

[67] Open vSwitch, https://goo.gl/6PUc73.

[68] Hostpad, https://github.com/hschaa/hostapd.git.

[69] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Mar. 2008.

[70] PCEngines Alix System Boards, http://pcengines.ch/alix.htm.

[71] tshark, https://goo.gl/TaTU.

[72] Ettus Research, https://kb.ettus.com/USRP2.

[73] Cacti, https://www.cacti.net/.

[74] S. Zehl, A. Zubow, and A. Wolisz, "hMAC: Enabling Hybrid TDMA/CSMA on IEEE 802.11 Hardware," Telecommunication Networks Group, Technische Universität Berlin, Tech. Rep. TKN-16-004, November 2016.

[75] Linux Advanced Routing and Traffic Control, http://lartc.org/manpages/tc.txt.

[76] P. Dely, A. Kassler, and N. Bayer, "Openflow for wireless mesh networks," in *Proc. of ICCCN*, July 2011, pp. 1–6.

[77] Cisco, "The Cisco Visual Networking Index Global Mobile Data Traffic Forecast, update 2017–2021," White Paper, 2016. [Online]. Available: https://goo.gl/7pdlTC

[78] I. Broustis, K. Pelechrinis, D. Syrivelis, S. V. Krishnamurthy, and L. Tassiulas, "Quantifying the overhead due to routing probes in multi-rate WMNs," in *Proc. of IEEE WCNC*, April 2010, pp. 1–6.

[79] J. Schulz-Zander, S. Schmid, J. Kempf, R. Riggio, and A. Feldmann, "LegoFi the Wi-Fi building blocks: The case for a modular Wi-Fi architecture," in *Proc. of ACM MobiArch*, 2016, pp. 7–12.

[80] A. Detti, C. Pisa, S. Salsano, and N. Blefari, "Wireless mesh software defined networks (wmSDN)," in *Proc. of IEEE WiMob*, 2013, pp. 89–95.

[81] D. Sajjadi, M. Tanha, and J. Pan, "A comparative study of channel switching latency for conventional and SDN-based routing in multi-hop multi-radio WMNs," in *Proc. of IEEE CCNC*, 2016, pp. 330–334.

[82] D. Sajjadi, R. Ruby, M. Tanha, and J. Pan, "Fine-grained access provisioning via joint gateway selection and flow routing on SDN-aware Wi-Fi mesh networks," in *Proc. of IEEE WiMob*, Oct 2017, pp. 1–10.

[83] H. Huang, P. Li, S. Guo, and W. Zhuang, "Software-defined wireless mesh networks: architecture and traffic orchestration," *IEEE Network*, vol. 29, no. 4, pp. 24–30, July 2015.

[84] A. Abujoda, D. Dietrich, P. Papadimitriou, and A. Sathiaseelan, "Software-defined wireless mesh networks for Internet access sharing," *Comput. Networks*, vol. 93, pp. 359–372, 2015.

[85] A. Amokrane, R. Langar, R. Boutaba, and G. Pujolle, "Online flow-based energy efficient management in wireless mesh networks," in *Proc. of IEEE GLOBECOM*, Dec 2013, pp. 329–335.

[86] M. Garetto, P. Giaccone, and E. Leonardi, "On the effectiveness of the 2-hop routing strategy in MANETs," in *Proc. of IEEE ICC*, June 2007, pp. 3108–3113.

[87] S. Chu and X. Wang, "MIMO-aware routing in wireless mesh networks," in *Proc. of IEEE INFOCOM*, March 2010, pp. 1–9.

[88] B. He, B. Xie, and D. P. Agrawal, "Internet gateway deployment optimization in a multi-channel multi-radio wireless mesh network," in *Proc. of IEEE WCNC*, March 2008, pp. 2259–2264.

[89] F. Li, Y. Wang, X.-Y. Li, A. Nusairat, and Y. Wu, "Gateway placement for throughput optimization in wireless mesh networks," *Mobile Networks and Appl.*, vol. 13, no. 1-2, pp. 198–211, 2008.

[90] Q. Xin and Y. J. Wang, "Gateway selection scheme for throughput optimization in multi-radio multi-channel wireless mesh networks," in *Proc. of MSN*, 2009, pp. 187–195.

[91] B. Aoun, R. Boutaba, Y. Iraqi, and G. Kenward, "Gateway placement optimization in wireless mesh networks with QoS constraints," *IEEE J. Sel. Areas Commun*, vol. 24, no. 11, pp. 2127–2136, Nov 2006.

[92] K. Papadaki and V. Friderikos, "Gateway selection and routing in wireless mesh networks," *Comput. Networks*, vol. 54, no. 2, pp. 319–329, 2010.

[93] D. Banfi, O. Mehani, G. Jourjon, L. Schwaighofer, and R. Holz, "Endpoint-transparent multipath transport with software-defined networks," in *Proc. of IEEE LCN*, Nov 2016, pp. 307–315.

[94] M. Fu and F. Wu, "Investigation of multipath routing algorithms in software defined networking," in *Proc. of ICGI*, Aug 2017, pp. 269–273.

[95] M. Rademacher, F. Siebertz, M. Schlebusch, and K. Jonas, "Experiments with OpenFlow and IEEE 802.11 Point-to-Point links in a WMN," in *Proc. of ICWMC*, 2016, pp. 100–105.

[96] M. Tanha, D. Sajjadi, and J. Pan, "Demystifying failure recovery for software-defined wireless mesh networks," in *Proc. of IEEE NetSoft, PVE-SDN Workshop*, June 2018, pp. 488–493.

[97] P. Patil, A. Hakiri, Y. Barve, and A. Gokhale, "Enabling software-defined networking for wireless mesh networks in smart environments," in *Proc. of IEEE NCA*, Oct 2016, pp. 153–157.

[98] A. Hurtado-Borràs, J. Palà-Solé, D. Camps-Mur, and S. Sallent-Ribes, "SDN wireless backhauling for small cells," in *Proc. of IEEE ICC*, June 2015, pp. 3897–3902.

[99] Y. Peng, L. Guo, Q. Deng, Z. Ning, and L. Zhang, "A novel hybrid routing forwarding algorithm in sdn enabled wireless mesh networks," in *Proc. of HPCC*, Aug 2015, pp. 1806–1811.

[100] M. Labraoui, M. M. Boc, and A. Fladenmuller, "Software defined networking-assisted routing in wireless mesh networks," in *Proc. of ACM IWCMC*, 2016, pp. 377–382.

[101] H. Li, J. Zhang, Q. Hong, H. Zheng, and J. Zhang, "Digraph-based joint routing and resource allocation in software-defined backhaul networks," in *Proc. of IEEE International Workshop on CAMAD*, June 2017, pp. 1–5.

[102] Y. Le, L. Ma, W. Cheng, X. Cheng, and B. Chen, "Maximizing throughput when achieving time fairness in multi-rate wireless LANs," in *Proc. of IEEE INFOCOM*, March 2012, pp. 2911–2915.

[103] Ericsson, "Ericsson Mobility Report: On the Pulse of the Networking Society," Tech. Rep., November 2016. [Online]. Available: http://goo.gl/UozTZH.

[104] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," *Wireless networks*, vol. 11, no. 4, pp. 471–487, 2005.

[105] G. P. McCormick, "Computability of global solutions to factorable nonconvex programs: Part I: Convex underestimating problems," *Mathematical Programming*, vol. 10, no. 1, pp. 147–175, 1976.

[106] D. Papadimitriou, D. Colle, and P. Demeester, "Mixed-integer optimization for the combined capacitated facility location-routing problem," in *Proc. of IEEE DRCN*, March 2016, pp. 14–22.

[107] D. A. Spielman and S.-H. Teng, "Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time," *Journal of the ACM*, vol. 51, no. 3, pp. 385–463, 2004.

[108] K. H. Borgwardt, *The Simplex Method: a probabilistic analysis*. Springer Science and Business Media, 2012, vol. 1.

[109] R. A. Fisher, F. Yates *et al.*, *Statistical Tables for Biological, Agricultural and Medical Research*. Edinburgh, 1938.

[110] D. S. Hochbaum, Ed., *Approximation Algorithms for NP-hard Problems*. Boston, MA, USA: PWS Publishing Co., 1997.

[111] G. Even, M. Rost, and S. Schmid, "An approximation algorithm for path computation and function placement in SDNs," *arXiv preprint arXiv:1603.09158*, 2016.

[112] M. Mitzenmacher and E. Upfal, *Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis*. Cambridge university press, 2017.

[113] J. Pan, Y. T. Hou, L. Cai, Y. Shi, and S. X. Shen, "Topology control for wireless sensor networks," in *Proc. of ACM Mobicom*, 2003, pp. 286–299.

[114] Linux Foundation Wiki, https://goo.gl/25x8zV.

[115] R. Santos and A. Kassler, "Small cell wireless backhaul reconfiguration using software-defined networking," in *Proc. of IEEE WCNC*, March 2017, pp. 1–6.

[116] Mininet WiFi, https://goo.gl/3sUAqJ.