AMERICAN UNIVERSITY OF BEIRUT

MULTI-CHANNEL IEEE 802.11 MAC LAYER SCHEDULING FOR ULTRA-DENSE IOT NETWORKS

by
BASSEL ABOU ALI MODAD

A thesis
submitted in partial fulfillment of the requirements
for the degree of Master of Engineering
to the Department of Electrical and Computer Engineering
of the Faculty of Engineering and Architecture
at the American University of Beirut

Beirut, Lebanon
July 16, 2019

AMERICAN UNIVERSITY OF BEIRUT

MULTI-CHANNEL IEEE 802.11 MAC LAYER SCHEDULING FOR ULTRA-DENSE IOT NETWORKS

by

BASSEL ABOU ALI MODAD

Approved by:

Dr. Zaher Dawy, Professor Electrical and Computer Engineering	Advisor
Dr. Hassan Ali Artail, Professor Electrical and Computer Engineering	Member of Committee
Dr. Ibrahim Abou Faycal, Professor Electrical and Computer Engineering	Member of Committee

Date of thesis defense: July 16, 2019

AMERICAN UNIVERSITY OF BEIRUT

THESIS, DISSERTATION, PROJECT RELEASE FORM

Student Name: Abou Ali Modad Bassel
Last First Middle

☒ Master's Thesis ○ Master's Project ○ Doctoral Dissertation

☐ I authorize the American University of Beirut to: (a) reproduce hard or electronic copies of my thesis, dissertation, or project; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes.

☒ I authorize the American University of Beirut, to: (a) reproduce hard or electronic copies of it; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes after: (One) ___ **year from the date of submission of my thesis, dissertation or project.**
Two ___ years from the date of submission of my thesis , dissertation or project.
Three ___ years from the date of submission of my thesis , dissertation or project.

Bassel 16/7/2019
Signature Date

This form is signed when submitting the thesis, dissertation, or project to the University Libraries

Acknowledgements

I would like to thank General Electric Research for the funding of the inital project. I would also like to thank my advisor Dr. Zaher Dawy, for his help and guidance throughout the thesis, Dr. Elias Yaacoub for his help and useful insights, and Dr. Abdallah Kassir for his collaboration and help in the practical coding implementation part of the thesis.

I would also like to thank my family for their support and encouragement throughout these two years. I would like to thank my friends: Nathalie Al Khatib, for her support and encouragement. Mohammad Ali Mhaidly for the interesting technical debates about my work, which gave me a different perspective on it, and for the good times of rest outside of academics and research. Hussam Abdel Khalek for the continual support in all areas in and out of academia, the good times of rest and mental distractions, and the great conversations. And Alaa Khaddaj for the very interesting, stimulating and intellectually challenging debates we had about academics, and for his help in some parts of the initial mathematical formulation.

An Abstract of the Thesis of

Bassel Abou Ali Modad for Master of Engineering
Major: Electrical and Computer Engineering

Title: Multi-Channel IEEE 802.11 MAC Layer Scheduling for Ultra-Dense IoT Networks

Over the last decades, wireless networks have experienced major growth in their enabling technologies and global penetration. One of the more recent components of this evolution is the Internet of Things (IoT), where different types of everyday objects are connected to the internet for purposes such as sensing and monitoring with applications that span a wide range of vertical sectors such as e-health, environment, and transportation. The IoT paradigm is seeing an exponential growth and is predicted to reach tens of billions of devices by 2030. To meet the high IoT connectivity demands, the IEEE 802.11 WiFi protocol is constantly being improved, yet performance bottlenecks still exist especially in ultra-dense network scenarios such as concerts, sports events, exhibitions, etc. In this thesis, we propose a practical approach that is compliant with the IEEE 802.11 standard to accommodate periodic IoT traffic in scenarios with very large number of users and critical delay-sensitive application data. The proposed approach combines data/control channel separation with optimized time division scheduling, including an extension to maintain a certain level of fairness among different IoT devices. Using NS-3 network simulation, the proposed approach is shown to achieve notable performance gains compared to the IEEE 802.11n WiFi standard in various scenarios with different system and design parameters.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Wireless networks have experienced increasing growth ever since their conception, with massive increases in the number of users each year and an increase in data traffic. These conditions demand an increase in capacity and supported users in the networks. Most of the traffic is generated indoors, thus wireless IEEE 802.11 WiFi networks are important in addressing the capacity issues, along with having the advantage of being cost effective and easily deployable. WiFi traffic is expected to account for 53% of global internet traffic by 2021 according to a 2018 Cisco forecast [1]. It is projected that the number of WiFi capable devices will be much greater than the number of cellular capable ones. Most of the devices are expected to be Internet of Things (IoT) devices characterized by their low data rates, low power consumption, small packet sizes, delay tolerance and their massive number. These devices are used for many applications such as sensing, monitoring, surveillance, healthcare and smart appliances [2]. Recently, however, there has been an increase in IoT applications where the data is critical and delay intolerant, such as wearable medical monitoring (e.g., heart rate, blood pressure, temperature and muscle activity). Coupled with the increase in device numbers, these new constraints require that delays due to medium access, network protocols, as well as packet collisions be minimal to guarantee a certain quality-of-service (QoS) to the users.

In this perspective, according to the current 802.11 medium access control (MAC) protocol, all devices contend to use the medium via the random access scheme: Carrier-Sensing-Multiple-Access with Collision Avoidance (CSMA/CA). The 802.11 MAC protocol is divided into an association stage between access points (APs) and mobile stations (STAs) with authentication, and a data transmission stage based on contention and CSMA/CA. Briefly, the association stage consists of a four-way handshake between the STA and AP where authentication and association happen. The data transmission stage consists of contention for medium access using a random back-off counter after it is sensed free followed by data transmission then waiting for the reception of an acknowledgement (ACK) frame. If the latter is not received, collision is assumed and the frame is retrans-

mitted. CSMA/CA offers a reasonable tradeoff between scalability, robustness and the need for a centralized controller. This random access creates a bottleneck and greatly increases congestion in dense IoT scenarios which is detrimental to the critical, delay intolerant applications.

The theoretical efficiency of CSMA/CA scheme is around 60 to 70% [3, Ch. 4.1.2], but in practice the efficiency is observed to be less than 50% [3, Ch. 4.1.2]. Clearly this scheme cannot be used as it is in dense environments, and smarter schemes with better time resource allocation are required. Although several amendments to the 802.11 protocol have been made, only recently did they address the issue of massive IoT connectivity and changing the medium access protocol.

In this thesis, we consider the problem of enhancing the 802.11 MAC protocol for scalability in dense IoT environments without modifying the core functionalities of the standard. This means that the only modifications allowed are settings of certain MAC parameters (such as RTS/CTS threshold, contention window, and retry counters) and transmission of a few additional bytes in control frames. We assume that the data to be transmitted is small in size, not exceeding a few hundred bytes, and is delay-sensitive. We propose a multi-channel hybrid CSMA/CA with adaptive Time-Division-Multiple-Access (TDMA) enhancement, where one channel is used only for user association using CSMA/CA and the others are used for data transmission using TDMA to avoid interfering with the associating users. We formulate the scheduling on the TDMA channels as an optimization problem where we find which users out of the associated ones can be scheduled and their optimal slot time. In addition to that, we address the problem of fair scheduling on the TDMA channel and propose two different algorithms for fairnes; the first being a proportionally fair algorithm, and the second being a time-slot allocation based fairness algorithm. Our proposed algorithms are unique in that they are standard compliant, requiring minimal additions in some frame structures, are geared towards dense IoT environments, minimize delays, use non-overlapping WiFi channels to enhance scalability in addition to using adaptive time slot durations to accommodate more users.

This thesis is organized as follows. In chapter 2, we present the background on the IEEE 802.11 protocol, its important amendments, and a summary on the existing literature on MAC protocol scalability and fair scheduling in wireless networks. In chapter 3, we present our system model and assumptions. We then present the problem to solve and our proposed solution in chapter 4. Chapter 5 details the implementation and simulation of the protocol and the simulation software we used. Results are presented in chapter 6 along with their analysis. Finally, we conclude in chapter 7 and present possible extensions to our work.

Chapter 2

Literature Review

The objective of this chapter is to investigate some previous work related to the thesis topic. We start by presenting background information on the 802.11 MAC protocol and ongoing improvements on it. Then we present the literature on different modified MAC protocols proposed to improve WiFi performance for our applications and similar ones. Finally, we present the literature on scheduling problems.

2.1 IEEE 802.11 Standard

2.1.1 Protocol Background

The 802.11 MAC protocol consists of an association stage and a data transmission stage [4]. In the association stage, the STA goes through three phases: not authenticated or associated, authenticated but not associated, authenticated and associated. The process starts with the STA sending a broadcast probe request frame to all APs to discover nearby networks, indicating its supported data rates and capabilities. The APs check if they have at least one common data rate with the STA, then they send a probe response frame indicating the network name (SSID), AP supported rates, encryption used, and other capabilities. The STA chooses the most compatible AP, sends an ACK frame, then the STA and AP exchange two authentication frames. Now the STA is authenticated but not yet associated, it sends an association request frame containing its chosen encryption types, and other 802.11 capabilities. The AP sends an ACK, checks if these capabilities match its own, creates an association ID (AID) for the device and sends it back in an association response frame whose status code field is changed to "success" [4]. The AID field is a 16 bit field in the header of control frames that plays the role of both frame duration and ID. When used to indicate AID, the two most significant bits are set to 1. The maximum AID allowed in the standard is 2007 [4]. Finally the STA sends an ACK after receiving the response

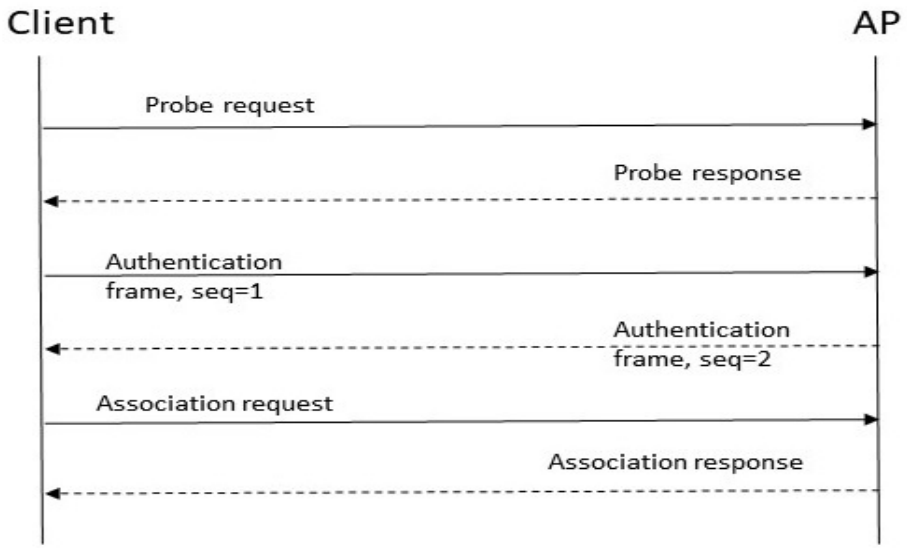


Figure 2.1: 802.11 Association Stage

and the data transmission stage starts. The stage is illustrated in Figure 2.1.

In the data transmission stage, CSMA/CA is used by the STAs to manage medium access. CSMA/CA uses a binary exponential back-off counter when the channel is sensed idle for more than DIFS (Distributed Inter-Frame Spacing) time ($50\mu s$), the counter is chosen at random uniformly between 0 and a contention window size $CW - 1$, where CW ranges from CW_{min} to CW_{max} [4]. The countdown is followed by a Request-to-Send/Clear-to-Send (RTS/CTS) frames handshake with a wait of SIFS (Short Inter-Frame Spacing) $10\mu s$ after each. The RTS frame contains a duration field which indicates to the AP the amount of time the STA will use the medium. If other devices sense the medium during an RTS/CTS handshake they set their Network Allocation Vector (NAV) counter to the duration specified and refrain from sensing the medium until the counter expires. After reception of the CTS by the STA using the medium and a wait of SIFS, data transmission starts followed by an acknowledgement. If the medium is sensed busy during back-off, the counter is frozen and resumes after DIFS seconds from the end of the on-going transmission. For the devices that can sense the RTS/CTS transmission but cannot demodulate the frame and read the duration field, they wait EIFS (Extended Inter-Frame Spacing) time before sensing the medium again [4]. For each unsuccessful transmission CW is doubled until it reaches CW_{max}. For every successful transmission CW is reset to CW_{min}. If no acknowledgement is received after SIFS seconds from the end of data transmission, the sender assumes the frame was collided and enters back-off again. The stage is briefly illustrated in Figure 2.2.

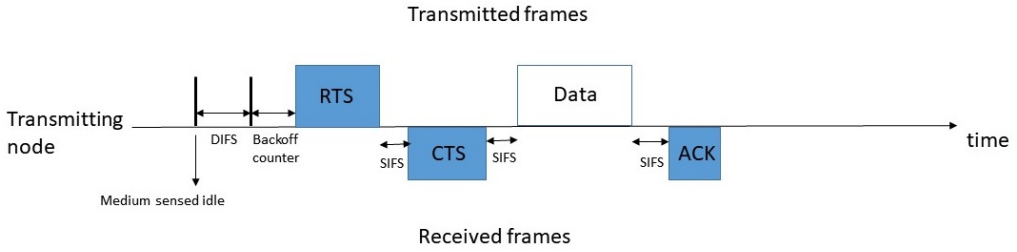


Figure 2.2: 802.11 CSMA/CA Data Stage

2.1.2 Amendments

The IEEE 802.11 protocol is always being updated to meet the new requirements of wireless applications. Many amendments exist that mostly aim to increase user throughput through PHY layer enhancements, however, only recently did the amendments aim to improve throughput for dense environments by enhancing the MAC medium access. Specifically, the 802.11ah amendment published in 2017 addresses massive registration of Machine-to-Machine (M2M) devices [5], and the 802.11ax amendment to be published in 2019 is expected to use Orthogonal Frequency Division Multiple Access (OFDMA) to enhance medium access efficiency [6].

First, the 802.11ah standard proposes a slotted CSMA/CA mechanism to efficiently register a massive number of M2M devices using frequencies below 1 GHz [4]. It modifies the AID field to allow up to 8191 ($2^{13} - 1$) devices on the sub-1GHz bands, however, the number of STAs using the traditional 2.4 or 5 GHz is still limited to 2007. To reduce signaling, the sub-1GHz STAs are organized in hierarchical groups according to the bits in the AID field, where each group is assigned a time period (called restricted access window, or RAW), waits for its associated beacon, then each STA in the group contends for the channel during specific uplink slots and receives data from the AP in downlink slots [5]. This is known as the Centralized Authentication Control (CAC) mechanism.

Second and more importantly, the 802.11ax amendment operates on 2.4 and 5 GHz bands as well as those between 1 and 7 GHz when they become available. It is expected to bring a four times increase in throughput over the current 802.11ac and be used for dense scenarios [6]. It is expected to use collision-free MAC protocols such as CSMA with enhanced collision avoidance (CSMA/ECA) which uses a deterministic back-off counter after successful transmissions [6]. OFDMA and multi-user MIMO (MU-MIMO) in uplink and downlink directions are also expected to be used. The former technique divides the channel bandwidth into

smaller channels and different time slots called Resource Units (RUs), then assigns subsets of them to different users, which makes them able to transmit without collision [7]. MU-MIMO enables spatial multiplexing of STAs and APs using multiple antennas in both uplink and downlink directions, which also enables multiple users to transmit at the same time.

The protocol has two modes of operation: a single-user (SU) mode where STAs and AP exchange data one at a time after successful CSMA/ECA, and a multi-user (MU) mode that uses the new OFDMA and MU-MIMO features [8]. In the MU mode the AP decides how to allocate the OFDMA resources and the number of RUs. Resource allocation happens after each OFDMA frame, which consists of a Trigger Frame (TF) and subsequent uplink and downlink data transmission [8].

An STA can only transmit if it is assigned an RU by the AP. Most scheduling information, such as Modulation and Coding Scheme, bandwidth, number of spatial streams and RU assignment, are moved to the PHY frame header to reduce the amount of signaling. There are two ways to access the medium in uplink OFDMA, namely deterministic access and random access. The random access is used when an STA is unassociated or the AP does not have information about the STA's buffer. The AP specifies which RUs the STAs compete for in the Trigger Frame for random access TF-R. The competition uses a backoff procedure: Each STA has a backoff counter chosen at random between 0 and OFDMA-CW. If the counter value is less than the number of RUs allocated for random access (say, M), the STA randomly selects one RU and transmits its frame. Otherwise, the STA decrements its counter by M and waits for the next TF-R. If the transmission fails, the STA doubles CW until CW_{max}, if it is successful, CW is reset to the minimum (maximum and minimum values of CW are set by AP in beacons) [9].

After acquiring an RU, the STAs use deterministic OFDMA access on those RUs in all subsequent frames unless a change is made by the AP. However, random access on specific RUs is still used for small frames necessary for resource allocation, such as buffer status report (BSR) and channel state information (CSI) [8].

With all the benefits of 802.11ax, a great increase in complexity is required on both the STA and AP sides especially for scheduling and resource allocation. This issue is more complex than in LTE (where OFDMA is also used) due to the strict requirements on RUs [8].

2.2 MAC Protocol Modifications for Scalability

Most work that does not involve creating new PHY and MAC layers for WiFi modifies the existing CSMA/CA or proposes hybrid centralized-distributed schemes. In this section we present a summary of the literature on these MAC protocol

modifications for IEEE 802.11 and other standards.

In [10] the authors propose an enhancement to 802.11ah in the form of a hybrid slotted CSMA/CA-TDMA protocol for registration of a massive number IoT devices. It also takes into account simultaneous registration of a large number of devices. The protocol uses a slotted contention period where devices use CSMA/CA in smaller time slots to transmit authentication requests, and a slotted TDMA period which is also divided into smaller slots to receive authentication responses and association request/response frames. The contention window size in CSMA/CA is chosen according to a Sift geometric probability distribution which favors larger values of CW, its advantage is that as the number of devices increases, they are more likely to select a larger CW value, thus avoiding congestion at the beginning of CSMA/CA slots. The protocol also finds the optimal duration of the contention period, TDMA period and optimal parameters of the CAC mechanism. Simulation results showed a significant reduction in registration time compared to 802.11ah, 64% reduction in registration time compared to contention free transmission.

In [11], a traffic adaptive 802.15.4 MAC protocol for multi-hop IoT networks is proposed. It uses duty-cycled communication via variable duration periodic wake-up periods to transmit one packet in every period using CSMA/CA. To transmit extra packets, dynamic TDMA slot number allocation is used based on the transmitting node's queue length (hence the adaptability to traffic). Multi-channel communication is used to re-transmit failed packets in the next wake-up period.

In [12], the authors propose a hybrid MAC protocol for M2M networks with heterogeneous applications. The protocol is divided into two stages, the first one is a fair priority-based channel access using p-persistent CSMA, and the second stage is a TDMA data transmission stage with fixed slot durations where only the devices whose transmission requests were ACKed in the CSMA stage are granted slots. The devices are organized into groups, each with a contention probability which directly corresponds to its priority. To achieve fairness between nodes, after each failed channel access the node increments its probability, hence its channel access priority, by a factor raised to the number of failed accesses and resets it to the original value after each success. The authors derive the optimal initial contention probability, incremental factor and durations of both stages, that maximize channel utility.

In [13], a similar hybrid MAC for virtualized WLANs is presented with the goal of the scheduling is to maximize the total network throughput in each superframe. In virtualized WLANs, users from different service providers can share the same network infrastructure. The problem is formulated as an optimization solved at the AP at each superframe with constraints on the number of TDMA users and access time of each service provider. The optimization is solved by an iterative algorithm using Complimentary Geometric Programming (CGP). A beacon frame is transmitted at the start of the superframe to indicate TDMA

scheduling and p values of CSMA. Results showed a doubling in network throughput (measured in packets per superframe) relative to p-persistent CSMA.

An approach other than hybrid TDMA-CSMA schemes is presented in [14], it is an opportunistic medium access scheme for 802.11 designed to improve throughput. It also addresses different packet priorities. The scheme is an enhancement of CSMA/CA based on PHY layer carrier sensing of different transmissions, each node sets one of received signal strengths as reference and changes its back-off parameters (CW_{min} and Arbitration Inter-Frame Spacing $AIFS$) when it hears a transmission close to the reference. It is assumed that each signal sensed corresponds to a different device on the network. The reference selection process is based on listening on the channel for a time t and collecting a set of signal strengths, then computing their average and choosing the one closest to the average as the reference. The back-off parameters change according to the packet priority, where those with higher priority are given smaller CW_{min} and $AIFS$. Results showed good improvement of throughput and 60% reduction of collisions compared to legacy 802.11, however, the collisions were still high even for small numbers of users (10 and 60 users).

In [15], the authors enhance the 802.11ah MAC by making the AP estimate the number of active STAs at each beacon interval, then finds the optimal RAW duration using the estimated number and estimated medium access success probability. Results showed a success probability greater than 60% for as much as 5000 devices.

An analytical framework for modeling hybrid MAC protocols is presented in [16], it uses a Markov Decision Process (MDP) to determine the best policy: whether a node sends its buffered frames (more than one) during the slotted CSMA/CA period, TDMA period, during both, or defer to the next superframe. The MDP is extended to take into account channel fading. Its goal is to minimize energy consumption and maximize throughput of each device, by making the reward in the MDP a function of throughput, buffer state, energy consumption and bandwidth cost to penalize nodes that occupy time slots with no frames to transmit. Results showed an improvement over usual hybrid MAC in terms of energy consumption and packet delivery ratio, however it requires more computational effort and information about buffer states and packet arrival rates of all nodes.

We notice that most of the proposed protocols in the literature use a hybrid CSMA-TDMA on the same channel and divide the superframe into two stages. To achieve more scalability and minimize collisions, in this thesis we dedicate TDMA data transmission to the entire superframe, moving the CSMA/CA part to another non-overlapping channel. Furthermore, we make use of the third non-overlapping channel which is also used for TDMA. The CSMA/CA channel will be used for association only, then devices switch to the TDMA transmission channels at the beginning of the next superframe. Each TDMA slot will be of variable length according to each user's channel conditions.

2.3 Scheduling Problems

In wireless networks, the aim of Scheduling and Resource Allocation (SRA) is to achieve optimized performance such that a certain network performance metric is maximized, subject to constraints on users and resources [17, 18, 19]. Such a metric is represented by a utility function, usually denoted by $U_i(r_i)$ for user i with decision variable r_i, which expresses a certrain requirement, e.g., energy consumption [16], [20], delay or latency [21], sum-throughput [22], or any combination of those while some others are regarded as constraints.

The network utility function is represented by $U(\mathbf{r})$, where $\mathbf{r}$ represents the vector of decision variables for all users. $U(\mathbf{r})$ is related to the individual utilities by either their sum or their average. We assume for the rest of this thesis that the relation is the sum of individual utilities.

$$U(\mathbf{r}) = \sum_{i=1}^{N} U_i(r_i) \tag{2.1}$$

Therefore, SRA problems are formulated as:

$$\max_{\mathbf{r}} \quad U(\mathbf{r}) \tag{2.2}$$

With additional constraints depending on the problem.

The most common network metric used is sum-throughput over all users. Throughput maximization problems are formulated as maximizing the sum-rate of the network where the individual rates are given by the Shannon capacity [17], or using discrete rates as in [23]. They can also be formulated as maximizing the throughput region [18]. The individual utility function is $U_i(R_i) = R_i$, where R_i is the rate. The constraints can be either on delay, number of resources allocated, probability of outage, or any combination of those. Due to the combinatorial nature of these problems, their complexity is usually NP-Hard since they require to exhaustively try all combinations. They are usually solved using sub-optimal algorithms. For example in [17], the problem of RU allocation for sum-throughput maximization in 802.11ax OFDMA is studied. A relaxed version of the problem, by removing the constraint that each user has at most one RU, is solved optimally using a divide-and-conquer algorithm. Then, a greedy sub-optimal algorithm and a recursive scheduling algorithm are proposed to solve the original problem.

The problem with sum-throughput maximization is that users with higher acheiveable rate (better channel) are always included in the scheduling, which results in the exclusion of those with lower rates (worse channels), thus favoring the better users and resulting in unfairness. Therefore, proportionally fair (PF) scheduling is used [24], [25], where the network objective function is maximizing the geometric mean of user rates, i.e., the product of the rates. The idea comes from game theory, where users compete for resources in a Nash Bargaining Problem [25]. This is equivalent to maximizing the sum of the logarithms of the rates

[18], and is a result used widely in the literature, e.g., [24],[26],[27]. The utility function of each user then becomes:

$$U_i(R_i) = \ln(R_i) \tag{2.3}$$

Another form of fair SRA is max-min fairness [24], which tries to maximize the utility for the users whose utility function is minimal first, then proceeding with the rest, thus prioritizing those with worse conditions. A rate allocation is max-min fair, if for any user i increasing the rate R_i results in a decrease of another user j's rate R_j [28]. It is shown that max-min fairness is achieved when the utility function is of the form [24]:

$$U_i(R_i) = -\frac{R_i^{-\alpha}}{\alpha}, \quad \alpha > 0 \tag{2.4}$$

where α determines the degree of fairness, and having α go to ∞ achieves max-min fairness [28].

As for SRA for minimal energy consumption, The work in [29] addresses resource allocation and scheduling for minimal energy consumption in MIMO-OFDMA systems with QoS constraints on each user. The QoS constraints are treated as making the sum of resource blocks allocated to a user greater than a minimum rate required to serve the user. The goal is to find the optimal resource allocation for all users such that the bits-per-Joules metric is maximized.

All the above problems can be solved either analytically [23], or using sub-optimal low-complexity algorithms [26]. We notice that none of the above works explicitly optimize for the largest possible number of served users, and most of the protocols are not based on practical modifications of existing wireless standards. In this thesis, we formulate our problem in a way that incorporates rate allocation, maximizes the number of users and at the same time be practically feasible, as will be shown in the next two chapters. We also address the fairness issue in ultra-dense scenarios using two different approaches.

Chapter 3

Network Model and Operation

The network model we consider is a single-hop star topology network with one AP and N STAs that are allowed to move inside the AP's transmission range. The devices are indexed by $i = 1, ..., N$. The topology is shown in Figure 3.1. The association stage is separated from the data transmission stage by placing each on different non-overlapping channels in order to minimize interference and collisions with the transmitted data, and achieve more scalability. Three orthogonal WiFi channels are used: one for association and two for data transmission. Devices use CSMA/CA on the association channel to get an association ID (AID) then switch over to their data assigned transmission channel and start sending their data on the time slot corresponding to their AID periodically. An adaptive TDMA scheme is used on the data channels where time is divided into superframes (or cycles) of duration T_c seconds known by all STAs. Each superframe is then slotted into time slots of variable duration T_{s_i} seconds where $i = 1, ..., N$. Devices use the time slot corresponding to their AID. We allow the time slots to be of different lengths to take into account the different channel conditions for each device. At the end of each superframe, there is a beacon period of duration T_{beacon} seconds before the beginning of the next superframe, during which the AP sends a beacon frame advertising any changes in user scheduling and synchronizing the devices' clocks. The superframe structure is shown in Figure 3.2.

3.1 Physical Layer Model

At the physical layer, the wireless channel is assumed to have flat block Rayleigh fading, it remains fixed during each time slot but varies independently from one time slot to another, as in [16]. The channel statistics for user i in their time slot are denoted as G_i and include the exponential distribution of power and distance-dependent pathloss. Since in TDMA there is only one active user per time slot, we assume that the only interference is from other networks, which is intercell interference denoted as I_{inter}, thus the received uplink Signal-to-Interference-plus-

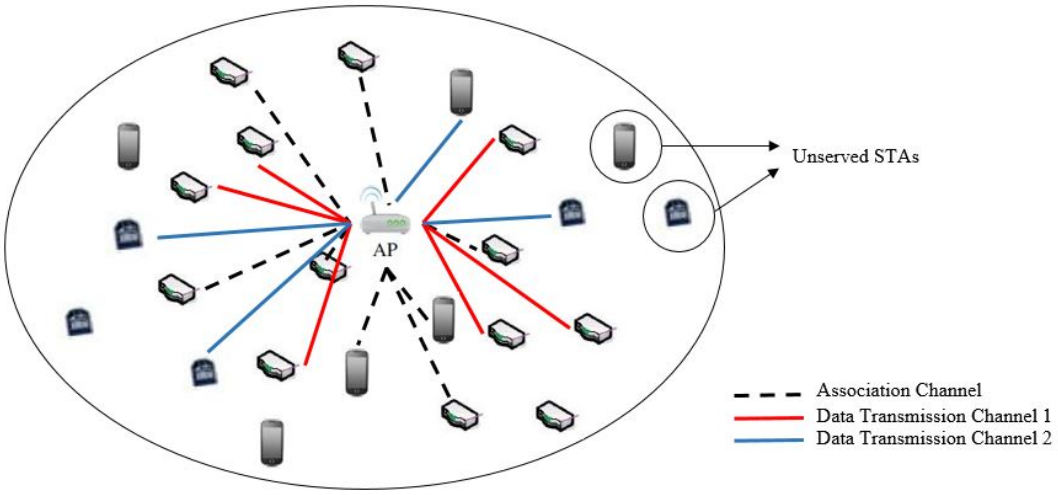


Figure 3.1: System Model

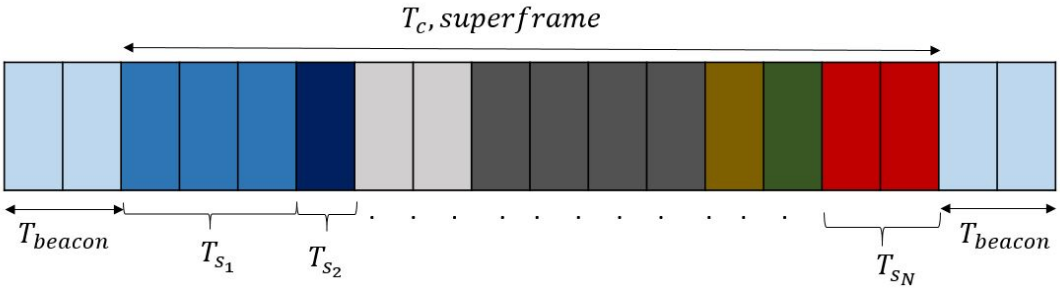


Figure 3.2: Superframe structure

Noise Ratio (SINR) at the AP, of user i is denoted as γ_i and given by

$$\gamma_i = \frac{P_i G_i}{\sigma^2 + I_{inter}} \tag{3.1}$$

Where P_i is the transmitted power of user i. We assume that users transmit at the same maximal power P_{max} in all their time slots, as in [27] and [30]. σ^2 is the additive white noise power. I_{inter} is a non-deterministic term equal to the sum of received powers from all interferers (outside STAs or APs): $I_{inter} = \sum_{k \in \mathcal{I}} P_k G_k$, $\mathcal{I}$ being the set of interferers. Due to the block fading assumption, this term does not change during the same time slot. For a user to have a successful transmission, the SINR at the AP should be greater than a certain threshold γ_{th}:

$$\gamma_i = \frac{P_{max} G_i}{\sigma^2 + \sum_{k \in \mathcal{I}} P_{max} G_k} \geq \gamma_{th} \tag{3.2}$$

The value of the threshold is dependent on many factors such as the bit rate, modulation, coding scheme, target bit error rate [27]. In our work, we take the threshold to be common for all users, and set it as the smallest SNR value required to achieve the smallest possible PHY data rate. The relation between the SNR and discrete data rate is given by a table mapping each achievable rate to its minimum required SNR.

In the downlink (DL) direction, we assume that the AP always sends its frames (ACK and beacon) at the lowest possible rate $R_{min} \in \mathcal{R}$ and highest power P_{max} to account for the worst channel conditions, ensure successful transmission of its frames and make T_{ACK} and T_{beacon} constant [30].

The STAs transmit their data periodically one at a time in their assigned slot every T_c seconds with one packet of fixed small size D bytes every superframe transmitted at a physical layer data rate R_i, $i = 1, ..., N$. The data rate R_i takes values in the set $\mathcal{R}$ of allowed rates in the 802.11 PHY protocol used, where each of them corresponds to a different modulation and coding scheme (MCS) number. We assume that the network is in a saturated condition, meaning that all users always have data to send in their time slot, and a user does not send only if the AP decides it.

3.2 MAC Layer Model

At the MAC layer, we define the slot time T_{s_i} to be the time required to complete one frame transmission and completely receive the corresponding ACK frame, which includes the inter-frame spacings DIFS and SIFS, and an additional guard time T_G that depends on the PHY protocol used [10]. RTS/CTS handshakes are disabled by default since the size of the frame is small and we assume is always less than the RTS/CTS threshold. Retransmissions and back-off counter

are disabled as well (the latter can disabled by setting $CW_{min} = CW_{max} = 0$), since these features are not required in a TDMA system.

$$T_{s_i} = T_{ACK} + DIFS + SIFS + \frac{D}{R_i} + T_G \tag{3.3}$$

T_{s_i} cannot be smaller than $T_{s_{min}}$ which corresponds to using the highest data rate R_{max} and best channel conditions. The relation between T_c and T_{s_i} is given by:

$$\sum_{i=1}^{N} T_{s_i} \leq T_c \tag{3.4}$$

3.2.1 Frames Structure

As shown in Figure 3.2, the superframe has variable number of time slots of variable length each, preceded by a beacon period where the modified beacon frame is transmitted. We use the structure of the beacon defined in the standard and change it by appending at its end N fields indicating the MCS number of each user. The length of each field is equal to $\lceil \log_2(MCS_{max}) \rceil + 1$ bits, where MCS_{max} is the highest MCS number available. If a device is not assigned a transmission during a superframe, its field is set to all 1's which does not map to any MCS number, thus knowing it should wait for the next beacon. The structure of the modified beacon frame is shown in Figure 3.3.

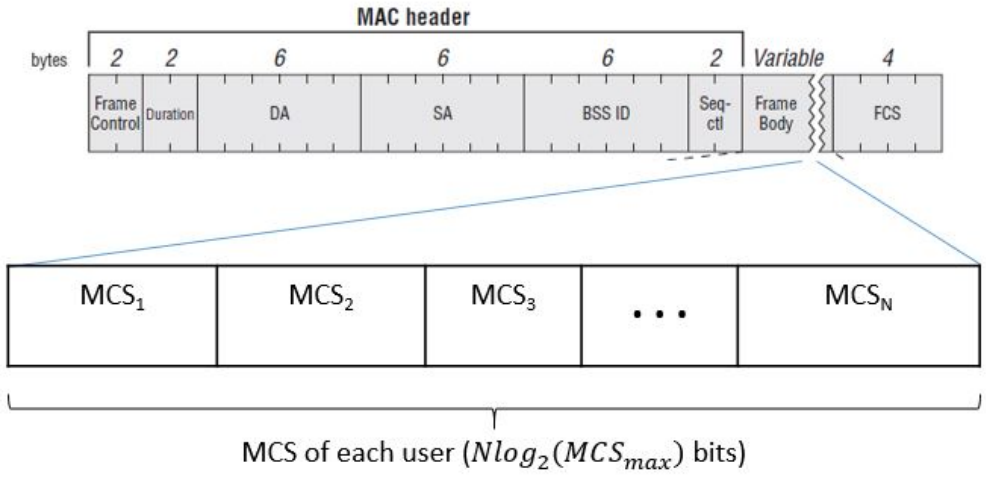


Figure 3.3: Beacon Frame Structure

3.3 Network Operation

The network operates as follows. When first entering the network, devices use CSMA/CA to contend for the medium and associate with the AP which grants them an available unique Association ID (AID) ranging from 1-2007 as defined in the standard, and assigns each device to one of the data channels.

After acquiring an ID, the devices switch over to their respective data transmission channel and wait for the reception of the beacon frame to synchronize their clocks to the Timestamp field in the beacon. Each device reads its corresponding MCS field based in its AID to get its assigned rate for the next superframe. All devices also read the MCS fields preceding their own and map the other devices' MCS to calculate the start time of their own slot. This approach is used because the time slots are not of equal length and thus a device cannot know its transmission start time using only its AID and the fixed slot time. The starting time of a device i is calculated using:

$$T_{start_i} = TimeStamp + \sum_{k=1}^{i-1} T_{s_k} \tag{3.5}$$

Where T_{s_k} is calculated by mapping the MCS to a rate R_k and using equation (3.3). Then, devices wait until T_{start_i}, start transmission, and wait until the next beacon. The operation of the network is illustrated in Figures 3.4 and the state diagrams of the STA and AP are shown in Figure 3.5 and 3.6. The blue color represents the association channel and the green color represents the data channel. There is no arrow between the association response and the scheduling states in Figure 3.6 since they are two independent processes running on two different channels. The STA first starts up, immediately enters the association sequence with the AP, repeats it as long as unassociated, then switches over to the data channel. On the data channel it waits for the beacon, processes it by reading its specific MCS field, if that field maps to a valid rate, it sends its packet at its specific time slot. Otherwise, it waits until the next beacon. We assume that the AP can perfectly estimate the SINR after reception of a frame. On each channel, The decision to schedule which users can transmit during the next superframe and at which rates is made by solving an optimization problem that aims to maximize the number of users admitted and minimize their time slots by assigning their optimal rates, subject to constraints on individual SNR and length of superframe. The problem formulation is discussed in the following chapter.

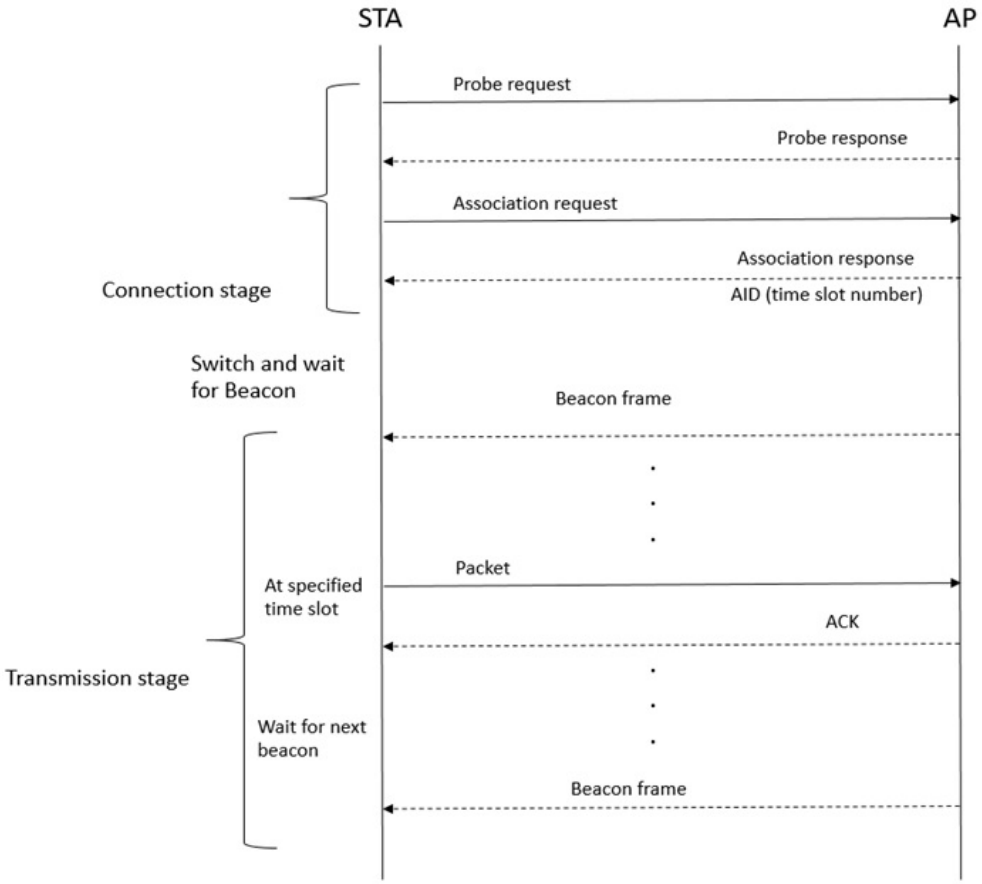


Figure 3.4: Network operation diagram

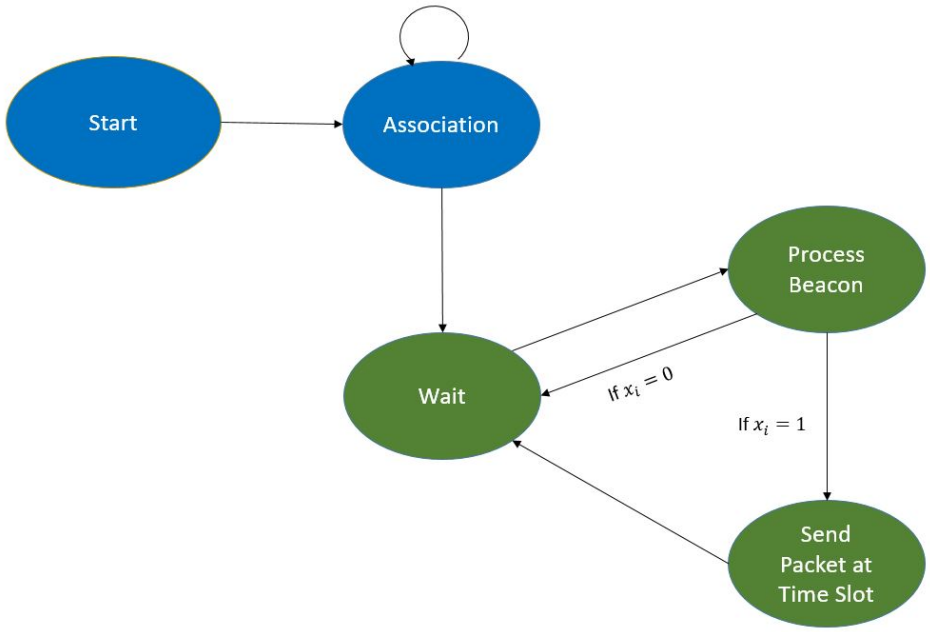


Figure 3.5: STA state diagram. Yellow: Association channel. Green: Data channel

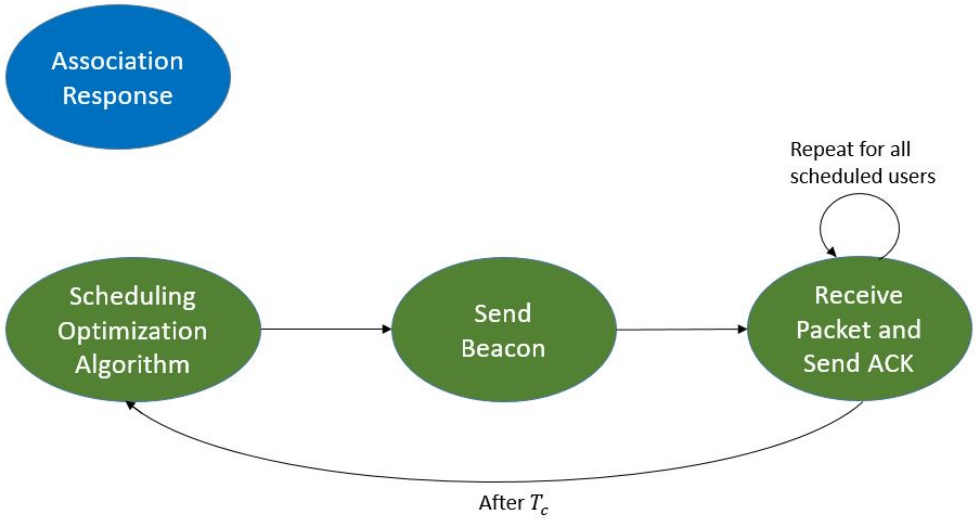


Figure 3.6: AP state diagram. Yellow: Association channel. Green: Data channel

Chapter 4

Problem Formulation and Solution Approach

4.1 Problem Formulation

The problem we want to solve on each data channel is to maximize the number of served users while minimizing their slot times according to their channel conditions, under constraints on the individual SNR, discrete nature of the data rates, and total superframe duration. We define x_i as the binary variable that indicates whether user i is served. The objective function can be written as:

$$\max \quad \sum_{i=1}^{N} x_i(1 - T_{s_i}) \tag{4.1}$$

Where the multiplication by the second factor serves to minimize the time of each assigned user. From the T_s equation in (3.3), the objective function is a function of $\frac{1}{R_i}$, therefore the problem can be formulated as follows:

$$\max_{R_i, x_i} \quad \sum_{i=1}^{N} x_i(1 - T_{\text{MAC}} - \frac{D}{R_i}) \tag{4.2}$$

$$\text{s.t.} \quad \sum_{i=1}^{N} x_i(T_{\text{MAC}} + \frac{D}{R_i}) \leq T_c \tag{4.3}$$

$$x_i(\gamma_i - \gamma_{th}) \geq 0 \qquad \forall i \tag{4.4}$$

$$R_i \in \mathcal{R} \qquad \forall i \tag{4.5}$$

$$x_i \in \{0, 1\} \qquad \forall i \tag{4.6}$$

where T_{MAC} is a term that groups all the times in equation (3.3). The objective (4.2) maximizes the total number of active users and minimizes the total time slot duration. The first constraint (4.3) ensures that all active users fit in one

superframe. Constraint (4.4) ensures each selected user meets the minimum SNR requirement. The third constraint (4.5) forces the selection of a data rate that is allowed by the PHY according to the available MCS. Finally, constraint (4.6) ensures the selection variables x_i are binary. This optimization problem is solved after every superframe and the optimal R_i's and x_i's are sent to the users.

4.2 Problem Complexity

The problem formulated contains integer variables R_i and x_i, where the latter are binary. The constraint (4.4) and objective (4.2) contain nonlinear functions of R_i, namely γ_i a function of the nonlinear rate mapping and $1/R_i$. Therefore the problem is an integer nonlinear programming problem. It is similar to the 0-1 Knapsack problem. In the general Knapsack problem, we are given a set of items each with a weight w_i and a value v_i, and we need to determine the optimal number x_i of each item to include such that the total value $\sum_{i=1}^{N} v_i x_i$ is maximal and the total weight $\sum_{i=1}^{N} w_i x_i$ is less or equal to a certain limit W. The 0-1 Knapsack problem constrains the number of each item to be at most 1. In our case, $v_i = 1 - T_{MAC} - \frac{D}{R_i}$, $w_i = T_{MAC} + \frac{D}{R_i}$, and $W = T_c$. The problem falls under integer programming problems and is proved to be NP-Hard [31]. In addition to the constraint on SNRs and that w_i and v_i are nonlinear functions of the decision variable R_i, our problem is at least as hard as the 0-1 Knapsack problem, making it NP-Hard. No computationally efficient solution can be found for large number of devices and we will have to resort to using heuristic or sub-optimal algorithms to solve the problem.

4.3 Solution Approach

4.3.1 Simplified Formulation

The formulation in equation (4.2) does not make use of the SNR-MCS mapping rate adaptation defined in the standard, thus making the problem more difficult to solve and not fully reflective of the real scenario and standard operation. Therefore, since the estimated SNRs of the users are fixed after every transmission and their optimal choice is directly related to the standard mapping table, we first algorithmically assign the rates based on this mapping, then use these optimal values R_i^* as given in the optimization, reducing the number of decision variables

and constraints by half (x_i's are left only). The simplified formulation is:

$$\max_{x_i} \quad \sum_{i=1}^{N} x_i \tag{4.7}$$

$$\text{s.t.} \quad \sum_{i=1}^{N} x_i (T_{\text{MAC}} + \frac{D}{R_i^*}) \leq T_c \tag{4.8}$$

$$x_i(\gamma_i - \gamma_{th}) \geq 0 \qquad \forall i \tag{4.9}$$

$$x_i \in \{0, 1\} \qquad \forall i \tag{4.10}$$

The problem is now an Integer Linear Programming (ILP) problem and is also an NP-Hard problem, but can be solved more easily than the previous one and better approximated using heuristic algorithms as well.

4.3.2 Proposed Scheduling Algorithm

The usual methods used to solve ILPs are usually solved by branch-and-bound methods [32]. This method uses implicit enumeration, an intelligent enumeration scheme where only a small number of solutions out of all possible exponentially many combinations are explicitly evaluated. The method involves linear programming relaxations on the initial problem which is divided into easier subproblems (branching) whose solutions are used as upper and lower bounds for the initial one (bounding). A subproblem branch is eliminated if it is infeasible, its solution is optimal (used to tighten bounds on initial problem), or its upper bound is less than the lower bound on the initial problem [32]. Branch-and-bound, along with its improved version branch-and-cut, are powerful and exact methods to solve ILPs, however they are time and computation consuming since the number of branches can grow exponentially if an initial candidate solution is found but its branches might not give an improvement [32]. Therefore, due to the delay-sensitive nature of the data and the computational constraints of most typical APs, we will resort to heuristic methods which are faster and can find sub-optimal solutions.

Our heuristic algorithm is based on three ideas: exclusion of users below the minimum threshold, per-cycle rate adaptation using the estimated SNR and the SNR-MCS mapping, and fitting users into the cycle each according to their optimal needed time slot. The steps can be summarized as follows.

1. For all users set those with SNR less than γ_{th} to $x_i = 0$, otherwise $x_i = 1$.
2. For all users with $x_i = 1$ find in which region their SNR lies and assign the corresponding rate R_i^* of that region.
3. Sort the users in ascending order of SNR.

4. Check if the constraint $\sum_{i=1}^{N} x_i(T_{MAC} + \frac{D}{R_i^*}) \leq T_c$ is violated. If not, done. Else:

5. While the constraint is violated, find the user with minimal SNR and set the corresponding x_i to 0.

6. Return the lists $\boldsymbol{x}, \boldsymbol{R}$.

The heuristic is more formally described in Algorithm 1.

Algorithm 1: Heuristic Scheduling Algorithm

Input: N, list of SNRs $\boldsymbol{\gamma}$, γ_{th}, T_c
Output: $\boldsymbol{x}, \boldsymbol{R}$

```
1  for i = 1 to N do
2      if γ_i < γ_th then
3          x_i = 0
4      else
5          x_i = 1
6  for users with x_i = 1 do
7      Map γ_i to R_i^*
8  Sort users in ascending order of γ_i
9  t = Σ_{i=1}^{N} x_i (T_MAC + D/R_i^*)
10 for i = 1 to N do
11     if t > T_c then
12         x_i = 0
13         R_i = ∞
14         t = Σ_{j=1}^{N} x_j (T_MAC + D/R_j^*)
15     else
16         return x, R
17 return x, R
```

The first two for loops have complexity $\mathcal{O}(N)$. Step 8 has complexity $\mathcal{O}(NlogN)$. Step 9 computes the current total time in $\mathcal{O}(N)$. Then in the final for loop we iterate over all sorted users and remove those with the smallest rate in $\mathcal{O}(1)$ since the list is sorted, while computing the total time again in $\mathcal{O}(N)$. If the total time satisfies the constraint then we have reached a feasible allocation and return $\boldsymbol{x}, \boldsymbol{R}$. If we iterate over all users and still did not satisfy the constraint then we return the allocation we currently have in step 17. This process has complexity $\mathcal{O}(N^2)$ in the worst case since the constraint computation takes $\mathcal{O}(N)$ and it

Table 4.1: SNR to MCS partial mapping table in 802.11n

MCS index	**Modulation**	**Coding Rate**	**Data Rate (BW=20MHz) (GI=800ns)**	**Min SNR (dBm) (BW=20MHz)**
1 Spatial Stream				
0	BPSK	1/2	6.5	2
1	QPSK	1/2	13	5
2	QPSK	3/4	19.5	9
3	16-QAM	1/2	26	11
4	16-QAM	3/4	39	15
5	64-QAM	2/3	52	18
6	64-QAM	3/4	58.5	20
7	64-QAM	5/6	65	25
2 Spatial Streams				
8	BPSK	1/2	13	2
9	QPSK	1/2	26	5
10	QPSK	3/4	39	9
11	16-QAM	1/2	52	11
12	16-QAM	3/4	78	15
13	64-QAM	2/3	104	18
14	64-QAM	3/4	117	20
15	64-QAM	5/6	130	25

is repeated for all N users at worst. Therefore the algorithm has a complexity of $\mathcal{O}(N^2)$ at the worst case. Note that even if the users were not sorted, the algorithm would still have $\mathcal{O}(N^2)$ complexity since finding the minimum is done in $\mathcal{O}(N)$ followed by another $\mathcal{O}(N)$ computation. We choose to sort the users to reduce computations in the constraint checking.

The next step is to map the rates to an MCS number using the standard-defined table, which is partially shown in Table 4.1 for 802.11n. A rate of infinity corresponds to an MCS field of all 1's which does not map to anything, thus the STA knows it should not transmit during this cycle.

The cases where returning users (those who have not been granted a slot in the current cycle but want to transmit during the next one) and new users arrive present the issue of having no prior SNR knowledge to base the scheduling decision on. This is fixed by assuming at each cycle that these users are granted $x_i = 1$ and $R_i = R_{min}$, the smallest achievable rate in the mapping table, then running the algorithm.

4.3.3 Fair Scheduling Algorithms

The formulation in (4.2) is equivalent to maximizing the sum-throughput in the network, since user admission is based on SNR value and the objective function represents the total active users. This approach, as stated in Section 2.3, leads to unfair rate and time slot allocation since those with the best conditions will be always served. Therefore, we use two approaches for fair resource allocation. The first approach uses the usual PF utility function $U_i(R_i) = \ln(R_i)$, and the second uses fairness in the number of time slots allocated to each user over a window of length K cycles, since time is our main resource. The utility of each user is computed every cycle and the window slides as time progresses. The utility function for the second approach is:

$$U_i^{(L)}(x_{i,l}) = \frac{\sum_{l=L-K}^{L-1} x_{i,l}}{\sum_{i=1}^{N} \sum_{l=L-K}^{L-1} x_{i,l}} \tag{4.11}$$

where the variable $x_{i,l}$ represents the activity of user i in cycle l and is binary. The superscript L denotes the cycle at which the function is computed. The term in the numerator represents the total time slots allocated to user i over the window of length K. The term in the denominator represents the total active time slots in the past window.

Our proposed scheduling algorithm is based on the one proposed in [26]. At each scheduling instance (each cycle) we aim to maximize the marginal PF utility defined as

$$\Lambda_{i,l} = \ln(R_{i,l}) - \ln(R_{i,avg}) \tag{4.12}$$

where $R_{i,avg}$ is the mean rate over the last window. This is done to see the gain in utility if the user is served and allocated the specific rate $R_{i,l}$ at this cycle. We start by finding the user with highest marginal utility and allocate their rate according to the standard mapping-table, then check if the resulting sum of time slots does not exceed T_c, if it does the user's rate becomes 0 and is not served in the following cycle. We then remove the user from the list and repeat the process until all users are decided upon or we run out of resources (total time exceeds T_c).

For the time-slot-based fairness, we proceed in the same way except that we do not compute marginal utility, but the utility function itself (4.11) at each cycle, and we iteratively find the users with minimal utility. This algorithm is similar to a max-min approach since we are prioritizing those with the least relative number of time slots first. Usual max-min algorithms use the data rate as the variable [24] and try to maximize the metric in Equation (2.3). We propose a version that is based on the time slot allocation with a different metric. The PF algorithm is described in Algorithm 2, and the time-slot-based algorithm in Algorithm 3.

Algorithm 2: Proportional Fairness Scheduling Algorithm

Input: N, list of SNRs $\boldsymbol{\gamma}$, γ_{th}, T_c, $\mathcal{U}$ set of all users
Output: $\boldsymbol{x_l}, \boldsymbol{R_l}$

```
for i = 1 to N do
    if γ_i < γ_th then
        x_{i,l} = 0
        U = U − {u_i}
for u_i ∈ U do
    Map γ_i to R_{i,l}
    Compute Λ_{i,l} = ln(R_{i,l}) − ln(R_{i,avg})
while U ≠ ∅ do
    Find u_{i*} satisfying i* = argmax_i Λ_{i,l}
    t = Σ_{i=1}^{N} x_{i,l}(T_MAC + D/R_{i,l})
    if t ≤ T_c then
        x_{i*,l} = 1
        U = U − {u_{i*}}
    else
        x_{i*,l} = 0
        R_{i*,l} = 0
        U = U − {u_{i*}}
return x_l, R_l
```

Algorithm 3: Time-Slot-Based Fairness Scheduling Algorithm

Input: N, list of SNRs $\boldsymbol{\gamma}$, γ_{th}, T_c, $\mathcal{U}$ set of all users

Output: $\boldsymbol{x_l}, \boldsymbol{R_l}$

for $i = 1$ *to* N **do**
  if $\gamma_i < \gamma_{th}$ **then**
    $x_{i,l} = 0$
    $\mathcal{U} = \mathcal{U} - \{u_i\}$
for $u_i \in \mathcal{U}$ **do**
  Map γ_i to $R_{i,l}$
  Compute $U_i^{(L)}(x_{i,l}) = \frac{\sum_{l=L-K}^{L-1} x_{i,l}}{\sum_{i=1}^{N}\sum_{l=L-K}^{L-1} x_{i,l}}$
while $\mathcal{U} \neq \emptyset$ **do**
  Find u_{i^*} satisfying $i^* = \text{argmin}_i\ U_i^{(L)}(x_{i,l})$
  $t = \sum_{i=1}^{N} x_{i,l}(T_{MAC} + \frac{D}{R_{i,l}})$
  if $t \leq T_c$ **then**
    $x_{i^*,l} = 1$
    $\mathcal{U} = \mathcal{U} - \{u_{i^*}\}$
  else
    $x_{i^*,l} = 0$
    $R_{i^*,l} = 0$
    $\mathcal{U} = \mathcal{U} - \{u_{i^*}\}$
return $\boldsymbol{x_l}, \boldsymbol{R_l}$

Algorithm 2 has complexity $\mathcal{O}(N^2)$ since finding the maximum of N users is repeated until all are served. The utility computation in Algorithm 3 has complexity $\mathcal{O}(N^2K)$ and the allocation loop has complexity $\mathcal{O}(N^2)$, however K is constant, hence Algorithm 3 has complexity $\mathcal{O}(N^2)$ as well. The implementation of the algorithms and simulation scenario will be presented in the following chapter.

Chapter 5

Implementation and Simulation

5.1 Network Simulator 3 (NS-3) Introduction

Simulations were done using Network-Simulator-3 (NS-3) [33], an open source discrete-event, system-level network simulator in C++, where events are scheduled according to an internal 64 bit clock. It is divided into hierarchical layers of abstraction that represent specific aspects of the network and device behavior (Figure 5.1). First, the core of the simulator consists of the components common to all protocols, hardware and networks such as callback functions, attributes of data types, debugging tools such as tracing attributes, logging and most importantly the simulator class which handles event scheduling. The second layer models nodes from their network interface (packets, MAC address, queues) to the physical layer models (e.g. 802.11ac, LTE PHY models). The next two layers implement the internet stack and its protocols such as IPv4, TCP, application layer protocols, propagation models and the mobility model attached to the node. Helper functions are available to facilitate access to these lower level models and the interconnection between them, for example a helper can be used to install a network interface on multiple nodes and install on it the MAC and PHY models. Finally, there are test classes to check the correct behavior of each model.

The simulator was run on a Linux Virtual Machine assigned 4 processors at 2.6GHz and 6 GB of RAM. NS-3 simulations are controlled by the Simulator class. All the network elements, channel allocations, protocols used (PHY, MAC, IPv4, TCP, etc.) and node mobility models need to be assigned before calling the Run function of the Simulator class. No dynamic channel switching is allowed during the simulation. All events need to be scheduled through the Simulator class's Schedule function which executes a certain function at a user-specified time delay from the current simulation time. Events in the simulator are executed sequentially based on their order of arrival in time, meaning that the execution of events cannot happen in parallel, the simulation time stops until the current event is finished before the next one starts and time progresses. If two events

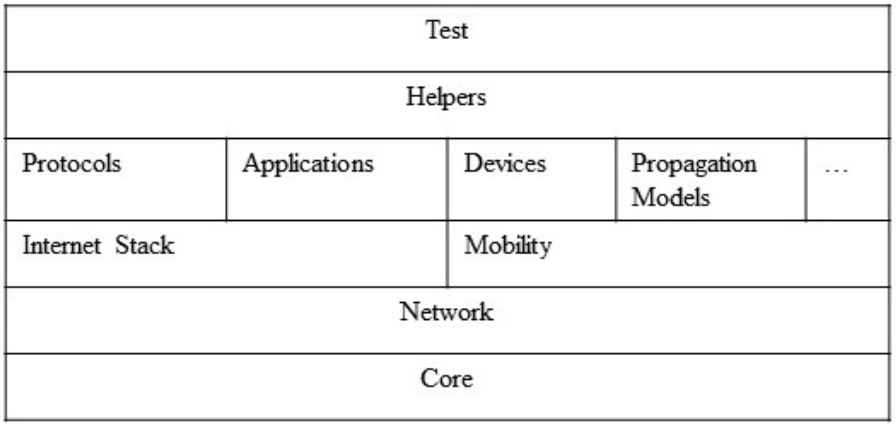


Figure 5.1: NS-3 hierarchical structure

are scheduled at the same time, the current implementation of NS-3 chooses at random which is processed first, which can lead to misleading results. However, this can be avoided using multiple iterations to average out the results. Since NS-3 is a system-level simulator, all relevant events at all protocol layers occur, meaning that if we were to simulate, for example, a UDP transmission over WiFi, all exchanges that happen in the real world scenario also happen in NS-3. A simulation ends if no more events are scheduled or the user-specified simulation time runs out.

5.2 Protocol Implementation

To implement our hybrid MAC protocol, there are two different approaches. The first one is creating new classes that inherit from the existing WiFi MAC and PHY implementations and modify them. This approach requires modifying most of the NS-3 implementation of the WiFi standard, which is a complicated and time consuming task requiring the modification of thousands of lines of code, hundreds of classes and modules that depend and interact with each other (as shown in Figure 5.2 and in [34]). The alternative approach is to control the behavior of the MAC and PHY layers from the Application class of NS-3, requiring only modifications of the needed MAC and PHY properties, then writing the scheduling, heuristic, and beacon transmission all from a higher layer. Note that we do not have full control over all the events in the network using this approach, so some frame exchanges defined in the standard might still happen as well as frame exchanges of higher layers. However, the advantage of this approach is the reduced complexity while maintaining a high level of fidelity to the real world scenario.

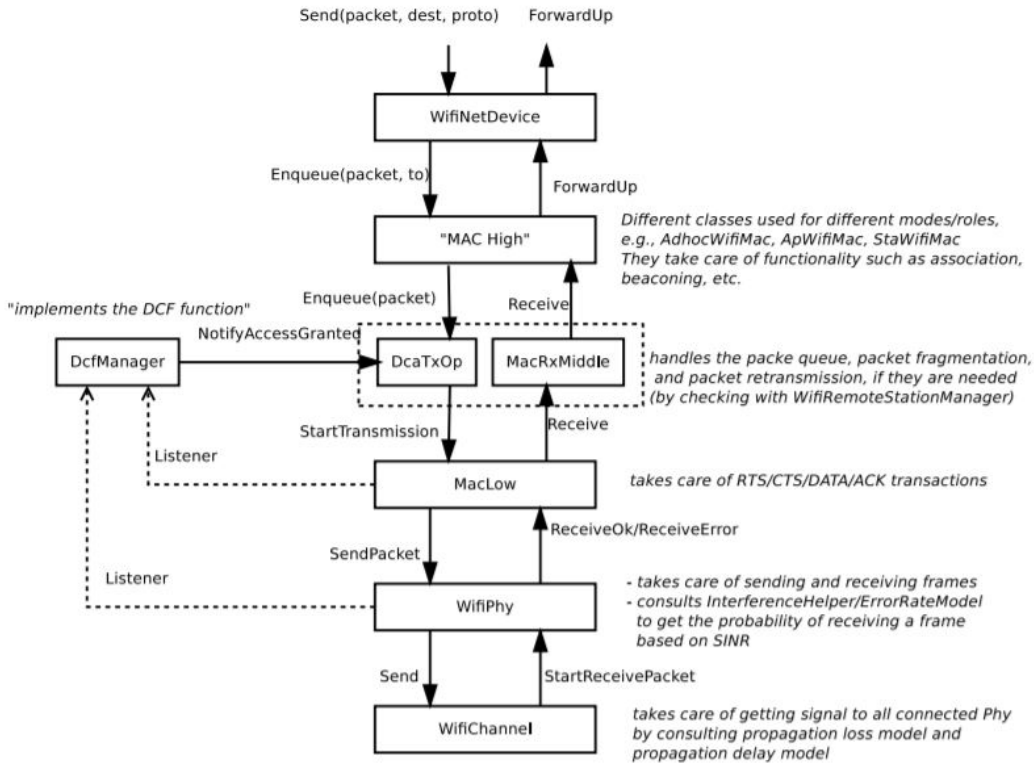


Figure 5.2: NS-3 802.11 Implementation Diagram [34]

Our approach is the following. Adding functions to the WifiPhy, MacLow, and MAC-High classes that permit modification of the transmission parameters (rate and contention window values) from a higher level, adding private variables to measure the SNR received and functions that extract it for use in the rate adaptation, calling these functions from the main application before scheduling any packet transmission, and adding additional control variables and structures to make sure these functions only get called for our application and ensure compatibility with other uses of the NS-3 WiFi Module.

The functions added to the PHY class manipulate the TxVector class that sets transmission parameter according to the given WiFiMode class. At every transmission we call a public member function of the WifiPhy class that takes as parameter the MCS value read from the beacon and changes the lower level WiFiPhy class TxVector private variable. Then, inside the PHY SendPacket function responsible for the actual PHY transmission, the transmission parameters (MCS which changes rate, bandwidth, spatial streams etc.) are read from the private TxVector member. To disable the backoff process (a step to effectively turning the second channel into TDMA), we schedule a function call right after the simulation start and before the application startup to set the CW values to

zero. On the receiver side, the only change is extracting the SNR of the received packet, done by adding a private variable in the WiFiPhy class and setting it to the SNR value inside the Receive function at the PHY. Then this provate variable is read at every reception through a public function of the WiFiPhy class called from the application level. This is done through a callback function that is activated every time a transmission happens, where inside this function we use the aforementioned SNR reading function.

On the AP side, the scheduling algorithm is first scheduled inside the StartApplication function after T_c seconds. Inside this function we call the SendBroadcast function which broadcasts the beacon to the devices (who do not start another transmission unless they receive the beacon and process it, which is done through a receive callback function on the STA side). After broadcasting the beacon, this function re-schedules the scheduling algorithm after T_c seconds, and the process repeats until the end of the simulation.

Performance metrics are measured through the use of private variables that get updated either at every reception, every transmission, or every second. To measure the packet delivery ratio, we also use the built-in variable tracking in NS-3, called TraceSource, which calls user defined functions when a specific event happens (such as PHY reception, PHY drop, MAC reception or drop, MAC frame enqueued or dequeued, some IP and transport layer events, etc.).

5.3 Simulation Setup

The simulation was set up as follows: we created N nodes and one AP running the 802.11n protocol at 2.4GHz with IPv4 and UDP. The nodes were placed randomly in a circular area of radius 20m around the AP according to a uniform distribution. The mobility model of the STAs is chosen to be a random walk inside the area boundaries ($400m^2$ square). The PHY channel model used the default network simulation channel model with propagation delay equal to the speed of light and log-distance pathloss model with reference pathloss of 46.677 dB at 1m. We added a Rayleigh fading model on top. The subnet mask is 255.255.248.0 which allows for a maximum of 2046 devices including the AP, which is more than enough for our case, since the maximum AID value is 2007.

Without loss of generality, we assumed all STAs and AP use only one spatial stream which gives us 8 rates to work with, per Table 4.1. The simulation is run for 20 cycles, each of length 1sec. Devices start randomly between time 0 and half the simulation time (i.e., 10sec). To get an accurate performance of the protocol and average out the randomness, we execute 10 iterations each with a different random seed. The relatively small number of iterations is justified by the fact that the measurements have a small variance and the fact that randomness is compensated by multi-user diversity [10], [35]. We do the same for comparison with legacy CSMA/CA and 802.11ax. However, an important note to keep in mind in the

simulations and results is that the most recent NS-3 implementation of 802.11ax is far from complete, lacking the main functionalities of OFDMA and MU-MIMO [36], and having only the PHY layer frames structures and CSMA/CA with higher CW_{min} values and reduced backoff slot time implemented. Therefore, the results obtained are not representative of real world 802.11ax performance, and are identical to the 802.11n results, so for the sake of not comparing with an incomplete implementation and drawing false conclusions, we will omit the 802.11ax results. We simulate for $N = \{1, 2, 5, 10, 50, 100, 200, 300, 500, 700, 800, 1000, 1500\}$. The simulation parameters are summarized in Table 5.1. Results are presented in the following chapter.

Table 5.1: Simulation parameters

Parameter	**Value**
Protocol	IEEE 802.11n at 2.4GHz with Hybrid Regular IEEE 802.11n at 2.4GHz IEEE 802.11ax at 2.4GHz
Area	Circle of radius 20m
Cycle Time T_c	1sec
Payload Size D	200B
SIFS	10μs
DIFS	50μs
Area	Circle of radius 20m
Channel Model	YANS Model: Propagation delay: speed of light, log-distance pathloss, Rayleigh fading
MAC Retry Limits	0
ACK Size	14 Bytes
ACK Transmission Rate	MCS0: 6.5Mbps
RTS/CTS	Disabled
Subnet Mask	255.255.248.0
Mobility Model	Random Walk
Internet Protocols	IPv4, UDP
Run Time T_{sim}	$20T_c$
Device Startup Time	$\mathcal{U}(0, T_{sim}/2)$

Chapter 6

Performance Results and Analysis

6.1 Performance Metrics

To assess the performance of our proposed hybrid protocol we compare it with 802.11 in the following metrics: packet delivery ratio, association-to-transmission delay, ratio of successful associations to total devices, ratio of successful transmissions to associated devices, and ratio of successful transmissions to total devices.

For fair comparison between 802.11 CSMA/CA and our hybrid protocol, we equalize the offered load OL per second (in Bytes/sec) seen by the AP for both protocols using:

$$OL = N_{u/s} N_{p/u} N_{B/p} \tag{6.1}$$

where $N_{u/s}$ is the number of users per second, $N_{p/u}$ is the number of packets per user in one transmission opportunity each second, and $N_{B/p}$ is the number of bytes per packet, thus giving OL in Bytes/sec. For our hybrid TDMA protocol $N_{u/s}^{(T)} = N$, $N_{p/u}^{(T)} = 1$ and $N_{B/p}^{(T)} = D$, which is 200B in our simulation. The superscript T indicates the TDMA protocol and C is for CSMA/CA. For CSMA/CA $N_{u/s}^{(C)} = N$ and $N_{B/p}^{(C)} = D$, which leaves $N_{p/u}^{(C)}$ as the unknown and is found by equating $OL^{(C)}$ and $OL^{(T)}$. Thus,

$$N_{p/u}^{(C)} = \frac{N_{u/s}^{(T)} N_{p/u}^{(T)} N_{B/p}^{(T)}}{N_{u/s}^{(C)} N_{B/p}^{(C)}} \tag{6.2}$$

which amounts to $N_{p/u}^{(C)} = 1$. Then we schedule in the simulation each user to transmit one packet in every second, with the delay between each user in the

same second being $1/N$ sec to have N user per second. As discussed above, the 802.11ax implementation is incomplete and results are identical to 802.11n, thus will not be shown.

6.1.1 Packet Delivery Ratio

The packet delivery ratio (PDR) is defined as the ratio of successfully received data packets to total generated data packets per device during the simulation time. To have an objective comparison of PDR vs N, we scale it by taking into account the users who did not associate and those who did not transmit.

6.1.2 Delay

We define the delay as the time between first attempting to associate and transmission of the first data packet, regardless of its reception or drop, in other words, the time it takes to be admitted into the network and sending the first packet. We find the average delay of all users for the entire simulation time and the empirical distribution of the delay.

6.1.3 Ratio of Associated Devices to Total Devices

This metric represents the ratio of devices that successfully associate and is used to measure how many devices are admitted to the network. We find the average of this metric for different N and its evolution over time.

6.1.4 Ratio of Successful Transmissions to Associated Devices

This metric is used to measure the ratio of devices granted a transmission opportunity and successfully deliver their packets. It is used to measure how many users out of those associated have a successful transmission opportunity (for the case of our protocol, how many out of the associated are granted time slots). We find the average of this metric for different N and its evolution over time.

6.1.5 Ratio of Successful Transmissions to Total Devices

This metric is similar to the above one, except that we compare with the total number of users instead of the associated ones only. We also find the average of this metric for different N and its evolution over time.

6.2 Results Comparison and Analysis

6.2.1 Packet Delivery Ratio

As seen in Figure 6.1, the scaled PDR starts off at 100% for small values of N for both protocols and is a decreasing function of N. At first all protocols have a 100% PDR due to the low traffic observed by the AP. The superiority of our protocol is shown for larger N where the PDR drops below 50% for $N = 1000$ only whereas it crosses that mark for $N = 200$ for 802.11n. The results of the 802.11n PDR are in alignment with what is shown in the literature, e.g., in [35], where the collision probability (thus, the PDR) for a saturated 802.11b network (network where all users always have packets to send) crosses the 50% mark for $N = 50$ and $CW = 32$. The reason the PDR drops for our protocol is the decrease in the number of users associated and those who did not successfully transmit. In addition, collisions can happen when the beacon frame is received in error leading to a wrong mapping of the data rates, which leads to sending at a rate the user's channel does not support. This can happen with probability p_e, the target bit error rate. Collisions can also happen if one user sends at a lower data rate that makes their time slot overlap with the subsequent user's, assuming the latter receives and decodes their MCS field correctly. In this case the probability of collision can be obtained by the following.

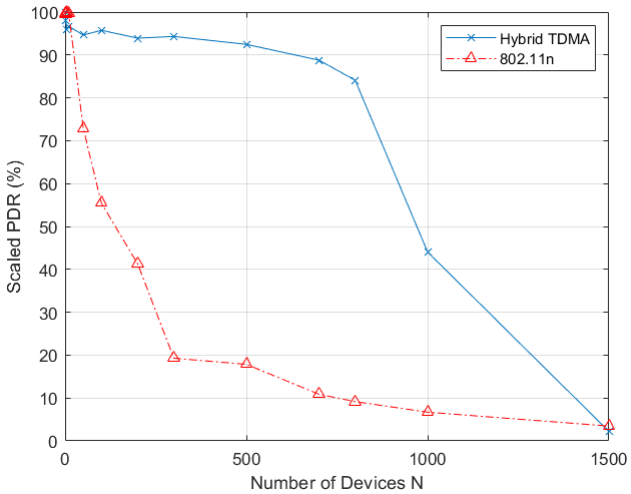


Figure 6.1: PDR performance comparison for CSMA/CA (802.11n) and Hybrid TDMA

For a user i receiving and decoding a beacon frame, define the events E_j^i, $j \in \{1, 2, 3, 4\}$, as:

- E_1^i: the event that user i receives the part of the beacon before their MCS field (denoted by U_i) with an error (can be more than one bit error) but U_i is received correctly.
- E_2^i: the event that user i receives the part of the beacon before U_i correctly but U_i is in error.
- E_3^i: the event that the part of the beacon before U_i is received in error and U_i is in error.
- E_4^i: the event that all fields up to and including U_i are received correctly.

Let $p_{e,i}$ be the probability of error of user i and let K be the size of the MCS field U_i in bits. We assume bit errors are independent, and each user's reception and decoding of the beacon is independent of all other users. Then, the probabilities of the above events are:

$$p(E_1^i) = p(U_i \, correct)p(U_1 \, to \, U_{i-1} \, incorrect) \tag{6.3}$$
$$= (1 - p_{e,i})^K (1 - (1 - p_{e,i})^{(i-1)K}) \tag{6.4}$$

$$p(E_2^i) = p(U_i \, incorrect)p(U_1 \, to \, U_{i-1} \, correct) \tag{6.5}$$
$$= (1 - (1 - p_{e,i})^K)(1 - p_{e,i})^{(i-1)K} \tag{6.6}$$

$$p(E_3^i) = p(U_i \, incorrect)p(U_1 \, to \, U_{i-1} \, incorrect) \tag{6.7}$$
$$= (1 - (1 - p_{e,i})^K)(1 - (1 - p_{e,i})^{(i-1)K}) \tag{6.8}$$

$$p(E_4^i) = p(U_i \, correct)p(U_1 \, to \, U_{i-1} \, correct) \tag{6.9}$$
$$= (1 - p_{e,i})^{iK} \tag{6.10}$$

These events sum to 1, which can be checked easily. Consider now the probability of collisions in the network due to beacon reception errors $p(collision)$. Clearly, $p(collision) = p(any \, user \, receives \, the \, beacon \, in \, error)$, with the right hand side being equal to $1 - p(all \, users \, receive \, the \, beacon \, correctly)$, and since the users decode independently, then:

$$p(collision) = 1 - \prod_{i=1}^{N} p(E_4^i) \tag{6.11}$$
$$= 1 - \prod_{i=1}^{N} (1 - p_{e,i})^{iK} \tag{6.12}$$

6.2.2 Delay

The average association-to-transmission delay is shown in Figure 6.2. At first, for low loads, 802.11n has a smaller delay, this is due to the time users spend waiting for the beacon and their assigned time slot in our hybrid protocol, whereas in 802.11 users can transmit after performing successful backoff when the medium is idle, which takes less than the waiting time for our protocol. As expected, the more users there are, the more time it takes for them to associate and switch channels to wait for the beacon reception. As for both 802.11 protocols, the delay saturates at around 5 sec for $N = 500$, which can be justified by looking at the number of associated users (in the next section), this number is very small and remains approximately the same for values larger than 500, meaning that almost the same number of users associate and are granted a transmission, hence the constant value. The reason users take longer to associate in all protocols is that there is a larger number of frames exchanged as N grows, thus the time during which the medium is busy increases. However, for our protocol, the switching to the data channel frees up the medium, hence reducing delays and facilitating associations on the control channel. The delay drops for $N = 1500$ because not all users associate and transmit, as seen in the ratio of associated users in the next section. Thus, the subset that manages to associate and switch to the data channel does not have to wait for the same time as the case for smaller N to transmit their frame, since this subset is smaller for $N = 1500$ than for other values, hence the smaller delay value.

As for the distribution, Figure 6.3a shows the empirical distribution for $N = 700$ for the protocols, and Figure 6.3b shows the distribution comparison for $N = 1000$. This empirical distribtion shows the delay for the entire network, i.e., for all users combined, not the individual delay per user. The distribution per user for 802.11 and CSMA/CA is shown to be exponential in the literature, e.g. in [37],[38]. Therefore, since the individual delay is exponential and we have a very large number of users in the network, the network delay follows a central limit theorem, thus having a normal distribution.

For the case of the hybrid protocol, we see that the delay is also normally distributed for the same reason as above, since devices initially use the CSMA/CA channel to associate but the subsequent delay to acquire a transmission opportunity and transmit the data is drastically reduced due to the separation of the data plane on a different channel and the TDMA medium access which is contention free. Clearly, the performance of the hybrid protocol in terms of delay is better than 802.11n, having a lower mean (as shown also in Figure 6.2) and less variance.

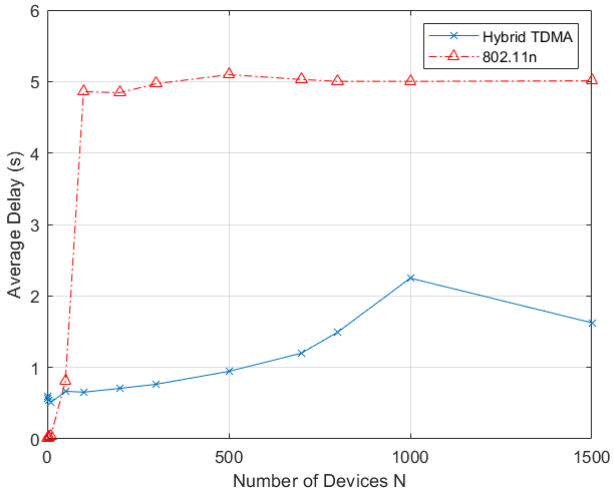
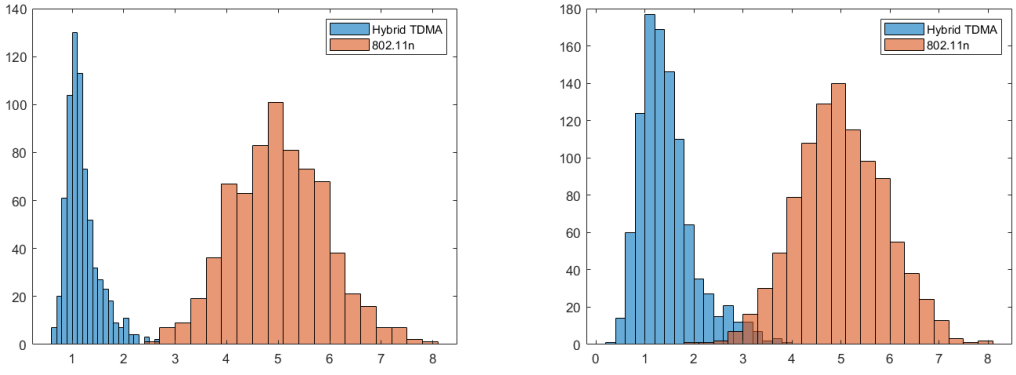


Figure 6.2: Delay performance comparison for CSMA/CA (802.11n) and Hybrid TDMA



(a) $N = 700$ (b) $N = 1000$

Figure 6.3: Delay distributions comparison for 802.11n and Hybrid TDMA for (a) $N = 700$ and (b) $N = 1000$

6.2.3 Ratio of Associated Devices to Total Devices

Figure 6.4 shows the average of this metric (over runs) at the end of the simulation vs N, Figure 6.5 and Figure 6.6 show its evolution over time for $N = 50, 100, 300, 800$, and 1000. During the start-up time ($t \leq 10$s) the number of associated users increases since devices start up randomly and switch to the data channels after successful association, which frees up the medium for new devices to associate, hence reducing the amount of contention on the control plane. Beyond $t = 10$s, most devices are associated, but the performance of our protocol decreases for gradually for larger N until $N = 1000$ where only 50% of users are associated. This is due to the larger number of exchanged frames and collisions on the association channel, but is still much better than CSMA/CA for $N = 1000$ where only around 2% of users are capable of associating. For CSMA/CA, worse performance is achieved even for smaller $N = 50$ due to the higher amount of contentions on the channel used for both association and data transmission, hence association request packets collide with data packets and association requests of other users.

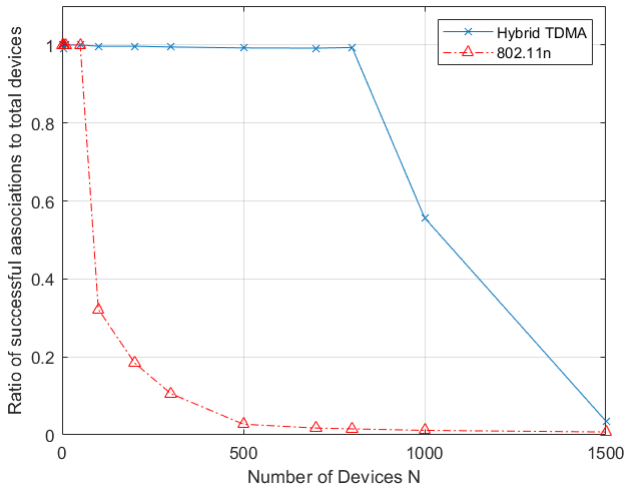


Figure 6.4: Associations ratio comparison for CSMA/CA (802.11n) and Hybrid TDMA

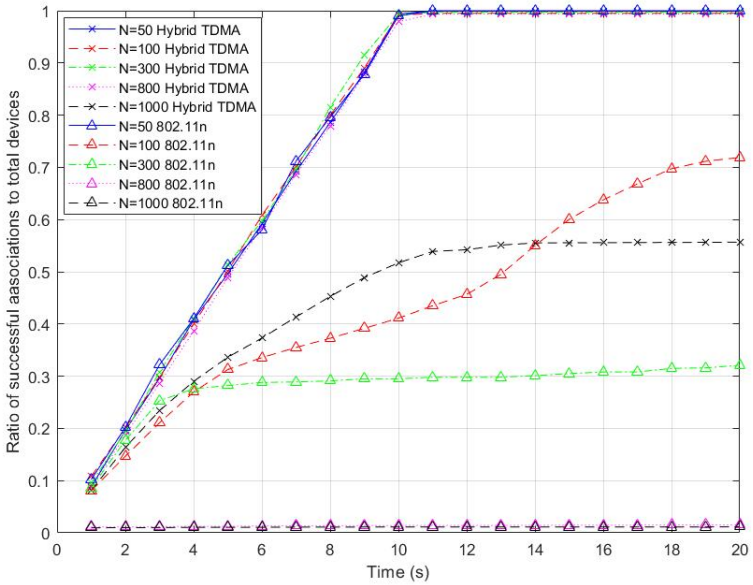
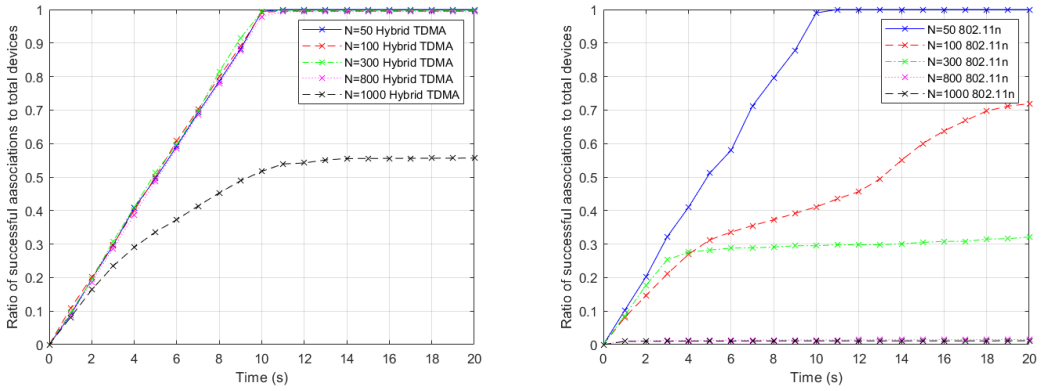


Figure 6.5: Associations ratio vs. time comparison for CSMA/CA (802.11n) and Hybrid TDMA



(a) Plots of Hybrid TDMA

(b) Plots of 802.11n

Figure 6.6: Detailed association ratio vs. time

6.2.4 Ratio of Successful Transmissions to Associated Devices

As shown in Figure 6.7, for our protocol, out of the users who associated, almost all of them are granted a transmission opportunity (a time slot) and successfully deliver their data. As expected, we see a worse behavior for CSMA/CA since the medium is saturated with other data transmissions and association frames, thus

the number of users who can successfully transmit their packet will be very low.

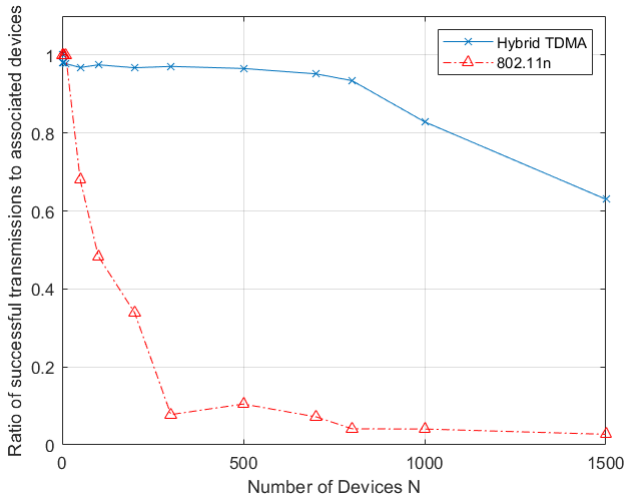


Figure 6.7: Successful transmissions out of associated users comparison for CSMA/CA (802.11n) and Hybrid TDMA

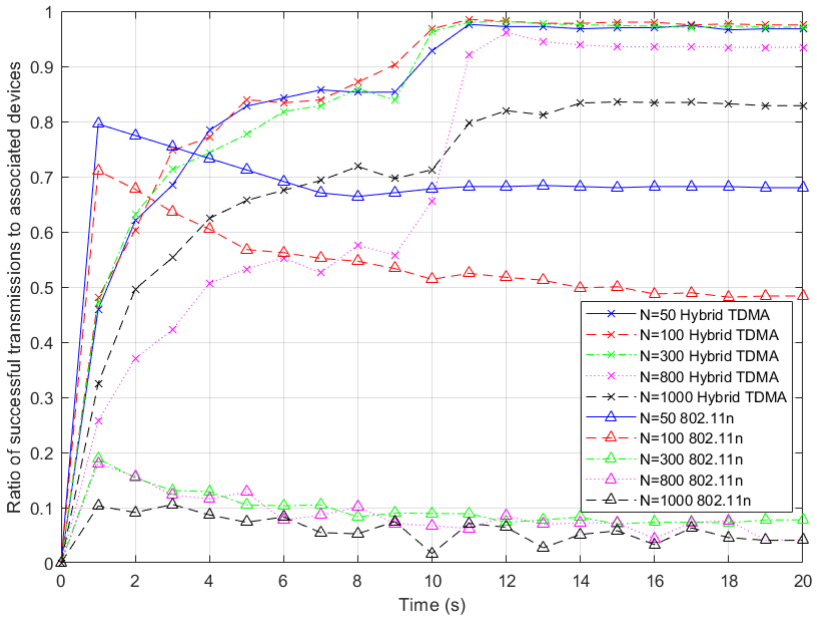
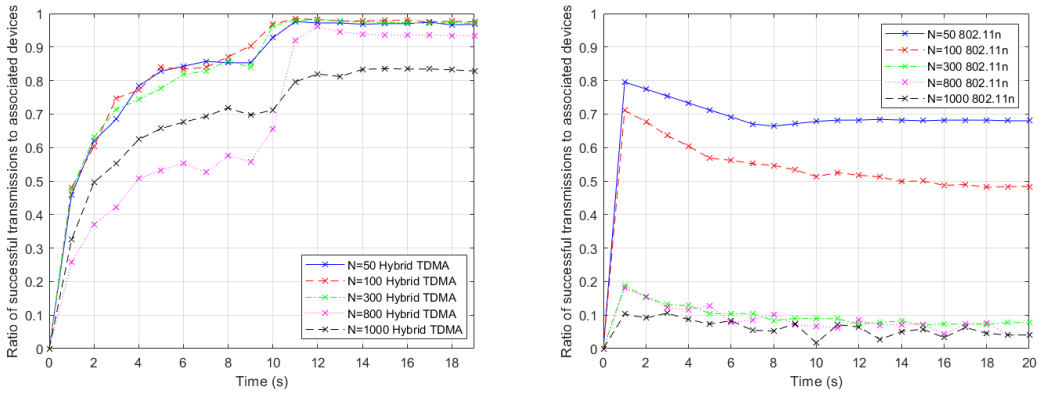


Figure 6.8: Successful transmissions out of associated users vs. time comparison for CSMA/CA (802.11n) and Hybrid TDMA



(a) Plots of Hybrid TDMA

(b) Plots of 802.11n

Figure 6.9: Detailed successful transmissions ratio vs. time

As for the evolution over time in Figure 6.8 and more detailed in Figure 6.9, for the hybrid protocol and for a specific N, the metric increases over time until $t = 10$s because the number of associated users increases and thus the number of those granted a slot is proportionately increasing. After the start-up period $t > 10$s, this ratio becomes close to 1 meaning almost all those associated are granted time slots.

For different values of N, and as N increases, the value of this metric becomes less due to the increased delays of association on the control channel, thus reducing the number of associated users and consequently those granted a time slot, which explains the decrease over different N in the period where $t \leq 10$s. After the start-up period, only for very large values of N (800 and 1000) does this ratio still increase, which is again due to the increased delay of association. During this stable period, we see that the value of this metric decreases for different N due to the scheduling algorithm, where not all users can be granted a time slot in a cycle.

As for CSMA/CA, we see that the number of users who successfully transmit a packet out of the associated ones decreases for a specific N due to the saturation of the medium with packet transmissions. The decreasing trend of the plots can further be explained by the fact that the number of successfully served users (those who are granted a transmission opportunity and successfully transmit) does not increase at a high rate in the CSMA/CA network due to the medium saturation, and as some users manage to associate, there is no guarantee on their packets being successfully transmitted, whose number might stay the same or decrease depending on the collision with other data packets. The number of associated users however, increases at a slightly faster rate as seen in the previous subsection, thus the overall ratio decreases with time. When the number of associated users almost stabilizes after the start-up period, the ratio of successfully served users also becomes stable. For increasing values of N, the metric decreases

as expected and the performance becomes progressively worse. The reason is the increased collisions, which lead to a decrease in successful transmissions.

6.2.5 Ratio of Successful Transmissions to Total Devices

The comparison is shown in Figure 6.10. For our protocol, out of the total users, almost all of them are granted a transmission opportunity and successfully deliver their data by the end of the simulation time. This trend decreases for larger N due to the scheduling algorithm, where not all users can be granted a time slot in a cycle. For CSMA/CA a much worse performance is achieved, due to the same reasons stated in the above subsection.

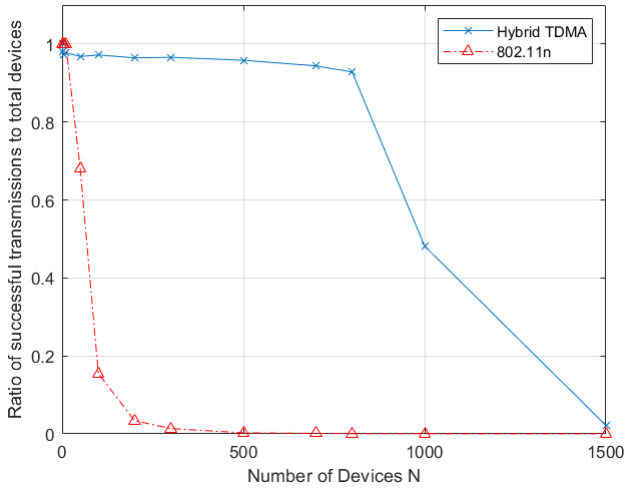


Figure 6.10: Successful transmissions out of total users comparison for CSMA/CA (802.11n) and Hybrid TDMA

As for the evolution over time in Figure 6.11 and more detailed in Figure 6.12, we observe a similar trend as Figure 6.5, where the number of users who are granted a transmission opportunity decreases as the load N increases for both CSMA/CA and the hybrid protocol, the only difference is that instead of comparing to the associated subset, we are comparing to the total users. Also, for a specific N we note the same explanations as the previous metric as to the shape and trend of the curves for the hybrid protocol. For the CSMA/CA protocol for a specific small N, the trend is increasing rather than decreasing, as opposed to Figure 6.8. This is due to the comparison with a fixed N (hence the denominator of the ratio does not increase), and the increase is only due to that of the number of users successfully served. The results for small N (50 and 100) make sense since the load is relatively small and only 70% and 15% of users connected to one AP are successfully served by the end of the simulation.

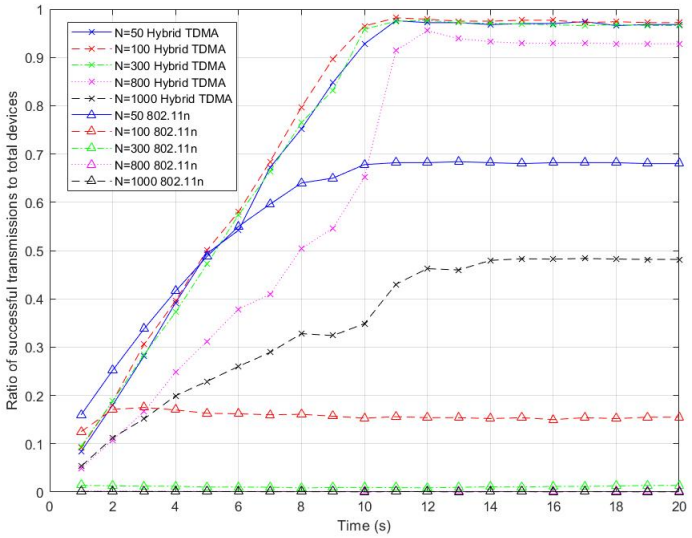
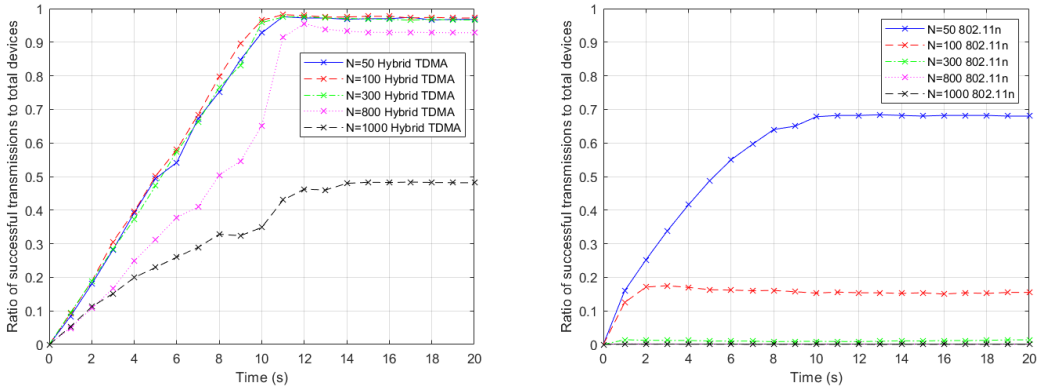


Figure 6.11: Successful transmissions out of total users vs. time comparison for CSMA/CA (802.11n) and Hybrid TDMA



(a) Plots of Hybrid TDMA

(b) Plots of 802.11n

Figure 6.12: Detailed successful transmissions to total devices ratio vs. time

6.3 Fair Scheduling Performance

For comparing the different algorithms' fairness, we compute the sum-throughput and marginal sum-of-log-throughput metrics of all users, averaged over time (60 cycles) and runs. The results are shown in Figures 6.13 and 6.14 for $N = 300, 800,$ and 1000. We also show the distribution of the rates for all three algorithms for all N in Figure 6.15. Clearly, the PF algorithm has superior fairness compared to

the other two as expected, as seen in Figure 6.14, where the sum-log-throughput metric is plotted, and the greedy algorithm performs better than the other two in its own sum-throughput metric, as shown in Figure 6.13. The Time-Slot (TS) based algorithm has similar performance in terms of sum rates than the greedy algorithm, but achieves slightly better rate allocation fairness than the latter. This is due to the scheduling where the users who have had many time slots in previous cycles, not being assigned in subsequent ones, hence their assigned rate is 0 and they do not transmit. This proves that while the proposed algorithm does not achieve the best fairness, it still outperforms the greedy version while using a metric that does not depend on the rates. As for the distribution of the rates, we clearly see a skewed distribution to higher rates for the greedy algorithm, meaning that users with better channel conditions are favored most of the time and allocations of high rates happens more frequently. On the other hand, the rate distributions for the PF and TS based algorithms are more spread out, more so for the PF version, meaning that allocations of high rates is not as frequent as the greedy algorithm and hence more fairness, which is reinforced by Figure 6.14.

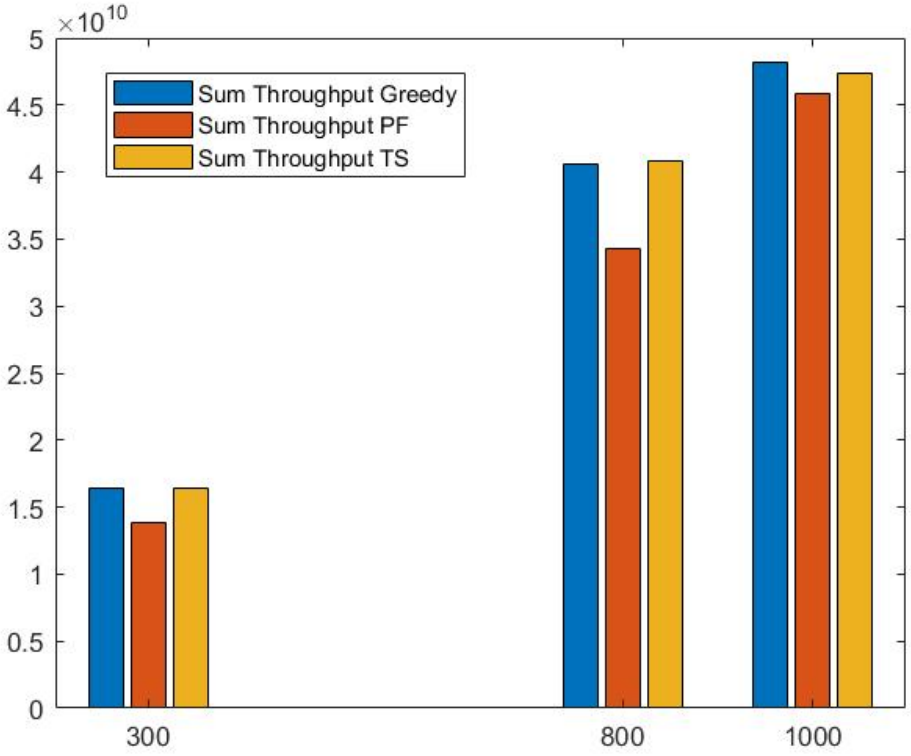


Figure 6.13: Sum of throughputs metric for $N = 300, 800, 1000$ for Greedy, PF, Time-slot based algorithms

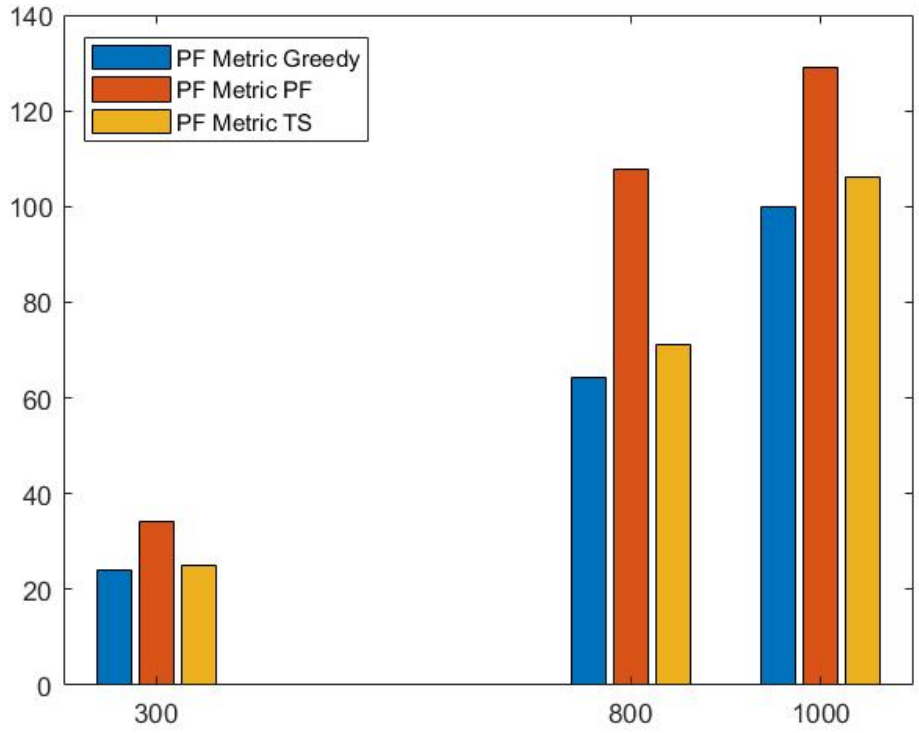
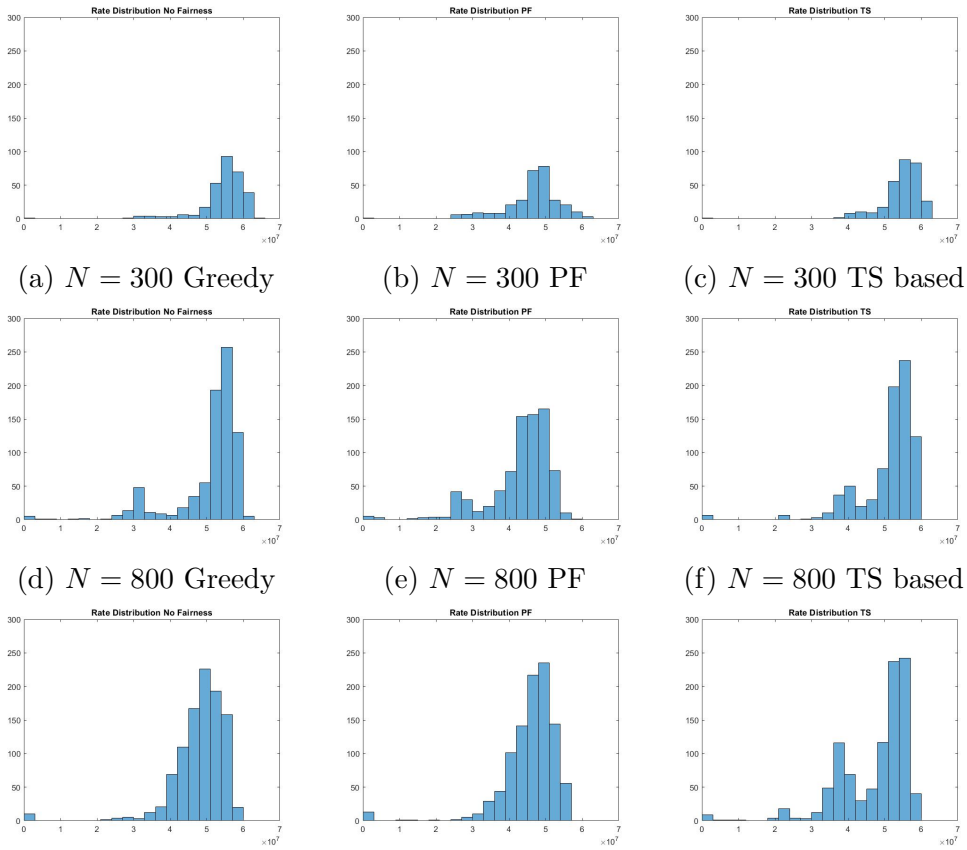


Figure 6.14: Sum of log throughputs metric for $N = 300, 800, 1000$ for Greedy, PF, Time-slot based algorithms



(a) $N = 300$ Greedy (b) $N = 300$ PF (c) $N = 300$ TS based

(d) $N = 800$ Greedy (e) $N = 800$ PF (f) $N = 800$ TS based

(g) $N = 1000$ Greedy (h) $N = 1000$ PF (i) $N = 1000$ TS based

Figure 6.15: Rate distribution comparison for the Greedy, PF, Time-slot based algorithms for $N = 300, 800, 1000$. First, second, and third rows respectively

6.4 Comparison with Two-Channel 802.11

As seen from the above results, the hybrid protocol's distinct advantage is the control-data plane separation, i.e., the extra channel. In this section, we compare the performance of our protocol, in the same metrics, to a 802.11n protocol with two active channels where CSMA/CA and the usual association and data transmission processes are used. The users split between the channels in an arbitrary way, placing $N/2$ on the first and the rest on the second. This is equivalent to having two APs with $N/2$ users on each channel since the protocol runs independently on both. Therefore, the results would scale by a factor of 2 for each N, meaning that, for example, the results obtained for $N = 100$ using one channel will be the results for $N = 200$ using both channels, therefore simulations are not needed. The results are plotted against N and are shown in Figures 6.16 to 6.20. Clearly, there is an improvement in performance, but there comes a point where each channel cannot handle additional load, thus it is better

to resort to a collision-free medium access scheme such as TDMA. This scaling in performance is linear in the number of orthogonal channels due to the same reasons stated above. However, we are only limited to three of those channels in 802.11, so there is a limit to the performance improvements of CSMA/CA and a collision-free protocol becomes better.

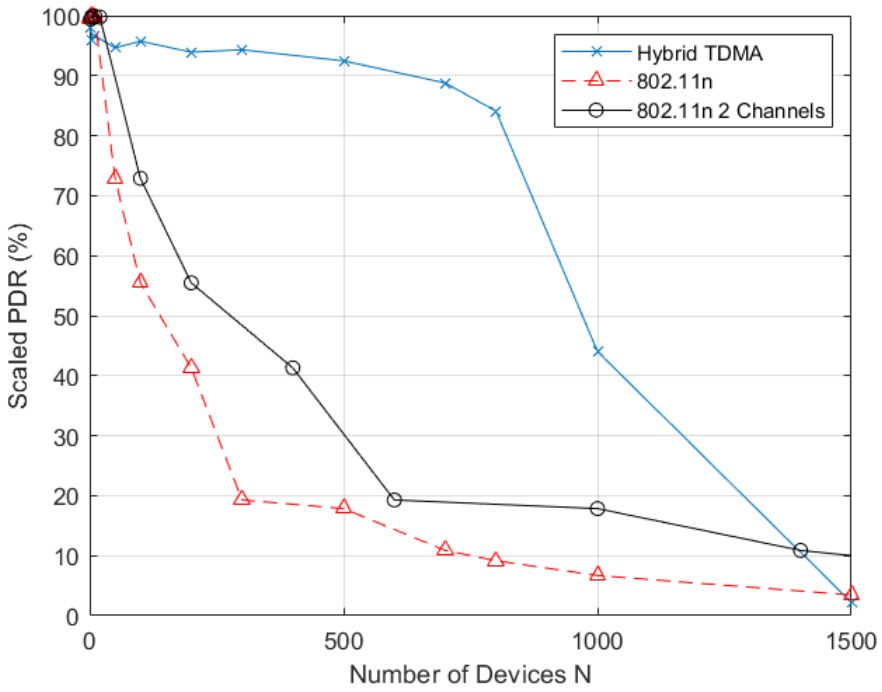


Figure 6.16: PDR comparison with 2 channels 802.11n

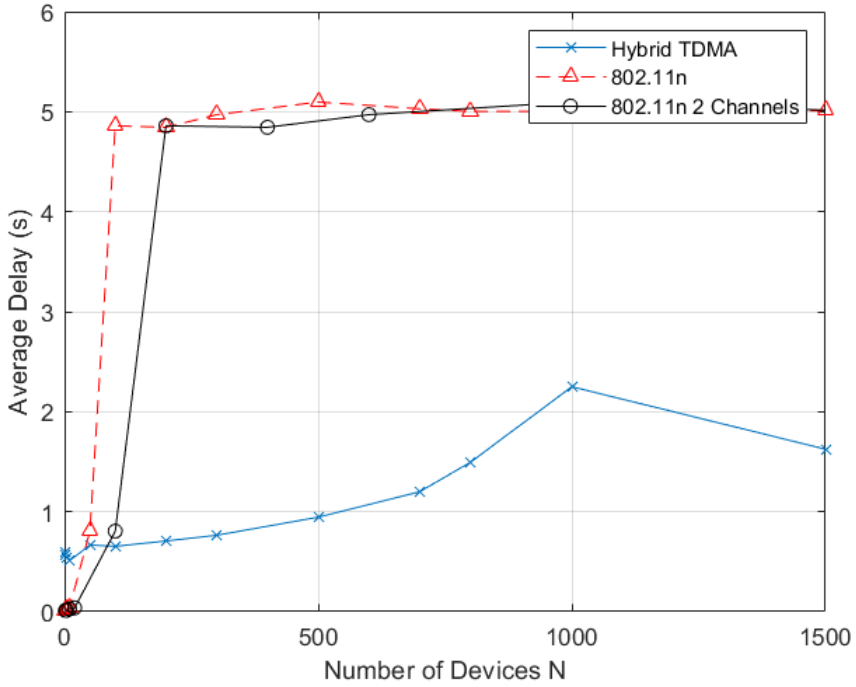


Figure 6.17: Delay comparison with 2 channels 802.11n

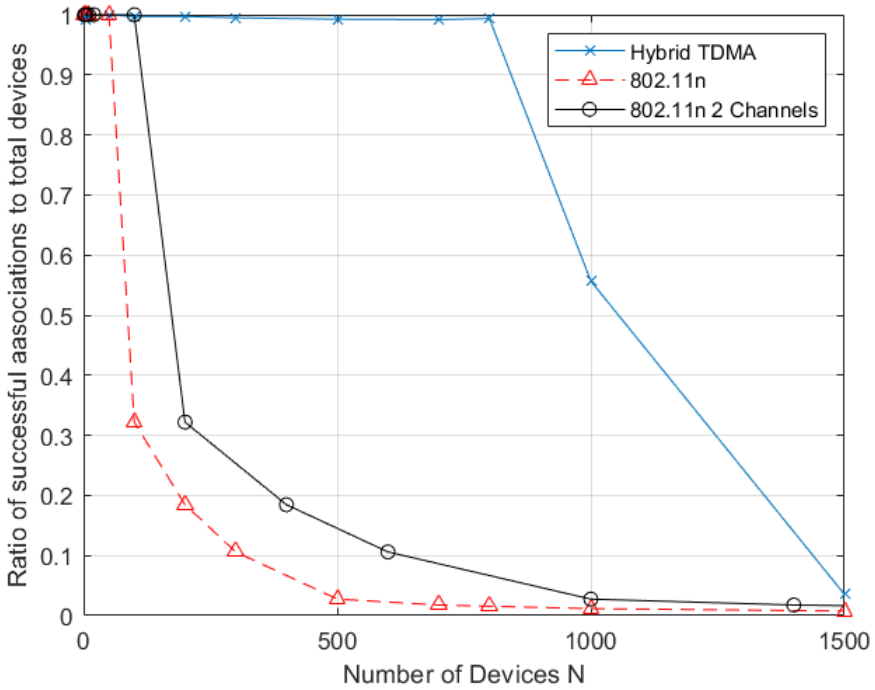


Figure 6.18: Association ratio comparison with 2 channels 802.11n

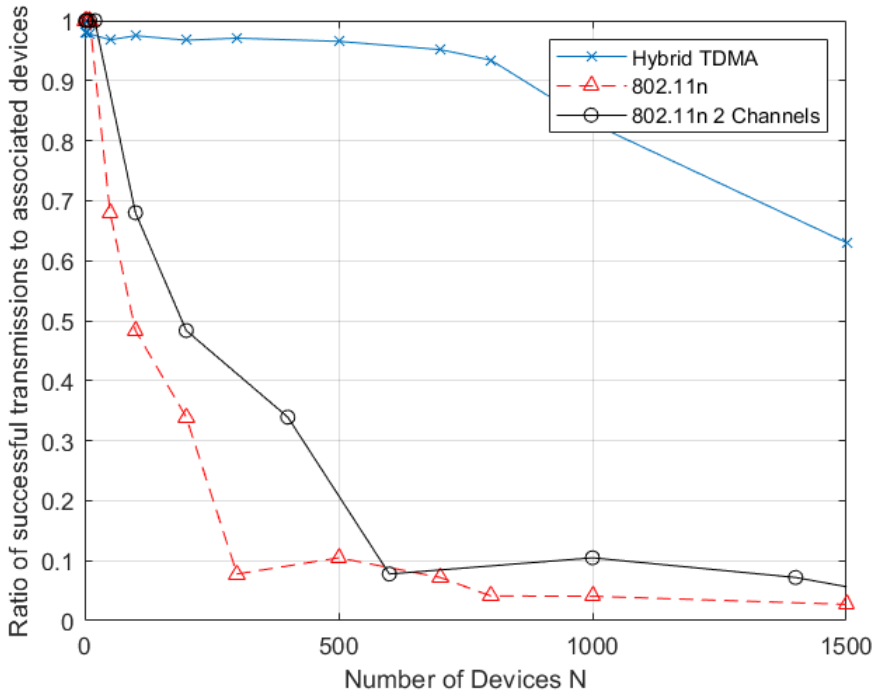


Figure 6.19: Successful transmission to associated users with 2 channels 802.11n

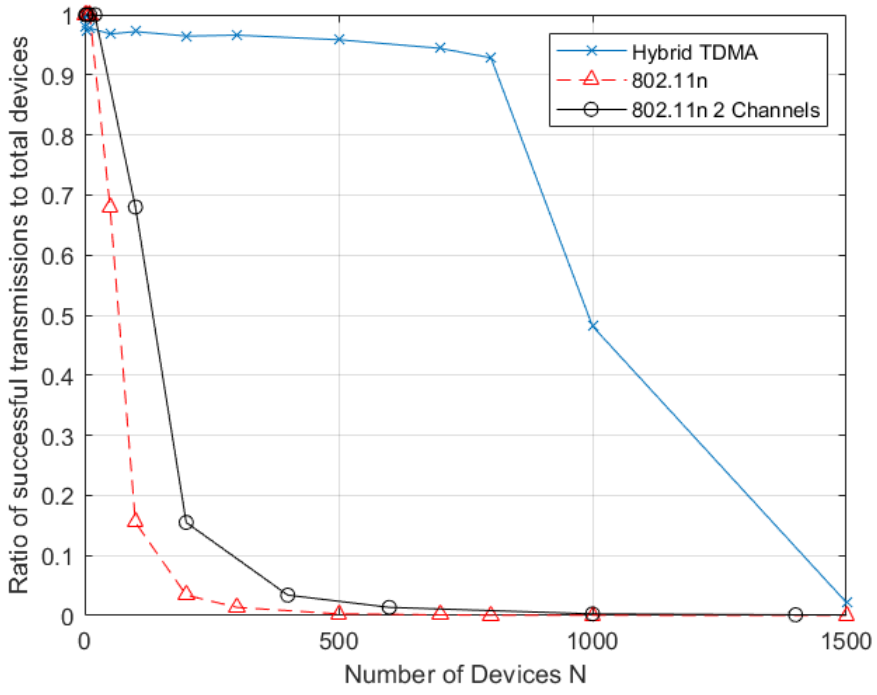


Figure 6.20: Successful transmission to total users with 2 channels 802.11n

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In summary, we presented in this thesis a practical hybrid TDMA/CSMA protocol based on standard compliant modifications of the IEEE 802.11 protocol, control-data plane separation on orthogonal channels, and scheduling. We analyzed the performance of our protocol and proved that it outperforms currently used CSMA/CA based random access in commercial 802.11 standards, as it can handle up to 1000 devices on a single AP reliably whereas 802.11 can handle much less. We also proposed and analyzed two fairness algorithms for improved user scheduling and rate allocation.

The results showed significant improvement in the packet delivery ratio, delay, associated users and served users. The proposed fairness algorithms' performance showed improved rate allocation over the initial scheduling algorithm. Comparison with a two-channel 802.11 protocol, for a fair evaluation of both, also showed the superiority of our protocol.

7.2 Discussion

In light of the above results, a possible question is whether results would have changed if the more recent 802.11ac standard was used instead of 802.11n. Both protocols use the same medium access scheme CSMA/CA, the only additions in 802.11ac are the increased spatial streams, increased bandwidth and modulation order, hence increased rates, and the DL MU-MIMO at the AP. No modifications of the MAC protocol are made, since the goal of this standard is to increase throughput while keeping backwards compatibility and coexistence with previous versions of the standard. Therefore, if we use one spatial stream in the simulation, the results will not change significantly for 802.11ac, and for the hybrid protocol the only difference will be two additional rates in the MCS-SNR mapping table due to the higher modulation order, which will not affect the results significantly

and the same performance gains compared to 802.11ac will be present.

Even if we extend the results to 802.11n or 802.11ac with multiple spatial streams for our hybrid protocol, this will only reduce the time slot duration of those with more spatial streams, which will not affect the number of associated users, delay and packet delivery ratio. The number of users granted a time slot at every cycle (compared to the associated users and total users) will improve only for very large values of N where the scheduling algorithm runs out of resources when one spatial stream is used. However, this increase will not be drastic, since for $N = 1000$ the ratio of users granted a slot to associated ones is around 0.8, which might increase to 1 since the slot times will be divided by the number of spatial streams, thus granting more free time in the cycle. As for the number of granted users to total users, the metric will not increase by a drastic amount since there will still be many unassociated users not having access to one of the TDMA data channels. The fairness results will not change, only the x-axis values will change to the higher rates. Therefore, the same conclusions, comparisons and analysis will be drawn from the results if more than one spatial stream was used.

Regarding the scheduling algorithms used: these algorithms' running time can be improved from $\mathcal{O}(N^2)$ to $\mathcal{O}(N \log N)$, by sorting users in descending order of SNR (algorithm 1), descending order of utility functions (algorithm 2), or ascending order of utility (algorithm 3), then looping over all users while computing the individual slot time of each and accumulating the sum until we reach T_c. The users not looped over, would not be served in the next cycle. This leads to $\mathcal{O}(N \log N)$ complexity. The reduction would greatly improve running time, but a similar scheduling result would be achieved, since in our implementation we are also eliminating users with large time slots, however in a less efficient way. The reduction in time would reduce the overhead of the beacon period compared to the cycle time, but the overhead of sending the large beacon frame would still be present.

One would notice a similarity in features between our proposed solution and the 802.15.4 protocol, which is the MAC protocol at the base of Zigbee. The distinguishing factors between them are, first, that 802.15.4 cannot handle user densities as large as the ones we presented. In the literature, the work in [39] studies its performance in dense wireless sensor networks and shows the degradation in performance (in terms of PDR and association delay) at a relatively low node density of 120, whereas our solution's performance degrades starting at 1000 devices. Another distinguishing factor is the large adoption of WiFi technology: most, if not all, devices come with built-in WiFi hardware not 802.15.4 hardware.

7.3 Future Work

The straightforward extension of this work is to adapt the hybrid protocol to fully compliant 802.11ax protocol, where we would have to adapt the scheduling

algorithms to the version of OFDMA used in 802.11ax with MU-MIMO. Since this standard uses a collision free uplink and downlink access in OFDMA, we expect its performance without any modifications to be much better than the presented 802.11n and incomplete 802.11ax. Its performance would still need to be compared to our hybrid protocol, see where some improvements can be made and try to make these improvements standard compliant.

Since the UL transmissions in 802.11ax are already organized in frames, the problem formulation will be to allocate RUs such that a utility function is maximized (which could be the PF utility or sum rates utilities among others). We would also need to take into account how many OFDMA frames can be fit into each cycle, denoted as K. There has been some work done on theoretical performance of 802.11ax OFDMA schedulers in [40] with different utility functions. The authors proposed an iterative algorithm to find the best RU configuration (number of RUs and size of each in terms of subcarriers) and best common MCS for all users, then they solved the RU allocation problem, formulated as an assignment problem, using the Hungarian Algorithm in polynomial time. Our work however, would assign different MCS values for each user depending on the RU size (as defined in the 802.11ax drafts, higher MCS values are only supported on RUs of bigger size). But since the UL OFDMA transmissions need to be of the same size to be put in the common OFDMA UL frame, users whose total transmission time is less than the longest time would need to use padding (also defined in the standard). Since in the 802.11ax standard, buffer status report is sent using CSMA/CA, we would need to disable it since we are assuming users always have data to send. Also, since the standard defines RUs specific for random access for users who are not allocated any RU yet, we could start with a static RU allocation for users since random access will cause collisions, then proceed with dynamic allocations.

We index each RU by the variable j, each user by i, and each OFDMA frame by k. We assume there are N users competing for M RUs, and that the OFDMA frame duration is at most equal to T_c. Let $x_{i,j}^{(k)}$ be the binary variable that indicates if user i is allocated RU j during OFDMA frame k. Let the utility function for each user i in RU j be $U_{i,j}^{(k)}$ indexed by the same variables. The optimization would start by assuming the maximum k is $\lfloor \frac{T_c}{T_{frame,max}} \rfloor$, where $T_{frame,max}$ is the longest OFDMA frame duration including Trigger Frame, Block ACK, and data transmission at the lowest MCS for all users. Then at the end of

each OFDMA frame k, the AP will solve the following optimization problem:

$$\max_{x_{i,j}^{(k)}} \quad \sum_{i=1}^{N}\sum_{j=1}^{M} x_{i,j}^{(k)} U_{i,j}^{(k)} \tag{7.1}$$

$$\text{s.t.} \quad \sum_{i=1}^{N} x_{i,j}^{(k)} \leq 1 \qquad \forall j \tag{7.2}$$

$$\sum_{j=1}^{M} x_{i,j} \leq 1 \qquad \forall i \tag{7.3}$$

$$x_{i,j}^{(k)} \in \{0,1\} \qquad \forall i,j \tag{7.4}$$

Where the first constraint restricts each user to have only one RU, and the second constraint restricts each RU to be allocated to one user only. This is an assignment problem which can be solved by the Hungarian Algorithm [40]. To find the best RU configuration and MCS of each user, we proceed as the authors did in [40] by iteratively trying all configurations and MCS values, use the Hungarain Algorithm each time, but excluding the configurations where an additional split of RUs would give worse performance. However, for each user and RU we would need to try all MCS values to find the optimal one, hence more complexity.

Finally, we would also need to keep track of which users are given an RU in each frame in order to prioritize them in the next frames (all within the same cycle), hence why the indexing with k. The PF fairness would naturally extend from the above simply by substituting $U_{i,j} = \frac{R_{i,j}}{Q_i}$ [40], where $R_{i,j}$ is the rate of user i in RU j (depends on MCS number and RU size), and Q_i is the amount of data sent by user i divided their time.

Another direction for this work is to incorporate different data packet sizes for different users, different priorities of data, or even more than one packet per cycle. This would make the optimization problem formulation more complicated, and the complexity of the scheduling algorithm would definitely increase. Some users would require multiple time slots to finish their packet transmissions or it would be fragmented into different TDMA cycles. Also, queue length of each user must be taken into account, along with how many transmission opportunity each user received in the past. This would require a buffer status report sent by each user after their data transmission or during a specific buffer report period, all at the cost of extra overhead and more complex standard modifications. This can also be extended to users who do not have any data to send, thus relaxing our initial assumption and making the protocol more general. A user without any data will report a buffer length of 0 to the AP and will not be taken into consideration in the scheduling algorithm.

A third direction, which is more experimental, is to implement a MAC hardware testbed and test our protocol in real scenarios. This has its own complications, from clock hardware synchronization for accurate time slots and reimplementing most of the 802.11 MAC features, which is a very time consuming and tedious task because of their extremely large number. We would expect a performance slightly worse than the simulated results due to the complications stated and due to external interference from other WiFi sources which will cause a drop in PDR and will increase delays since the channel might be sensed busy due to the interference. The limitation of this direction is the scale to which we can go, since it is very hard and not cost efficient to get around 1000 testbeds and program them just to test the performance of our protocol.

Appendix A

Abbreviations

IEEE	Institute of Electrical and Electronics Engineers
MAC	Medium Access Control
PHY	Physical Layer
CSMA/CA	Carrier Sensing Multiple Access with Collision Avoidance
CSMA	Carrier Sensing Multiple Access
CW	Contnetion Window
TDMA	Time Division Multiple Access
UL	Uplink
DL	Downlink
OFDMA	Orthogonal Frequency Division Multiple Access
MIMO	Multiple Input Multiple Output
MU-MIMO	Multi-User MIMO
IoT	Internet of Things
M2M	Machine-to-Machine
LTE	Long Term Evolution
STA	802.11 Station
AP	Access Point
AID	Association ID
ACK	Acknowledgment Frame
SIFS	Short Interframe Spacing
DIFS	DCF Interframe Spacing
DCF	Distributed Coordination Function
SRA	Scheduling and Resource Allocation
MCS	Modulation and Coding Scheme
SNR	Signal-to-Noise Ratio
SINR	Signal-to-Interference-and-Noise Ratio
ILP	Integer Linear Programming
PF	Proportionally Fair
UDP	User Datagram Protocol
IP	Internet Protocol

IPv4	Internet Protocol Version 4
PDR	Packet Delivery Ratio

Bibliography

[1] Cisco, "Cisco visual networking index complete traffic forecast (2016 – 2021) Q&A." https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/qa_c67-482177.html, June 2017.

[2] S. C. Mukhopadhyay, "Wearable sensors for human activity monitoring: A review," *IEEE Sensors Journal*, vol. 15, no. 3, pp. 1321–1330, 2015.

[3] A. Ghosh, J. Zhang, J. G. Andrews, and R. Muhamed, *Fundamentals of LTE*. Upper Saddle River, NJ, USA: Prentice Hall Press, 1st ed., 2010.

[4] "IEEE standard for information technology–telecommunications and information exchange between systems local and metropolitan area networks–specific requirements - part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," *IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012)*, pp. 1–3534, 2016.

[5] T. Adame, A. Bel, B. Bellalta, J. Barcelo, and M. Oliver, "IEEE 802.11ah: the WiFi approach for M2M communications," *IEEE Wireless Communications*, vol. 21, no. 6, pp. 144–152, 2014.

[6] B. Bellalta, "IEEE 802.11ax: High-efficiency WLANs," *IEEE Wireless Communications*, vol. 23, no. 1, pp. 38–46, 2016.

[7] J. Lee, "OFDMA-based hybrid channel access for IEEE 802.11ax WLAN," in *Proc. 14th Int. Wireless Communications Mobile Computing Conf. (IWCMC)*, 2018.

[8] E. Khorov, A. Kiryanov, A. Lyakhov, and G. Bianchi, "A tutorial on IEEE 802.11ax high efficiency WLANs," *IEEE Communications Surveys Tutorials*, vol. 21, no. 1, pp. 197–216, 2019.

[9] D. Deng, S. Lien, J. Lee, and K. Chen, "On quality-of-service provisioning in IEEE 802.11ax WLANs," *IEEE Access*, vol. 4, pp. 6086–6104, 2016.

[10] N. Shahin, R. Ali, and Y. T. Kim, "Hybrid slotted-CSMA/CA-TDMA for efficient massive registration of IoT devices," *IEEE Access*, vol. 6, pp. 18366–18382, 2018.

[11] S. Zhuo and Y. Q. Song, "GoMacH: A traffic adaptive multi-channel MAC protocol for IoT," in *2017 IEEE 42nd Conference on Local Computer Networks (LCN)*, 2017.

[12] Y. Liu, C. Yuen, X. Cao, N. U. Hassan, and J. Chen, "Design of a scalable hybrid mac protocol for heterogeneous m2m networks," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 99–111, 2014.

[13] A. D. Shoaei, M. Derakhshani, S. Parsaeefard, and T. Le-Ngoc, "Efficient and fair hybrid TDMA-CSMA for virtualized green wireless networks," in *Proc. IEEE 84th Vehicular Technology Conf. (VTC-Fall)*, 2016.

[14] A. Franco, S. Bastani, E. Fitzgerald, and B. Landfeldt, "Omac: An opportunistic medium access control protocol for IEEE 802.11 Wireless networks," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, 2015.

[15] C. W. Park, D. Hwang, and T. Lee, "Enhancement of IEEE 802.11ah MAC for M2M communications," *IEEE Communications Letters*, vol. 18, no. 7, pp. 1151–1154, 2014.

[16] B. Shrestha, E. Hossain, and K. W. Choi, "Distributed and centralized hybrid CSMA/CA-TDMA schemes for single-hop wireless networks," *IEEE Transactions on Wireless Communications*, vol. 13, no. 7, pp. 4050–4065, 2014.

[17] K. Wang and K. Psounis, "Scheduling and resource allocation in 802.11ax," in *Proc. IEEE INFOCOM 2018 - IEEE Conf. Computer Communications*, 2018.

[18] A. Pantelidou and A. Ephremides, "The scheduling problem in wireless networks," *Journal of Communications and Networks*, vol. 11, no. 5, pp. 489–499, 2009.

[19] A. Pantelidou and A. Ephremides, "A cross-layer view of optimal scheduling," *IEEE Transactions on Information Theory*, vol. 56, no. 11, pp. 5568–5580, 2010.

[20] M. Karaca, S. Bastani, B. E. Priyanto, M. Safavi, and B. Landfeldt, "Resource management for OFDMA based next generation 802.11 Wlans," in *Proc. 9th IFIP Wireless and Mobile Networking Conf. (WMNC)*, 2016.

[21] A. Mohan, A. Chattopadhyay, and A. Kumar, "Hybrid MAC protocols for low-delay scheduling," in *Proc. IEEE 13th Int. Conf. Mobile Ad Hoc and Sensor Systems (MASS)*, 2016.

[22] D. Malak, H. Huang, and J. G. Andrews, "Throughput maximization for delay-sensitive random access communication," *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 709–723, 2019.

[23] E. Yaacoub, A. M. El-Hajj, and Z. Dawy, "Ergodic sum-rate maximization in OFDMA uplink with discrete rates," in *Proc. European Wireless Conf. (EW)*, pp. 57–62, 2010.

[24] E. Yaacoub and Z. Dawy, "Fair optimization of video streaming quality of experience in LTE networks using distributed antenna systems and radio resource management," *Journal of Applied Mathematics*, vol. 2014, 2014.

[25] E. Yaacoub and Z. Dawy, "A survey on uplink resource allocation in OFDMA wireless networks," *IEEE Communications Surveys Tutorials*, vol. 14, no. 2, pp. 322–337, 2012.

[26] E. Yaacoub and Z. Dawy, "Proportional fair scheduling with probabilistic interference avoidance in the uplink of multicell OFDMA systems," in *Proc. IEEE Globecom Workshops*, pp. 1202–1206, 2010.

[27] A. Pantelidou and A. Ephremides, "What is optimal scheduling in wireless networks?," in *Proceedings of the 4th Annual International Conference on Wireless Internet*, WICON '08, (ICST, Brussels, Belgium, Belgium), ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008.

[28] and Guocong Song and Ye Li, "Cross-layer optimization for OFDM wireless networks-part i: theoretical framework," *IEEE Transactions on Wireless Communications*, vol. 4, no. 2, pp. 614–624, 2005.

[29] X. Xiao, X. Tao, and J. Lu, "Energy-efficient resource allocation in LTE-based MIMO-OFDMA systems with user rate constraints," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 1, pp. 185–197, 2015.

[30] D. Bankov, A. Didenko, E. Khorov, V. Loginov, and A. Lyakhov, "IEEE 802.11ax uplink scheduler to minimize, delay: A classic problem with new constraints," in *Proc. and Mobile Radio Communications (PIMRC) 2017 IEEE 28th Annual Int. Symp. Personal, Indoor*, 2017.

[31] A. Ephremides and T. V. Truong, "Scheduling broadcasts in multihop radio networks," *IEEE Transactions on Communications*, vol. 38, no. 4, pp. 456–460, 1990.

[32] D.-S. Chen, R. G. Baston, and Y. Dang, *Applied Integer Programming: Modeling and Solution*. Wiley, 2010.

[33] "The network simulator - ns-3." Available: https://www.nsnam.org/.

[34] "The ns-3 wi-fi module documentation," 2016. Available: https://www.nsnam.org/bugzilla/attachment.cgi?id=2230.

[35] H. L. Vu and T. Sakurai, "Collision probability in saturated IEEE 802.11 networks," in *Australian Telecommunication Networks and Applications Conference, Australia*, 2006.

[36] "Design documentation - model library," 2019.

[37] H. Zhai, Y. Kwon, and Y. Fang, "Performance analysis of IEEE 802.11 MAC protocols in wireless LANs," *Wireless communications and mobile computing*, vol. 4, no. 8, pp. 917–931, 2004.

[38] Q. Wang, K. Jaffrès-Runser, J.-L. Scharbarg, C. Fraboul, Y. Sun, J. Li, and Z. Li, "A thorough analysis of the performance of delay distribution models for IEEE 802.11 DCF," *Ad Hoc Networks*, vol. 24, pp. 21 – 33, 2015.

[39] E. E. Petrosky, A. J. Michaels, and D. B. Ridge, "Network scalability comparison of IEEE 802.15.4 and receiver-assigned CDMA," *IEEE Internet of Things Journal*, pp. 1–1, 2019.

[40] D. Bankov, A. Didenko, E. Khorov, and A. Lyakhov, "OFDMA uplink scheduling in IEEE 802.11ax networks," in *2018 IEEE International Conference on Communications (ICC)*, pp. 1–6, 2018.