Performance Optimization in Large-Scale 802.11n Wireless Local Area Networks

A Dissertation Presented

by

Dawei Gong

to

The Graduate School

in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in

Computer Engineering

Stony Brook University

May 2014

Stony Brook University

The Graduate School

Dawei Gong

We, the dissertation committee for the above candidate for the Doctor of Philosophy degree, hereby recommend acceptance of this dissertation.

Yuanyuan Yang – Dissertation Advisor
Professor, Department of Electrical and Computer Engineering

Sangjin Hong – Chairperson of Defense
Professor, Department of Electrical and Computer Engineering

Dantong Yu
Adjunct Professor, Department of Electrical and Computer Engineering

Samir R. Das
Professor, Department of Computer Science

This dissertation is accepted by the Graduate School.

Charles Taber
Dean of the Graduate School

Abstract of the Dissertation

Performance Optimization in Large-Scale 802.11n Wireless Local Area Networks

by

Dawei Gong

Doctor of Philosophy

in

Computer Engineering

Stony Brook University

2014

Recently, wireless local area networks (WLANs) have become an indispensable part of our daily life. To enhance the rate, range and reliability of WLANs, the IEEE 802.11n standard has introduced several new technologies, such as multiple input multiple output (MIMO), channel bonding and frame aggregation. However, the performance of WLANs is often unsatisfactory. In this dissertation, we study performance optimization in large-scale 802.11n WLANs, aiming at improving network throughput, reducing transmission failures, and enhancing the reliability and efficiency of link-layer multicast.

In WLANs, clients need to associate with an access point (AP) to access the network. An AP and its associated clients operate on the same channel. The performance of WLANs can be adversely affected if too many clients associate with the same AP,

or nearby APs operate on overlapping channels. The problem becomes more severe in 802.11n WLANs because of the channel bonding mechanism, which combines two adjacent 20MHz channels together for data transmissions, and the presence of heterogeneous 802.11a/b/g/n clients. We first introduce mathematical models to estimate the client throughput in 802.11n WLANs. Based on these models, we propose AP association and channel assignment algorithms to maximize the network throughput. We further present low-complexity algorithms that minimize interference and contentions on high-rate clients in order to improve network performance.

Another factor that affects performance in 802.11n WLANs is transmission failures, which are often caused by varying channel conditions. To ensure high reliability, failed frames are automatically retransmitted in WLANs. However, due to the temporal and spatial correlation of channel errors, retransmissions for a failed frame may also fail at a high probability. To address this issue, we design a cooperative retransmission protocol for 802.11n WLANs, where each node dynamically selects a neighbor that can overhear its transmission to help retransmitting. If the direct transmission fails, the selected neighbor retransmits the failed sub-frames of the aggregated frame. Transmission failures can also be caused by collisions, especially in WLANs that have heavy traffic load or a large number of clients. An AP can operate in the point coordination function (PCF) mode and poll its associated clients in turn, so as to achieve contention-free transmission. Nevertheless, the client polling in neighboring BSSs may still collide with each other due to the lack of coordination. Based on this observation, we study high-throughput collision-free client polling in large-scale WLANs that operate in the PCF mode.

For many multimedia applications, e.g., video streaming, video conference, the sender needs to transmit every frame to multiple recipients, which could place tremendous traffic loads on WLANs.

Link-layer multicast is a promising technology to greatly reduce this type of traffic loads thanks to the broadcast nature of the wireless medium. However, it is rarely used in practice due to the lack of reliability and efficiency. By taking advantage of smart antennas, we set up a system for multicast in 802.11n WLANs, with the objective to delivering multicast frames to all multicast clients reliably and efficiently. We have carried out extensive simulations and experiments, and the results show that the proposed schemes can significantly boost the network performance of large-scale 802.11n WLANs.

To my mother, Meixian Yang, for her love.

Contents

List of Figures

List of Tables

Acknowledgements

This dissertation would not have been accomplished without the support and encouragement from so many people.

First and foremost, I would like to express my deep gratitude to my advisor, Prof. Yuanyuan Yang, for her invaluable guidance and strong support. Throughout the course of dissertation, she always gave me constructive comments and feedback for improvement. I am so thankful to have had a great opportunity to work closely with her on networking research projects for six years. She mentored me on almost everything related to research and professional development. From the numerous meetings with her, I have learned not only the knowledge in the networking research field, but also what it means for one to have high level of academic professionalism. Her advice will continue to help me in my future career development.

I would also like to thank my dissertation committee members, Prof. Sangjin Hong and Prof. Dantong Yu in the department of Electrical and Computer Engineering, and Prof. Samir R. Das in the department of Computer Science. Thanks for their precious time and insightful suggestions to improve the quality of my dissertation.

I am also grateful to my colleagues Dr. Min Yang, Dr. Miao Zhao, Dr. Lin Liu, Dr. Xi Deng, Ji Li, Zhexi Pan, Zhiyang Guo, Zhemin Zhang, Cong Wang, Jun Duan, Zhenhua Li and Hao Li in Mobile Computing Laboratory for their friendship and all of their help over years.

I would also like to express my gratitude to many individuals in the department who have made my study at Stony Brook University wonderful and memorable. Graduate Program Coordinator, Rachel Ingrassia, have been especially helpful.

I would also like to thank Dr. Huayan Amy Wang, Dr. Allen Miu and Dr.

Joshua Zhao at Ruckus Wireless Inc, for sharing their expertise on WLANs with me. I am also thankful to Prof. Hewu Li at Tsinghua University, for his useful advices on devising and improving algorithms in the dissertation. I am very grateful to Prof. Fenghua Li and Prof. Qian Wu at Tsinghua University, for their friendship and mentorship.

I am indebted to my parents and my sister for their support during my doctoral study. Without their unconditional love, I would not have come to the United States to pursue a doctoral degree. My mother always told me to focus on study, even in her last few months of life. My father is my mentor whenever I need advice on either study or life. My sister is always willing to listen to my stories, and share her options with me. A special thank goes to my wife, Qian Hu, who is now a professor at the University of Central Florida. Her love, encouragement and support has provided me strength to complete the doctoral study, and will continue to be my source of energy in the wonderful journey of life. Thank you all!

Chapter 1

Introduction

This chapter explains the motivation, optimization objectives, open challenges, and contributions of the dissertation.

1.1 Motivation

Recently, IEEE 802.11 [1] based wireless local area networks (WLANs) have been widely deployed at homes, schools, airports, enterprises, etc., to provide wireless network access. Moreover, with the popularity of smartphones and tablets, WLANs have also been deployed by mobile network carriers in large scale, to offload the explosively growing data traffic from mobile networks. On the other hand, in the IEEE 802.11n standard, several new technologies have been introduced to improve the range, rate and reliability of WLANs. First, multiple-input multiple-output (MIMO) technology is used to boost physical rates and reliability by transmitting multiple spatial streams simultaneously or exploring spatial diversity. In addition, the maximum coding rate is increased from 3/4 to 5/6 and a short guard interval (400ns) between orthogonal frequency-division multiplexing (OFDM) symbols is introduced to improve spectrum efficiency and thus maximum physical rates. Furthermore, a channel bonding technology, also known as 40MHz channel, is applied to further enhance physical rates through combining two non-overlapping 20MHz channels together for data transmissions. At the MAC layer, frame aggregation mechanism is employed such that multiple frames are aggregated into a single

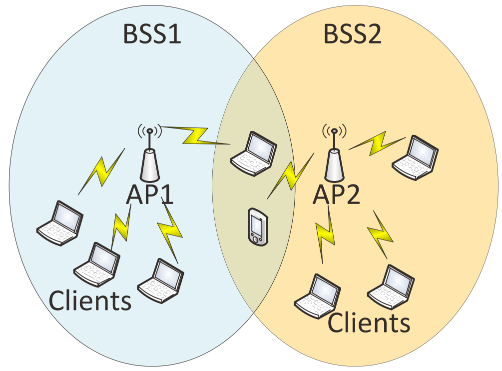


Figure 1.1: A typical deployment of wireless local area networks.

frame before transmission. Each aggregated frame is acknowledged by a block ACK frame, in which a bitmap is used to acknowledge all sub-frames. In this way, both the MAC overhead and random backoff period due to carrier sense multiple access with collision avoidance (CSMA/CA) are greatly reduced.

As shown in Fig. 1.1, a WLAN typically consists of multiple access points (APs), which are connected to the Internet or an Intranet via wired connections. WLAN clients need to associate with an AP to access the network. An AP and its associated clients are referred to as a basic service set (BSS). Each AP operates on a specific channel in the 2.4GHz or 5GHz frequency band. A client uses the same channel as its associated AP. In large-scale 802.11n WLANs, APs are often densely deployed to provision anytime anywhere wireless connections. As a result, neighboring BSSs may interfere with each other if they operate on the same channel, as all clients in them need to contend and share the channel. Moreover, a client is often in the coverage of multiple APs and needs to choose one of them to associate with. By default, a client associates with the AP that has the best signal quality.

The performance of a large-scale 802.11n WLAN is highly related to the resource allocation strategies, including both AP association decisions for clients and channel assignments for APs. Previous studies [2, 3] have shown that WLAN clients tend to stay in particular areas of network due to various reasons, such as proximity to power outlets, attending a conference, etc. Thus with the default best-signal association strategy, APs in the hotspots are overloaded while APs in other areas are underutilized. In addition, due to the limited number of channel resources, it is a challenging task to assign channels to all APs such that the network performance is optimized. Extensive research effort has been devoted to AP association [4–8, 11, 12] and channel assignment [29–32, 34, 35] for legacy 802.11a/b/g WLANs. However, these existing resource allocation schemes may not lead to optimal network performance in 802.11n WLANs, since the impact of heterogeneous clients, as well as new features in 802.11n WLANs have not been considered.

In WLANs, a station can transmit frames at a series of different data rates to accommodate various channel conditions. The throughput of all clients in a BSS and neighboring BSSs that operate on the same channel is dominated by the client that has the lowest data rate, as the CSMA/CA mechanism ensures that each client has an equal opportunity to access the wireless medium in the long term, regardless of its data rate [36]. In addition, when coexisting with 802.11a/b/g clients, 802.11n stations need to use longer preambles or even RTS/CTS to protect their transmissions from corruptions. As a result, low-rate, legacy clients can seriously affect the network throughput of high-rate clients, which raises new challenges to the AP association problem. Moreover, the scarcity of channel resources becomes more evident in 802.11n WLANs because of the channel bonding mechanism. It has been shown via experiments that transmissions over bonded channels are prone to interference [41, 42], even if the interference is on only one of the two combined channels. On the other hand, legacy clients can significantly undermine the benefits of channel bonding, since they can transmit on only one half of the bonded channel and occupy the channel for a very long time.

Another factor that affects WLAN performance is frame retransmissions. Due to varying channel conditions and external interference, frames transmitted in wireless networks have a much higher error rate than in wired networks. To ensure high reliability, failed frames are retransmitted automatically in WLANs. Such retransmissions would occupy the wireless medium and thus degrade network performance. Moreover, channel errors and interference in WLANs tend to be in burst and temporal related, thus the retransmissions following an unsuccessful transmission may fail at a high probability as well, which would further deteriorate the performance of WLANs. On the other hand, due to the broadcast nature of the wireless medium, the neighbors of the sender may successfully overhear a frame that fails to reach the destination. If a neighbor instead of the sender retransmits the failed frame, the receiver may receive it with a much higher probability because of the spatial diversity of wireless medium. Based on this observation, several cooperative retransmission protocols [49, 50, 52, 53, 57–59] have been proposed for legacy WLANs. However, none of these schemes have considered how to cooperative retransmit only the failed sub-frames of aggregated frames in 802.11n WLANs. This could undermine the benefits of cooperative retransmissions, as it is unnecessary to retransmit the sub-frames that have already been received by the receiver.

The performance of WLANs could also be severely affected by collisions. A WLAN typically operates in the distributed coordination function mode, where all stations access the wireless medium using the CSMA/CA mechanism. With CSMA/CA, a station that has pending traffic first senses the wireless medium for a distributed inter-frame space (DIFS) period. If the medium is free during the DIFS period, the station back offs a random number of time slots before transmitting a frame. A collision occurs if multiple stations choose the same backoff duration. As the traffic load or the number of clients grows, the collision probability increases drastically, which results in poor WLAN performance, as a significant amount of medium access time is wasted by collisions. A WLAN can also operate in the point coordination function mode, in which each AP polls its associated clients in a centralized manner, to achieve contention-free transmissions. This mechanism is effective in a single BSS. However, collisions may still occur among transmissions from nearby BSSs, due to the lack of coordination among APs. Although there are a few existing algorithms [67–70] on coordinating client polling among nearby BSSs, they either achieve this goal by scarifying network throughput or have prohibitive complexity and overhead.

For many multimedia applications (e.g., video streaming, video conference), the network support of multicast traffic is critical to achieve satisfactory performance, as each packet needs to be delivered to multiple recipients. Link-layer multicast in WLANs [81] is a promising technology for these applications, since a multicast transmission from the AP can reach all associated clients thanks to the broadcast nature of wireless medium. However, link-layer multicast is rarely used in deployed WLANs since it is neither reliable nor efficient. Different from unicast frames, multicast frames are not acknowledged in WLANs, which leaves the reliability of multicast open, since the frames may be corrupted by channel errors and collisions. Moreover, although the maximum physical data rate in 802.11n WLANs is as high as 600Mbps, multicast frames are transmitted at the basic data rate (6Mbps) to reach clients that have poor channel conditions. One approach [77, 78] to improving the reliability of multicast in WLANs is to enable RTS/CTS and automatic retransmission, where each multicast client sends an ACK after receiving a multicast frame, and the AP retransmits the frame if any client fails to receive the frame. Another approach [79] is to translate multicast frames into separate unicast frames to each client, such that retransmissions and rate adaptation are applied automatically for each client. However, the network performance of the above approaches decreases drastically as the number of multicast clients grows, since the overhead is proportional to the number of clients. On the other hand, it has been shown in practice [86] that the performance of unicast traffic can be greatly improved by the application of smart antennas, which can strengthen the signal quality in desired direction and mitigate interference in other directions by manipulating the radiation pattern. Thus it is interesting to study how to improve the multicast performance in WLANs by using smart antennas.

1.2 Optimization Objectives and Challenges

We now introduce the systematical objectives of performance optimization in large-scale 802.11n WLANs.

- **Proportional Client Throughput.** The CSMA/CA mechanism provides throughput fairness to all clients at the cost of low efficiency. Clients that have good channel condition should not be penalized by clients with poor channel condition. Therefore, it is expected that the throughput of a client is proportional to its physical data rate, which reflects its channel condition.

- **Improved Network Throughput.** The network throughout, which is defined as the aggregated throughput of all clients in the network, is a good metric to assess the network performance. It is expected to have high network throughput in 802.11n WLANs to satisfy the tremendous bandwidth requirements of mobile and multimedia applications.

- **Reduced Transmission Failures.** Transmission failures can be caused by channel errors and collisions. A failed transmission and the follow re-transmissions would occupy the wireless medium for extra time, leading to increased medium access delay and decreased throughput for all clients sharing the medium. It is expected to have low transmission failure rate in order to optimize network performance in 802.11n WLANs.

- **Reliable and Efficient Multicast.** Multicast is essential for many multimedia applications. To achieve satisfactory client experience, multicast traffics should be delivered to all subscribed clients at a high packet reception ratio. Multicast traffics should also be transmitted efficiently, so as to satisfy the rapidly growing bandwidth requirement of multimedia applications, and to minimize the impact on coexisting unicast traffics. It is expected to provision reliable and efficient link-layer multicast in 802.11n WLANs to significantly reduce the bandwidth consumption of multimedia applications.

Given their unique characteristics and universal deployment, we need to handle the following open technical challenges and issues in the performance optimization of WLANs.

- **Limited Channel Resources.** The number of channel resources for WLANs is very limited. For example, there are only three orthogonal 20MHz channels in the 2.4GHz frequency band. Thus it is difficult to assign channels to APs in WLANs, as neighboring BSSs may interfere with each other if operated on overlapped channels. The task becomes more challenging in 802.11n WLANs because of the channel bonding mechanism, which combines two adjacent 20MHz channels together to use high data rates. Therefore, how to assign channels to maximize network throughput is a challenging issue in large-scale 802.11n WLANs.

- **Time-Varying Channel Conditions.** The channel condition of static clients in WLANs is time-varying due to the multi-path effects of signal propagation and external interference. Moreover, a number of clients in WLANs are semi-static, i.e., they tend to move to new locations periodically. Therefore, the network performance of WLANs would become sub-optimal if the performance optimizing schemes are not executed timely with the channel variations. On the other hand, the network performance can also be adversely affected by the overhead of the optimizing schemes if they are executed too frequently. Therefore, how to balance the freshness of channel conditions and the overhead of performance optimization is an open issue in WLANs.

- **Coexistence of Heterogeneous Clients.** As aforementioned, in 802.11n WLANs, the client that has the lowest data rate dominates the throughput of all other clients in a BSS. In addition, 802.11n WLANs are backward compatible to support low-rate 802.11a/b/g clients. The performance of 802.11n WLANs would be further degraded since 802.11n transmissions need to use long preamble or RTS/CTS to protect from corruptions from 802.11a/b/g transmissions. Hence, how to conduct channel assignment and AP association in WLANs with heterogeneous clients so as to optimize network performance is an open and realistic issue.

- **Implementation Considerations.** Numerous WLANs have been deployed worldwide, and billions of mobile devices are equipped with WLAN radio. For a performance optimization scheme to be applicable in realistic WLANs, the scheme has to be compatible with the IEEE 802.11 standard. Moreover, the scheme should also have low complexity, since the computing capability of most APs is very limited. Therefore, it is important to consider the feasibility when designing performance optimization schemes for 802.11n WLANs.

1.3 Contributions

The contributions of this dissertation can be summarized as follows.

- **On-line AP Association Algorithms for 802.11n WLANs with Heterogeneous Clients**. We explore AP association for 802.11n WLANs with heterogeneous clients (802.11a/b/g/n). We first present a Markov model to estimate the uplink and downlink throughput of clients, and formulate the AP association problem into an optimization problem. The objective is to provide each client a bandwidth proportional to its usable data rate. Based on this Markov model, we propose an on-line AP association algorithm, under the condition that each client can acquire timely information of all clients associated with nearby APs. Furthermore, for WLANs where APs are densely deployed, we provide another on-line AP association algorithm with lower complexity, which takes full advantage of 802.11n transmissions by simply associating different types of clients with different APs. We have conducted extensive simulations and experiments to validate the proposed algorithms. The results show that our algorithms can significantly improve both 802.11n throughput and aggregated network throughput under various network scenarios, compared to previous AP association schemes. Our experiments also confirm the effectiveness of the algorithms in enhancing network throughput, maintaining proportional fairness among clients and balancing load among APs.

- **Distributed Channel Assignment Algorithms for 802.11n Wireless LANs with Heterogeneous Clients.** We study channel assignment in 802.11n WLANs with heterogeneous clients. We first present the network model, interference model, and throughput estimation model to estimate the throughput of each client. We then formulate the channel assignment problem into an optimization problem, with the objective of maximizing overall network throughput. Since the problem is NP-hard, we give a distributed channel assignment algorithm based on the throughput estimation model. We then present another channel assignment algorithm with lower complexity, and aim at minimizing interference experienced by high-rate, 802.11n clients. We have carried out extensive simulations to evaluate the proposed algorithms. Simulation results show that our algorithms can significantly improve the network throughput of 802.11n WLANs, compared with other channel assignment algorithms.

- **An Efficient Cooperative Retransmission Protocol for 802.11n WLANs.** We propose an efficient cooperative retransmission MAC (CAR-MAC) protocol for WLANs, which utilizes new features of 802.11n and is compatible with standard 802.11n transmissions. In CAR-MAC, all nodes periodically broadcast a C-Beacon message to release their retransmitting capability, and each node selects a cooperative node based on received C-Beacon messages. If some sub-frames in the aggregated frame from the sender fail to reach the destination, the cooperative node retransmits the failed sub-frames together with its own new sub-frames, such that overhead from cooperative retransmissions is amortized by normal frame transmissions. We have theoretically analyzed the improvement on network throughput brought by CAR-MAC protocol. In addition, we have conducted extensive simulations to evaluate CAR-MAC protocol under various channel conditions. Both theoretical and simulation results show that the proposed protocol can greatly improve network throughput and reduce frame retransmissions, compared with the 802.11n standard and existing cooperative retransmission schemes.

- **High-Throughput Collision-Free Client Polling in Large-Scale WLANs.** We study client polling in multi-AP WLANs, with the objective of providing high-throughput, collision-free channel access for each client, and maximizing network capacity. We first give a WLAN framework in which the PCF of all APs is coordinated and clients are polled in a time slotted manner. We then formulate client polling into a time slot allocation problem and propose a collision-free polling scheme consisting of three procedures: (1) a basic polling procedure that determines the minimum number of time slots required to poll every client once to obtain the polling frequencies of all clients; (2) a complementary polling procedure that makes extra polls for APs that have idle time slots without causing collisions, to improve spatial reuse of the network; (3) a backup poll selecting procedure that finds backup clients to poll in case the current polled client has no data to transmit, to utilize the otherwise wasted bandwidth. We have conducted extensive simulations and compared it with two existing schemes. The simulation results show that the proposed scheme can provide high throughput and uniform channel access time for all clients, while boosting spatial reuse 2 to 3 times compared to other schemes.

- **Link-Layer Multicast in 802.11n WLANs with Smart Antennas.** We consider link-layer multicast in 802.11n WLANs with smart antennas to improve multicast performance. We partition clients into several groups, then select an antenna pattern from smart antennas and a multicast rate for each group, and transmit the same frame to each group. We first examine the gain of smart antennas and reliability of various 802.11n data rates for multicast in indoor WLANs via experiments. We then present the system model for multicast over smart antennas and formulate the problem into a mixed integer program. After that, we propose an optimal algorithm for the mixed integer program, under the condition that the packet reception ratio (PRR) of all antenna patterns and data rates is known for every client. As clients join and leave the network frequently and the wireless channel is time-varying, we also propose an on-line algorithm that is able to adapt the partition of clients, antenna pattern and multicast rate for each group dynamically, based on PRR reports from clients. We have implemented the on-line algorithm on off-the-shelf WLAN products and conducted extensive experiments to evaluate the performance. The results show that the proposed algorithm can significantly improve multicast throughput compared to other strategies, and at the same time guarantee high PRR for all clients.

Our work combines algorithm and protocol design, mathematical modeling, theoretical analysis, simulation evaluation, and experiment validation techniques to carry out comprehensive studies on the above issues. The proposed research will have a significant impact on both fundamental research on performance optimization in large-scale WLANs, and product development in the WLAN industry. The proposed performance optimization schemes can be applied to existing 802.11n WLANs, as well as the forthcoming 802.11ac WLANs.

1.4 Dissertation Outline

The rest of the dissertation is organized as follows. Chapter 2 proposes two on-line algorithms for AP association in 802.11n WLANs with heterogeneous clients. Chapter 3 presents two distributed channel assignment algorithms for 802.11n WLANs, with the objective to maximizing network throughput. Chapter 4 introduces an efficient cooperative retransmission protocol for 802.11n WLANs, where neighbors help retransmit failed sub-frames in aggregated frames. Chapter 5 studies high-throughput collision-free client polling in WLANs operating in the PCF mode. Chapter 6 sets up a smart antenna based system, to achieve reliable and efficient link-layer multicast in 802.11n WLANs. Finally, Chapter 7 concludes the dissertation.

Chapter 2

AP Association in WLANs with Heterogeneous Clients

2.1 Introduction

In this chapter, we study the problem of AP association in 802.11n WLANs with heterogeneous clients. First, we show via experiments that the performance of 802.11n transmissions could be severely affected by the association decisions of legacy 802.11a/b/g clients. We then introduce a network model for AP association, and develop a bi-dimensional Markov model to estimate client throughput in a BSS with heterogeneous clients. To describe the gain of the association, we define the MAC efficiency of a client as its achievable throughput divided by the optimal data rate it can use. We formulate the problem of AP association into an optimization problem, aiming at maximizing the MAC efficiency of all clients proportionally. To provide practical solutions, we propose an on-line AP association algorithm, named FAME, which maximizes the minimum MAC efficiency of the network when making association decisions. The algorithm is based on the bi-dimensional Markov model and requires timely knowledge of all clients in nearby BSSs. For WLANs where APs are densely deployed, we further propose another on-line algorithm with lower complexity, called Categorized AP association algorithm, which takes full advantage of 802.11n transmissions by associating different types of clients with different APs. We have conducted extensive simulations and experiments to evaluate the proposed algorithms under various network scenarios. The results demonstrate that both algorithms significantly outperform the compared schemes in terms of 802.11n throughput and network throughput.

The remainder of the chapter is organized as follows. Section 2.2 reviews the related work. Section 2.3 discusses the new challenges of AP association in 802.11n WLANs. Section 2.4 introduces the Markov model to estimate client throughput and formulates the AP association problem into an optimization problem. Section 2.5 presents two on-line algorithms. Section 2.6 evaluates the performance of the proposed algorithms via simulations. Section 2.7 further implements and validates the proposed algorithms in a WLAN testbed. Finally, Section 2.8 concludes the chapter.

2.2 Related Work

There have been a number of AP association schemes in the literature for load balance among APs. Various metrics were used in these studies to determine the AP load. In the AP association scheme proposed in [5], the sum of the reciprocal of data rates from associated clients is used to estimate the load of an AP. In [6] and [7], load balancing AP association was formulated into non-cooperative games, where the estimated packet access delay is used as the cost utility. It was proved in [6] that a Nash equilibrium can be achieved, and both centralized and localized algorithms were proposed in [7] to reach the equilibrium. In addition, the estimated file download time is adopted to indicate the AP load in the AP association scheme for web browsing in [8]. On the contrast, the effect of hidden terminals was considered in [9] as the main reason for AP performance degradation and used as the metric for AP association. Furthermore, the traffic intensity of clients was regarded as another impacting factor to AP load in the AP association scheme in [10]. However, it was pointed out [11] that greedy selection of the least-loaded AP does not guarantee optimal AP association, and the weighted sum of estimated throughput and usable data rate is used as the metric to make association decisions. In addition, it could be challenging to apply these schemes in realistic WLANs, since most of them require modifications to WLAN clients, which is not feasible for WLAN operators. Thus, in [12] a cell-breathing scheme was proposed for AP association, where APs balance their load by adjusting the transmitting power of Beacon frames, requiring no change at clients.

AP association has also been jointly considered with channel assignment and power assignment problems in WLANs. In [13], Gibbs sampler based algorithms were proposed for the joint AP association and channel assignment problem, with the objective of minimizing global interference and transmission delays. In [14], the joint problem of AP association and channel assignment was further studied from the perspective of a non-cooperative game, aiming at minimizing the aggregated packet transmission time for all clients. Moreover, an AP association algorithm for multi-channel WLANs was presented in [15], in which clients first select a best-signal AP on each channel to form a subset, then choose the AP that offers the highest throughput from this subset. Joint AP association and channel assignment in IEEE 802.11n WLANs was first studied in [40]. However, it mainly focuses on the impact of channel bonding while leaving the impact of conventional 802.11a/b/g clients untouched. AP association was also jointly explored with power assignment and bandwidth allocation in [16] and [17], respectively.

However, most of the above AP association schemes achieve throughput fairness among clients at the cost of reduced network throughput, due to aforementioned performance anomaly. Given that clients need to choose different physical data rates to adapt to various channel conditions, it is more desirable to achieve *airtime fairness* among clients, that is, all clients have generally the same medium access time regardless of their data rates. In the AP association schemes from [18, 19], it was proved that airtime fairness can be achieved by implementing proportional fairness. Nevertheless, These schemes do not specify how to implement airtime or bandwidth allocation in WLANs to realize proportional fairness. To implement proportional fairness, AP association was jointly explored with rate adaptation and contention resolution in [20]. In particular, clients choose their associated APs, adjust their physical data rates and minimum contention window sizes to reach proportional fairness. To alleviate performance anomaly and achieve airtime fairness, several approaches other than AP association have also been proposed in prior work. In the WLAN virtualization scheme presented in [21], a controller was deployed to allocate uplink airtime to clients fairly, while clients regulate their uplink traffic based on commands from the controller. In [22], airtime usage control mechanisms based on enhanced distributed channel access (EDCA) were proposed, where stations control their airtime usage by choosing appropriate arbitration interframe space (AIFS) and the contention window size. Similarly, in the time fair CSMA (TFCSMA) scheme from [23], each station chooses a target throughput based on its optimal physical data rate, and adapts its minimum contention window size dynamically according to the ratio of measured throughput and the target throughput.

2.3 Challenges of AP Association in 802.11n WLANs

In 802.11n WLANs, clients consist of 802.11n clients and conventional 802.11b and 802.11a/g clients, as the 802.11n standard is backward compatible. For convenience, we use *legacy clients* to refer to the conventional 802.11a/b/g clients in the rest of the chapter. A high throughput mixed (HT-mixed) preamble or an explicit protection mechanism, such as RTS/CTS, or CTS-to-self, needs to be used by 802.11n clients at the presence of 802.11a/g and 802.11b clients, respectively. The performance of 802.11n clients with these preambles and protections was studied in [24]. It was shown that with the RTS/CTS protection, the MAC efficiency of 802.11n is only 12%, implying that most of time the wireless medium is wasted. Furthermore, as discussed in the last section, all clients in multi-rate WLANs generally have the same throughput, which is dominated by the client with the lowest data rate. Such performance anomaly also holds in 802.11n WLANs with legacy clients [25]. A bi-dimensional Markov model for multi-rate WLANs in [26] further validates this performance anomaly theoretically.

The performance of 802.11n WLANs with heterogeneous clients highly depends on the strategy of AP association. Take the 802.11n WLAN in Fig. 2.1 as an example, where clients 1, 2 and 3 are of 802.11g, 802.11b and 802.11n types, respectively. We assume that client 1 is already associated with AP a and client 3 is associated with AP b. Based on the RSSI-based AP selection strategy, client 2 can be associated with either AP as both APs have the same

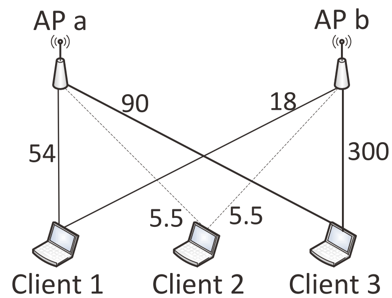


Figure 2.1: An example WLAN, where lines denote the potential associations between APs and clients, and numbers by the links are corresponding data rates.

Table 2.1: Throughput of different AP association strategies

AP for Client2	**Client3 Frame Size**	**Throughput**			
		Client1	**Client2**	**Client3**	**Aggregate**
AP *a*	1.5KB	3.9Mbps	3.0Mbps	21.0Mbps	27.9Mpbs
AP *b*	1.5KB	15.8Mpbs	3.0Mbps	5.6Mbps	24.4Mbps
AP *a*	30KB	3.9Mbps	3.0Mbps	180Mbps	186.9Mbps
AP *b*	30KB	15.8Mbps	2.8Mbps	53Mbps	71.6Mbps

data rate. On the other hand, based on the least-load AP association strategy, client 2 will associate with AP b, since the load of AP b is lower by using either the reciprocal of the data rate or average packet delay as the metric. However, much higher aggregated throughput can be achieved if client 2 is associated with AP a. Such inefficiency becomes more evident when frame aggregation is enabled. Table 2.1 lists the throughput of each client and the entire network for different AP associations of client 2. From the table, we can see that the AP association strategy in 802.11n WLANs has a significant impact on the throughput of 802.11n clients as well as the aggregated throughput.

2.4 Throughput Estimation and Problem Formulation

As shown in the previous section, network throughput in 802.11n WLANs is highly related to AP association decisions. In this section, we first present a network model for a 802.11n WLAN with heterogeneous clients, and discuss the constraints on AP association. We then analyze the time to transmit a frame by various clients. After that, we introduce a bi-dimensional Markov model to estimate the uplink and downlink throughput of each client. Finally, we formulate the AP association problem into an optimization problem, aiming at providing each client the throughput that is proportional to its usable data rate. In this way, the AP loads are balanced while the network throughput is boosted.

2.4.1 Network Model

We consider a WLAN consisting of multiple APs and a number of clients. Let set A denote the set of APs and set N denote the set of clients. Each AP has a limited coverage area and all clients are randomly distributed in the field. We assume that there are sufficient channel resources and each AP is assigned with a channel that is orthogonal with the channel of other APs in the neighborhood. For client i, we use a variable $t_i = 0, 1, 2, 3$ to denote that it is an 802.11a, 802.11b, 802.11g or 802.11n client, correspondingly. As 802.11a stations operate on 5GHz band and 802.11b/g stations operate on 2.4GHz band, we will not consider 802.11a clients in this chapter, though the network model can be applied directly to 802.11a clients. We define a variable $u_i \in (0, 1)$ to denote the probability that client i has packets to transmit to the AP, which will be referred to as *uplink traffic* in the rest of chapter. Similarly, we define a variable $d_i \in (0, 1)$ to denote the probability that the AP has packets to transmit to client i, which will be referred to as *downlink traffic*.

We use set R to denote the set of data rates supported by IEEE 802.11n standard. As aforementioned, 802.11n is backward compatible with 802.11a/b/g standards, thus the data rates of 802.11a/b/g clients are a subset of R. We assume that a client can estimate the optimal data rate for downlink trans-

Table 2.2: Notations used in formulation of AP association problem

Symbol	Semantics
A	Set of all APs
N	Set of all clients
R	Set of data rates supported by 802.11n
m	Number of backoff stages for DCF
W_0	Minimum contention window size for DCF
t_i	Type of client i
o_a	Operation mode of AP a
a_i	Associated AP for client i
N_a	Set of clients associated with AP a
N_{a_i}	Set of clients sharing the same AP with client i
$r_{i,a}$	Optimal data rate between client i and AP a
L_i	Average frame length for client i
d_i	Downlink traffic probability of client i
u_i	Uplink traffic probability of client i
q_i	Probability of station i having packets to transmit
$T_s(i)$	Frame transmission time for client i
$T_c(i)$	Collision detection time for client i
$b_{l,k}(i)$	Stationary probability of client i at state (l,k)
τ_i	Transmission probability of client i
p_i	Transmission failure probability of client i
$p_s(i)$	Successful transmission probability of client i
δ	Duration of a time slot
S_i^{up}	Estimated uplink throughput of client i
S_i^{down}	Estimated downlink throughput of client i
α_i	MAC efficiency of client i

mission from an AP by measuring the RSSI of Beacon packets from the AP. We further assume that the channel condition is symmetric thus downlink and uplink transmissions between a client and an AP have the same optimal data rate. For client i and AP a, we use $r_{i,a} \in R$ to denote the uplink and downlink data rates between them. For simplicity, we assume that channel changes slowly, and thus the optimal data rate $r_{i,a}$ remains unchanged during AP association. We also assume the uplink traffic and downlink traffic for client i have the same average frame length, and use variable L_i to denote it. The notations used in this chapter are summarized in Table 2.2.

2.4.2 Association Constraints

We now discuss the constraints that a client has to satisfy to associate with an AP. For each client i, we use a variable $a_i \in A$ to denote its associated AP. In addition, we define set N_a as the set of clients associated with AP a.

$$N_a = \{i | i \in N, a_i = a\}$$

If client i is not within the coverage area of AP a, the potential data rate $r_{i,a}$ would be zero. Thus the following condition must be met.

$$r_{i,a_i} > 0$$

Then we simply use variable $r_i = r_{i,a_i}$ to represent the data rate between client i and its associated AP a_i. As aforementioned, transmissions to and from 802.11n clients need to use HT-mixed preambles if there are 802.11a/g clients in the network. Moreover, protection mechanism is required by 802.11g/n transmissions if 802.11b clients are associated with the same AP. Hence we define an operation mode variable o_a for each AP a to indicate the necessity of HT-mixed preambles and protections, which is defined as

$$o_a = \min\{t_i | \forall i \in N_a\}$$

2.4.3 Frame Transmission Time for Various Clients

In this subsection, we analyze the time for a client to successfully transmit a frame or detect a collision, which will be used to estimate the client throughput in later subsections.

In 802.11 WLANs, all stations follow the distributed coordination function (DCF) to access the wireless medium. With DCF, a station first listens to the medium for a DCF inter frame space (DIFS) period if the station has pending traffic. If the medium is idle during DIFS, the station selects a random backoff time from the contention window to postpone its transmission. In case the channel becomes busy again during the backoff period, the station stops counting down its backoff timer and waits for the channel to be idle. Otherwise, the station begins its transmission after the backoff period. If the data frame

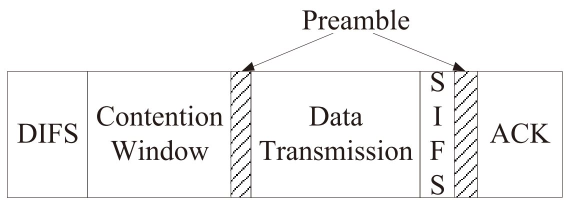


Figure 2.2: Timing of an IEEE 802.11 frame transmission.

needs to be protected from legacy clients, the station first transmits a RTS or CTS-to-self frame. After that, the station transmits the data frame. A preamble is transmitted ahead of the data payload such that the receiver can acquire the coding and modulation schemes for the payload. After successfully receiving the data frame, the receiver sends back an ACK frame, beginning with a preamble as well, after waiting for a short inter frame space (SIFS) period. The timing of an 802.11 frame transmission is shown in Fig. 2.2.

Note that for a specific client i, the frame transmission time of uplink and downlink traffic is identical, since the uplink and downlink traffic has the same data rate based on the assumption that the channel between a client and an AP is symmetrical. Thus, after obtaining the medium access opportunity, the frame transmission time to or from client i can be expressed as

$$\begin{aligned} T_s(i) = & T_{DIFS} + T_{prot}(i) + T_{pre}(i) + T_{SIFS} \\ & + T_{data}(i) + T_{SIFS} + T_{ack}(i) \end{aligned}$$

If a collision occurs, the required time for client i to detect the collision is

$$\begin{aligned} T_c(i) = & T_{DIFS} + T_{prot}(i) + T_{pre}(i) + T_{SIFS} \\ & + T_{data}(i) + T_{ack\ timeout}(i) \end{aligned}$$

In the above equations, T_{DIFS}, $T_{prot}(i)$, $T_{pre}(i)$, T_{SIFS}, $T_{data}(i)$, $T_{ack}(i)$ and $T_{ack\ timeout}(i)$ respectively stand for the DIFS duration, protection time, preamble time, SIFS period, data payload transmission time, ACK transmission time and ACK timeout time. DIFS duration and SIFS duration are constants. The payload transmission time for a client can be determined via dividing its average frame length by the data rate for the association. In addition, the protection time and preamble time for a client are specified in the 802.11n standard given the client type and the operation mode of its association AP.

2.4.4 Throughput Estimation

In this subsection, we first present a bi-dimensional Markov model to describe the DCF behavior of stations in 802.11n WLANs with heterogeneous clients. We then estimate the uplink and downlink throughput of each client, using the frame transmission time and the transmission and collision probabilities derived from the Markov model.

Fig. 2.3 shows the proposed bi-dimensional Markov model. In this model, we use $m+1$ backoff stages to describe the exponential backoff behavior of DCF for transmissions and retransmissions, where m is a constant value specified by the 802.11 standard. At each back off stage $l, 0 \leq l \leq m$, the station randomly chooses a back off value from the contention window size, W_l, which is given as follows

$$W_l = \begin{cases} W_0, \ l = 0; \\ 2^l W_0, \ 1 \leq l \leq m \end{cases}$$

where W_0 is the minimum contention window size at stage 0. Thus, a station is at state (l, k) in the Markov model if it is in backoff stage l, $0 \leq l \leq m$, and its contention window size is $k, 0 \leq k < W_l$. Moreover, a station is at state $(-1, 0)$ if it has no packet to transmit when the channel is free.

We also use a variable q_i to denote the probability that station i has packets to transmit. Thus for any client $i \in N$, q_i equals its uplink traffic probability u_i. Meanwhile, for an AP $a \in A$, q_a is the summed downlink traffic probability of all of its associated clients. q_a is rounded to 1 if the summation is greater than 1.

Initially, station i is at the idle state $(-1, 0)$. If the station has no packet to transmit in the next time slot, it transits back to state $(-1, 0)$. The transition probability is $(1 - q_i)$, as q_i is the probability that station i has packets to transmit. Otherwise, station i transits to a random state $(0, k)$ at backoff stage 0. The transition probability is q_i/W_0 as the traffic probability of station i is q_i while the contention window size is randomly selected from $[0, W_0 - 1]$. If

$k \neq 0$, station i transits from state $(0, k)$ to state $(0, k-1)$ with probability 1 in the next time slot. Otherwise, station i transmits a packet in the next time slot. If the transmission fails, the station transits to a random state $(1, k)$ in the next time slot with probability p_i/W_1, where p_i is the transmission failure probability of station i. If the transmission is successful, the station transits back to state $(-1, 0)$. The transition probability is thus $(1 - p_i)$. In general, a station transits from state (l, k) to state $(l, k-1)$ in the next time slot if $k > 0$. When $k = 0$, the station tries to transmit a frame. It transits back to state $(-1, 0)$ if the transmission is successful; otherwise, it transits to a random state at backoff stage $l + 1$.

Let $s(t)$ and $c(t)$ be the stochastic processes of backoff stage and backoff counter for station i, then its stationary probability at state (l, k) would be

$$b_{l,k}(i) = \lim_{t \to \infty} P\{s(t) = l, c(t) = k\}, 0 \leq l \leq m, 0 \leq k < W_l$$

Similarly, the stationary probability at the idle state can be expressed as

$$b_{-1,0}(i) = \lim_{t \to \infty} P\{s(t) = -1, c(t) = 0\}$$

For any two states, (l, k) and (l', k'), we use condition probability $P\{l', k'|l, k\}$ to denote the one-step transition probability from state (l, k) to state (l', k'). Thus, the one-step transition probability for station i at state (l, k) is

$$\begin{cases} P\{-1, 0| -1, 0\} = 1 - q_i & \\ P\{0, k| -1, 0\} = q_i/W_0 & 0 \leq k < W_0 \\ P\{-1, 0|l, 0\} = 1 - p_i & 0 \leq l \leq m \\ P\{l+1, k|l, 0\} = p_i/W_{l+1} & 0 \leq l < m, 0 \leq k < W_{l+1} \\ P\{l, k-1|l, k\} = 1 & 0 \leq l \leq m, 1 \leq k < W_l \\ P\{m, k|m, 0\} = p_i/W_m & 0 \leq k < W_m \end{cases} \tag{2.1}$$

By the chain regularities of the Markov model, the stationary probability

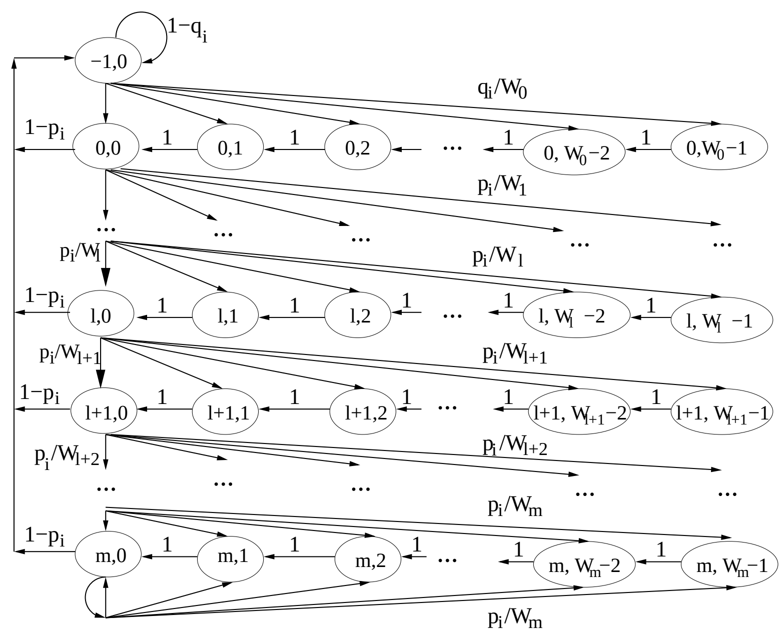


Figure 2.3: Bi-dimensional Markov chain model for client i with saturated traffic, where W_l is contention window size at state l and p_i is the transmission failure probability of client i.

of station i at state (l, k) is

$$b_{l,k}(i) = \frac{W_l - k}{W_l} \begin{cases} q_i \cdot b_{-1,0}(i) & l = 0 \\ p_i \cdot b_{l-1,0}(i) & 0 < l < m \\ p_i \cdot [b_{m-1,0}(i) + b_{m,0}(i)] & l = m \end{cases} \tag{2.2}$$

By (2.1) and (2.2), $b_{l,k}$ can be expressed as a function of $b_{-1,0}$, transmission failure probability p_i and traffic probability q_i. On the other hand, according to the normalization condition for a stationary Markov chain, the summation of stationary probabilities for all states should be equal to 1, which can be formally expressed as

$$b_{-1,0}(i) + \sum_{l=0}^{m} \sum_{k=0}^{W_l} b_{i,k}(i) = 1$$

where $b_{-1,0}(i)$ can be derived from

$$b_{-1,0}(i) = \frac{2(1-p_i)(1-2p_i)}{2(1-2p_i)(1-p_i+q_i) + q_i W_0((1-p_i)+(2p_i)^m(1-2p_i))}$$

Furthermore, we use variable τ_i to denote the transmission probability of station i. Then τ_i is equal to the summed probabilities that the counter equals zero at all backoff stages, which can be formally expressed by

$$\tau_i = \sum_{l=0}^{m} b_{l,0}(i) = \frac{2q_i(1-2p_i)}{2(1-2p_i)(1-p_i+q_i) + q_i W_0((1-p_i)+(2p_i)^m(1-2p_i))}$$

For any client $i \in N$, its probability of transmitting a frame without colliding with its associated AP and other clients associated with the same AP is equal to the probability that client i transmits while all other clients and the AP are not transmitting. We define set N_{a_i} to be the set of clients associated with the same AP of client i. Then the successful transmission probability of client i is

$$p_s(i) = \tau_i \cdot (1-\tau_{a_i}) \cdot \prod_{j \in N_{a_i}} (1-\tau_j)$$

For any AP $a \in A$, its successful transmission probability is equal to the probability that AP a transmits while none of its associated clients are transmitting, which can be expressed as follows

$$p_s(a) = \tau_a \cdot \prod_{i \in N_a} (1-\tau_i)$$

A transmission failure can be caused by either a collision or a channel error. Note that transmission failures caused by channel errors can be neglected in our model, as we have assumed that each station selects the optimal data rate based on the channel condition. In other words, we assume that all transmission failures are caused by collisions. Then for any client $i \in N$, its transmission failure probability p_i is equal to the probability that its associated AP or at least one of other clients associated with the same AP is transmitting, that is,

$$p_i = \tau_i \cdot [1 - (1 - \tau_{a_i}) \cdot \prod_{j \in N_{a_i}} (1 - \tau_j)]$$

For any AP $a \in A$, its transmission failure probability p_a is equal to the probability that at least one of its associated clients transmits while it is transmitting, which can be expressed as

$$p_a = \tau_a \cdot [1 - \prod_{i \in N_a} (1 - \tau_i)]$$

The theoretical throughput for client i can then be expressed as the length of successfully transmitted payload divided by the average duration of a time slot $T_{avg}(a_i)$ for all stations in the same BSS, that is,

$$S_i = \frac{p_s(i) \cdot L_i}{T_{avg}(a_i)}$$

Similarly, the throughput for AP a can be expressed as

$$S_a = \frac{p_s(a) \cdot L_a}{T_{avg}(a)}$$

Note that for AP a, it needs to transmit frames to all associated clients that have downlink traffic. We assume that the AP determines the receiving client of a transmission proportional to the downlink traffic probability of all clients. Then the expected frame length of AP a can be given by

$$L_a = \frac{\sum_{i \in N_a} d_i \cdot L_i}{\sum_{i \in N_a} d_i}$$

The average time slot $T_{avg}(a)$ for AP a and all its associated clients can be further expressed as the summation of three expected slot durations

$$T_{avg}(a) = T_I(a) + T_S(a) + T_c(a)$$

where $T_I(a)$, $T_S(a)$ and $T_C(a)$ stand for the expected durations of an idle time slot, a successful frame transmission and a transmission failure due to collisions, respectively, for AP a and its associated clients.

The probability that AP a and its associated clients are not transmitting can be represented as

$$P_I(a) = (1 - \tau_a) \cdot \prod_{i \in N_a} (1 - \tau_i)$$

Thus the average duration of an idle time slot in the BSS where AP a resides is

$$T_I(a) = P_I(a) \cdot \delta$$

where δ is the duration of a DCF backoff time slot.

The expected duration of a successful downlink transmission for AP a can be given by

$$T_s^{down}(a) = \frac{\sum_{i \in N_a} d_i \cdot T_s(i)}{\sum_{i \in N_a} d_i}$$

Then the expected duration of a successful transmission for AP a and its associated clients is

$$T_S(a) = p_s(a) \cdot T_s^{down}(a) + \sum_{i \in N_a} p_s(i) \cdot T_s(i)$$

To determine the expected collision duration of an AP and its associated clients, we first sort them according to their collision duration T_c. For an AP a, its collision duration for downlink traffic depends on the data rate and frame length of the destination of a transmission. For simplicity, we will use the expected collision duration of all clients as the downlink collision duration, that is

$$T_c^{down}(a) = \frac{\sum_{i \in N_a} d_i \cdot T_c(i)}{\sum_{i \in N_a} d_i}$$

Then assume that station i only collides with other stations in the same BSS that have a shorter collision duration. In other words, a collision between any two stations $i, j, (T_c(i) < T_c(j)$ will be counted by station j only, rather than both of them, when we calculate the expected collision duration of their BSSs. Then the collision probability of a station i can be regarded as the probability that station i transmits and at least another station with a shorter collision duration transmits simultaneously, while all stations with a longer collision duration are not transmitting. Then if the collision duration $T_c(i)$ of client i is less than the collision duration $T_c(a_i)$ of its associated AP, its collision probability can be rewritten as

$$p'_i = \tau_i \cdot \left[1 - (1-\tau_{a_i}) \cdot \prod_{j \in N_{a_i}}^{T_c(j) \le T_c(i)} (1-\tau_j)\right] \cdot \prod_{j \in N_{a_i}}^{T_c(j) > T_c(i)} (1-\tau_j)$$

If the collision duration $T_c(i)$ of client i is greater than the collision duration $T_c(a_i)$ of its associated AP, its collision probability is

$$p'_i = \tau_i \cdot (1-\tau_{a_i}) \cdot \left[1 - \prod_{j \in N_{a_i}}^{T_c(j) \le T_c(i)} (1-\tau_j)\right] \cdot \prod_{j \in N_{a_i}}^{T_c(j) > T_c(i)} (1-\tau_j)$$

Similarly, the collision probability of AP a can be rewritten as

$$p'_a = \tau_a \cdot \left[1 - \prod_{i \in N_a}^{T_c(i) \le T_c(a)} (1-\tau_i)\right] \cdot \prod_{i \in N_a}^{T_c(i) > T_c(a)} (1-\tau_i)$$

Then the expected collision duration for AP a and its associated clients is

$$T_C(a) = p'_a \cdot T_C(a) + \sum_{i \in N_a} p'_i \cdot T_c(i)$$

Finally, based on above equations, the estimated uplink throughput for client i can be represented as

$$S_i^{up} = \frac{p_s(i) \cdot L_i}{T_I(a_i) + T_S(a_i) + T_C(a_i)} \tag{2.3}$$

For an AP a, the estimated downlink throughput for all of its associated clients can be given by

$$S_a^{down} = \frac{p_s(a) \cdot \sum_{i \in N_a} (d_i \cdot L_i)}{\sum_{i \in N_a} d_i \cdot [T_I(a) + T_S(a) + T_C(a)]}$$

Accordingly, the estimated downlink throughput for client i can be expressed as

$$S_i^{down} = \frac{d_i}{S_{a_i}^{down} \cdot \sum_{j \in N_{a_i}} d_j} \tag{2.4}$$

2.4.5 Formulation of AP Association Problem

As analyzed in the previous subsection, the downlink and uplink throughput of a client is related to not only its own traffic load and data rate, but also the client type, traffic load and data rates of all other clients in the same BSS. The composition of a BSS is eventually determined by the AP association strategy. Our goal of AP association is to provide each client the throughput proportional to its data rate. In this way, the throughput of a 802.11n client is determined by its signal quality to its associated AP, rather than the throughput of legacy clients in the same BSS. Then both 802.11n throughput and overall throughput can be boosted. In this subsection, we first define a MAC efficiency for each client as the metric to evaluate an AP association decision. After that, we formulate the AP association problem into an optimization problem.

The MAC efficiency of a client should reflect both the uplink throughput and downlink throughput of the client. In addition, the MAC efficiency should reflect the traffic load of a client as well, since it is inaccurate to say a client has poor MAC efficiency simply because it has no data to transmit and thus has throughput close to zero. Moreover, the data rate a client may use should be considered in the MAC efficiency as well, because it determines the highest achievable throughput of a client. Therefore, we define the MAC efficiency for client i as

$$\alpha_i = \frac{S_i^{up} + S_i^{down}}{\min\{1, u_i + d_i\} \cdot r_{i,a_i}} \tag{2.5}$$

Given an AP association assignment, the clients in each BSS of the network are determined. Then the uplink and downlink throughput of each client can be estimated using the equations in the previous subsection. Accordingly, the MAC efficiency of each client can be determined. Then the AP association problem becomes the problem of finding an association assignment among all association possibilities, so that each client receives satisfactory MAC efficiency. As the utility function of summed logarithmic of the MAC efficiency of all clients can guarantee proportional fairness among all clients, we will use it as the objective function in the optimization.

We can now formulate the AP association problem into an optimization problem. Given a WLAN consisting of multiple 802.11n APs and a number of heterogeneous clients, find an AP association assignment, such that the MAC efficiency of all clients is proportionally maximized, while all association constraints are satisfied. The MAC efficiency of each client is determined using the estimated throughput from the previous subsection. The optimization problem can be described as follows

Maximize

$$\sum_{\forall i \in N} \log \alpha_i$$

Subject to

$$a_i \in A, \forall i \in N \tag{2.6}$$

$$\tau_i = \frac{2q_i}{2(1-p_i+q_i) + q_i W_0 (2p_i)^m + \dfrac{q_i W_0 (1-p_i)}{1-2p_i}} \tag{2.7}$$

$$T_I(a) = \delta \cdot (1-\tau_a) \cdot \prod_{i \in N_a} (1-\tau_i) \tag{2.8}$$

$$T_S(a) = p_s(a) \cdot T_s^{down}(a) + \sum_{i \in N_a} p_s(i) \cdot T_s(i) \tag{2.9}$$

$$T_C(a) = p'_a \cdot T_C(a) + \sum_{i \in N_a} p'_i \cdot T_c(i) \tag{2.10}$$

$$S_i^{up} = \frac{p_s(i) \cdot L_i}{T_I(a_i) + T_S(a_i) + T_C(a_i)} \tag{2.11}$$

$$S_a^{down} = \frac{p_s(a) \cdot \sum_{i \in N_a} (d_i \cdot L_i)}{\sum_{i \in N_a} d_i \cdot (T_I(a) + T_S(a) + T_C(a))} \tag{2.12}$$

$$S_i^{down} = \frac{d_i}{S_{a_i}^{down} \cdot \sum_{j \in N_{a_i}} d_j} \tag{2.13}$$

$$\alpha_i = \frac{S_i^{up} + S_i^{down}}{\min\{1, u_i + d_i\} \cdot r_{i,a_i}} \tag{2.14}$$

In the above formulation, constraint (2.7) specifies the transmission probability for each station. Constraints (2.8), (2.9) and (2.10) determine the expected duration of an idle slot, successful transmission, failed transmission due to collision, respectively, for all clients associated with AP a. The estimated uplink throughput, downlink throughput and MAC efficiency for each client are given in constraints (2.11), (2.13) and (2.14).

Note that in the above formulation, there is an integrity constraint on x_i variables. Also, the problem is non-linear and non-concave. Thus, the complexity of this optimization problem grows exponentially as the network size increases. Since it is difficult to solve the optimization problem directly, in the next section, we will propose an on-line AP association algorithm based on the client throughput and MAC efficiency estimated in this section. In addition, for WLANs where APs are densely deployed, we will propose another algorithm with lower complexity than the first algorithm.

2.5 AP Association Algorithms

In this section, we propose two on-line AP association algorithms for 802.11n WLANs with heterogeneous clients. The first algorithm is called *FAir Mac Efficiency (FAME)* algorithm, in which each client achieves the throughput proportional to its usable data rate, by associating with the AP that maximizes the minimum MAC efficiency of all associated clients. The second algorithm is referred to as *Categorized* algorithm, in which the potential of 802.11n transmissions is fully exploited by associating different types of clients with different APs. FAME algorithm has no preference on the network structure while Categorized algorithm performs the best in WLANs where APs are densely deployed. The algorithms are described in detail in following subsections.

2.5.1 FAME AP Association Algorithm

We first present FAME algorithm, with the primary objective to maximizing the minimum MAC efficiency of all clients. The MAC efficiency of a client can also be regarded as the fraction of medium time used by the client, since it has been defined as the ratio of achievable throughput to the usable data rate of the client. Thus, by maximizing the minimum MAC efficiency during

AP association, all clients would have generally the same airtime. On the other hand, some low-rate legacy clients may do not have sufficient bandwidth to fulfill application demands, if their airtime is the same as other clients with much higher data rates. The AP association strategy to maximize the minimum throughput of all clients is more desirable in such scenarios. To take advantage of both strategies, a weight is assigned to the MAC efficiency in FAME and each client associates with the AP that maximizes the weighted MAC efficiency of all clients. For client i, its weighted MAC efficiency α_i^w can be given by

$$\begin{aligned}\alpha_i^w &= (1 + w(r_{i,a_i} - 1)) \cdot \alpha_i \\ &= (1 + w(r_{i,a_i} - 1)) \cdot \frac{S_i^{up} + S_i^{down}}{\min\{1, u_i + d_i\} \cdot r_{i,a_i}}\end{aligned} \tag{2.15}$$

where w is a balancing factor ranging from 0 to 1. When w is 0, the weighted MAC efficiency is the MAC efficiency itself and then FAME algorithm provides airtime fairness. When w is 1 and a client has saturated traffic, its weighted MAC efficiency equals the client throughput and then FAME algorithm provides throughput fairness. w can also take an intermediate value between 0 and 1 to reach a balance between airtime fairness and throughput fairness. w is set to 0 be default as the primary goal of FAME is that each client achieves the throughput proportional to its physical data rate.

In FAME algorithm, each AP broadcasts a MAC efficiency entry for every associated client in the *Beacon* frame, including the client type, current data rate, average frame length, traffic probability, etc. In addition, a client estimates the potential data rate for each associable AP by measuring the RSSI of Beacon frames from all nearby APs. The client then determines the minimum weighted MAC efficiency of each nearby AP, by taking the MAC efficiency entries into the Markov model presented in the previous section. After that, the client associates with the AP that maximizes the minimum weighted MAC efficiency. The corresponding AP then updates its entries of MAC efficiency accordingly to include the new client. In the worst scenario, a client is in the coverage area of all APs in set A and an AP has at most $|N|$ associated clients. Then the time complexity of FAME algorithm is $O(|A| \cdot |N|)$, assuming that the time to estimate the minimum MAC efficiency of an AP is proportional to the number of associated clients.

FAME algorithm can be implemented in WLANs by extending the Beacon and *Associate Request* frames defined in the 802.11 standard. As discussed earlier, each AP maintains the MAC efficiency entries for all associated clients and broadcasts them in the Beacon frames. In particular, an AP acquires the uplink data rate, downlink data rate, and the average length of aggregated frames of a client from recently received and transmitted frames. An AP estimates the downlink traffic probability of a client from the number of pending frames to that client, while the client estimates the uplink traffic probability from the number of pending frames to the AP. After receiving the Beacon frames of all nearby APs, A client associates with the AP that maximizes the minimum weighted MAC efficiency by sending an extended Associate Request frame, including its own uplink traffic probability. To cope with client mobility, every client scans other channels to receive Beacon frames from nearby APs every a few seconds, so as to acquire updated MAC efficiency entries. A client then runs the FAME algorithm to determine whether the minimum MAC efficiency can be improved by associating with a different AP. If so, the client measures the received signal strength of a few more Beacon frames from the new AP. The client associates with the new AP only if the signal strength becomes stronger or at least remains the same, so as to avoid unnecessary associations when moving away from the new AP. The pseudo code of FAME algorithm is given in Table 2.3.

2.5.2 Categorized AP Association Algorithm

As discussed in the previous subsection, FAME algorithm requires the data rate and traffic load information of all clients in nearby BSSs to make association decisions. Thus each client should re-execute FAME algorithm whenever the channel condition or traffic load of other clients changes, to ensure it is associated with the best AP. This may affect ongoing applications and place extra computing overhead on clients. In this subsection, we present Categorized algorithm, which is much less sensitive to network variation.

Categorized algorithm minimizes the impact of legacy clients on 802.11n

Table 2.3: Fair MAC efficiency AP association algorithm

Input:
Set of APs A
Set of clients N
Client Type Vector $T = \{t_i | \forall i \in N\}$
Weight factor w
Output:
AP Association Matrix $X = \{x_{i,a} | \ \forall i \in N, a \in A\}$
Algorithm:
for each client $i \in N$
  for each AP $a \in A$
    if client i receives Beacon frame from a
      Add AP a into subset A_i;
      Determines $r_{i,a}$ between client i and AP a;
    end if
  end for
  Determine probability of uplink traffic u_i;
  for each AP $a' \in A_i$
    Send a Probe Request to AP a';
    Receive a Probe Response from AP a';
    Let $N_{a'}$ be the set of clients associated with a';
    Determine MAC efficiency of all clients in $N_{a'} \bigcup i$;
    $\alpha'_a = \min\{\alpha_j | j \in N_{a'} \bigcup i\}$;
    Weighted MAC efficiency $\alpha_a^{w\prime} = (1 + w(r_{i,a'} - 1)) \cdot \alpha'_a$;
  end for
  $c = \arg\max_{a'} \{\alpha_a^{w\prime} | a' \in A_i\}$;
  Associate client i with AP c;
  $x_{i,c} = 1$;
  Update MAC efficiency entries for AP c;
end for

transmissions by associating different types of clients with different APs. APs are categorized by the type of their preferred clients. The rationale behind this is that in dense WLAN deployments, every client is within the coverage of multiple APs most of time. Thus a client is able to use satisfactory data rates by associating with a categorized AP, which does not necessarily have the best signal quality. In this way, performance degradation of 802.11n transmissions resulted from coexistence with legacy clients in the same BSS can be greatly alleviated.

Initially, all APs are not assigned to any category. When a new client i joins the network, it first checks whether it is within the coverage area of one or more APs that prefer the type of client i. If more than one APs qualify, client i associates with the AP that has the best signal quality. If there is no such an AP, client i examines whether it is covered by APs that have not been categorized yet. If so, client i associates with the AP resulting in the maximum data rate for the association and the AP updates its category to the type of client i. Clearly, it cannot be guaranteed that each client is able to associate with an AP with matched preference, especially when APs are deployed sparsely. In such a case, client i associates with the AP whose minimum data rate is closest to the data rate of client i if associated. The reason is that if the preamble and protection overhead caused by legacy clients is inevitable, we want to minimize the data rate difference among all clients associated with the same AP. In the worst case, a client is covered by all APs in set A. For each AP, the client takes constant time to determine whether the AP is the best choice, given the category and the minimum data rate among all associated clients of the AP. Therefore, the complexity of Categorized algorithm is $O(|A|)$.

Similar to FAME algorithm, Categorized algorithm can be implemented on deployed WLANs by extending the *Beacon* frame of 802.11. In the Beacon frame, an additional category field and a new data rate field are appended. The category field indicates the preferred client type of an AP, while the new data rate field specifies the lowest data rate used by all associated clients of the AP, for both uplink and downlink traffics. After receiving Beacon frames from all associable APs, a client can make association decisions based on simple comparisons. Each client also scans other channels every a few seconds to receive Beacon messages from nearby APs, so as to cope with client mobility. The client then executes the Categorized algorithm to determine whether there exists a better AP to associate with. Similar to FAME algorithm, the client measures the signal strength of a few more Beacon frames from the new AP, and associates with the new AP only if the signal strength becomes stronger or remains the same. An AP resets its preference to uncategorized if all of its associated clients disassociate from it. The pseudo code of the algorithm is given in Table 2.4.

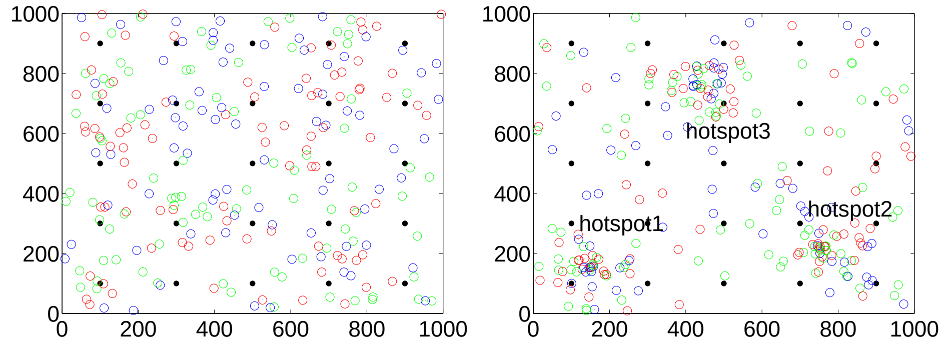
2.6 Simulation Results

In this section, we evaluate the performance of the proposed AP association algorithms via simulations, and compare them with the RSSI-based and the max-min throughput association schemes. We first evaluate the performance of the proposed algorithms in terms of network throughput, MAC access delay and fairness among clients in WLANs where all clients are stationary. After that, we examine their performance in WLANs where there exist mobile clients.

In the simulation, 25 APs are deployed in a $1000 \times 1000m^2$ field, and the APs are evenly placed on a 5×5 grid to provision fully coverage to the field. As co-channel interference among neighboring BSSs is not considered in this chapter, we assign an orthogonal 40MHz channel to each AP to eliminate potential interference. An AP uses one part of its assigned 40MHz channel for transmissions to/from legacy clients. A number of clients are deployed in the field and the type of each client is randomly selected among 802.11b, 802.11g and 802.11n. The collision-aware rate adaptation (CARA) algorithm [28] is employed by for rate adaptation, as it differentiates transmission failures resulted from channel variations from transmission failures caused by collisions, and adapts data rate only for channel variations. For 802.11b and 802.11g transmissions, all data rates specified in the IEEE 802.11b and 802.11g standards are used in CARA. For 802.11n transmissions, only single spatial-stream, short guard interval (SGI) data rates at 40MHz are used, since the CARA algorithm does not support adaptation between single spatial-stream and double spatial-stream data rates. However, it should be pointed out that other rate

Table 2.4: Categorized AP association algorithm

Input:
Set of APs A
Set of clients N
Client Type Vector $T = \{t_i|\ \forall i \in N\}$
Output:
AP Association Matrix $X = \{x_{i,a}|\ \forall i \in N, a \in A\}$
Algorithm:
for each AP $a \in A$
Set its category c_a to zero;
end for
for each client $i \in N$
for each AP $a \in A$
if client i receives Beacon frames from AP a
Get the category c_a of AP a;
Get the lowest data rate r_a^{min} among clients of AP a;
Add AP a into subset A_i;
end if
end for
if $\exists A_i' \subset A_i$ such that $A_i' = \{a \in A_i|c_a = t_i\}$
$b = \arg\max_a \{r_{i,a}|a \in A_i'\}$;
Associate client i with AP b;
else if $\exists A_i'' \subset A_i$ such that $A_i'' = \{a \in A_i|c_a = 0\}$
$b = \arg\max_a \{r_{i,a}|a \in A_i''\}$;
Associate client i with AP b;
Set the category c_b of AP b to t_i;
else
$b = \arg\min_a \{r_{i,a} - r_a^{min}|a \in A_i\}$;
Associate client i with AP b;
end if
$x_{i,b} = 1$;
end for



(a) Uniform distribution. (b) Hotspot distribution.

Figure 2.4: An example of client distribution in the simulation. The black dots denote APs, and the colored dots denote clients. In particular, red, green and blue dots represent 802.11b, 802.11g and 802.11n clients, respectively.

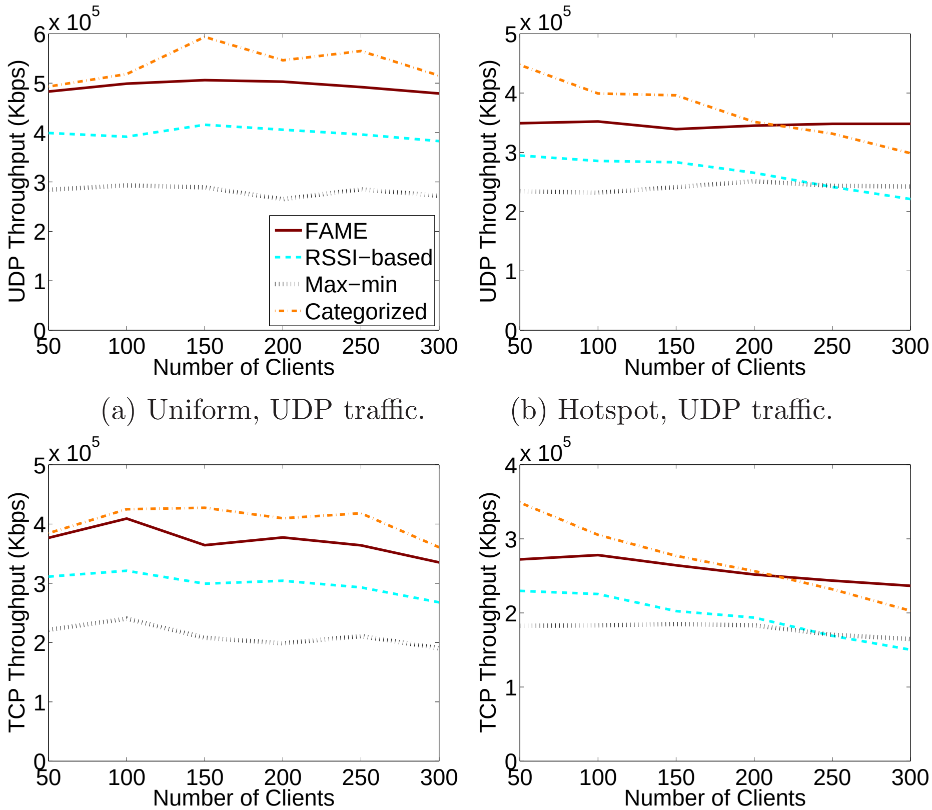
adaptation algorithms for WLANs could also be used in the simulation, as the proposed AP association algorithms do not rely on a specific rate adaptation algorithm. TCP and UDP traffics have been adopted separately in the simulation to emulate various applications. Every client always has saturated traffic in both uplink and downlink directions. For each network configuration, the simulation is run 300 seconds.

2.6.1 Performance in WLANs with Stationary Clients

In this subsection, we evaluate the network performance of the proposed algorithms in WLANs where all clients are static. As shown in Fig. 2.4, we consider two types of client distributions. (1) Uniform: all clients are randomly distributed over the entire field; (2) Hotspot: 75% clients are placed in three circle-shaped hotspot areas, while other clients are randomly distributed in the rest of the field.

We first study the network throughput of FAME and Categorized algorithms under various client densities. The UDP throughput under uniform and hotspot client distributions is plotted in Fig. 2.5(a) and Fig. 2.5(b), respectively, where the number of clients increases from 50 to 300 in a step of 50. We can see that both FAME and Categorized algorithms can achieve much higher UDP throughput than the compared schemes under both types of client distribution, regardless of the client density. When 200 clients are uniformly distributed in the field, the UDP throughput of FAME and Categorized algorithm is 25% and 36% higher than that of RSSI-based algorithm, and 92% and 107% higher than that of max-min algorithm, respectively. This can be attribute to the fact that the impact of legacy clients, especially the protection overhead from RTS/CTS transmissions, is alleviated in the proposed algorithms. We can also observe that the Categorized algorithm leads to the highest UDP throughput under uniform client distribution, as most 802.11n transmissions are isolated from 802.11b/g transmissions while no AP is overloaded. However, under hotspot distribution, the UDP throughput of Categorized algorithm drops below that of FAME algorithm when the number of clients grows beyond 200. The UDP throughput of RSSI-based algorithm also decreases as the client density increases. This is because APs in the hotspots may become overloaded when the client density is high, as the AP load is not considered in Categorized and RSSI-based algorithms when making association decisions. The TCP throughput under uniform and hotspot client distributions is plotted in Fig. 2.5(c) and Fig. 2.5(d). The TCP throughput in both cases is lower than the UDP throughput, as the transmission of TCP ACK consumes medium access time and intensifies collisions. Nevertheless, both proposed algorithms achieve much higher TCP throughput than the compared schemes for similar reasons.

We then examine the average *MAC access delay* of the proposed algorithms, so as to assess the intensity of medium access contention, the transmission overhead, and the applied data rates of different AP association algorithms. Here MAC access delay is defined as the duration from the time that a packet enters the queue at the MAC layer to the time that the packet is transmitted. The average MAC access delay for UDP traffic in uniform client distribution and hotspot client distribution is given in Fig. 2.6 (a) and Fig. 2.6 (b). It can be noted that the MAC access delay of all AP association algorithms increases along with the number of clients, because the contention intensity within each BSS is high when there are many clients in it. In the uniform distribution case, RSSI-based algorithm has the shortest MAC access delay, since every client is associated with the AP that has the best signal quality and thus uses a high data rate, while the traffic load is in general evenly distributed among all APs. The MAC access delay of FAME and Categorized algorithms is very



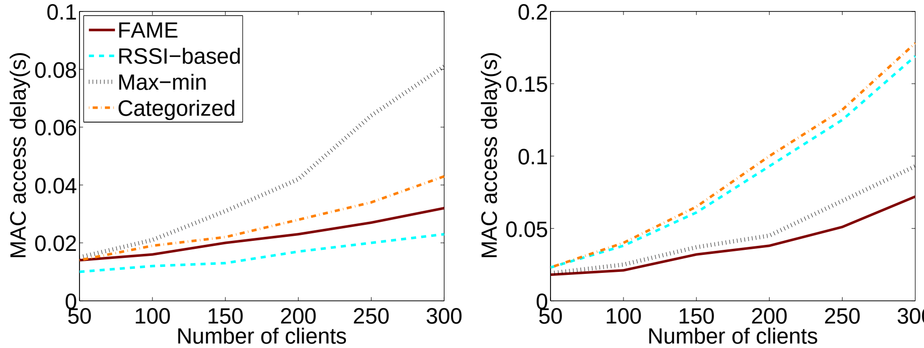
(a) Uniform, UDP traffic.

(b) Hotspot, UDP traffic.

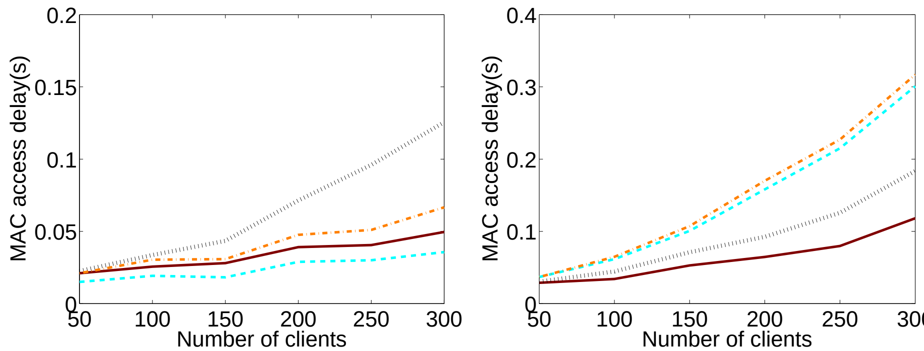
(c) Uniform, TCP traffic.

(d) Hotspot, TCP traffic.

Figure 2.5: Network throughput under uniform and hotspot client distributions with respect to various client densities.



(a) Uniform, UDP traffic. (b) Hotspot, UDP traffic.

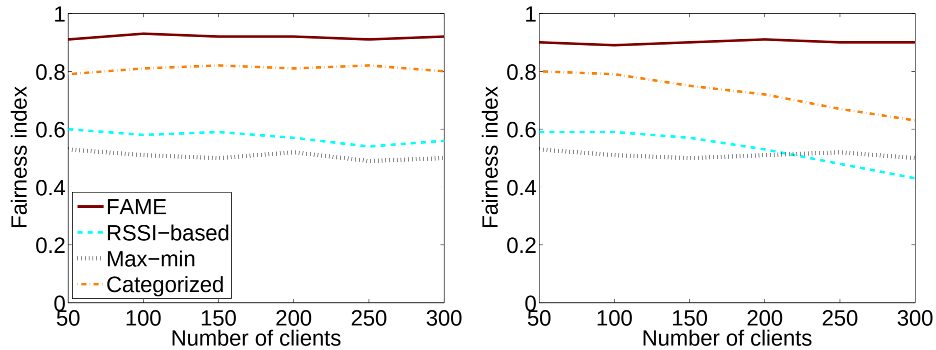


(c) Uniform, TCP traffic. (d) Hotspot, TCP traffic.

Figure 2.6: Average MAC access delay under uniform and hotspot client distributions with respect to various client densities.

close to that of RSSI-based algorithm and lower that of max-min algorithm, indicating the benefits of avoiding protection overhead when applicable. On the contrast, under hotspot distribution, FAME has the shortest MAC access delay while Categorized has the longest MAC access delay. Moreover, max-min has similar MAC access delay to FAME, while RSSI-based algorithm has similar MAC access delay to Categorized. The reason behind this is that only FAME and max-min algorithms are capable of associating clients in hotspots to APs out of the hotspots. Similar results can be observed for TCP traffic in Fig. 2.6 (c) and Fig. 2.6 (d), except that the MAC access delay of all algorithms is longer because each station need to transmit both TCP data and TCP ACK packets.

Next, we evaluate the fairness of the proposed AP association algorithms. As aforementioned, the primary objective of our proposed algorithms is that



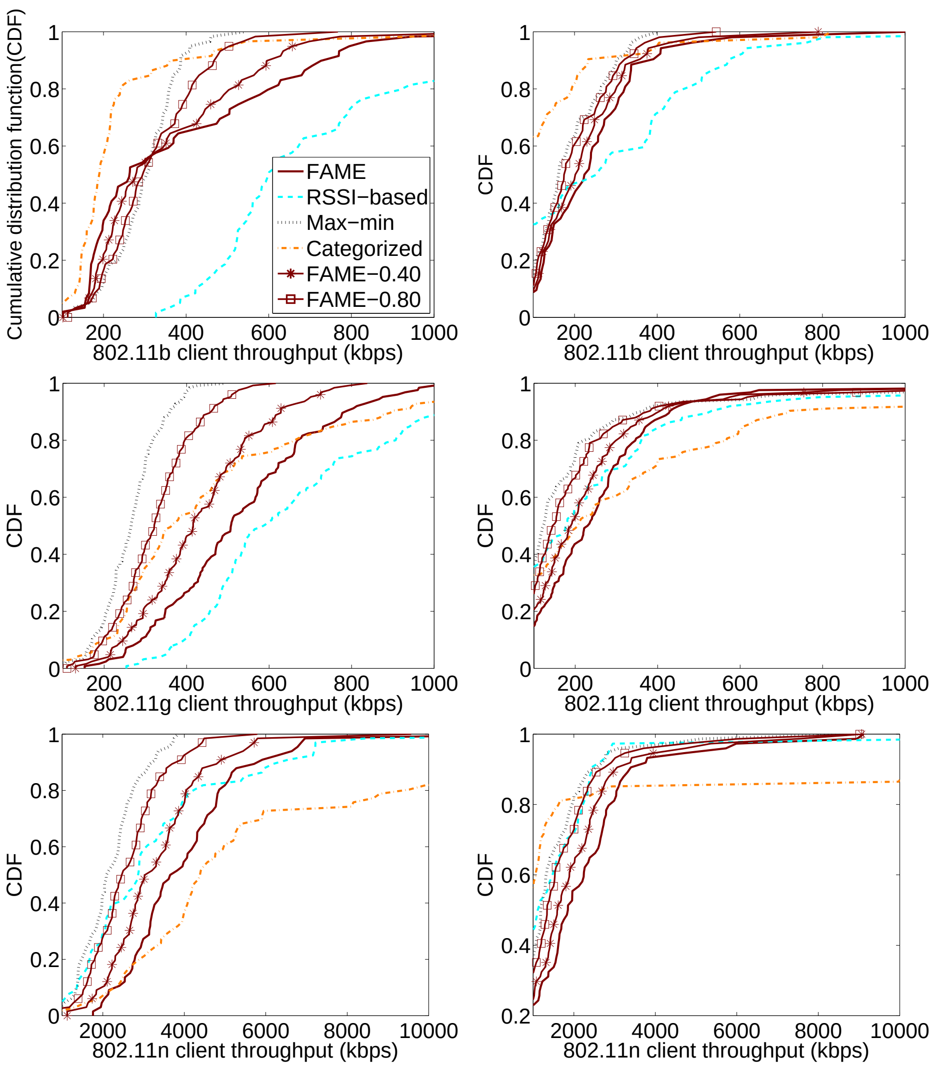
(a) Uniform distribution. (b) Hotspot distribution.

Figure 2.7: Fairness index of MAC efficiency under various client densities.

each client achieves the throughput that is proportional to its physical data rate. In other words, we want to achieve fairness among the MAC efficiency of all clients. We adopt the fairness index from [27] to quantify the fairness of MAC efficiency. The fairness index $F(\alpha)$ of MAC efficiency is defined as

$$F(\alpha) = \frac{(\sum_{i \in N} \alpha_i)^2}{|N| \sum_{i \in N} \alpha_i^2} \tag{2.16}$$

where N is the set of clients and α_i is the MAC efficiency of client i (Equation (2.5)). $F(\alpha)$ takes a value from 0 to 1, and reaches 1 is all clients have the same MAC efficiency. The fairness index of MAC efficiency for UDP traffic is plotted in Fig. 2.7, where the number of clients varies from 50 to 300. We can observe that the fairness index of FAME algorithm is always above 0.9 under both uniform and hotspot distributions regardless of client density, indicating that 90% of clients achieve a fair MAC efficiency. The fairness of Categorized algorithm is also above 0.8 under uniform distribution, validating the effectiveness of isolating 802.11b, 802.11g and 802.11n transmissions from each other. The fairness index of RSSI-based and max-min algorithm is below 0.6. This is mainly because that 802.11g transmissions are severely suppressed by low-rate 802.11b transmissions and aggregated-frame 802.11n transmissions. We can also see that under hotspot distribution, the fairness index of Categorized algorithm degrades as the number of client increases. The reason is that the MAC efficiency of clients in the hotspots becomes very low when the client density

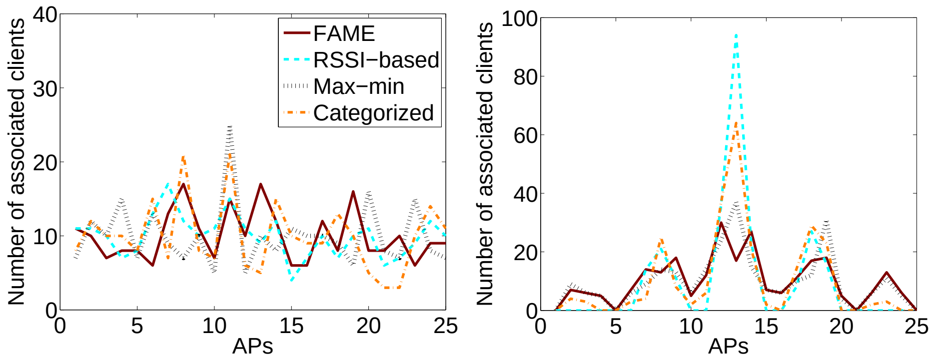


(a) Uniform distribution. (b) Hotspot distribution.

Figure 2.8: Cumulative distribution function (CDF) for throughput of 802.11b, 802.11g and 802.11n clients in an example WLAN with 200 clients.

is high, as the associated APs are overloaded. While the MAC efficiency of clients in the rest area of the field is much higher, since the associated APs are underutilized. The fairness index of RSSI-based algorithm is worse when the client density is high for the same reason. We have also evaluated the fairness index of the proposed algorithms for TCP traffic. In Fig. 2.7, Fig. 2.8 and Fig. 2.9, the results for TCP traffic is very similar to the results of UDP traffic and thus are not presented due to space limitation.

We also assess the fairness of the proposed algorithms and verify the effectiveness of the balancing factor in FAME algorithm by examining detailed client throughput in an example WLAN with 200 clients. The cumulative distribution function (CDF) of client throughput for UDP traffic is given in Fig. 2.8, where FAME-0.4 and FAME-0.8 denote variants of FAME algorithm where the balancing factor w is set to 0.4 and 0.8, respectively. Different types of clients are plotted separately for clear presentation. Under uniform distribution, it can be observed that although about 60% 802.11b clients have higher throughput if the max-min algorithm is used, over 90% of 802.11g and 802.11n clients have much higher throughput if FAME or Categorized algorithm is used. In addition, most 802.11b clients in FAME and Categorized algorithms can achieve at least one half of the throughput achieved by 802.11b clients in max-min algorithm, which is acceptable for most applications. On the other hand, even if the max-min algorithm is used, the average throughput of 802.11n clients (around 2.5Mbps) is still much higher than the average throughput of legacy clients (around 250Kbps). In other words, the max-min algorithm cannot provide throughput fairness among 802.11n clients and legacy clients, because of the frame aggregation mechanism of 802.11n. Under hotspot distribution, it can be noted that regardless of the client type, a large number of clients have very low throughput in Categorized and RSSI-based algorithms due to AP overload. FAME algorithm can greatly reduce the amount of such clients while ensuring that the throughput of each client is comparable to other AP association schemes. More importantly, it is obvious that as the balancing factor increases, the throughput of 802.11b clients increases while the throughput of 802.11n clients decreases. In particular, the CDF of FAME algorithm is very close to the CDF of max-min algorithm when the balancing factor equals 0.8. This validates that FAME algorithm can reach a balance



(a) Uniform distribution. (b) Hotspot distribution.

Figure 2.9: Number of associated clients on each AP in an example WLANs with 200 clients.

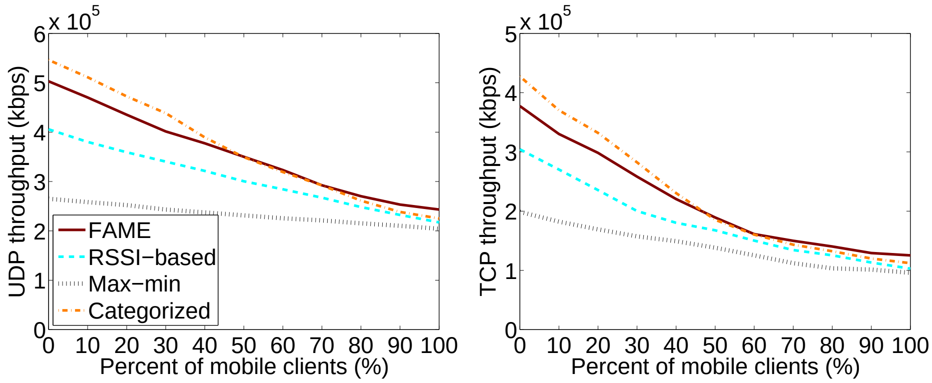
between airtime fairness and throughput fairness by choosing an appropriate balancing factor.

Finally, we evaluate the AP load distribution of various algorithms, by examining the number of associated clients on each AP in an example WLAN with 200 clients. The simulation results are shown in Fig. 2.9. Note that under uniform distribution (Fig. 2.9(a)), the number of associated clients on each AP is very close to the average value of all algorithms, although RSSI-based algorithm has a smaller variation. It is reasonable since the client distribution is the only factor affecting the load of each AP in that algorithm. On the other hand, as shown in Fig. 2.9(b), the load of RSSI-based and Categorized algorithms is not so balanced compared to other algorithms under hotspot distribution, which is because that the load on APs is not considered in RSSI-based and Categorized algorithms when making association decisions. Nevertheless, the maximum number of associated clients on an AP in Categorized is still much lower than that of RSSI-based algorithm (64 vs. 94), as some clients in the hotspots need to associate with far APs whose category matches their client types, if Categorized algorithm is used.

2.6.2 Performance in WLANs with Mobile Clients

In this subsection, we evaluate the network throughput of FAME and Categorized algorithms in WLANs with mobile clients. The mobility model for mobile clients is as follows: every mobile client randomly chooses a direction and a moving speed ranging from 0 to 5m/s, moves for a fixed duration t_1, and then pauses for a fixed duration t_2. The mobile client repeats this procedure until the end of the simulation. If the client moves into the boundary of the field during moving, it chooses a new direction and keeps moving at the same speed until duration t_1 is reached. If not otherwise specified, 50% clients are mobile and other clients are stationary. In addition, the moving duration t_1 and pausing duration t_2 are set to 6 and 4 seconds, respectively. All AP association algorithms are executed every 2 seconds.

We first study the network throughput of FAME and Categorized algorithms in WLANs with various percentages of mobile clients. The UDP and TCP throughput of all algorithms is plotted in Fig. 2.10, where the number of clients is fixed at 200 while the percent of mobile clients increases from zero to 100%. We can see that the network throughput of both TCP and UDP traffic decreases when the percent of mobile clients increases. This is because that mobile clients need to choose low data rates for transmissions to adapt to degraded channel quality, when they move to the boundary of neighboring BSSs. In addition, TCP throughput drops more drastically than UDP throughput as the percent of mobile clients grows, which can be attributed to the additive increasing multiplicative decreasing congest control mechanism of TCP traffic. When a client moves away from its associated AP, its TCP throughput decreases quickly if a few packets are dropped due to deteriorating channel conditions; when a client moves towards its associated AP, its TCP throughput increases slowly even if the rate adaptation algorithm has already chosen higher data rates for transmission. We can also observe that when the percent of mobile clients grows beyond 60%, the network throughput of FAME algorithm becomes higher than that of Categorized algorithm. The reason is that when most clients move around the field, some clients in Categorized algorithm may fail to find APs with matched category. Then they need to share the same AP with different types of clients, which undermines the throughput gain of isolating different types of clients. Nevertheless, the proposed algorithms always lead to higher network throughput than the compared schemes regardless of the percent of mobile clients, validating the benefits of minimizing protection overhead and encouraging airtime fairness.

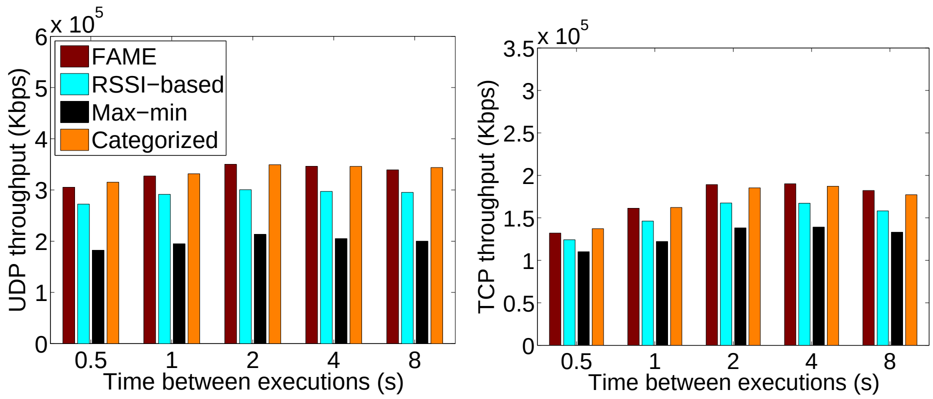


(a) UDP traffic. (b) TCP traffic.

Figure 2.10: Network throughput in a 200-client WLANs under various percentages of mobile clients.

We then evaluate the impact of the frequency at which the AP association algorithms are executed on network throughput. The network throughput of a WLAN with 200 clients is shown in Fig. 2.11, in which all AP association algorithms are executed every 0.5, 1, 2, 4, and 8 seconds, respectively. It can be noted that the network throughput increases when the duration between two consecutive executions of the AP association algorithms grows from 0.5 to 2 seconds. This is because that clients cannot transmit or receive data frames when they are scanning nearby APs in all channels for re-association. Thus the network throughput is reduced if the AP association algorithms are executed too frequently. In addition, the throughput improvement is more obvious for TCP traffic when the duration increases from 0.5 to 2 seconds, as frequent scanning of nearby APs may result in time out for some TCP packets and trigger retransmissions. On the other hand, the network throughput of the proposed algorithms decreases when the duration between two consecutive executions grows beyond 4 seconds. The reason behind this is that after moving to new locations, mobile clients remain associated with far APs and use low data rates for transmission, if the association algorithms are not executed timely.

From above simulation results, we can see that in WLANs where clients are uniformly distributed or there are few mobile clients, Categorized algorithm is a better choice for deployment, as it has much lower complexity, can lead



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

Figure 2.11: Impact of the duration between two consecutive executions of AP association algorithms on network throughput.

to higher network throughput while the throughput of most legacy clients is acceptable. In WLANs where clients follow the hotspot distribution or there are a number of mobile clients, FAME algorithm performs the best, as it can distribute load among APs, and provides all clients generally the same medium access time by maximizing the minimum MAC efficiency. More importantly, FAME algorithm can allocate more medium access time to legacy clients than 802.11n clients if necessary, by plugging in a positive balancing factor into FAME algorithm. In the same WLAN, both algorithms can be deployed simultaneously and the best one can be activated according to the dynamic characteristics of the network.

2.7 Experimental Results

In this section, we further verify the performance of the proposed algorithms in a WLAN testbed. As shown in Fig. 2.12, three 802.11n APs are deployed in two adjacent rooms as the testbed, with each AP operating on channel 1, channel 6 and channel 11 on 2.4Ghz frequency band, respectively. In addition, two 802.11b clients (indexed by 1 to 2), three 802.11g clients (indexed by 3 to 5) and five 802.11n clients (indexed by 6 to 10) are randomly placed in the two rooms. The wireless signal can penetrate the wall between the two rooms without obvious degradation. We run the test at midnight when no traffic is

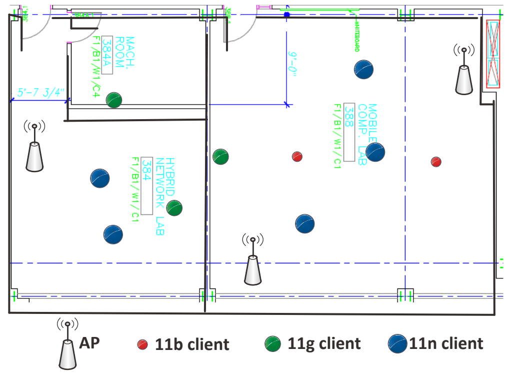
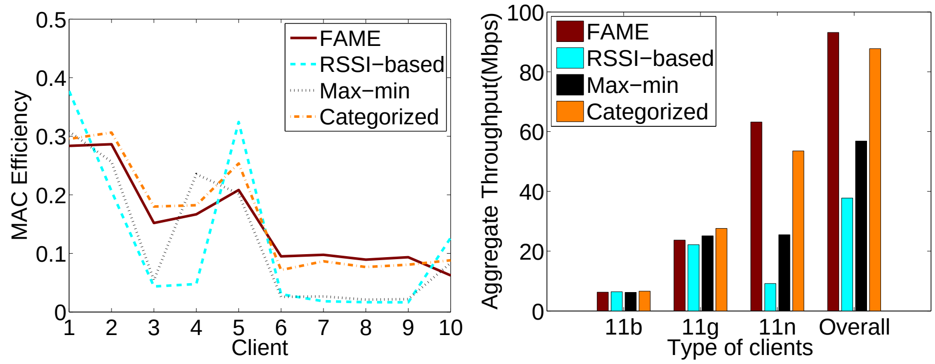


Figure 2.12: Locations of APs and various clients for the 802.11n testbed.

observed over the deployed WLAN of the building, thus external interference can be neglected. During the test, all clients transmit saturated UDP traffic to a PC, which is connected to all three APs via Gigabit Ethernet. The PC also transmits UDP traffic to every client. We have two sets of test scenarios. The frame aggregation feature of 802.11n clients is disabled in one scenario and enabled in the other.

We first study MAC efficiency and aggregated throughput for all clients when the frame aggregation feature on 802.11n clients is disabled. The test results are shown in Fig. 2.13. We can see that by using the proposed AP association algorithms, the MAC efficiency of all clients is more balanced, especially for clients of the same type. Note that the MAC efficiency of 802.11n clients is much lower than 802.11b clients regardless of AP association algorithm, which is inevitable since without frame aggregation, 802.11n clients take much less time to transmit data payload, while their transmitting overhead at the MAC and physical layers is identical to or even higher than that of legacy clients. As for aggregated throughput, both the overall throughput and the 802.11n throughput are greatly improved by the proposed algorithms, since the protection overhead for 802.11n clients is completely avoided. Another interesting observation is that FAME provides higher aggregated throughput for 802.11n clients compared with Categorized algorithm. This is because that with FAME, 802.11n clients share APs with 802.11g clients, resulting in more balanced load among all APs, at the cost of slightly increased preamble

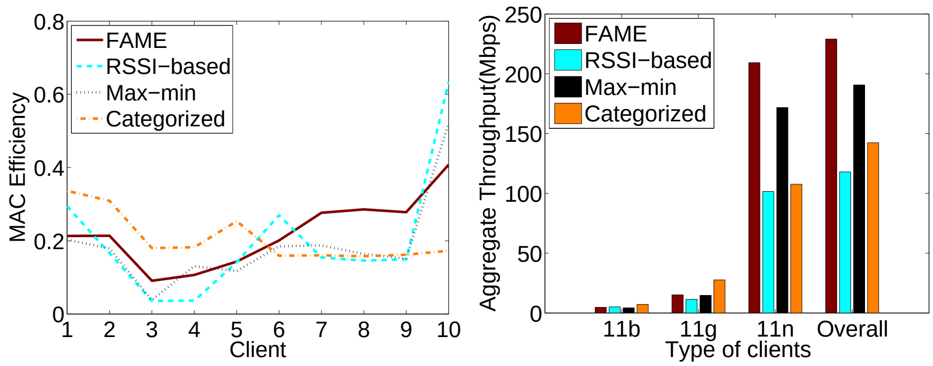


(a) MAC efficiency. (b) Aggregated throughput.

Figure 2.13: MAC efficiency and aggregated throughput for the testbed when frame aggregation of 802.11n clients is disabled.

durations for 802.11n clients.

We now evaluate the proposed algorithms when frame aggregation on 802.11n clients is enabled, which is the default option in reality. The MAC efficiency and aggregated throughput for all clients are plotted in Fig. 2.14 (a) and (b). Similar to the previous scenario, the proposed algorithms can have more balanced MAC efficiency among all clients. However, different from the previous scenario, we notice that the MAC efficiency of 802.11n clients is comparable to or even higher than that of legacy clients for all AP association algorithms this time. The reason is that 802.11n clients can transmit many data packets in each transmission by aggregating them into one 802.11 frame, although they have the same opportunity to access the medium as other clients and even need extra overhead to protect their transmissions. In contrast, legacy clients transmit one data packet in each transmission. Note that the MAC efficiency of a 802.11g client would be very poor if it shares the AP with 802.11b and 802.11g clients, because both 802.11b and 802.11n clients occupy the medium for much longer time for each obtained transmission opportunity. This is confirmed by the aggregated 802.11g throughput of the algorithms shown in Fig. 2.14 (b). Only for Categorized algorithm, the throughput of 802.11g clients is proportional to their data rates, because the medium is not shared with either 802.11b or 802.11n clients. Furthermore, 802.11n clients can achieve high throughput even if they share an AP with



(a) MAC efficiency. (b) Aggregated throughput.

Figure 2.14: MAC efficiency and aggregated throughput for the testbed when frame aggregation of 802.11n clients is enabled.

802.11b clients, with the help of frame aggregation. For the overall aggregated throughput, FAME outperforms all other algorithms by minimizing the protection overhead for 802.11n clients, reducing the impact of 802.11n and 802.11b clients on 802.11g clients, and balancing network load at the same time.

2.8 Conclusions

In this chapter, we have studied AP association for IEEE 802.11n based WLANs with heterogeneous clients, in particular, addressed the new challenges introduced by the high data rates and frame aggregation mechanism of 802.11n. We first showed via experiments that 802.11n throughput and overall network throughput can be severely affected by legacy clients. After that, we presented a bi-dimensional Markov model to estimate client throughput, and formulated the AP association problem into an optimization problem. Based on the Markov model introduced in the problem formulation, we provided an AP association algorithm with which each client achieves the throughput proportional to its data rate. For WLANs where APs are densely deployed, we further provided a simple but effective AP association algorithm that guarantees the performance of 802.11n transmissions by associating different type of clients with different APs. Finally, we conducted extensive simulations and experiments to assess their performance. Our simulation and experimental results demonstrate that not only high network throughput, but also fairness and balanced load can be achieved by the proposed algorithms.

Chapter 3

Channel Assignment in WLANs with Heterogeneous Clients

3.1 Introduction

In this chapter, we study channel assignment in 802.11n WLANs with heterogeneous clients, aiming at maximizing the overall network throughput. We first present a novel network model for channel assignment in 802.11n WLANs, considering the new channel bonding and frame aggregation mechanisms, as well as the impact of legacy clients. We then discuss the interference relationship among clients from nearby BSSs, and develop an analytical model to estimate the throughput of each client. Based on this analytical model, we formulate the channel assignment problem into an integer linear program (ILP), which is NP-hard. After that, we present a distributed channel assignment algorithm, where each AP iteratively updates its channel to maximize the estimated local network throughput. Furthermore, we propose another low-complexity channel assignment algorithm, with the objective to minimizing interference suffered by high-rate 802.11n clients. We have carried out extensive simulations to evaluate the proposed algorithms. As will be seen from the simulation results, our proposed algorithms significantly outperform the compared algorithms in terms of network throughput.

The remainder of the chapter is organized as follows. Section 3.2 reviews the related work. Section 3.3 introduces the network model and formulates the channel assignment problem into an integer linear program. Section 3.4 presents two distributed algorithms. Section 3.5 evaluates the performance of the proposed algorithms. Finally, Section 3.6 concludes the chapter.

3.2 Related Work

There has been some work in the literature on channel assignment based on vertex coloring algorithms, where each vertex represents a BSS and two vertices are connected if their corresponding BSSs interfere with each other. In [29] and [30], the weight of a vertex is defined as the traffic demand of the corresponding BSS. The objective of channel assignment is to minimize the maximum channel utilization ratio, which is defined as the fraction of time that a channel is occupied by wireless transmissions. In the channel assignment algorithm in [31], for each edge in the graph, a weight is defined as the total number of clients in the corresponding BSSs, and the objective is to minimize the overall weight of the graph. A similar channel assignment scheme was proposed in [32], where the expected transmission delay due to interference from a neighboring BSS is regarded as the weight of the edge. However, the interference model in the above schemes is inaccurate, as clients in the same BSS are geographically distributed and thus sense distinct interference.

There have also been some channel assignment algorithms in which interference experienced by each client is considered individually. In the channel assignment algorithm in [33], each client maintains an interference set including all the conflicting BSSs. A random compaction scheme was proposed to maximize the number of conflict-free clients. In [34], a weight is defined for each BSS to reflect the traffic demand as well as interference degree of every client in the BSS. The objective is to minimize the overall weight so as to maximize the network throughput. It has been shown via simulations in [37] that the overall network performance can be enhanced by these client-centric schemes [33, 34], compared with aforementioned schemes [29–32] where interference is examined at the BSS level.

Besides above channel assignment algorithms for WLANs, a number of channel assignment algorithms have been proposed for mobile and cellular networks as well. In [38], an efficient distributed algorithm was proposed for dynamic channel allocation in mobile and cellular networks, where the channels are grouped by the number of cells in a cluster and each group of channels cannot be shared concurrently within the cluster. Moreover, in [39], an overview of various channel allocation algorithms was presented, and the major channel allocation protocols in each category were discussed. However, these algorithms cannot be directly applied to WLANs for two reasons. First, in cellular networks, different mobile hosts in the same cell need to be assigned different channels; while in WLANs, all clients in the same BSS contend and share the same channel with the CSMA/CA mechanism. After acquiring the channel access opportunity, a client transmits data on all sub-carriers of the channel by using the OFDM technology. Second, in cellular networks, neighboring cells should operate on orthogonal channels to avoid mutual interference; while in WLANs, neighboring BSSs can operate on overlapped channels.

In the meanwhile, some research efforts have been devoted to improving the performance of 802.11n WLANs [40, 43–47]. In [43] and [44], the performance of two different frame aggregation mechanisms: aggregation of MAC service data units (A-MSDU) and aggregation of MAC protocol data units (A-MPDU) are modeled and compared. The optimal aggregation sizes under various channel conditions were derived as well. An aggregated frame is acknowledge by a block ACK (BACK) frame, which includes a bitmap to indicate the receiving information of all sub-frames. Corrupted sub-frames in an aggregated frame can be retransmitted in a greedy way by aggregating the corrupted sub-frames together with new data frames. They can be also retransmitted in a conservative way by aggregating only corrupted sub-frames. It was shown in [45] that the greedy scheme can lead to better network performance. In addition, a MIMO rate adaptation algorithm was proposed in [46] for 802.11n WLANs, where single spatial-stream rates are distinguished from double spatial-stream rates. In [40], a joint channel assignment and AP association scheme for 802.11n WLANs was presented. Nevertheless, the impact of legacy clients is not studied in the above schemes. It was shown in [47] that the throughput of 802.11n transmissions drops drastically when coexisting with legacy clients.

3.3 System Models and Problem Formulation

In this section, we describe the network model for channel assignment in 802.11n WLANs with heterogeneous clients, estimate the uplink and downlink throughput for each client, and formulate the channel assignment problem into an optimization problem.

3.3.1 Network Model

Consider an 802.11n WLAN consisting of a set of APs. Each AP is associated with a number of clients, including both 802.11n and legacy clients. We use sets A and N to denote the set of APs and the set of clients, respectively. For each AP $a \in A$, we define $N_a \subseteq N$ as the subset of clients associated with it. For each client $n \in N$, we use variable a_n to denote its associated AP. We further define a variable t_n to indicate the client type, which takes an enumerated value from set $\{a, b, g, n\}$. We assume that the channel condition is symmetric and varies slowly, as most clients in WLANs are static or semi-static. Hence there is an optimal data rate between each client and its associated AP. For client n, we use variable r_n to denote its optimal data rate. Note that the optimal data rate of 802.11n clients is also related to bandwidth of the channel assigned to their associated AP. Furthermore, the frame length and traffic direction of a client depends on its application type, e.g., voice applications typically have short, bi-directional frames. In the rest of the chapter, we use *downlink traffic* to refer to the traffic from APs to clients, and *uplink traffic* to refer to the traffic from clients to APs. We use variable l_n^{up} to denote the average frame length for uplink traffic from client n, and variable l_n^{down} to denote the average frame length for downlink traffic to client n.

Legacy clients can only transmit or receive one frame each time, while 802.11n clients can use frame aggregation to transmit or receive multiple frames at each medium access opportunity. The number of sub-frames in an aggregated frame of 802.11n clients is called *frame aggregation level* and is limited by the maximum frame length, i.e., 64KB, as well as the common practice of maximum transmission duration, e.g., 4ms. The actual aggregation level of a client depends on both the data rate of the client and the number of frames in the queue. If there are plenty of frames in the queue, the maximum

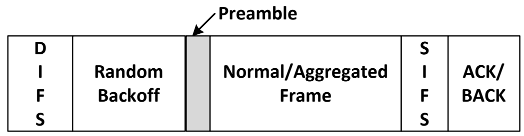
Table 3.1: Interference matrix for 802.11n channels at 2.4GHz

Channels	**1 (20)**	**6 (20)**	**11 (20)**	**1_6 (40)**	**6_11 (40)**
1 (20)	1	0	0	1	0
6 (20)	0	1	0	1	1
11 (20)	0	0	1	0	1
1_6 (40)	1	1	0	1	1
6_11 (40)	0	1	1	1	1

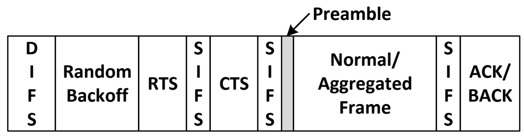
number of frames allowed by the transmission duration will be aggregated. Otherwise, all frames for the clients in the queue will be aggregated. The aggregation level of legacy clients can be regarded as 1 if they have pending traffic. If a client has no pending traffic, its aggregation level is 0. We define variable f_n^{up} and f_n^{down} to represent the average aggregation level of client n for uplink and downlink traffic, respectively. Then the average length of an aggregated frame of client n for uplink and downlink traffic equals $l_n^{up} \cdot f_n^{up}$ and $l_n^{down} \cdot f_n^{down}$, respectively.

Note that the definition of the available channel set for an 802.11n network in this dissertation is different from that in the literature. Typically, the available channel set is defined as the collection of all non-overlapping channels in the frequency band, and there is no interference between any two channels in the set. In an 802.11n WLAN, a bonded 40Mhz channel interferes with any other channel that overlaps the 40Mhz bandwidth. Therefore, we define an available channel set K, including both the non-overlapping 20Mhz channels and the bonded 40Mhz channels. We further define an interference matrix I to denote the interference relationship between any two channels in K. Specifically, the available channel set at 2.4GHz frequency band is defined as $K = \{1, 6, 11, 1\_6, 6\_11\}$, where 1_6 and 6_11 stand for the 40MHz channels bonded by channel 1 and channel 6, channel 6 and channel 11, respectively. The corresponding interference table is given in Table 3.1. We then use variable $k_a \in K$ to denote the channel assigned to AP a.

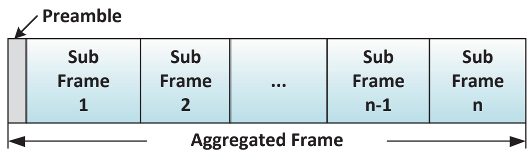
Stations in an 802.11n WLAN use the CSMA/CA mechanism to access the wireless medium. As shown in Fig. 3.1, if a station has pending traffic, it first senses the wireless medium for a distributed inter frame space (DIFS) period. If the medium is busy during this period, the station holds its transmission until the current transmission finishes, and restarts to sense the medium. Oth-



(a) Frame transmission in 802.11n WLANs without RTS/CTS.



(b) Frame transmission in 802.11n WLANs with RTS/CTS.



(c) Structure of an 802.11n aggregated frame.

Figure 3.1: Time components of legacy and 802.11n frame transmissions.

erwise, it defers its transmission for a random period of time to avoid collisions. The station then exchanges RTS/CTS frames with the receiving station if the transmission could be corrupted by legacy clients or hidden terminals. After that, the station transmits a data frame or an aggregated data frame, beginning with a preamble. If the receiving station successfully receives a normal frame, it sends back an ACK frame after a short inter frame space (SIFS) period. If the receiving station receives an aggregated frame, it sends back a BACK frame after a SIFS period. As discussed earlier, the type of preamble and the necessity of RTS/CTS for 802.11n stations depend on the coexistence of legacy clients in the same BSS. We define the *operation mode* o_a of AP a as a function of types of its associated clients. Then the type of preamble and the necessity of RTS/CTS of each client can be determined from the operation mode of its associated AP. Accordingly, we define the transaction time of a station as the period from sensing the wireless medium to successfully receive the ACK/BACK frame. The transaction time of a client can be expressed as

$$T_{tran} = T_{difs} + T_{cont} + T_{prot} + T_{pre} + T_{data} + T_{sifs} + T_{ack/back} \tag{3.1}$$

where T_{difs} denotes the DIFS period, T_{sifs} denotes the SIFS period, T_{cont}

denotes the random backoff period for contention, T_{prot} denotes the duration of RTS/CTS exchanges, T_{pre} denotes the preamble duration, T_{data} denotes the duration of transmitting a regular data frame or an aggregated frame, and $T_{ack/back}$ denotes the duration of an ACK or BACK frame, respectively. Note that the uplink transaction time may be different from the downlink transaction time for the same client due to various application requirements on traffic loads.

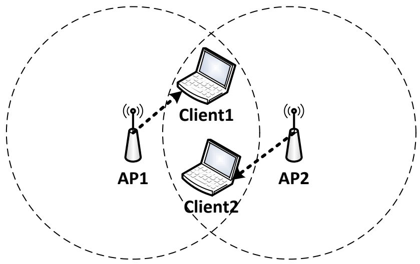
In the above equation, T_{difs}, T_{sifs}, T_{ack} and T_{back} are constants, T_{prot} and T_{pre} can be determined from the operation mode of the associated AP. For example, T_{prot} of 802.11n clients equals 430μs when coexisting with 802.11b clients; otherwise, it is zero. T_{data} can be derived from the optimal data rate, average frame length and average aggregation level of a station. Therefore, T_{data} for uplink traffic of client n can be given by

$$T_{data}(n) = \frac{8l_n^{up} \cdot f_n^{up}}{r_n} \tag{3.2}$$

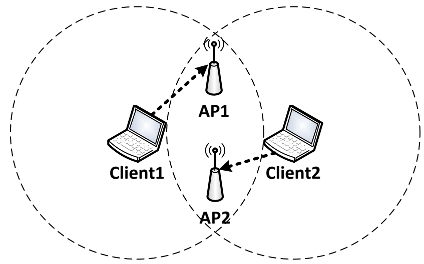
The T_{data} for downlink traffic of client n can be determined similarly. T_{cont} depends on the number of contending stations and will be discussed in detail later.

3.3.2 Interference Model

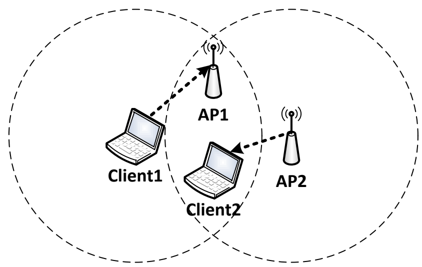
In this chapter, we consider interference from the perspective of clients. For any two clients from neighboring BSSs, we say they interfere with each other if one client or its associated AP can sense the transmission of the other client or the associated AP of the other client. This interference model applies to both uplink and downlink traffic, since each data frame needs to be acknowledged by an ACK frame. For any two clients $m, n \in N$, we use a binary variable $i_{m,n}$ to denote their interference relationship. $i_{m,n}$ is one if they interfere with each other; otherwise, it is zero. Two clients need to contend and share the wireless medium if they interfere with each other and their BSSs are assigned with overlapping channels. Two interfering clients may further experience the hidden terminal problem, which may lead to severe transmission failures. As shown in Fig. 3.2, for downlink traffic, the hidden terminal problem occurs if the associated APs of two interfering clients are not within the carrier sense



(a) Downlink traffic.



(b) Uplink traffic.



(c) Mixed traffic.

Figure 3.2: Examples of the hidden terminal problem for downlink, uplink and mixed traffic. (a) AP 1 is not in the carrier sense range of AP 2. (b) Client 1 is not in the carrier sense range of client 2. (c) Client 1 is not in the carrier sense range of AP 2.

range of each other. Similarly, for uplink traffic, the hidden terminal problem occurs if two interfering clients are not within the carrier sense range of each other. For the mixed traffic scenario where one client has uplink traffic while the other client has downlink traffic, the hidden terminal problem occurs if the transmitting client is not in the carrier sense range of the transmitting AP. We assume that RTS/CTS are enabled for clients that are affected by hidden terminals.

3.3.3 Throughout Estimation Model

In this subsection, we derive a model to estimate the throughput for each client in an 802.11n network, using the network model and interference model in previous two subsections. In this way, the throughput of the entire network can be estimated, and the objective of channel assignment becomes to maximize the estimated overall network throughput.

We can determine the transaction time of each client from Equation (3.1). Hence, if the medium is always sensed free, the throughput of a client can be simply derived by dividing the aggregated frame length by the transaction time of the client. As aforementioned, the random backoff duration T_{cont} in Equation (3.1) depends on the number of contending stations and the collision probability. Given C contending stations, the contention duration can be approximated using equations from [36], expressed as follows

$$\begin{aligned} p_{col}^{C} \approx& 1 - \left(1 - \frac{1}{CW_{min}}\right)^{C-1} \\ T_{cont}^{C} \approx& T_{slot} \cdot \frac{1 + p_{col}^{M}}{2C} \cdot \frac{CW_{min}}{2} \end{aligned} \tag{3.3}$$

In the above equations, p_{col}^{C} and T_{cont}^{C} stand for the collision probability and contention duration with C stations, while T_{slot} and CW_{min} are both constants, standing for the duration of a time slot and the minimum contention window size, respectively.

To transmit uplink traffic, a client needs to share the wireless medium with other stations in the same BSS, including both clients and its associated AP. It also needs to share the wireless medium with other interfering clients and APs in nearby BSSs if they are assigned with overlapping channels. Whether a client interferes with an AP depends on the destination of the transmission from the AP. However, it is impossible to accurately predict the destination of the next downlink transmission from an AP at a specific time, as it depends on the downlink traffic loads of all clients and the queue scheduling strategy of the AP. For simplicity, we say that a client interferes with an AP from neighboring BSSs, if the client interferes with at least one client associated with the AP. Then the number of contending stations for client n (including client n itself) can be derived by the following equation

$$C_n = 1 + |N_{a_n}| + \sum_{m \in N, m \notin N_{a_n}} I(k_{a_m}, k_{a_n}) \cdot i_{m,n} + \sum_{b \in A, b \neq a_n} I(k_{a_n}, k_b) \cdot \max_{m \in N_b} \{i_{m,n}\}$$

in which the first term stands for the associated AP of client n, the second term stands for all clients in the same BSS, the third term stands for contending clients in nearby BSSs, and the fourth term stands for contending APs from nearby BSSs. By plugging C_n into Equation (3.3), the random backoff duration T_{cont} of client n can be derived.

Furthermore, when all stations have pending traffic, the transmission alternates among stations in the long term since each station has an equal opportunity to access the medium with CSMA/CA. Then after transmitting a frame, a client needs to wait for the transmissions of all other contending stations before transmitting its own frame again. We use $T_{tran}^{avg}(a)$ to denote the average transaction time of AP a for downlink traffic, which will be discussed in detail later. Then the duration between two transmissions from client n can be given by

$$T(n) = T_{tran}^{avg}(a_n) + \sum_{m \in N_{a_n}} T_{tran}^{up}(m) + \sum_{m \in N, m \notin N_{a_n}} I(k_{a_m}, k_{a_n}) \cdot i_{m,n} \cdot T_{tran}^{up}(m)$$
$$+ \sum_{b \in A, b \neq a_n} I(k_{a_n}, k_b) \cdot \max_{m \in N_b} \{i_{m,n}\} \cdot T_{tran}^{avg}(b)$$

Accordingly, the estimated uplink throughput $S^{up}(n)$ of client n can be approximated as the average length of its aggregated frames, divided by the duration $T(n)$ between two consecutive transmissions from it, that is,

$$S^{up}(n) = \frac{l_n^{up} \cdot f_n^{up}}{T(n)} \tag{3.4}$$

To transmit downlink traffic, an AP has to contend the wireless medium with its associated clients. It also has to contend the wireless medium with interfering clients and APs from nearby BSSs. Similar to uplink traffic, we say that an AP interferes with a neighboring AP if at least one pair of their associated clients interfere with each other. In addition, an AP interferes with a client from neighboring BSSs if at least one of its associated clients interferes with the client from neighboring BSSs. Then the number of contending stations for AP a (including AP a itself) can be expressed as

$$C_a = 1 + |N_a| + \sum_{b \in A, b \neq a} I(k_a, k_b) \cdot \max_{m \in N_a, n \in N_b} \{i_{m,n}\} + \sum_{n \in N, n \notin N_a} I(k_{a_n}, k_a) \cdot \max_{m \in N_a} \{i_{m,n}\}$$

where the first term represents the AP itself, the second term represents the associated clients of the AP, the third term represents interfering APs in nearby BSSs, and the fourth term represents interfering clients from nearby BSSs. Then the random backoff duration $T_{cont}(a)$ of AP a can be derived by plugging C_a into Equation (3.3).

Different from uplink traffic, the transaction time of an AP may change every time, because an AP is usually associated with multiple clients and it needs to transmit downlink traffic to them in turns. As an AP can transmit to only one client at each medium access opportunity, we assume that the AP maintains an individual queue for each client and alternates the transmission to all clients in a round-robin fashion. Then the average transaction time for AP a is

$$T_{tran}^{avg}(a) = \frac{\sum_{n \in N_a} T_{tran}^{down}(n)}{|N_a|}$$

Similar to clients, an AP needs to share the wireless medium and alternate transmissions with all other contending stations. Then after transmitting a frame, an AP waits for the transmissions of all other contending stations before transmitting another downlink frame. The duration between two consecutive transmissions from AP a can be given by

$$\begin{aligned} T(a) = T_{tran}^{avg}(a) + \sum_{n \in N_a} T_{tran}^{down}(n) + \sum_{b \in A, b \neq a} I(k_a, k_b) \cdot \max_{m \in N_a, n \in N_b} \{i_{m,n}\} \cdot T_{tran}^{avg}(b) \\ + \sum_{n \in N, n \notin N_a} I(k_{a_n}, k_a) \cdot \max_{m \in N_a} \{i_{m,n}\} \cdot T_{tran}^{up}(n) \end{aligned} \tag{3.5}$$

Then the overall downlink throughput of AP a can be approximated as the average frame length of all associated clients with downlink traffic, divided by the duration $T(a)$ between two consecutive transmissions from AP a, that is,

$$S^{down}(a) = \frac{\sum_{n \in N_a} l_n^{down} \cdot f_n^{down}}{|N_a| \cdot T(a)}$$

Accordingly, for a client n, its downlink throughput can be approximated as its frame length, divided by the duration between two consecutive transmissions from its associated AP to it, which equals $|N_{a_n}| \cdot T(a_n)$. This is because

Table 3.2: List of notations used in channel assignment problem formulation

A	Set of 802.11n APs
N	Set of heterogeneous clients
K	Set of available channels
I	Interference matrix among available channels in K
N_a	Subset of clients associated with AP a
a_n	Associated AP of client n
o_a	Operation mode of AP a
t_n	Type of client n
r_n	Optimal data rate between client n and its AP
l_n	Average frame length of client n
f_n	Average frame aggregation level of client n
k_a	Channel assigned to AP a
$i_{m,n}$	Indicator whether clients m, n interfere with each other
$T_{tran}^{down}(n)$	Transaction time of client n for downlink traffic
$T_{tran}^{up}(n)$	Transaction time of client n for uplink traffic
$T_{tran}^{avg}(a)$	Average downlink transaction time for clients of AP a
T_m	Duration between two transmissions from station m
$S^{up}(n)$	Estimated uplink throughput of client n
$S^{down}(n)$	Estimated downlink throughput of client n

the associated AP needs to not only share the medium with contending stations, but also rotate the transmission opportunity among all clients. The downlink throughput of client n can be expressed as

$$S^{down}(n) = \frac{l_n^{down} \cdot f_n^{down}}{|N_{a_n}| \cdot T(a_n)} \tag{3.6}$$

It should be mentioned that the throughput degradation caused by collisions is not considered in the throughput estimation model, as the primary objective of this chapter is to optimize the channel assignment of the network rather than accurately predict the throughput.

3.3.4 Formulation of Channel Assignment Problem

The channel assignment problem in 802.11n WLANs with heterogeneous clients can be formally described as follows. Given a WLAN consisting of a set A of 802.11n APs, a set N of heterogeneous clients, and a set K of available channels. Each AP is associated with a subset $N_a \in N$ of clients. Assign a channel to each AP, such that the overall network throughput is maximized.

We summarize the notations used in the problem formulation in Table 3.2. The channel assignment problem can be formulated into an optimization problem, aiming at maximizing the network throughput, which can be expressed as

Maximize

$$\sum_{\forall n \in N} (S^{up}(n) + S^{down}(n))$$

Subject to

$$\begin{aligned}
&k_a \in K, && \forall a \in A && (3.7)\\
&T_{tran}(n) = f(r_n, l_n, f_n, o_{a_n}, k_{a_n}), && \forall n \in N && (3.8)\\
&T(n) = T^{avg}_{tran}(a_n) + \sum_{m \in N_{a_n}} T^{up}_{tran}(m) && &&\\
&+ \sum_{m \in N, m \notin N_{a_n}} I(k_{a_m}, k_{a_n}) \cdot i_{m,n} \cdot T^{up}_{tran}(m) && &&\\
&+ \sum_{b \in A, b \neq a_n} I(k_{a_n}, k_b) \cdot \max_{m \in N_b} \{i_{m,n}\} \cdot T^{avg}_{tran}(b), \forall n \in N && && (3.9)\\
&S^{up}(n) = \frac{l^{up}_n \cdot f^{up}_n}{T(n)}, && \forall n \in N && (3.10)\\
&T(a) = T^{avg}_{tran}(a) + \sum_{n \in N_a} T^{down}_{tran}(n) && &&\\
&+ \sum_{b \in A, b \neq a} I(k_a, k_b) \cdot \max_{m \in N_a, n \in N_b} \{i_{m,n}\} \cdot T^{avg}_{tran}(b) && &&\\
&+ \sum_{n \in N, n \notin N_a} I(k_{a_n}, k_a) \cdot \max_{m \in N_a} \{i_{m,n}\} \cdot T^{up}_{tran}(n), \forall a \in A && && (3.11)\\
&S^{down}(n) = \frac{l^{down}_n \cdot f^{down}_n}{|N_{a_n}| \cdot T(a_n)}, && \forall n \in N && (3.12)
\end{aligned}$$

In the above formulation, Equation (3.7) ensures that each AP is assigned with a channel from set K; Equation (3.8) combines Equations (3.1), (3.2) and (3.3), indicating that the transaction time of a client depends on its data rate, its average frame length, its average aggregation level, the operation mode and channel bandwidth of its associated AP. Equations (3.9), (3.10), (3.11) and (3.12) are used to estimate the uplink and downlink of each client. This optimization problem is an integer linear program (ILP), which is well-known to be NP-hard. Therefore, we will present two distributed algorithms in the next section to provide sub-optimal solutions.

3.4 Channel Assignment Algorithms

As shown in the previous section, the interference relationship between any two clients from different BSSs is needed to estimate the throughput of a BSS. This information is essential to our proposed channel assignment algorithms. Therefore, in this section, we first introduce a protocol to obtain the interference relationship among clients from neighboring BSSs. We then propose a distributed channel assignment algorithm, named as *throughput-maximizing channel assignment (TMCA)* algorithm, in which each AP estimates the network throughput in its neighborhood using the throughput estimation model from the previous section, and iteratively updates its channel until the local network throughput cannot be further improved. After that, we present another distributed channel assignment algorithm, referred to as *lexicographical interference minimization (LIM)* algorithm, where each client is assigned a priority according to its client type and data rate, and the objective is to minimize the interference experienced by high-priority clients. TMCA algorithm can lead to higher network throughput while LIM algorithm has a lower complexity. The proposed protocol and algorithms are described in detail in the following subsections.

3.4.1 Protocol for Local Information Exchange

In most of current WLAN deployments, APs are connected to a wireless controller or the Internet via local area network (LAN) links. This is also the case for 802.11n WLANs, as wireless backhaul links are more prone to interference and congestion due to the boosted throughput from each AP. Therefore, as shown in Fig. 3.3, we assume that each AP is directly connected to a LAN infrastructure and the LAN interfaces of neighboring APs are within the same broadcast domain. Under this assumption, information exchange among neighboring BSSs is carried out through LAN broadcast, thus the transmission failure due to collision and poor channel condition in the protocol can be ignored. The protocol for local information exchange includes three phases: client list and measurement schedule announcement phase, carrier sense and report phase, and the interference and hidden terminal announcement phase.

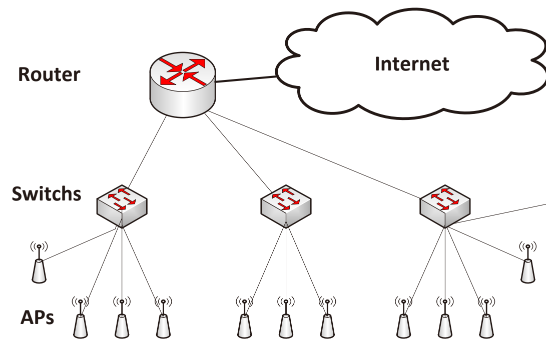


Figure 3.3: A typical example of 802.11n WLAN deployments, in which each AP is connected to an access switch LAN interface. All access switches are connected to the wireless controller or the Internet via core switches or routers.

Client List and Measurement Schedule Announcement

In this phase, each AP broadcasts a client list and measurement schedule (CLMS) message including the associated client information over the LAN interface, such that other APs in the neighborhood can obtain the type, data rate, average frame length, and average aggregation level of its associated clients. Furthermore, to determine the interference among clients, the AP schedules a measurement message for every station in its BSS (including the AP itself), thus other clients in neighboring BSSs can determine whether the station transmitting the measurement message is in its carrier sense range by sensing the wireless medium at the scheduled time. Each AP chooses a random base time and chooses the time for its clients increasingly. The measurement schedule for associated clients is included in the CLMS message as well. A CLMS message is given in Table 3.3 as an example. After receiving a CLMS message, an AP rebroadcasts it in the wireless medium such that its associated clients can obtain the measurement schedule. In addition, an AP updates its schedule if it has not broadcast the CLMS message while its scheduled time

Table 3.3: An example of the client list and measurement schedule message.

MAC Address	Type	Measurement Schedule
00:18:E7:D3:9C:B3	802.11n AP	05:12:17
00:20:A6:CA:44:74	802.11n Client	05:12:19
00:23:69:D8:F8:EA	802.11g Client	05:12:21
...	...	...

conflicts with the schedule in the received messages. All measurement messages are transmitted in the wireless medium over the default channel.

Carrier Sense and Report

At the end of the last phase, each station is aware of the measurement schedule of all other stations in the neighborhood. In this phase, each station constructs a carrier sense list including all the stations within its carrier sense range. At each scheduled measurement time a station determines whether the transmitting station is within the carrier sense range by sensing the wireless medium. If the medium is busy, the transmitting station is added into its carrier list. A station broadcasts its own message at the scheduled time. After completing the measurement of all scheduled stations, a client sends a report message to its associated AP including its carrier sense list. Based on the carrier sense lists from its associated clients and itself, an AP can derive a set of interfering clients and hidden terminals for each client associated with it.

Interference and Hidden Terminal Announcement

Given the channel assignment of neighboring BSSs and the information acquired in the last two phases, an AP can estimate the throughput of its associated clients based on the proposed system models. However, to obtain a sub-optimal channel assignment in a distributed manner, each AP should be able to estimate the throughput of neighboring BSSs, so as to choose the best channel assignment. Hence in this phase, each AP broadcasts an interference and hidden terminal (IHT) message via the LAN interface, including the set of interfering clients and hidden terminals for each associated client. After receiving the IHT messages from neighboring BSSs, an AP is capable of estimating the throughput of its own BSS and its neighbors.

This local information exchange procedure should be performed periodically to reflect the variance of client association, channel condition and traffic load.

3.4.2 Throughput-Maximizing Channel Assignment Algorithm

We now describe the distributed throughput-maximizing channel assignment (TMCA) algorithm for 802.11n WLANs with heterogeneous clients, where each AP aims at maximizing the local network throughput, which is defined as the overall throughput of all BSSs in the neighborhood. TMCA algorithm works in a distributed manner, as each AP is capable of estimating the throughput of every neighboring BSSs. Initially, each AP randomly chooses a channel from the available channel set and broadcasts a channel announcement (CA) message via its LAN interface. After receiving a CA message, an AP first estimates its local throughput by taking the interference relationship among clients into the throughput estimation model in the previous section. Then it checks whether the local network throughput can be improved if assigning a different channel. If so, it assigns the channel that improves the local throughput most and broadcasts a CA message via the LAN interface. Otherwise, it keeps waiting for CA messages from other APs. The pseudo code of TMCA algorithm is given in Table 3.4.

The TMCA algorithm is triggered repeatedly at each AP by the received CA messages from other APs. An AP stops triggering the channel assignment of neighboring APs when it cannot further improve its local throughput and stops broadcasting CA messages. The TMCA algorithm terminates when no further CA message is broadcast. Note that the throughput is enhanced every time and the maximum throughput is limited by the channel capacity, thus the algorithm will provably terminate.

TMCA algorithm can be implemented in the user space of APs and clients. As discussed earlier, an AP needs the optimal data rate, average frame length and average aggregation level of associated clients to estimate their uplink and downlink throughput. An AP acquires the optimal data rate at 20MHz and

Table 3.4: Throughput-maximizing channel assignment algorithm

Input:
Transaction time of all clients in nearby BSSs
Interference relationship among clients in nearby BSSs
Output:
Channel assignment k_a for each AP $a \in A$
Algorithm:
1: **for each** AP $a \in A$
2: Assign a random channel $k \in K$
3: Broadcast a CA message
4: Idle and wait for CA messages
5: **If** receiving a CA message
6: Determine local throughput S
7: Determine the maximal local throughput S'
8: If assigned another channel k'
9: **If** $S' > S$
10: Assign k' to AP a
11: Broadcast a new CA message
12: **end if**
13: **end if**
14: Go back to step 3
15: **end for**

40MHz channels of a client from the rate adaptation procedure. It can further predict the average frame length and average aggregation level by examining the lengths and number of packets to the client in the queue. Similarly, a client can obtain the data rate, average frame length and aggregation level from the rate adaptation procedure and the queue at the MAC layer. The client can send a message to the AP to report such information. Moreover, an exponential moving average function can be applied on the collected data to limit the impact of traffic burst or rate adaptation due to temporarily external interference.

3.4.3 Lexicographical Interference Minimization Algorithm

As discussed in the previous subsection, TMCA algorithm requires the data rate and traffic load information of clients to make channel assignment decisions. Thus TMCA algorithm needs to be re-executed whenever the channel condition or traffic load of clients changes, to ensure the highest network throughput is achieved. Furthermore, TMCA needs to deploy a procedure at clients to operate properly, which may be difficult for some portable devices. Thus in this subsection, we present the lexicographical interference minimization (LIM) algorithm, which has a lower complexity and can be deployed without modifying clients.

In LIM, clients are categorized into groups according to their types and data rates. The rationale is to minimize the interference experienced by high-rate 802.11n clients, such that they can achieve a throughput that is proportional to their data rates. Then each AP organizes all clients in the neighborhood into a vector called *interference-free client vector*, in which high-rate 802.11n clients are placed at the beginning so as to reduce the interference they may experience. 802.11b clients with low data rates are placed at the end of the vector. Initially, each AP randomly assigns a bonded channel to itself and broadcasts a CA message via the LAN interface. After receiving a CA message, an AP determines the new inference-free client vector of its neighborhood. Then the AP checks whether the vector can be lexicographically improved by assigning a different channel to itself. If so, it chooses the channel that boosts the vector most and broadcasts a new CA message. The bonded channels are preferred at a tie to take advantage of channel bonding as much as possible. If the AP cannot improve the vector by changing its own channel, it waits for CA messages from other APs.

As the same iteration triggering (receiving of CA messages) and stopping conditions from TMCA algorithm are used, LIM algorithm will provably terminate as well. The pseudo code of LIM algorithm is given in Table 3.5.

Note that LIM algorithm can be easily implemented on APs, since it only needs type and data rates of clients in the neighborhood. An AP obtains the type information of all clients by looking up its association table. In

Table 3.5: Lexicographical interference minimization algorithm

Input:
  Type and data rate of all clients in nearby BSSs
  Interference relationship among clients in nearby BSSs
Output:
  Channel assignment k_a for each AP $a \in A$
Algorithm:
for each AP $a \in A$
  Assign a random bonded channel $k \in K$
  Broadcast a CA message
  Idle and wait for CA messages
  If receiving a CA message
    Determine the interference-free client vector $\overrightarrow{V}$
    Determine the maximum interference-free client vector $\overrightarrow{V}'$
    If assigned another channel k'
    If $\overrightarrow{V}' > \overrightarrow{V}$
      Assign k' to AP a and broadcast a CA message
    end if
  end if
  Go back to step 3
end for

addition, an AP can acquire the data rate of a client from recently received and transmitted frames. An AP can then disseminating the collected information to neighboring APs by broadcasting via the LAN interface.

3.5 Performance Evaluations

In this section, we evaluate the performance of the proposed channel assignment algorithms via simulations, and compare them with the CFAssign-Rac algorithm from [33], which outperforms most of other channel assignment algorithms in the literature. We also compare with a random channel assignment scheme. Furthermore, we implement and compare with a centralized version of the TMCA algorithm, named TMCA-C, where each AP is aware of global information and thus can estimate the overall network throughput. In the following subsections, we first introduce the network simulator and network configurations used in the simulations. We then evaluate the proposed algorithms in terms of network throughput in WLANs where all clients are stationary. After that, we further assess the network performance of the proposed algorithms in WLANs with mobile clients. Finally, we examine the impact of the number of available channels and the frequency of channel assignment on network performance.

3.5.1 Simulation Configurations

We carry out the simulations using the network simulator-3 (NS-3), which is an upgrade of the widely used network simulator (NS-2), since it has been shown in [48] that NS-3 requires less simulation runtime and memory usage than NS-2 for the same set of simulations. NS-3 already provisions MAC layer and physical layer modules for IEEE 802.11a/b/g standards. We implement the A-MPDU frame aggregation module, block ACK module, and 802.11n data rates to support 802.11n transmissions. In addition, we employ the log-distance path loss model as the propagation model, as it considers both propagation attenuation and shadow fading. The path loss of this model can be expressed as follows

$$PL = PL_0 + 10\lambda \log_{10} \frac{d}{d_0} + X_g$$

where PL is the path loss measured in Decibel (dB), d_0 is the constant reference distance, PL_0 is the path loss at the reference distance, λ is the path loss exponent, d is the transmission distance, and X_g is a Gaussian random variable representing shadow fading. Note that the path loss between a client and its associated AP is not necessarily symmetric, as the shadow fading parameter for each direction is randomly generated. In our simulations, the path loss exponent is 3, the reference distance is 1m, and the path loss at the reference distance is 46.7dB. The Gaussian random variable has a mean of zero and a standard deviation of 4dB.

In this simulations, 25 APs are deployed in a $1000 \times 1000m^2$ field. The APs are randomly scattered around a 5×5 grid such that the field is fully covered while the interference among APs is not deterministic. A number of clients are distributed in the field, and each client associates with the AP that has the highest signal strength. If not otherwise specified, all APs operate at the

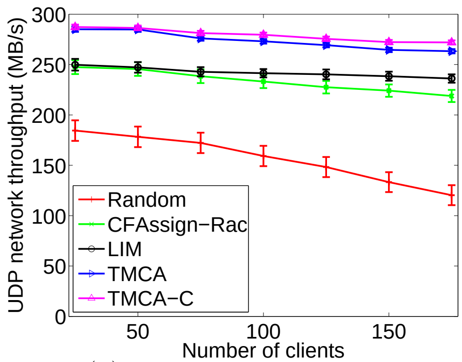
2.4GHz frequency band and there are three orthogonal 20MHz channels. In addition, all clients are static and the type of each client is randomly selected among 802.11b, 802.11g and 802.11n. The transmitting power level of all stations (APs and clients) is set to 17dBm, and the clear channel assessment threshold for carrier sense is fixed at -96dBm. It should be pointed out that NS-3 handles packet receptions and collisions totally different from NS-2. In NS-2, if the received signal strength is above a threshold and a station does not sense another transmission during a packet receiving process, the packet is regarded as successfully received. On the other hand, if another packet arrives when a station is receiving a packet, a collision is reported. In NS-3, the success of a packet reception depends on the signal to noise-plus-interference ratio (SINR), the packet length, the coding rate and the modulation scheme (physical data rate). When receiving a packet, the SINR is first determined. Then the SINR and packet length are plugged into the coding and modulation scheme to derive the packet error rate. Note that different physical data rates have different minimum SINR thresholds. Hence, we do not specify a transmission range in the simulations, as packets transmitted at low data rates require lower SINR and thus can be decoded by further stations. Furthermore, all transmissions arriving later than the current reception are regarded as interferences. Thus if the ratio of the signal strength from current reception to the signal strength of other transmissions is greater than the SINR threshold, the current reception may still succeed. Otherwise, a collision is reported. All stations utilize the rate adaptation algorithm in [46] to adapt their data rates according to packet success ratio.

In the simulations, TCP and UDP traffic are applied separately, where large file transfer is used for TCP traffic while constant bit rate (CBR) flows are used to generate saturated and unsaturated UDP traffic. For clear presentation, uplink and downlink traffic will be examined and plotted separately. If not otherwise specified, the proposed algorithms are triggered every 30 seconds. However, an AP switches its channel only if the new channel can improve local network throughput or reduce the interference-free client vector lexicographically. The impact of switching channel on network performance is very limited, as the 802.11h standard provisions a mechanism for dynamic channel switching. If an AP needs to switch channel, it broadcasts the new channel and the scheduled switching time in the following Beacon message. Then at the scheduled time, both the AP and its associated clients switch to the new channel simultaneously. The simulation is run 300 seconds for each network configuration. In the following subsections, each plotted result is the average of 100 runs. The 95% confidence interval is provisioned as well.

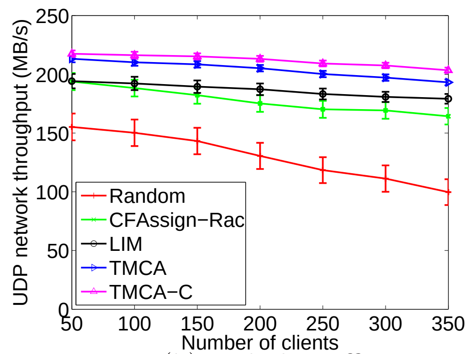
3.5.2 Performance in WLANs with Stationary Clients

In this subsection, we evaluate the proposed algorithms in terms of UDP and TCP network throughput in WLANs where all clients are stationary. We first examine the network throughput of the algorithms under various client densities. We then assess the algorithms under different traffic load. After that, we evaluate the proposed algorithms in WLANs where there are only 802.11b, 802.11g and 802.11n clients, respectively.

We first study the UDP network throughput of the proposed algorithms under different client densities, where all UDP flows are saturated and the number of clients increases from 50 to 350 in a step of 50. The simulation results for downlink and uplink traffic are plotted in Fig. 3.4(a) and Fig. 3.4(b), respectively. We can see that the UDP network throughput of all algorithms drops slowly in both traffic scenarios as the number of clients increases. This is because that all stations always have pending traffics and thus a higher number of clients results in a higher collision probability. Notably, TMCA algorithm has a great advantage over the compared algorithms in network throughput, regardless of the client density. This validates that the estimated local network throughput is a better metric for channel assignment than the number of interference-free clients. We can also observe that when the client density is low, the performance of LIM algorithm is close to the CFAssign-Rac algorithm, as both algorithms can eliminate the interference for most clients. However, as the number of clients grows, the advantage of LIM over CFAssign-Rac becomes more evident. In addition, we can see that the variation of UDP throughput for LIM is smaller than that of CFAssign-Rac algorithm, and the lower end of the confidence interval bar for LIM is higher than the higher end of the confidence interval bar for CFAssign-Rac when the number of clients is high. This can be attributed to the fact that more high-rate 802.11n clients can transmit without interference from neighboring BSSs. Furthermore, we can note that the UDP throughput of TMCA is close to the throughput of TMCA-C, indicating that our distributed implementation of the algorithm is not at the cost of significant performance degradation. The UDP throughput in the downlink traffic scenario is obviously higher than that in the uplink traffic scenario, which can be explained as that the contention level and collision probability is generally lower when only APs need to access the medium to transmit downlink traffic.



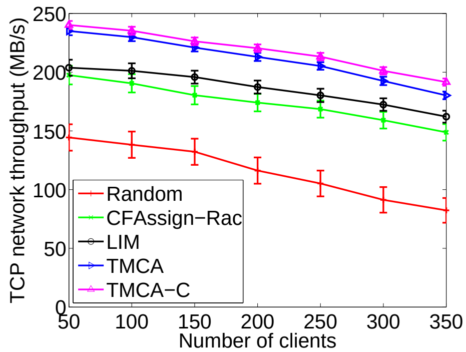
(a) Downlink traffic.



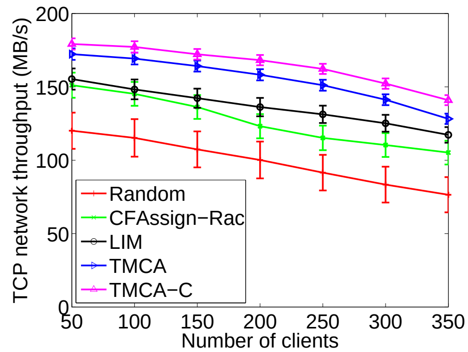
(b) Uplink traffic.

Figure 3.4: UDP network throughput of various algorithms under different numbers of clients.

Next, we examine the TCP network throughput of the proposed algorithms under different client densities. The simulation results are shown in Fig. 3.5. It can be observed that the TCP network throughput is lower than the UDP network throughput in both downlink and uplink scenarios for the following reasons. First, TCP traffic is bi-directional since the receiving end of a TCP flow needs to send TCP ACKs to guarantee reliable transmissions. Although TCP ACKs are of small sizes, the contention overhead for them at the MAC layer results in lower MAC efficiency, especially when RTS/CTS are applied to avoid collisions from legacy clients or hidden terminals. Second, the bi-directional property of TCP traffic leads to higher collision probabilities, which in turn results in frame retransmissions and may even cause timeouts for TCP ACKs. When a TCP ACK timeout occurs, the transmission rate of TCP flows is reduced in half to avoid congestions. Nevertheless, our proposed TMCA and LIM algorithms always outperform the compared algorithms, and the throughput gain is more remarkable when the client density is high. Moreover, the TCP throughput of TMCA algorithm is close to the TCP throughput of its centralized version, TMCA-C.



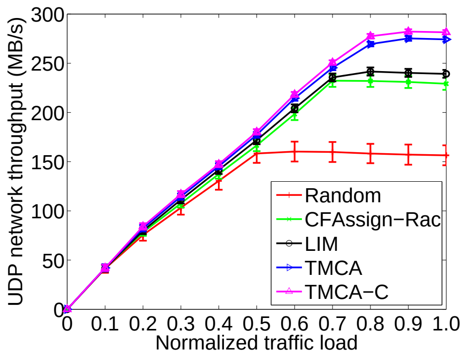
(a) Downlink traffic.



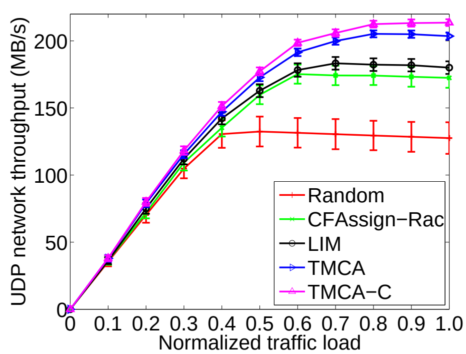
(b) Uplink traffic.

Figure 3.5: TCP network throughput of various algorithms under different numbers of clients.

We further evaluate the network throughput of the proposed algorithms under different traffic loads. We only vary the traffic load of UDP flows here, since TCP flows will automatically increase their flow rate until reaching the channel capacity. As the maximum achievable throughput of a client is related to its physical data rate, we define the *normalized traffic load* of a client as the ratio of its data rate at the transportation layer to its data rate at the physical layer. The UDP network throughput under various traffic loads is plotted in Fig. 3.6, where the number of clients is fixed at 200 and the normalized traffic load increases from 0 to 1 in a step of 0.1. When the traffic load is low, all channel assignment algorithms have generally the same UDP network throughput. This is because that although the compared Random and CFAssign-Rac algorithms may result in higher contention and interference intensity for some clients, all clients still have sufficient medium access time to fully transmit their traffic. As the traffic load increases, the UDP network throughput of all algorithms grows until the network is saturated. It is obvious that the saturation point of TMCA and LIM algorithms appears later than the compared algorithms, indicating that the proposed algorithms can lead to higher network capacity. Similar to Fig. 3.4, the UDP network throughput for downlink traffic is higher than that for uplink traffic, since the collision probability is higher when all clients need to access the medium so as to transmit uplink traffic.



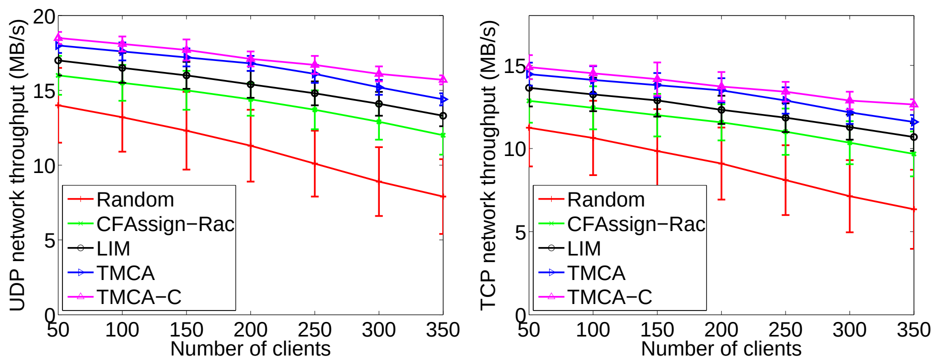
(a) Downlink traffic.



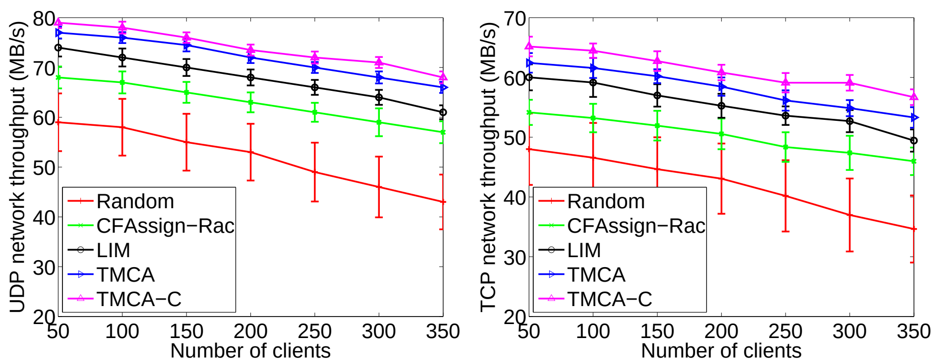
(b) Uplink traffic.

Figure 3.6: UDP network throughput of various algorithms under different traffic loads.

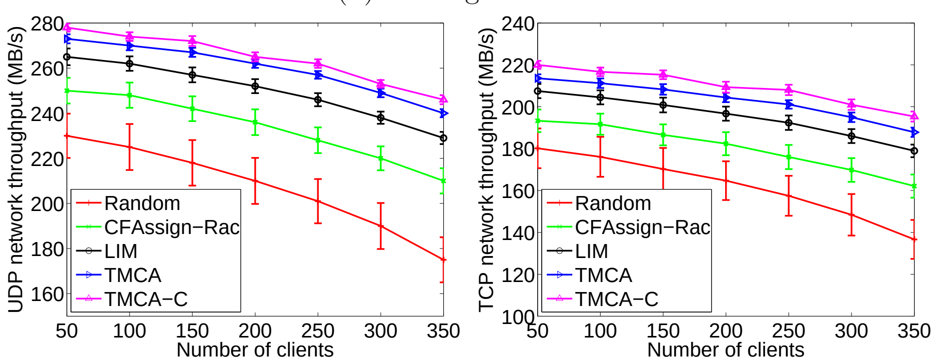
Although our algorithms are designed for WLANs with heterogeneous clients, we are also interested in their performance in WLANs where all clients are of the same type. The uplink network throughput for UDP and TCP traffics in 802.11b, 802.11g and 802.11n WLANs is shown in Fig. 3.7(a), Fig. 3.7(b) and Fig. 3.7(c), respectively. Similar to previous results, TMCA algorithm leads to the highest network throughput in 802.11b, 802.11g and 802.11n WLANs, validating our earlier statement that the estimated network throughput is a better metric than the number of interference-free clients for channel assignment. More importantly, it is evident that the performance gap between TMCA and LIM is reduced, while the advantage of LIM over CFAssign-Rac is more evident when all clients in the WLANs are of the same type. The reason behind this is that high-rate 802.11g and 802.11n clients no longer have to share the channel with low-rate 802.11b clients, neither do they need to use RTS/CTS to protect their transmissions from corruptions by 802.11b transmissions. Then the strategy of minimizing interference of high-rate clients is better reflected in the overall network throughput. These results indicate that our proposed algorithms can be applied to both heterogeneous and homogeneous WLANs.



(a) 802.11b WLANs.



(b) 802.11g WLANs.



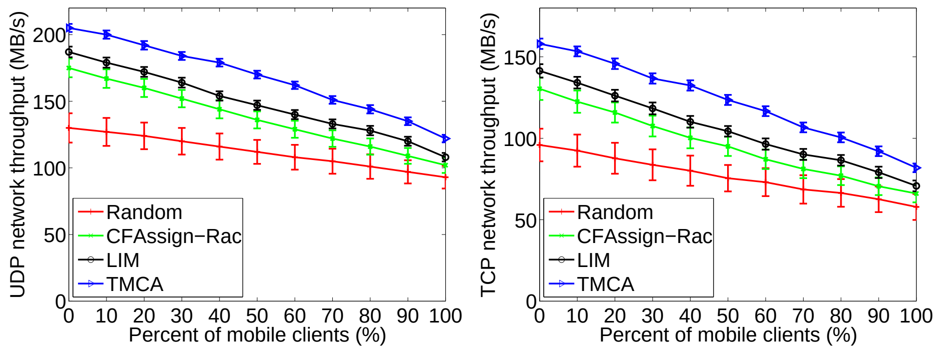
(c) 802.11n WLANs.

Figure 3.7: Uplink UDP and TCP throughput of various algorithms for 802.11b, 802.11g and 802.11n WLANs.

3.5.3 Performance in WLANs with Mobile Clients

In this subsection, we evaluate the network throughput of TMCA and LIM algorithms in WLANs with mobile clients. As mobile clients in WLANs are typically semi-static, we adopt the following mobility model: every mobile client randomly chooses a direction and moves at a constant speed v for a fixed duration t_1, and then pauses for a fixed duration t_2. The mobile client repeats this procedure until the end of the simulation. If a mobile client moves into the boundary of the field, it chooses a new direction and keeps moving until duration t_1 is reached. If not otherwise specified, the moving speed of mobile clients is 1m/s, the moving duration and pausing duration are both set to 5 seconds. Note that the network throughput of TMCA-C will not be presented in this subsection since no optimal results can be derived as the interference and data rates of mobile clients vary.

We first study the network throughput of the proposed algorithms in WLANs with different percentages of mobile clients. The uplink UDP and TCP network throughput is shown in Fig. 3.8, where the number of clients is fixed at 200 while the percent of mobile clients varies from zero to 100%. It can be observed that both UDP and TCP network throughput decreases as the percent of mobile clients increases. The reason is that when a mobile client moves to the boundary of a BSS, it has to use low data rates to transmit packets to its associated AP. In addition, the mobile client may need to disassociate with the current AP and associates with another AP that has higher signal strength. This handoff overhead also contributes to the performance degradation. It can also be noted that TCP throughput drops faster than UDP throughput as the percent of mobile clients grows, which can be attributed to the additive increasing multiplicative decreasing congestion control mechanism of TCP. As a client moves away from its associated AP, its TCP throughput decreases in a multiplicative manner if several packets are dropped due to deteriorating channel conditions. As a client moves closer to its associated AP, its TCP throughput increases in an additive manner even if a higher data rate has already been chosen at the physical layer. It is notable that the advantage of TMCA and LIM to the compared algorithms is relatively small when the percent of mobile clients is high. This is because that the interference relationship and optimal physical data rates of mobile clients change when they move around the field. Then the channel assignment derived from TMCA and LIM become out-of-date sometimes, as the proposed algorithms are performed every 30 seconds, while mobile clients move every 5 seconds. We will further study the impact of the frequency of performing channel assignment on network throughput in next subsection.

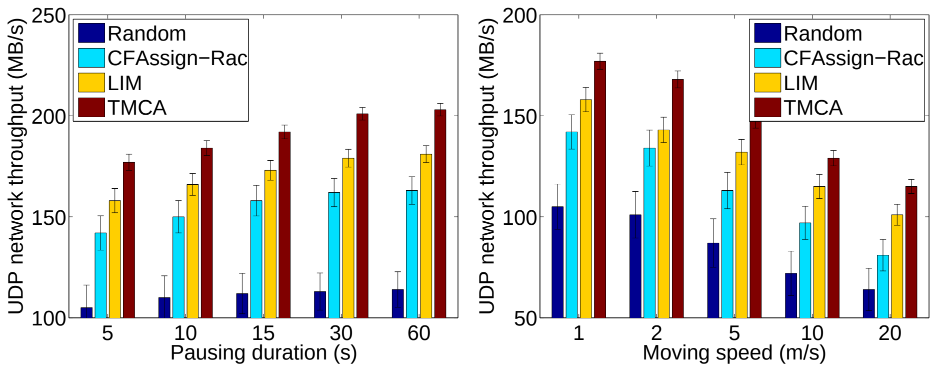


(a) UDP network throughput. (b) TCP network throughput.

Figure 3.8: Uplink network throughput of various algorithms in WLANs with various percentages of mobile clients.

Next, we assess the proposed algorithms in WLANs where mobile clients have different pausing durations and moving speeds. In this set of simulation, the number of clients is 200 and the percent of mobile clients is fixed at 50%. From Fig. 3.9 (a), we can see that as the pausing duration t_2 of mobile clients increases from 5 seconds to 30 seconds, the uplink UDP throughput of all algorithms grows for the following reasons. First, the employed rate adaption algorithm can probe more data rates and choose the optimal data rate for mobile clients when they pause at a location for a long time. More importantly, the interference and data rates information collected by TMCA, LIM and CFAssign-Rac algorithms stays valid for a longer time when the pausing duration is long. Accordingly, their derived channel assignments are effective for a longer time, which is validated by the observation that the network throughput of TMCA, LIM and CFAssign-Rac grows faster than the throughput of Random algorithm as the pausing duration increases. We also examine the proposed channel assignment algorithms by varying the moving speed of mobile clients from $1m/s$ to $20m/s$. We can note that the uplink

UDP throughput of all algorithms decreases drastically as the moving speed increases, which can be attributed to multiple factors: AP association, rate adaptation and channel assignment. Different from 3G/4G cellular networks, WLANs are mostly used indoor and each AP usually has a very limited coverage area. Thus when a mobile client moves at a high speed, the rate adaption algorithm can hardly use high data rates due to rapidly varying channel conditions. In addition, the handoff overhead could be substantial since the client roams among multiple BSSs frequently. Moreover, the interference information used by channel assignment algorithms will be affected adversely as well. Nevertheless, such high-speed moving scenarios are rarely observed in WLANs, as clients are usually static or move at a very low speed relatively from the perspective of the AP, even in trains or airplanes that provision WLAN access.



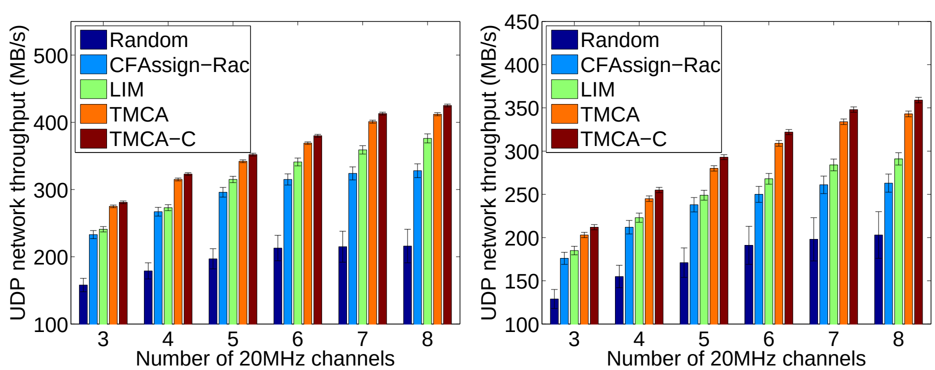
(a) Throughput vs. Pausing duration. (b) Throughput vs. Moving speed.

Figure 3.9: Uplink UDP throughput of various algorithms given different pausing durations and moving speeds for mobile clients.

3.5.4 Impact of System Parameters on Network Performance

In this subsection, we further assess the impact of critical system parameters on network performance. We first evaluate the proposed algorithms given different number of available channels. We then examine the network throughput of WLANs when the proposed algorithms are executed in different frequencies.

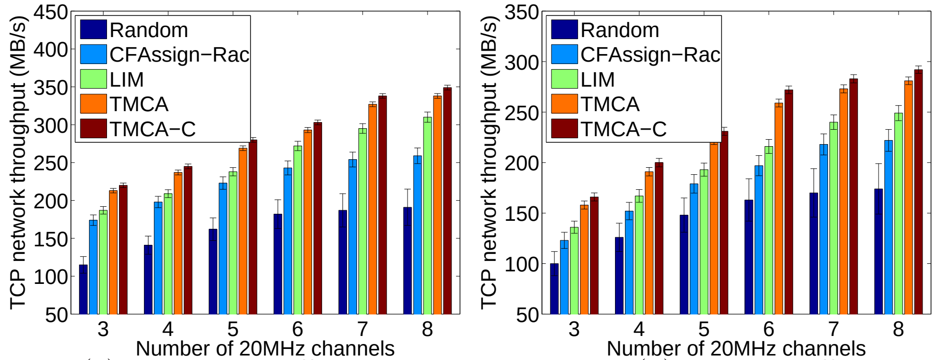
We first study the impact of the number of non-overlapping 20MHz channels on network throughput. The simulation is conducted at the 5GHz frequency band, which provisions much more non-overlapping 20MHz channels compared to the 2.4GHz frequency band. Hence each client in the WLAN is either an 802.11a or an 802.11n client. The UDP network throughput and TCP network throughput under different numbers of available channels are shown in Fig. 3.10 and Fig. 3.11 respectively, where the number of clients is fixed at 200 while the number of non-overlapping 20MHz channels varies from 3 to 8. It can be observed that the UDP and TCP throughput of TMCA, LIM and CFAssign-Rac algorithms increases notably as the number of 20MHz channels grows from 3 to 6. However, when the number of channels grows beyond 6, the throughput improvement of the CFAssign-Rac algorithm becomes marginal. The reason is that most co-channel interference among clients in the simulations is already eliminated when there are 6 channels, while CFAssign-Rac does not assign bonded channels to BSSs to enable high data rates for 802.11n clients. On the contrary, the proposed TMCA and LIM algorithms can further improve the network throughput by taking advantage of channel bonding in certain BSSs, without causing interference to nearby BSSs. Nevertheless, the improvement on network throughput of the proposed algorithms is relatively small when the number of channels increases from 7 to 8, which validates our earlier discussion that the benefits of channel bonding in 802.11n WLANs are limited by the coexistence of legacy clients.



(a) Downlink traffic.   (b) Uplink traffic.

Figure 3.10: UDP network throughput of various algorithms given different numbers of non-overlapping 20MHz channels.

After that, we evaluate the impact of the frequency of channel assignment on network performance. Note that the compared CFAssign-Rac algorithm

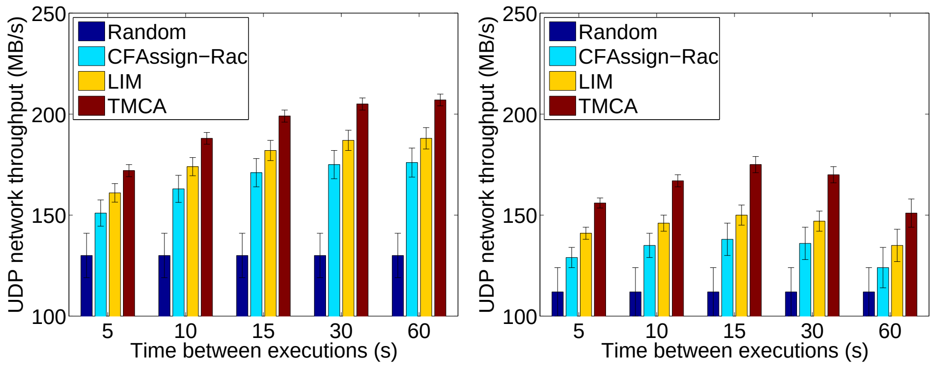


(a) Downlink traffic. (b) Uplink traffic.

Figure 3.11: TCP network throughput of various algorithms given different numbers of non-overlapping 20MHz channels.

also has to be executed periodically to maintain updated interference information. The uplink UDP throughput in WLANs with stationary clients is presented in Fig. 3.12 (a), where the number of clients is fixed at 200 while the channel assignment algorithms are performed every 5s, 10s, 15s, 30s and 60s, respectively. We can see that the network throughput of Random algorithm seldom changes with the channel assignment frequency, since the algorithm does not have extra coordination or measurement overhead, while the delay of switching channels inside a BSS is negligible thanks to the 802.11h. On the contrary, the network throughput of all other algorithms decreases when the channel assignment frequency increases. This is because that when all clients are stationary, the interference and channel conditions of all clients are relatively stable. Then the network throughput will be affected if channel assignments are executed very frequently, as a substantial amount of medium access time is occupied by the measurement frames. Moreover, clients also need to send frames to their associated APs to report their uplink traffic load if the TMCA algorithm is employed.

We also examine the impact of channel assignment frequency on network throughput in WLANs with mobile clients. The simulation results are plotted in Fig. 3.12(b), where there are 50% of mobile clients, and the moving speed, moving duration and pausing duration is $1m/s$, 5s and 5s, respectively. Similar to Fig. 3.12 (a), the network throughput of the Random algorithm does not change along with the channel assignment frequency for the same reason. We can clearly observe a tradeoff between the benefits of refreshing channel



(a) WLANs with stationary clients.

(b) WLANs with mobile clients.

Figure 3.12: Impact of the frequency of channel assignment on uplink network throughput.

assignment frequently and the cost of measuring and reporting overhead. The network throughput increases when the duration between two consecutive executions of the channel assignment algorithms grows from 5s to 15s. This is because that although interference information of some mobile clients may become stale when the duration is long, the performance degradation caused by inaccurate interference information is outweighed by the reduction of measurement overhead. However, when the duration grows beyond 15s, the network throughput begins to decrease, indicating the importance of maintaining fresh interference and data rate information. In practice, the optimal channel assignment frequency also depends on the client density, the percentage of mobile clients, the mobility model of the mobile clients, etc.

From above simulation results, we can see that the TMCA algorithm always leads to the highest network throughput, regardless of the type and direction of traffic. The LIM algorithm also performs better than the compared algorithms and can bring satisfactory network throughput, especially when there are sufficient non-overlapping channels. Moreover, the LIM algorithm has a lower complexity and is easy to be deployed. Therefore, both algorithms can be deployed in the same WLAN, and the best one can be activated according to the traffic load, the available channel resources and the feasibility of installing the TMCA procedure on clients.

3.6 Conclusions

In this chapter, we have studied channel assignment in 802.11n WLANs with heterogeneous clients, focusing on the benefits and challenges of new features in 802.11n, such as channel bonding, frame aggregation, and the impact of conventional 802.11a/b/g clients. We first presented a network model to estimate the uplink and downlink throughput of each client. We then formulated the channel assignment problem into an optimization problem, with the objective to maximizing overall network throughput. As the optimization problem is NP-hard, we proposed a distributed algorithm for channel assignment, where each AP updates its channel iteratively to improve the estimated local network throughput. After that, we proposed another channel assignment algorithm, in which APs try to minimize the interference experienced by high-rate 802.11n clients. Finally, we conducted extensive simulations to evaluate the proposed algorithms. Simulation results have shown that our proposed algorithms can significantly improve UDP and TCP network throughput under various network scenarios, compared with other channel assignment algorithms.

Chapter 4

A Cooperative Retransmission Protocol for 802.11n WLANs

4.1 Introduction

In this chapter, we propose a cooperative aggregated retransmission MAC (CAR-MAC) protocol for 802.11n WLANs. We first introduce a distributed scheme to dynamically select a cooperative node for each pair of sender and destination. In this scheme, all nodes periodically broadcast a *C-Beacon* message to neighbors, including overhearing capability, packet error rate, data rate for transmissions, etc. Senders choose a cooperative node for each destination based on received C-Beacon messages. In CAR-MAC protocol, a sender specifies its cooperative node explicitly when transmitting an aggregated frame. The selected cooperative node overhears the aggregated frame and the following block ACK frame from the destination to determine whether it is necessary to help retransmit corrupted sub-frames in the aggregated frame. The destination acknowledges an aggregated frame with a block ACK frame, including the receiving status of all sub-frames. If some sub-frames in the aggregated frame fail to reach the destination, the cooperative node aggregates the failed sub-frames into a new frame and retransmits it to the destination. The contributions of CAR-MAC are in three-fold. First, CAR-MAC takes advantage of frame aggregation and block ACK mechanisms of 802.11n for cooperative retransmissions and is fully compatible with the 802.11n standard. Second, CAR-MAC does not intensify collisions in the network, as cooperative nodes retransmits frames only after receiving a block ACK frame, which indicates that the failed sub-frames are corrupted by channel errors rather than collisions. Third, the overhead of CAR-MAC is negligible, especially in WLANs with heavy load, as cooperative nodes can aggregate retransmitting sub-frames together with their own data frames to the destination, such that the overhead of cooperative retransmissions is amortized. We theoretically analyze the improvement on network throughput brought by CAR-MAC. We also conduct extensive simulations to evaluate CAR-MAC under various channel conditions. Both numerical and simulation results show that CAR-MAC protocol can significantly improve network performance, compared with the 802.11n standard and previous cooperative retransmission schemes.

The remainder of the chapter is organized as follows. Section 4.2 briefly reviews previous cooperative communication schemes for WLANs, and gives an overview of new features in the IEEE 802.11n standard. Section 4.3 describes the proposed scheme for cooperative node selection and the CAR-MAC protocol in detail. Section 4.4 analyzes the performance of CAR-MAC theoretically. Section 4.5 provides the simulation results, followed by the conclusions in Section 4.6.

4.2 Related Work

Cooperative communications have exhibited great potential in improving the spectrum efficiency as well as reliability of wireless networks by exploring the broadcast nature and spatial diversity of wireless medium [49]. In [50], several cooperative strategies, including amplify-and-forward, decode-and-forward and compress-and-forward, were outlined, and the outage probabilities of these strategies were analyzed theoretically. Cooperative nodes can participate in the communication between a sender and a destination by either persistently relaying received frames to the destination, or helping retransmit a frame only if the direct transmission fails. Several cooperative relaying schemes have been presented in [51–54] to enhance the spectrum efficiency. However, these schemes cannot guarantee the reliability of data transmissions.

On the other hand, cooperative retransmission can improve the spectrum efficiency as well as the reliability of wireless communications. It is assumed that cooperative nodes can overhear the original transmission thus it is unnecessary to transmit the frame to the cooperative node explicitly. In some cooperative retransmission schemes, only one cooperative node is selected to help retransmit failed frames. In [55], a cooperative node is preselected for each sender-destination pair. The cooperative node retransmits the frame if an ACK timeout occurs. In the CD-MAC proposed in [56], each sender selects its cooperative node based on the received signal strength from neighbors. If the direct transmission fails, the sender and the cooperative node retransmit the frame simultaneously using the distributed space time coding (DSTC) scheme. In [57], neighbors of a sender compete to help retransmit a failed frame. Each neighbor determines its backoff duration based on the overhearing link quality and cancels its retransmission attempt after another neighbor begins retransmission. A decentralized partially observable Markov decision process (DEC-POMDP) model was given in [58] for selecting cooperative node, given imperfect channel state information.

In some cooperative retransmission schemes, multiple neighbors of the sender cooperatively retransmit a failed packet to further explore spatial diversity [59–61]. In [59], a cooperative group is preselected for each sender-destination pair. If a negative acknowledge is received from the destination, both the sender and all nodes in the cooperative group retransmit the packet simultaneously using DSTC. A cooperative retransmitting scheme was proposed for enhanced distributed channel access (EDCA)-enabled WLANs in [60]. In this scheme, a higher priority cooperative queue is maintained for every EDCA access class at each node. Each node caches the packets overheard from neighbors in these cooperative queues, and compete the medium to help retransmit a packet if the direct transmission fails. In [61], multiple cooperative nodes set up different retransmitting backoff durations according to their link qualities. Each cooperative node retransmits the packet when its backoff timer expires, until the packet is received by the destination. However, most diversity gains can be typically achieved with only one or two cooperative nodes. Thus the coordination overhead among multiple cooperative nodes may not be paid off by the benefits achieved.

Cooperative retransmissions can also be jointly explored with other techniques to mitigate the cooperation overhead. In [62] and [63], cooperative nodes help retransmit a frame even if they cannot fully receive the frame. The destination combines frames from both the sender and cooperative nodes to successfully decode the frame. Forward error correction (FEC) and cooperative retransmission are applied together in [64], where the cooperative nodes help calculate and transmit the parity bits of a packet gradually until the destination successfully decodes the packet. In [65], a data frame is divided into blocks and cooperative nodes help retransmit the failed blocks. However, the scheme is incompatible with the 802.11 standard and did not consider how to select and coordinate the cooperative nodes.

4.3 A Cooperative Aggregated Retransmission MAC Protocol

In this section, we present an efficient cooperative aggregated retransmission MAC (CAR-MAC) protocol for 802.11n WLANs. In CAR-MAC, each node selects only one cooperative node to help retransmit failed sub-frames if necessary. We first introduce a distributed scheme to dynamically select a cooperative node for all sender-destination pairs in the network. We then describe the CAR-MAC protocol in depth.

4.3.1 Selection of Cooperative Nodes

In CAR-MAC, each sender explicitly specifies a cooperative node to eliminate potential collisions among multiple cooperative retransmissions. In addition, a sender piggybacks the address of the cooperative node in the original data frame, so as to minimize the coordination overhead between the sender and the cooperative node. As shown in Fig. 4.1, multiple neighbors may be qualified to cooperatively retransmit failed sub-frames for a sender-destination pair at a time. However, due to the time-varying property of the wireless medium, no neighbor can always overhear the transmission from the sender and successfully deliver the retransmission. Thus, every time the neighbor that leads to the best overall network performance should be selected as the cooperative node.

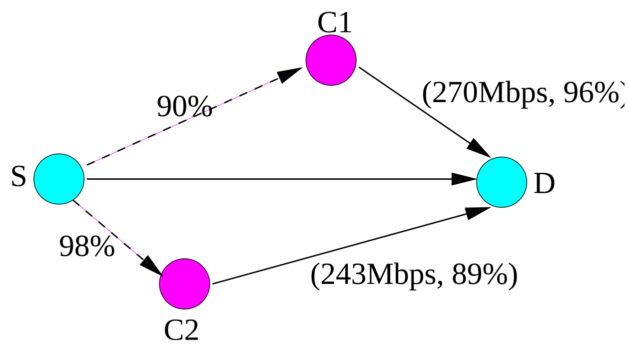


Figure 4.1: An example of selecting a cooperative node for a sender-destination pair in a wireless network. S, D, C1 and C2 denote the sender, destination, candidate 1 and candidate 2 of the cooperative node, respectively. Numbers above dashed lines represent the overhearing ratio. Numbers above solid lines stand for the physical data rate and sub-frame success ratio.

Multiple factors need to be considered when selecting a cooperative node from neighbors for a sender-destination pair. We first need to consider the *overhearing ratio* of a neighbor, which is defined as the ratio of the number of sub-frames that are successfully decoded by the neighbor, to the number of sub-frames that are included in all aggregated frames overheard by the neighbor. The overhearing ratio is a good metric to evaluate the channel quality between the sender and the neighbor. Second, we should consider the *sub-frame success ratio* P_s and the *collision ratio* P_c for data transmissions from the neighbor to the destination, as they reflect the channel quality between the neighbor and the destination. The sub-frame success ratio and collision ratio are discussed separately since the former is caused by channel errors while the later comes from collisions. A neighbor of high collision probability should not be selected as the cooperative node because all cooperatively retransmitted sub-frames by such a node would fail when collision occurs. Third, the adapted data rate R from the neighbor to the destination should be considered as well because it is highly related to the efficiency of the retransmission. We also consider the channel utilization ratio U sensed by the neighbor, to avoid intensifying the contention level of a heavily loaded neighborhood. We define the aforementioned factors as the *cooperative capabilities* of a node and have each node maintain an entry for all these capabilities.

In CAR-MAC, every node broadcasts a *C-Beacon* message periodically to notify the neighborhood of its cooperative capabilities. Similar to the *Beacon* message of 802.11 standard, C-Beacon is composed of information elements. The C-Beacon message has two types of information elements: *utilization element* and *cooperative neighbor element*. The utilization element includes the channel utilization ratio of the node. A node maintains a cooperative entry for each neighbor that it is willing to help retransmission. This entry includes the overhearing ratio from that neighbor, as well as the sub-frame success ratio, collision ratio and physical data rates of transmissions to that neighbor. The cooperative neighbor element for a node contains the cooperative entries of all neighbors. The overhead of C-Beacon messages is quite small for two reasons. First, C-Beacon messages are broadcast at a high data rate since they only need to reach neighbors with good channel qualities. Second, each node broadcasts a C-Beacon message every few seconds so as to reflect the change of channel conditions.

A sender determines its cooperative node based on the C-Beacon messages received from all neighbors. The neighbor that has the highest probability to successfully retransmit a message should be selected as the cooperative node. Also, as discussed earlier, nodes in a heavy traffic neighborhood should not be selected as cooperative nodes. Thus we define the rank for neighbor i as follows

$$Rank(i) = \frac{P_o(i) \cdot P_s(i) \cdot (1 - P_c(i))}{U(i)} \cdot \frac{R_i}{R_{max}} \tag{4.1}$$

where R_{max} is the maximum physical data rate used by all neighbors of the sender. The neighbor with the highest rank is selected as the cooperative node.

4.3.2 Description of CAR-MAC Protocol

In this subsection, we describe the cooperative aggregated retransmission MAC protocol in detail. In CAR-MAC, all nodes access the medium following the DCF scheme in 802.11 standard. Once obtaining the medium access, a sender aggregates multiple data packets into one large frame and transmits it to the destination. The sender also specifies the selected cooperative node in each sub-frame of the aggregated frame. The destination replies to the aggregated frame with a block ACK frame after a short inter-frame slot (SIFS) time. An overhearing neighbor caches the aggregated frame if it is the designated cooperative node; otherwise, it drops the frame after updating its overhearing

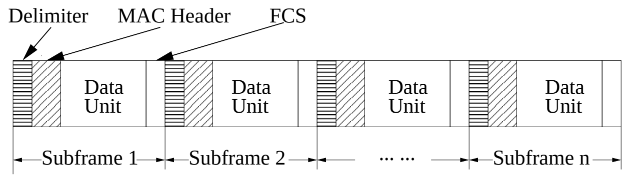


Figure 4.2: Structure of an aggregated frame in 802.11n WLANs.

ratio. If the block ACK frame implies that not all sub-frames are successfully received by the destination, the cooperative node aggregates all the failed sub-frames into a new frame, and retransmits it after a SIFS time. The destination replies to the cooperatively retransmitted frame with a *cooperative block ACK* frame. If all sub-frames retransmitted by the cooperative node are received by the destination, the sender moves on to subsequent data units. Otherwise, the sender retransmits the failed sub-frames after obtaining the medium access.

Different from most cooperative retransmission schemes in the literature, CAR-MAC does not intensify the collision level of the wireless network. First, only one cooperative node would help retransmit failed sub-frames thus there is no collision among multiple cooperative retransmissions. Second, the cooperative retransmission does not collide with the retransmission from the sender either, since the sender retransmits only if the cooperative retransmission fails and needs to compete for the medium. Third, the cooperative node will retransmit only if the transmission failure is caused by channel errors, but not by collisions. If a transmission failure is caused by channel errors, it is highly possible that the destination can still decode the MAC header of some sub-frames and then replies with a block ACK frame. In contrast, collisions often corrupt entire aggregated frames. Then both the sender and the cooperative node will observe a block ACK timeout. As the cooperative node retransmits only after receiving a block ACK, it will not help retransmit a frame corrupted by collisions.

CAR-MAC is compatible with the IEEE 802.11n standard. CAR-MAC uses the aggregated MAC protocol data unit (A-MPDU) scheme of 802.11n for frame aggregation. The data structure of an aggregated data frame is given in Fig. 4.2. We can see that every sub-frame in A-MPDU has its own MAC header, staring with a sub-frame delimiter. In the delimiter ahead of each sub-

2	2	6	6	2	10	4Bytes
Frame Control	Duration /ID	RA	TA	BA Control	BA Information	FCS

(a) Standard block ACK

2	2	6	6	2	10	6	10	4Bytes
Frame Control	Duration /ID	RA	TA	BA Control	BA Info	CA	Coop BA Info	FCS

(b) Cooperative block ACK

Figure 4.3: Data formats of a standard block ACK and a cooperative block ACK.

frame there are four reserved bits. CAR-MAC use these reserved bits to denote the cooperative type of a sub-frame. Moreover, there are four address fields in the MAC header of a sub-frame and the fourth address is unused in most scenarios. In CAR-MAC, the sender uses the fourth address field to specify the cooperative node of the aggregated frame. Moreover, the cooperative node uses this address field to specify the original sender of cooperatively retransmitted sub-frames.

To acknowledge the sub-frames retransmitted by cooperative nodes, we extend the block ACK frame of 802.11n to a cooperative block ACK frame. The data format of a standard block ACK frame is given in Fig. 4.3(a), in which RA denotes the MAC address of the receiver, TA denotes the MAC address of the transmitter, BA control is a 2-byte control field for block ACK, and the BA information field includes the starting sequence and the receiving bitmap of the aggregated frame. Similar to the delimiter in the aggregated frame, there are also reserved bits in the BA control field of block ACK. We extend these bits to distinguish a standard block ACK and a cooperative block ACK. In addition, we also introduce two new fields: a cooperative node address (CA) field and a cooperative information field. The format of a cooperative block ACK frame is shown in Fig. 4.3(b). In this way, the destination is able to simultaneously acknowledge the sub-frames of the senders retransmitted cooperatively and the new sub-frames of the cooperative node. These extensions do not alter the main structure of the aggregated frame and the block ACK frame. As a result, conventional 802.11n devices can coexist with CAR-MAC.

Cooperative retransmissions in CAR-MAC can be categorized into five cases: 1) Direct transmission succeeds; 2) Cooperative retransmission succeeds; 3) Cooperative retransmission with new sub-frames succeeds; 4) Cooperative retransmission fails; 5) Collision occurs. The data diagrams for all these cases are illustrated in Fig. 4.4, where S, C and D denote the sender, the cooperative node and the destination, respectively. We discuss these cases separately next.

Case 1: Both the destination and the cooperative node receive all sub-frames successfully, and the destination receives the block ACK frame. The cooperative node drops the cached sub-frames regardless whether it receives the block ACK frame or not. If a block ACK frame is received, the cooperative node knows that the destination has received all sub-frames. Otherwise, the cooperative node assumes that the destination fails to receive the aggregated frame due to collisions. As discussed earlier, the cooperative node does not retransmit cooperatively to avoid escalating the collision.

Case 2: The destination receives part of the aggregated data frame, and the cooperative node overhears all the sub-frames. In addition, both the sender and the cooperative node receive the block ACK. There is no pending traffic at the cooperative node. Thus the cooperative node aggregates and retransmits the failed sub-frames on behalf of the sender after a SIFS time. After receiving the retransmitted sub-frames, the destination replies with a cooperative block ACK frame.Both the sender and the cooperative node proceed to new transmission after receiving the cooperative block ACK frame.

Case 3: This case is similar to Case 2 except that the cooperative node also has pending traffic in its queue. To improve MAC efficiency, the cooperative node aggregates sub-frames to be retransmitted for the sender and its own new sub-frames. After receiving this mixed aggregated frame, the destination replies with a cooperative block ACK frame, acknowledging both the retransmitted sub-frames and original sub-frames from the cooperative node.

Case 4: In this case, the cooperative node also tries to retransmit failed sub-frames after receiving the block ACK frame. Nevertheless, the retransmitted frame may not be received by the destination due to channel errors or collisions. It is also possible that although all retransmitted sub-frames are received by the destination, the sender does not receive the cooperative block ACK. Since a cooperative block ACK is not received on time, the sender assumes that the cooperative node fails to retransmit the failed sub-frames. Thus the sender needs to retransmit the failed sub-frames. Note that the sender does not need to increase its contention window exponentially since the block ACK frame received earlier implies that the transmission failure is not caused by collisions.

Case 5: The destination receives some sub-frames of the aggregated frame and sends back a block ACK frame. The cooperative node receives the block ACK successfully but the sender fails to receive it. Thus the cooperative node retransmits the failed sub-frames and the destination replies with a cooperative block ACK. If the sender receives the cooperative block ACK, it realizes that it has missed a block ACK and proceed to transmit new data units. Otherwise, the sender assumes that the aggregated frame is dropped due to collisions. Thus it doubles its contention window and retransmits the entire aggregated frame after obtaining medium access.

To avoid being collided by potential hidden terminals, the cooperative node may exchange RTS/CTS control frames with the destination before cooperative retransmissions. As shown in Fig. 4.4(f), the cooperative node sends a RTS frame to the destination after a SIFS time from the block ACK frame, while the destination node replies to the RTS frame with a CTS frame. After that, the cooperative node retransmits the sub-frames as specified by CAR-MAC. Note that no modification to the RTS/CTS mechanism is needed.

From the above descriptions, we can see that CAR-MAC is simple but efficient in retransmitting failed sub-frames cooperatively. It is also robust to various error conditions since no complex state transition is introduced at the sender and the cooperative node. Note that cooperative nodes can adapt the ratio of cooperatively retransmitting traffic and their own traffic by dynamically adjusting the maximal number of retransmitting sub-frames in an aggregated frame. However, the topic is out of the scope of this chapter and will be investigated in our future work.

4.4 Performance Analysis

In this section, we analyze the performance of the proposed CAR-MAC protocol and compare it with the standard 802.11n transmissions. We will first

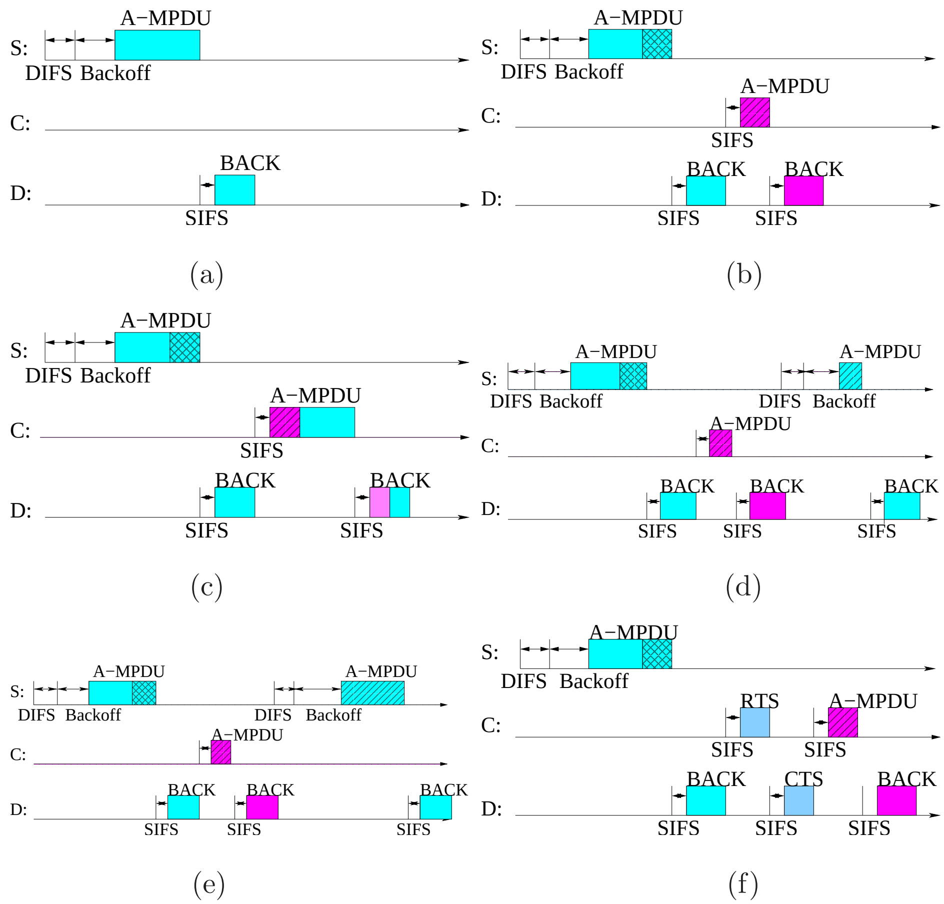


Figure 4.4: Various cases of cooperative retransmissions in CAR-MAC, where S, C and D denote the sender, cooperative node and the destination, respectively. Pink blocks represent cooperatively retransmitted sub-frames and cooperative block ACK, while blue blocks represent direct transmission and block ACK. (a) Case 1: Direct transmission succeeds. (b) Case 2: Cooperative retransmission succeeds. (c) Case 3: Cooperative retransmission with new sub-frames succeeds. (d) Case 4: Cooperative retransmission fails. (e) Case 5: Collision occurs. (f) RTS/CTS protects cooperative retransmission.

derive the approximate throughput of CAR-MAC on independent and identically distributed (i.i.d.) channels based on a Markov model. We will also give numerical results about the improvement on network throughput by CAR-MAC.

4.4.1 Throughput Analysis of CAR-MAC

In this subsection, we model the throughput of CAR-MAC using a bi-dimensional Markov model. A Markov model was proposed in [26] for conventional multi-rate WLANs, where nodes may have different traffic loads, physical data rates and payload sizes. However, in this model, transmission failures caused by channel errors cannot be distinguished from failures resulted from collisions, making it inapplicable to CAR-MAC. We extend this model to support the frame aggregation and block ACK features of 802.11n to describe the throughput of CAR-MAC. For simplicity, we assume all nodes are within the carrier sense range of each other and there is no hidden terminal in the network. Based on this assumption, the contention behavior of CAR-MAC is identical to the MAC protocol of 802.11n. This is because that both the cooperative retransmission frame and cooperative block ACK frame introduced in CAR-MAC are transmitted after a SIFS interval from the block ACK frame, making it unnecessary to compete for the medium before transmitting them. Thus they have no impact on the contention behavior of the WLAN.

According to the Markov model, every node in the WLAN has a stationary transmission probability and a stationary collision probability. We further assume that all nodes in the WLAN always have pending traffic to transmit. Hence the transmission probabilities and the collision probabilities of all nodes are identical to each other. Using the chain transition probability given in [26], the transmission probability τ for a node can be expressed as

$$\tau = \frac{2(1-2P_c)}{(1-2P_c)(W_0+1)+P_cW_0(1-(2P_c)^m)} \tag{4.2}$$

where P_c, W_0 and m stand for the collision probability, the initial contention window size and the number of back off stages, respectively.

Let the number of nodes in the WLAN be N, then the collision probability for a transmitting node is equivalent to the probability that as least another node transmits simultaneously, that is,

$$P_c = 1 - (1 - \tau)^{N-1} \tag{4.3}$$

For an aggregated frame transmitted by any node i, we assume that the channel errors for all sub-frames satisfy independent and identical distribution and define the sub-frame error probability as $P_e(i)$. Then the transmission of an aggregated frame can be described as a series of Bernoulli trials. We further define the average aggregation level for node i as $A(i)$. Therefore, when no collision occurs, the expected number of successfully received sub-frames $E^D_{avg}(i)$ for a direct transmission is

$$E^D_{avg}(i) = A(i) \cdot (1 - P_e(i)) \tag{4.4}$$

On the other hand, in CAR-MAC the cooperative node retransmits all failed sub-frames caused by channel errors. It is reasonable to assume that these retransmitted sub-frames can also be described as a series of Bernoulli trials. We denote the sub-frame error probability for the cooperative node of sender i as $P^C_e(i)$. Then the expected number of received sub-frames $E^C_{avg}(i)$ for a transmission with cooperative retransmission is equivalent to the summation of the number of sub-frames delivered by the sender and the number of sub-frames delivered by the cooperative node, that is,

$$E^C_{avg}(i) = E^D_{avg}(i) + (A(i) - E^D_{avg}(i)) \cdot (1 - P^C_e(i)) \tag{4.5}$$

If cooperative retransmission is not adopted, the theoretical throughput for node i can be expressed as the length of successfully delivered payload from the direct transmission divided by the average duration of a time slot T^D_{avg} for all nodes in the network. Thus throughput $S^D(i)$ for node i can be represented as

$$S^D(i) = \frac{(1 - P_c) \cdot E^D_{avg}(i) \cdot L}{T^D_{avg}} \tag{4.6}$$

where L is the average length of a sub-frame.

Similarly, the theoretical throughput $S^C(i)$ for node i in CAR-MAC can be defined as the length of payload delivered by both the sender and the cooperative node, divided by the average duration of a time slot for all nodes in CAR-MAC

$$S^C(i) = \frac{(1 - P_c) \cdot E^C_{avg}(i) \cdot L}{T^C_{avg}} \tag{4.7}$$

The average time slot duration for both CAR-MAC and standard 802.11n MAC can be further expressed as the summation of four expected time slots

$$T_{avg} = T_{idle} + T_{succ} + T_{col} + T_{error} \tag{4.8}$$

where T_{idle}, T_{succ}, T_{col} and T_{error} stand for the expected durations of an idle time slot, an aggregated frame that is fully received, a transmission failure due to collisions and an aggregated frame in which some sub-frames are corrupted by channel errors, respectively. As aforementioned, CAR-MAC does not affect the collision probability and transmission probability of nodes. Thus T_{idle} of CAR-MAC equals the T_{idle} of standard 802.11n MAC. Moreover, as the cooperative node does not retransmit in cases of successful transmissions and collisions, the T_{succ} and T_{col} of CAR-MAC are also identical to those of standard 802.11n MAC.

The expected duration of an idle time slot can be defined as the product of the probability that no node is transmitting and the duration of a back off time slot, that is,

$$T_{idle} = (1 - \tau)^N \cdot \delta$$

where δ denotes the duration of a back off time slot.

The duration of a successful transmission includes a DIFS interval, the time needed to transmit the aggregated frame, a SIFS interval and the time to transmit the block ACK frame. The time of the aggregated frame can be further divided into two parts: physical layer convergence protocol (PLCP) header and the transmission of the data payload. Moreover, the transmission time of the data payload is related to the aggregation level and the physical data rate. Putting all these together, the duration $T_{succ}(i)$ of a successful transmission for node i can be formally expressed as

$$T_{succ}(i) = T_{difs} + T_{sifs} + T_{plcp} + T_{back} + \frac{A(i) \cdot L}{R(i)}$$

in which T_{difs}, T_{sifs}, T_{plcp}, T_{back} and $R(i)$ represent the DIFS interval, the SIFS interval, the duration for a PLCP header, the duration of a block ACK frame and the physical data rate for node i, respectively.

Accordingly, the expected duration of a successful transmission in the WLAN is equal to the summation of the products of successful transmission probability and transmission duration for all nodes. The successful transmission probability for node i can be rewritten as the probability that only node i transmits and all sub-frames are successfully received. Thus T_{succ} can be given by

$$T_{succ} = \sum_{i=1}^{N} \tau(1-\tau)^{N-1}(1-P_e(i))^{A(i)} \cdot T_{succ}(i)$$

For standard 802.11n MAC, the duration of an erroneous transmission is the same as that of a successful transmission, since they are both acknowledged by a block ACK frame. We define $T^D_{error}(i)$ as the duration of an erroneous frame without cooperative retransmission for node i, thus

$$T^D_{error}(i) = T_{succ}(i)$$

For CAR-MAC, besides all the time components in $T^D_{error}(i)$, the duration of an erroneous transmission also includes the cooperative retransmitted frame, the cooperative block ACK and the two SIFS intervals between these frames. Thus, the duration of an erroneous transmission for node i in CAR-MAC, $T^C_{error}(i)$, is given as follows

$$\begin{aligned} T^C_{error}(i) &= T^D_{error}(i) + 2T_{sifs} \\ &+ T_{back} + T_{plcp} + \frac{(E^C_{avg}(i) - E^D_{avg}(i))L}{R^C(i)} \end{aligned}$$

Similar to the expected duration of successful transmissions, the expected duration of erroneous transmissions can be represented as the summation of the products of erroneous transmission probability due to channel error and

$T_{error}(i)$ for all nodes. The erroneous transmission probability for node i equals the probability that only node i transmits and at least one sub-frame is corrupted. Then the expected duration T^D_{error} for erroneous transmissions in standard 802.11n and the expected duration T^C_{error} for erroneous transmissions in CAR-MAC can be expressed as

$$\begin{aligned} T^D_{error} &= \sum_{i=1}^{N} \tau(1-\tau)^{N-1}(1-(1-P_e(i))^{A(i)})T^D_{error}(i) \\ T^C_{error} &= \sum_{i=1}^{N} \tau(1-\tau)^{N-1}(1-(1-P_e(i))^{A(i)})T^C_{error}(i) \end{aligned} \tag{4.9}$$

A collision is detected if a block ACK timeout occurs. Thus the time for node i to detect a collision is

$$T_{col}(i) = T_{difs} + T_{plcp} + T_{backtimeout} + \frac{A(i) \cdot L}{R(i)}$$

where $T_{backtimeout}$ is the duration for a block ACK timeout.

If collision occurs between two transmissions that are of different transmission times, the wireless medium would be occupied by the longer transmission. To determine the expected duration of a collision for all nodes in the WLAN, we first sort all nodes in an ascending order of their collision detection time $T_{col}(i)$. We then assume that node i collides only with other nodes that have a shorter transmission duration. In this way, the collision between any two nodes i and j, $(T_{col}(i) < T_{col}(j))$, will be only counted by node j, rather than by both of them, when we calculate the expected collision duration of the network. Then the collision probability for node i can be rewritten as

$$P_{col}(i') = (1-(1-\tau)^{i'-1}) \cdot \tau \cdot (1-\tau)^{N-i'}$$

where $i'(1 \leq i' \leq N)$ is the index of node i in the sorted list of all nodes.

Consequently, the expected duration of collision for all nodes is

$$T_{col} = \sum_{i'=1}^{N} P_c(i')T_{col}(i')$$

By plugging Equations (4.4), (4.8) and (4.9) into Equation (4.6), the throughput for node i in standard 802.11n networks is

$$S^D(i) = \frac{(1 - P_c)(1 - P_e(i)) \cdot A(i) \cdot L}{T_{idle} + T_{succ} + T_{col} + T^D_{error}} \tag{4.10}$$

Similarly, by plugging Equations (4.5), (4.8) and (4.9) into Equation (4.7), the throughput for node i in CAR-MAC is

$$S^C(i) = \frac{(1 - P_c)(1 - P_e(i) \cdot P^C_e(i)) \cdot A(i) \cdot L}{T_{idle} + T_{succ} + T_{col} + T^C_{error}} \tag{4.11}$$

4.4.2 Numerical Results for an Example WLAN

We now apply the above theoretical analysis to an example WLAN. We evaluate the overall network throughput of CAR-MAC and compare it with standard 802.11n transmissions. Consider a 10-node WLAN where all nodes are of saturated traffic. All parameters used in the calculation follow the standard and are listed in Table 4.1. By plugging these parameters into the equations given in the last subsection, we can derive the throughput of each node for both CAR-MAC and standard 802.11n. As both the sub-frame error probability and the physical data rate for cooperative retransmissions are considered when selecting cooperative nodes, we will discuss the impact of these two factors separately. We define *throughput ratio* as the ratio of network throughput of CAR-MAC to the network throughput of standard 802.11n transmission, to show the superior of CAR-MAC in boosting network throughput.

We first study the performance of CAR-MAC in terms of throughput ratio under different sub-frame error probabilities. All cooperative nodes retransmit at the same data rate as the sender in this case. In addition, the sub-frame error probability for all senders is a random variable with expected value $E(P_e)$ and the sub-frame error probability for all cooperative nodes is also a random variable with expected value $E(P^C_e)$. We fix $E(P^C_e)$ at 0.05 and increase $E(P_e)$ from 0.05 to 0.6. The numerical results for these configurations are plotted in Fig. 4.5(a), where the throughput ratio of average aggregation levels 5, 10, 15 and 20 are plotted separately. We can see that as $E(P_e)$ grows, the benefits of cooperative retransmissions become more significant. Specifically, when

Table 4.1: List of parameters used in numerical results of CAR-MAC.

Parameter	Value
Back Off Time Slot	9us
SIFS	10us
DIFS	28us
Block ACK Duration	28us
Block ACK Timeout	19us
Sub-frame Length	1400bytes
Number of nodes	10
Physical Data Rates	$\{15 \ldots, 300\}$Mbps
Expected Aggregation Level	5, 10, 15, 20

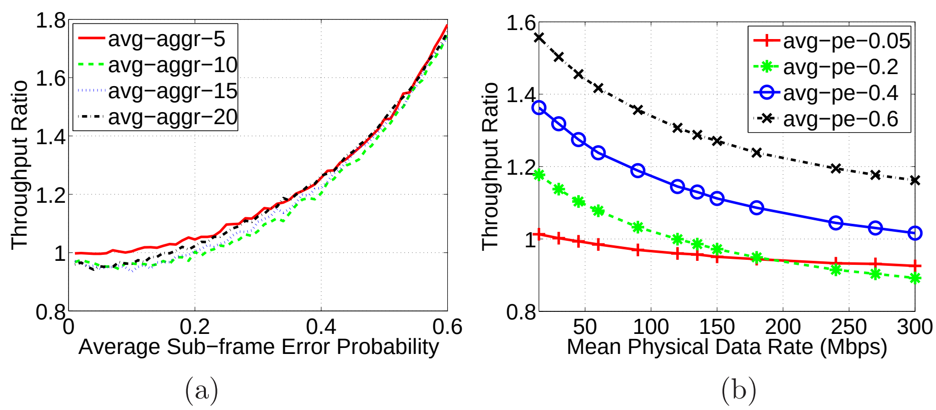


Figure 4.5: Network throughput ratio of CAR-MAC to standard 802.11n transmissions in an example WLAN. (a) Throughput ratio under different sub-frame error probabilities. (b) Throughput ratio under different physical data rates at the sender.

$E(P_e)$ equals 0.5, the network throughput is improved over 50% by CAR-MAC compared with standard 802.11n transmissions. We can also observe that the effects of CAR-MAC are more obvious when the expected aggregation level is low. This is reasonable because the network throughput of direct transmissions grows linearly with the aggregation level, but the number of sub-frames that need retransmission increases much slowly when the sub-frame error probability is small. Note that the gap of throughput ratios among different aggregation levels decreases as $E(P_e)$ increases.

We also examine the performance of CAR-MAC when the cooperative node is capable of retransmitting failed sub-frames at a higher data rate than the original sender. We assume that every cooperative node and the corresponding sender have the same sub-frame error probability. We set the physical data rate for all cooperative retransmissions to be $300Mbps$, while increasing the physical data rate of direct transmissions from $15Mbps$ to $300Mbps$. The numerical results are given in Fig. 4.5(b), where the scenarios of average sub-frame error probability being 0.05, 0.2, 0.4 and 0.6 are examined separately. We can note that when the difference of transmitting data rate between the senders and the cooperative nodes is larger, CAR-MAC has more benefits. The reason is that if a cooperative node has the same transmitting data rate and the same sub-frame error probability as the sender, cooperative retransmissions will have the same MAC efficiency as direct transmissions. We can also see that when the sub-frame error probability is 0.05, the network throughput of CAR-MAC is slightly lower than standard 802.11 transmissions. This is because that only few sub-frames need to be retransmitted in such situations, where the gain of retransmissions cannot cover the overhead. Fortunately, such performance degradation can be avoided in CAR-MAC since it allows the cooperative node to aggregate its own data units with cooperatively retransmitted sub-frames, so as to ensure high MAC efficiency.

4.5 Simulation Results

In previous section, we have shown that CAR-MAC can greatly boost the throughput of a WLAN, by assuming that cooperative nodes have either better sub-frame error probabilities or higher transmitting data rates than senders. But in real WLANs, it cannot be guaranteed that cooperative nodes always have better channel conditions than senders. Moreover, the overhead of selecting cooperative node is not discussed. In this section, we will thoroughly evaluate the performance of CAR-MAC under various network scenarios in terms of network throughput and packet delay via simulations. We will also explore the impact of C-Beacon messages on network throughput. As discussed earlier, 802.11n WLANs achieve high throughput by using frame aggregation and block ACK, which were not supported by existing cooperative retransmission schemes. Therefore, we will compare CAR-MAC with standard 802.11n transmissions, and a CD-MAC [56] like scheme, named *CD-MAC-aggr*, in which the cooperative node retransmits an entire aggregated frame if the frame is not fully received by the destination.

In the simulation, the WLAN is deployed over a $200 \times 200m^2$ field. All nodes are stationary and transmit at the maximum power level such that they can sense the transmission of each other, so as to avoid hidden terminal problems. In addition, all nodes operate in an ad hoc mode thus direct transmission between any two nodes is permitted. An optimal data rate is selected for each pair of nodes by taking their distance into the Ricean fading propagation model. Rate adaptation is disabled in the simulation, as sub-frame error probability can be affected by the employed rate adaptation algorithm. Each node sends constant bit rate (CBR) UDP traffic to all other nodes. If not otherwise specified, all nodes broadcast a C-Beacon frame every second. In addition, the traffic bit rate of each flow is set to 1Mbps.

We first examine the network throughput of CAR-MAC under various node densities. The simulation results are plotted in Fig. 4.6(a), where the number of nodes increases from 5 to 20 in a step of 2. It is noted that CAR-MAC always has higher network throughput than the compared schemes, regardless of the node density. Moreover, the advantage of CAR-MAC is more evident when the node density is high. In particular, the network throughput of CAR-MAC is 39% and 27% higher than 802.11n transmissions and CD-MAC-aggr respectively, when the number of nodes is 20. This is because in CAR-MAC cooperative nodes retransmit frames more efficiently than senders, and they only retransmit failed sub-frames instead of entire aggregated frames. For similar reasons, it can also be observed from Fig. 4.6(b) that CAR-MAC always leads to the highest network throughput under various traffic loads, where the number of nodes is fixed at 10 and the traffic load on each flow changes from 1Mbps to 2Mbps. It should be pointed out that the network throughput of all schemes begins to decrease when the number of nodes or the traffic load grows beyond a threshold, which can be attributed to collision penalty in a saturated network. Nevertheless, the turning point of CAR-MAC appears later than the compared schemes in both Fig. 4.6 (a) and (b), indicating that CAR-MAC is more beneficial in saturated networks.

We now study the average packet delay of CAR-MAC in WLANs with different number of nodes and various traffic loads. Fig. 4.7 shows the sim-

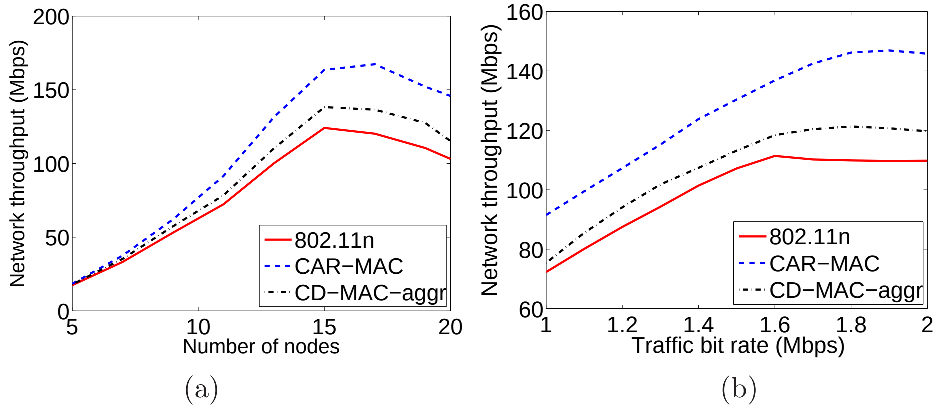


Figure 4.6: Network throughput of CAR-MAC under various node densities and traffic loads. (a) Network throughput vs. Number of nodes. (b) Network throughput vs. Traffic bit rate for each flow.

ulation results. We can see that when the number of nodes is small or the traffic load is light, the packet delay of all schemes is low as most packets are transmitted immediately rather than queued when they arrive at the MAC layer. As the node density or traffic load increases, the average packet delay of CAR-MAC grows more slowly than standard 802.11n transmissions. The reason is that CAR-MAC is able to reduce the number of retransmissions for failed sub-frames, so as to decrease their packet delay. Moreover, the packet delay of following packets in the queue is also shortened as the time occupied by retransmissions is reduced. We can also observe that CAR-MAC has a shorter packet delay than CD-MAC-aggr, because CD-MAC-aggr takes extra time to retransmit sub-frames that have already been successfully received. Note that the average packet delay increases drastically when the traffic in the network is saturated, as the data queues will overflow eventually and thus a large number of packets are dropped.

We also investigate the gains of aggregating the cooperatively retransmitted sub-frames together with new data units at the cooperative nodes. Fig. 4.8 gives the network throughput and the average packet delay of all nodes, in which CAR-MAC-mixed represents the strategy of mixing retransmitted sub-frames with new data units. From Fig. 4.8(a) we can observe that when the number of nodes is small, the network throughput of the mixed strategy is close to that of the basic CAR-MAC. This can be attributed to the fact that

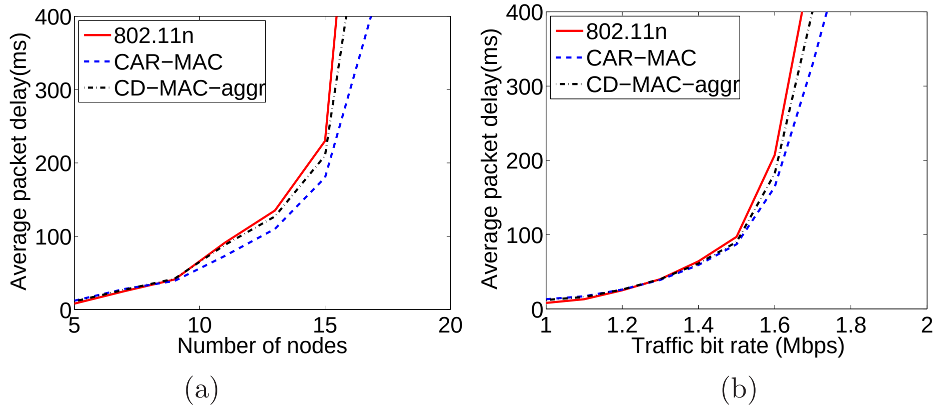


Figure 4.7: Average packet delay for CAR-MAC under various node densities and traffic loads. (a) Average packet delay vs. Number of nodes. (b) Average packet delay vs. Traffic bit rate for each flow.

the overhead of cooperative retransmissions is not reflected in the network throughput when the network load is unsaturated. When the number of nodes is large, the advantage of the mixed strategy is remarkable, indicating its capability of amortizing the overhead of cooperative retransmissions as well as alleviating contention intensity. The relatively low packet delay exhibited in Fig. 4.8(b) also verifies that the mixed strategy is capable of utilizing the wireless spectrum more efficiently.

Finally, we evaluate the impact of C-Beacon messages on the network throughout of normal 802.11n transmissions. Cooperative retransmissions are disabled in this case to exclude the throughput gain of cooperative retransmissions. The network throughput for standard 802.11n WLANs with C-Beacon messages is plotted in Fig. 4.9, where Standard, CBeacon-0.5s, CBeacon-1s, CBeacon-2s and CBeacon-5s stand for the scenarios that WLAN nodes do not broadcast C-Beacon messages, and broadcast a C-Beacon message every $0.5s$, $1s$, $2s$ and $5s$, respectively. We can see that the impact of C-Beacon messages on the overall network throughput is very limited, even when C-Beacon messages are broadcast every $0.5s$ and the network is under severe contention. This is because that C-Beacon messages are generated at a much lower frequency compared with the heavy data traffic. In addition, as C-Beacon messages are small packets broadcast at high data rates, they occupy the wireless medium for very short durations. More importantly, similarly to Beacon messages,

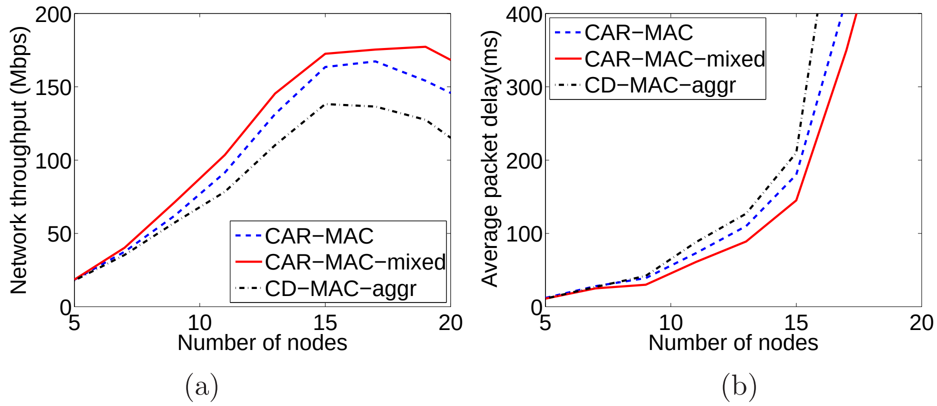


Figure 4.8: Benefit of aggregating new data units together with cooperatively retransmitted sub-frames in CAR-MAC. (a) Improvement in network throughput. (b) Reduction in average packet delay.

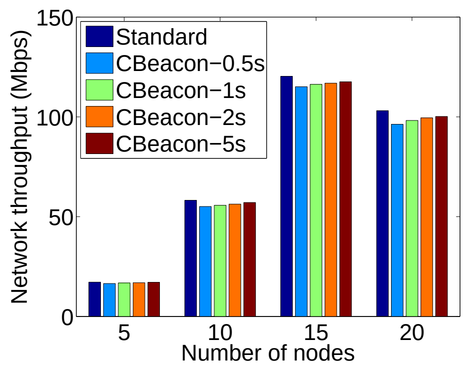


Figure 4.9: Impact of C-Beacon messages on standard 802.11n transmissions.

C-Beacon messages are buffered at a separate queue at each node, which has higher medium access priority than the data queues. Therefore, they will not escalate the contention level of a network, as they can always obtain the medium without causing a physical collision when competing with data frames.

4.6 Conclusions

In this chapter, we have proposed a cooperative retransmission MAC (CAR-MAC) protocol for IEEE 802.11n based WLANs, taking full advantage of frame aggregation and block ACK mechanisms of 802.11n. We first gave a distributed scheme to select cooperative nodes, such that each node may dynamically select a cooperative node for each destination. We then presented the CAR-MAC protocol in which the cooperative node only retransmits the failed sub-frames caused by channel errors. After that, we analyzed the theoretical throughput of the proposed protocol and derived numerical results based on the analysis. Finally, we conducted extensive simulations to evaluate the performance of the proposed protocol. Both numerical and simulation results show that the propose protocol boosts the network throughput and reduces the average packet delay significantly.

Chapter 5

High-Throughput Collision-Free Client Polling in WLANs

5.1 Introduction and Related Work

In this chapter, we consider client polling for large-scale WLANs operating in the point coordination (PCF) mode, to provide collision-free access for every client and utilize network capacity to the maximum extent. We first introduce a WLAN framework, in which CFPs of all APs are synchronized and divided into fixed-length time slots, and clients are polled in these time slots. We then formulate client polling into a time slot allocation problem and show that the problem is NP-complete. After that, we present a collision-free client polling scheme consisting of three procedures: (1) a basic polling procedure that determines the minimum number of time slots required to poll every client once to obtain the polling frequencies of all clients; (2) a complementary polling procedure that makes extra polls for APs that have idle time slots without causing collisions, to improve spatial reuse of the network; (3) a backup poll selecting procedure that finds backup clients to poll in case the current polled client has no data to transmit, to utilize the otherwise wasted bandwidth. Finally, we conduct extensive simulations to evaluate the performance of the proposed scheme. As will be seen from the simulation results, the proposed scheme outperforms the existing schemes significantly. Note that since the same channel is used for all APs in the network in our scheme, we can deploy multiple vertically overlapped networks in the same location, by assigning an orthogonal channel to each network and polling it independently, to greatly boost network capacity.

The rest of the chapter is organized as follows. In section 5.2, we introduce the background and related work. In Section 5.3, we give the system model used in our scheme and formulate client polling into an optimization problem. In Section 5.4, we present the proposed scheme to find the minimum number of time slots required to poll all clients, determine complementary polls for idle APs and prepare backup clients. In Section 5.5, we evaluate the performance of the proposed scheme and compare it with other schemes. Finally, we conclude the chapter in Section 5.6.

5.2 Background and Related Work

WLANs operate in the PCF mode to diminish contentions among clients. In this mode, an AP works as the *point coordinator* and periodically initiates contention-free periods (CFP), during which it polls and transmits data to its associated clients. CFPs alternate with contention periods (CP), during which clients also use DCF to compete for the medium. As shown in Fig. 5.1, a CFP begins with a *Beacon* frame that includes the maximum duration of the CFP, and all clients set their network allocation vector (NAV) to this period so they will not attempt to transmit during the CFP. The AP alternatively transmits data frames to clients, and polls clients by transmitting CF-Poll frames. After receiving a data frame, a client sends out a CF-ACK frame to acknowledge the reception. After receiving a CF-Poll frame, a client transmits a data frame if it has pending traffic, or a CF-Null frame if it has no frame to transmit. The CF-Poll and CF-ACK information can be piggybacked in data frames to reduce MAC overhead. The AP can terminate the CFP earlier than the broadcast duration by sending out a CF-End frame.

Due to the contention-free characteristics of PCF, QoS requirements of real-time services can be satisfied in WLANs with a single AP, by properly polling all associated clients. In IEEE 802.11e [66], PCF was extended to hybrid coordination function (HCF) to enhance QoS in WLANs. Moreover, a lightweight multi-priority PCF was presented in [67] to support QoS in WLANs, which is

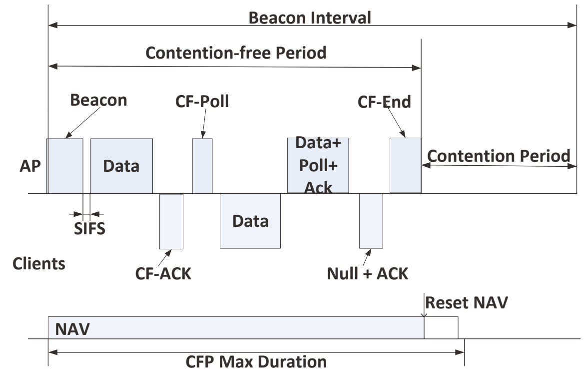


Figure 5.1: An example of PCF-based frame transmissions during a contention-free period.

less complex compared to HCF.

However, in multi-AP WLANs that operate in the PCF mode, there may be collisions among nearby BSSs due to the lack of inter-AP coordination, which may degrade the performance of real-time services. Similar to WLANs operating on the DCF mode, the effectiveness of assigning orthogonal channels to interfering BSSs is constrained by the limited channel resources. The multi-AP coordination problem was first studied in [68], where all clients in the network are grouped into collision-free sets, and the CFPs of all APs are synchronized. In each CFP, clients in one of the collision-free sets are polled. The shortcoming of this approach is that it assumes only clients in the overlapped area of BSSs cause collisions, ignoring the hidden terminal problem among nearby BSSs. In [69], a client polling framework, named *MiFi*, was proposed for multi-AP WLANs, in which CFPs at all APs are synchronized and divided into time slots. Interfering APs are assigned different time slots, so as to poll all clients without causing collisions and provide the same throughput for every client. Nevertheless, the assumptions in the AP interference model in this work is too strong thus places an unnecessary constraint on concurrent polls, while its throughput-fair objective is achieved by sacrificing the aggregate throughput of the network.

A dynamic client polling scheme for WLANs operating on the HCF mode was presented in [70], where the client with the earliest due time in its QoS requirement is polled first, and clients that do not interfere with the ongoing polls are polled simultaneously. One drawback of this approach is that it requires 802.11e implementation on clients. Moreover, the computation and communication overheads of this scheme are quite significant, since the coordinator has to make a polling decision dynamically for every frame transmission.

5.3 System Model and Problem Formulation

5.3.1 Network Model

We consider a WLAN consisting of a set of APs and a number of clients, and all APs are assigned the same channel. We use sets M and C to denote APs and clients, respectively. In the network, all APs operate in the PCF mode and are connected via a wired network to a *wireless controller*, which coordinates the connected APs and provisions the Internet access. We assume that the CFPs of all APs are of the same duration and are synchronized by the controller. In addition, we set the duration of a CP to the minimum value allowed by the 802.11 standard. This is to allow the transmission of WLAN control frames while restricting the transmission of data frames during CPs, since a busy medium caused by data transmission may defer the Beacon message and severely affect the synchronization of CFPs. Every client is associated with an AP that has the strongest received signal strength. For each client-AP association, an optimal physical data rate can be adapted according to the channel condition. Thus for each client i, we define a variable A_i to denote its associated AP, and a variable R_i to denote its optimal physical data rate. An example WLAN with such framework is given in Fig. 5.2.

We further assume that at every AP the CFP is divided into several time slots that are of the same length, and within each time slot the AP polls one client or transmits a frame to one client. As different clients may have diverse physical data rates and frame sizes, which depend on the channel condition and the application type, we set the length of a time slot as the required time to transmit a frame of size α between the client with the lowest data rate and its associated AP. For non-real-time traffic, the data frames of any client can be aggregated or fragmented before transmitting, such that the transmission time of these frames fits into the length of the time slot. Without loss of

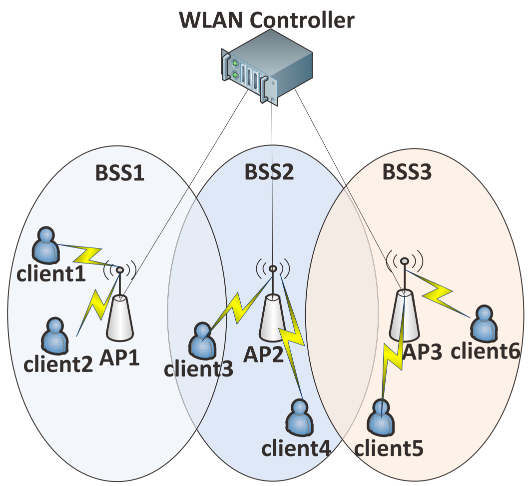


Figure 5.2: An example WLAN, where all APs are connected to the WLAN controller, $client1$ and $client2$ are associated with $AP1$; $client3$ and $client4$ are associated with $AP2$; $client5$ and $client6$ are associated with $AP3$.

generality, assume that client i has the lowest data rate, then for any other client j, the frame size would be $\frac{R_j}{R_i}\alpha$.

For real-time traffic, the delay introduced by frame aggregation may affect the application performance. Thus real-time frames should be transmitted without aggregation. As small packets are often used in real-time traffic, clients with higher data rates may not need an entire data slot to transmit a data frame. The corresponding APs will be idle till the beginning of the next time slot to stay synchronized with other APs. Frame size α can be shortened to reduce this idle time. However, the frame overhead ratio for clients with non-real-time traffic will be increased when the frame size is reduced. The wireless controller adjusts α according to the amount of real-time traffic in the network, so as to maximize the overall channel utilization.

5.3.2 Interference Model

In WLANs operating in the PCF mode, there is no interference between any two clients from the same BSS, since they are polled in different time slots. Two clients from neighboring BSSs would interfere with each other, if one client or its associated AP can sense the transmission from the other client or the corresponding AP. The reason is that as an ACK frame is needed for every data transmission, the loss of either the data frame or the ACK frame caused by interference will lead to a transmission failure. For example, in the WLAN in Fig. 5.2, both $client1$ and $client2$ interfere with $client3$, since $AP1$ and $client3$ can sense the transmission of each other; In addition, $client4$ interferes with $client5$ as they can sense the transmission of each other. We define a binary variable $I_{i,j}$ to describe the interference between any two clients i and j. The variable is set to one if two clients interfere; Otherwise, it is zero.

The interference area of a station (client or AP) may be spatially and/or temporally irregular, due to multi-path propagation, reflection and attenuation of wireless signals in an indoor WLAN environment. In our model, we determine the interference among stations by scheduling passive measurement in nearby BSSs when a station is transmitting, which is easy to implement with the assistance of the wireless controller. In particular, if a station is about to be idle in the subsequent time slots, the AP piggybacks a list of clients from neighboring BSSs that will be polled during these time slots in the CF-Poll message. The idle station measures the received power level of these transmissions and reports the sensed results to the controller when polled later. In this way, the controller can obtain and update the interference relationship among any two stations in the network.

5.3.3 Conflict Constraint for Concurrent Polling

Clients from the same BSS need to be polled in different time slots as only one client in the BSS is allowed to transmit or receive during a time slot. In addition, interfering clients from neighboring BSSs should be polled in different time slots to avoid transmission collisions. Let T_i denote the time slot assigned to client i, then such conflict constraint can be formally expressed as

$$T_i \neq T_j, \text{if } A_i = A_j \text{ or } I_{i,j} = 1, \forall i, j \in C$$

This conflict constraint of concurrent polling can also be described by a polling conflict graph $G = (V, E)$, where each vertex represents a client, and there is an edge between two vertices if they are from the same BSS or interfering with each other. Clearly, clients within a BSS form a clique in this graph. The polling conflict graph of the WLAN in Fig. 5.2 is given in Fig. 5.3.

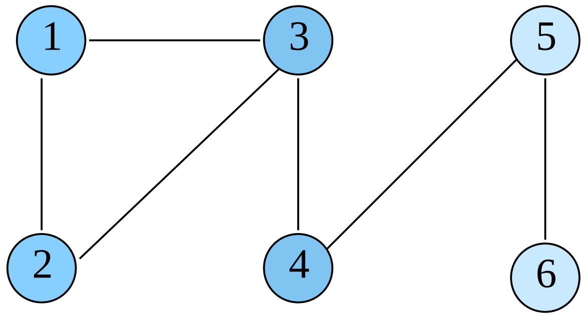


Figure 5.3: Polling conflict graph of the WLAN in Fig. 5.2.

5.3.4 Problem Formulation

We can now formally formulate client polling in PCF-based WLANs with multiple APs into an optimization problem as follows. Given a WLAN with a set of APs, each of which associated with a number of clients, and the interference relationship among clients from nearby BSSs, find the minimum number of time slots, K, to poll every client, such that interfering clients from neighboring BSSs and non-interfering clients from the same BSS are polled in different slots. The polling will be repeated in the sense that each client will be polled once every K time slots. This optimization problem can be formulated as the following integer program.

Minimize K

Subject to

$$1 \leq T_i \leq K, \qquad \forall i \in C \tag{5.1}$$

$$T_i \neq T_j, \text{ if } A_i = A_j, \qquad \forall i, j \in C \tag{5.2}$$

$$T_i \neq T_j, \text{ if } I_{i,j} = 1, \qquad \forall i, j \in C \tag{5.3}$$

The above problem is equivalent to the vertex-coloring problem in the polling conflict graph defined in the last subsection. In particular, coloring a vertex in color k in the graph is equivalent to allocating time slot k to the

Table 5.1: Optimal client polling for all APs in example WLAN

Time Slot	**AP1**	**AP2**	**AP3**
1	Client1	*Idle*	Client5
2	Client2	Client4	Client6
3	*Idle*	Client3	*Idle*

corresponding client in the network. Clearly, this is the *K-colorable* problem in graph theory, which is well-known to be NP-complete [71].

5.4 Collision-Free Client Polling

In this section, we propose a *Collision-Free Client Polling (CFCP)* scheme for multi-AP WLANs to increase throughput and achieve equal channel access time for every client. It consists of three procedures: (1) a basic polling procedure that determines the minimum number of time slots required to poll every client once to obtain the polling frequencies of all clients; (2) a complementary polling procedure that makes extra polls for APs that have idle time slots without causing collisions, to improve spatial reuse of the network; (3) a backup poll selecting procedure that finds backup clients to poll in case the current polled client has no data to transmit, to utilize the otherwise wasted bandwidth. The details of these procedures are given in the following subsections.

5.4.1 Basic Polling Procedure

Recall that we need to allocate K time slots for all APs to poll every client once in a collision-free manner, which is called *basic polls*. If all the clients are static and there is no joining or leaving client in the network, the optimal polling that uses the least amount of time slots can be derived by solving the aforementioned integer program using mathematical tools. An optimal polling for the WLAN in Fig. 5.2 is given in Table 5.1 for instance. However, in large-scale WLANs, where clients are dynamic and mobile, the time complexity of this approach is too high.

On the other hand, many heuristic algorithms have been proposed in the literature for the vertex coloring problem in graph theory. It has been shown in practice that some of these algorithms, such as the recursive largest first (RLF) [72] and backtracking sequential coloring (BSC) [73], can provide results close to the optimal ones. Thus in this chapter, the RLF algorithm will be applied on the polling conflict graph to determine the basic polls. The general idea of RFL is as follows. First, vertex c with the maximum degree is selected. Then all vertices that are not adjacent to c and have a maximum number of common neighbors with c are contracted into c until c is adjacent to all these vertices. Then, vertex c and all vertices that have been contracted into c are assigned a new color and removed from the graph. This step is repeated until all the vertices are removed from the graph. After running this procedure, a client polling matrix P is constructed, in which $p_{k,c}$ is set to one if client c is to be polled in time slot k, otherwise it is zero. The worst-case time complexity of this algorithm is $O(|V|^3)$.

5.4.2 Complementary Polling Procedure

Note that in basic polls many APs would be idle in some time slots. Take the optimal polling presented in Table 5.1 as an example, $AP2$ is idle in time slot 1, and both $AP1$ and $AP3$ are idle in time slot 3. Collision avoidance and the difference among the numbers of clients in various BSSs may lead to such idle slots. In particular, let c_{max} and c_{min} denote the maximum and minimum number of clients for all BSSs in the network, respectively, then the BSS with c_{min} clients will be idle in at least $c_{max} - c_{min}$ slots. APs can make extra polls during these time slots to fully utilize network capacity. In our example, $AP3$ can make an extra poll to either $client5$ or $client6$ in time slot 3. Such extra polls are called *complementary polls*.

We propose a complementary polling procedure in this subsection to find a set of complementary polls for each allocated time slot. In any time slot, the complementary polls should not conflict with the ongoing polls. Moreover, the complementary polls should satisfy the conflict constraint among themselves. Therefore, for time slot k, we first identify a set S_k consisting of clients that are not interfering with the ongoing polls. We then select a subset C_k from S_k such that all clients in this subset can be polled simultaneously. From the polling conflict graph perspective, we first need to derive a subgraph from the colored polling conflict graph, including all vertices that are not adjacent to vertices in color k, and all the edges connecting these vertices. After that, we find an independent set on the subgraph to be the set of complementary polls for time slot k.

For each time slot, the maximum independent set on the corresponding subgraph can be used as the set of complementary polls, so as to maximize spatial reuse in that time slot. However, this may not be the best strategy for two reasons. First, finding the maximum independent set on a graph is NP-complete [74] thus the time complexity is too high to derive such a set fast for WLANs with a large number of clients. Second, clients subjected to less interference are prone to be selected in the maximum independent set, leading to accumulated unfairness among clients after a set of complementary polls has been found for all time slots.

On the other hand, a maximal independent set (MIS) on the subgraph can also be used as the set of complementary polls for a time slot. In fact, as a variety of maximal independent sets exist on a graph, we can take the fairness among clients into consideration when choosing the complementary polls set from these MISs. Other QoS requirements can also be used as the metric to choose the set of complementary polls. The Bron-Kerbosch algorithm [75] has been widely used in practice to find all MISs on a graph. We adapt it in this procedure to find only the first 200 MISs when the subgraph is large, because the number of MISs increases exponentially as the size of subgraph grows. Specifically, we maintain a polled counter for each client and determine the complementary polls set for all time slots one by one. For each time slot, we choose the MIS that leads to the smallest difference among all polled counters as the complementary polls set. The pseudo code of this procedure is given in Table 5.2. The notations used in this procedure and the following procedure are summarized in Table 5.4.

5.4.3 Backup Poll Selecting Procedure

Note that a time slot for a BSS will be wasted if the polled client has no data frame to transmit or receive. This is common for clients with light traffic. To

Table 5.2: Complementary client polling procedure

Input:
  Polling Matrix P;
  Polling Conflict Graph G;
  Number of Allocated Time Slots K;
Output:
  Complementary Polling Matrix P';
Algorithm:
  Initialize all elements of polled counter vector $\overrightarrow{cnt}$ to 1;
  $P' = P$;
  for $k = 1{:}K$
    $O_k = \{i |\ i \in C;\ p_{k,i} = 1\}$;
    $S_k = \{j |\ j \in C;\ \forall i \in O_k, I_{i,j} = 0, A_i \neq A_j\}$;
    Subgraph $G_k = (S_k, E')$;
    Find a list of MISs, L_k, on G_k using Bron-Kerbosch algorithm;
    for each $MIS_l \in L_k$
      Temporary counter vector $\overrightarrow{tcnt} = \overrightarrow{cnt}$;
      for each i $\in MIS_l$
        $tcnt[i]$++;
      end for
      $D_l = \max(\overrightarrow{tcnt}) - \min(\overrightarrow{tcnt})$;
    end for
    $l' = \underset{MIS_l \in L_k}{\operatorname{argmin}} \{D_l\}$;
    for $i \in MIS_{l'}$
      $p'_{k,i} = 1$; $cnt[i]$++;
    end for
  end for

better utilize these time slots, we introduce a backup poll selecting procedure in this subsection, which determines a *backup client* for any polled client that has no pending traffic. For example, in the optimal polling in Table 5.1, in time slot 1, *client*2 and *client*6 can be selected as backup polls for *client*1 and *client*5, respectively, if they do not have pending traffic.

Similar to constructing complementary poll sets, in each time slot, the backup polls should not conflict with ongoing polls. The backup polls have to satisfy the conflict constraint among themselves as well. However, as it is difficult to predict whether a polled client has traffic or not, we cannot determine a backup poll set in advance for every time slot that works under all circumstances. Thus we define a *safe backup set* B_c for each client c, including any client whose neighbors in the polling conflict graph is a subset of the neighbors of client c. Then in each time slot, if a polled client c has no traffic, it is safe to poll any client in B_c instead, without violating the conflict constraint. Moreover, we define a backup counter for each client, recording the times that a client is used as backup poll. If a poll to client c is idle, the associated AP can instantly chooses the client that has the smallest backup counter from the safe backup set B_c, so as to maintain fairness among clients in the long term. Other metrics can be used as well when choosing a client from the backup poll set to achieve various QoS objectives. The pseudo code of this procedure is given in Table 5.3.

5.5 Performance Evaluations

In this section, we first study the effectiveness of the RLF algorithm for finding the minimum number of required polling time slots in various networks. We then evaluate the performance of the proposed collision-free client polling scheme by simulation in terms of minimum number of time slots to poll all clients, saturated throughput of each client, expected number of concurrently active BSSs, and successful ratio of finding backup polls. We compare our scheme with the aforementioned synchronized-PCF scheme [68] and the MiFi scheme [69].

In the simulation, a WLAN is deployed in a $1000 \times 1000m^2$ field and 25 APs are placed evenly on a 5×5 grid to provide fully coverage. Each client

Table 5.3: Backup poll selecting procedure

Input:
Association Vector A;
Interference Matrix I;
Idle poll $p_{k,c}$;
Output:
Backup poll $b_{k,c}$;
Algorithm:
Initialization:
for each $i \in C$
Neighbor set $N_i = \{j | \forall j \in C, A_i = A_j \text{ or } I_{i,j} = 1\}$;
end for
for each $i \in C$
Safe backup set $B_i = \{j | \forall j \in C, N_j \subset N_i\}$;
Backup polled counter $bcnt[i] = 0$;
end for
Entry:
Given idle poll $p_{k,c}$
$index = \underset{i \in B_c}{\operatorname{argmin}} \{bcnt[i]\}$;
$bcnt[index]{+}{+}, \; b_{k,c} = index$;
Return

Table 5.4: Notations used in the collision-free client polling scheme

Notation	Semantics
A	Association vector $A = \{A_i \| i \in C; \; A_i \in M\}$
I	Interference matrix $I = \{I_{i,j} \| i, j \in C\}$
G	Polling conflict graph
K	Number of time slots
P	Polling matrix
P'	Complementary polling matrix
L_k	List of MIS for subgraph G_k
$\overrightarrow{cnt}$	Polled counter vector
$\overrightarrow{tcnt}$	Temporary counter vector
$\overrightarrow{bcnt}$	Backup polled counter vector
D_l	Difference between the max and min of $\overrightarrow{tcnt}$ for $l \in L_k$
N_i	Set of neighbors on G for client i
B_i	Set of safe backup clients for client i

Table 5.5: Performance comparison among different time slot allocation algorithms

# Clients	**Max # Clients in a BSS**	**Required Time Slots**		
		Optimal	**RLF**	**BSC**
50	5	7	7	7
100	9	11	11	11
150	13	17	17	18
200	14	21	21	21
250	17	21	22	23
300	21	27	28	28

is randomly distributed in the field and associated with the closest AP. For simplicity, we assume that all stations have the same maximum transmission range and interference range, which are set to $120m$ and $180m$, respectively. The data rate of each client is selected from the data rate set of IEEE 802.11g standard according to its distance to the associated AP.

We first study the effectiveness of the RLF algorithm under different client densities, by comparing it with the BSC algorithm and the optimal results. The number of clients varies from 50 to 300. The results are given in Table 5.5, where the maximum number of clients in a BSS of the network is also given for comparison. We can see that the RLF algorithm always achieves the same or better results compared with the BSC algorithm, and its results are very close to the optimal ones under all client densities. We also note that the difference between the allocated time slots and the maximum number of clients in a BSS becomes more evident as the client density increases, indicating that more time slots are needed when the interference among neighboring BSSs becomes more severe.

Next, we compare the minimum number of required time slots of the proposed scheme with that of the synchronized-PCF scheme and the MiFi scheme. The number of clients is increased from 50 to 500 at a step of 50. The simulation results are plotted in Fig. 5.4(a), in which each point is the average result of 100 experiments. We can observe that our scheme constantly uses fewer time slots than MiFi, and the advantage becomes more evident as the number of clients increases. The synchronized-PCF scheme needs slightly fewer time slots than our scheme at the price of potential collisions due to hidden terminals from nearby BSSs, which is reflected in the corresponding throughput shown in Fig. 5.4(b).

Furthermore, we evaluate the saturated throughput of each client using the proposed CFCP polling scheme. The simulation results are given in Fig. 5.4(b), where the number of clients is set to 200 and every client always has data frames to transmit. The results are sorted in an ascending order of the physical data rate of clients. It is notable that the throughput of each client in our scheme is proportional to its data rate, and clients with the same data rate have the same throughput, since all clients have equal time for data transmissions. On the other hand, in synchronized-PCF, the throughput of many clients is close to zero due to collisions, though the throughput of a few clients are slightly higher than that of CFCP due to the relatively smaller number of allocated time slots. On the other hand, in MiFi all clients have the same throughput, which is the result of its throughput-fair design. However, it is clear that the client throughput of MiFi is lower than the expected client throughput of CFCP, as spatial reuse is not fully exploited by MiFi. Apparently, the throughput of each client is related to the number of clients and the association of clients in the network. To guarantee certain throughput for each client, an admission control mechanism is needed, which is out of the scope of this chapter.

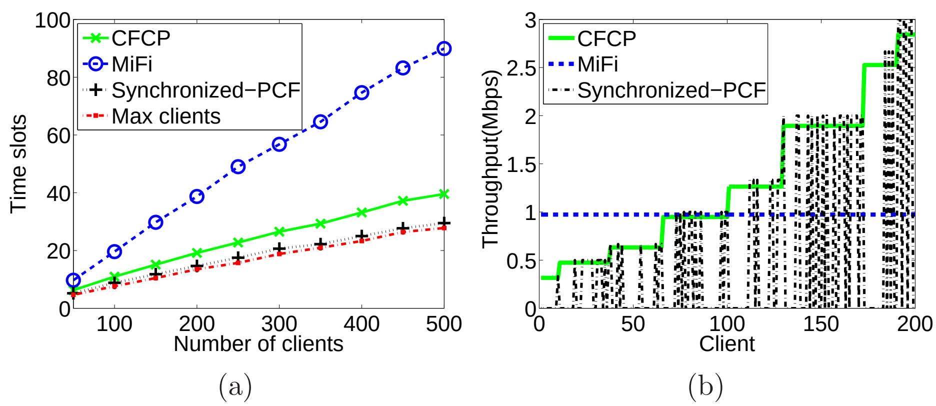


Figure 5.4: Number of allocated time slots and saturated client throughput for various client polling schemes. (a) Number of time slots vs. client density. (b)saturated throughput for every client.

To assess the spatial reuse property of CFCP, we examine the expected number of active BSSs in the network with basic polling under different client densities, and compare it with complementary polling and MiFi. As shown in Fig. 5.5(a), the expected number of active BSSs with basic polling is more than two times of that of MiFi when the client density is high. Moreover, with complementary polling, at least one half of the BSSs are active simultaneously on average regardless of the client density. Note that the spatial reuse level of our scheme increases as the client density grows, indicating that the exponentially boosted polling choices overweight the additional polling constraints when the number of clients is increased.

Finally, we evaluate the performance of the backup poll selecting procedure under different client densities, in terms of the percentage of polls that have backups and the percentage of clients that are used as backups. The simulation results are given in Fig. 5.5(b), where backup-coverage denotes polls that have backups and backup-usage denotes clients that are used as backups. We can observe that the percentage of polls with backups increases along with the client density, showing a benefit from user diversity to select backup polls. In particular, almost 80% polls have backups when the network has 500 clients. On the other hand, the percentage of clients used as backups increases at first as the client density grows, which can be attributed to the boosted user diversity as well. However, the growth becomes less apparent when the number of clients is greater than 200. The reason is that as we only choose backups from the safe backup sets, clients with the maximum neighbor set on the polling conflict graph will never be selected as backups.

5.6 Conclusions

In this chapter, we have studied collision-free client polling in PCF-based WLANs with multiple APs to provide guaranteed bandwidth and bounded access delay for real-time services. We have formulated the problem into a time slot allocation problem and given a polling scheme that polls every client using the minimum number of time slots. Moreover, we have provided a procedure that makes complementary polls for idle APs to enhance the utilization of network capacity. We have also presented a procedure to find backups for the polled clients that have no pending traffic. Simulation results show that the

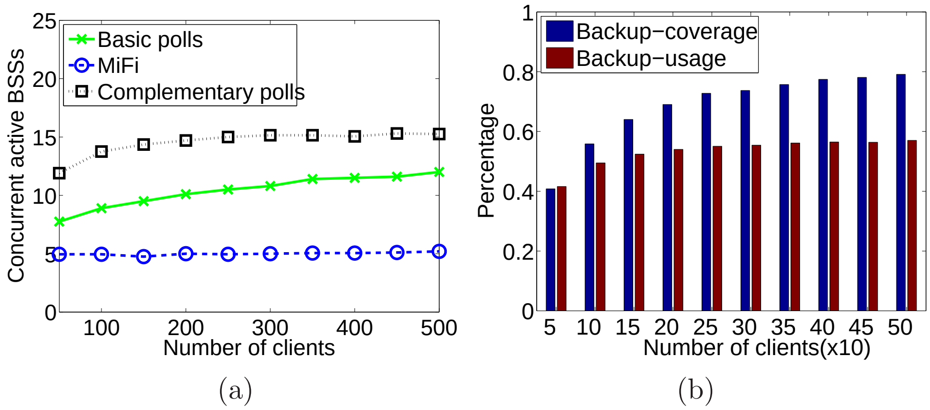


Figure 5.5: Effectiveness of complementary polling procedure and backup poll selecting procedure. (a) Expected number of simultaneously active BSSs vs. client density. (b) Percentage of polls with backups and percentage of clients used as backups vs. client density.

proposed scheme can provide high throughput and almost identical channel access time for each client and high spatial reuse level for the network.

Chapter 6

Link-Layer Multicast in WLANs with Smart Antennas

6.1 Introduction

In this chapter, we consider link-layer multicast in 802.11n WLANs where APs are equipped with smart antennas while clients have omni-directional antennas. A smart antenna is an array of antennas that transmit wireless signal at a series of different radiation patterns. By managing the radiation pattern, the wireless signal in desired directions can be enhanced while interference from other directions can be reduced [86]. By taking advantage of smart antennas, we partition multicast clients into several groups, and select an antenna pattern carefully for each group such that multicast frames can be transmitted to each group once at high data rates. We find through experiments that clients in close locations do not necessarily have the same optimal antenna patterns. In addition, multicast frames transmitted at 802.11n double-stream data rates are more prone to be affected by noise and interference. We formulate the problem of link-layer multicast over smart antennas into a mixed integer program, aiming at minimizing the aggregated transmission time of each multicast frame while ensuring high packet reception ratio (PRR) for all clients. For the case that PRRs for all antenna patterns and data rates are known, we propose an optimal algorithm to solve the mixed integer program. For practical deployment, we present an efficient on-line algorithm where the partition of clients, antenna pattern and data rate for each group are adapted dynamically based on PRR reports from clients. We conduct extensive experiments to evaluate the on-line algorithm and demonstrate that our algorithm can achieve high throughput than translating multicast frames into unicast and multicast over omni-directional antennas, while ensuring high PRR for all clients.

The rest of the chapter is organized as follows. Section 6.2 reviews the related work. Section 6.3 studies the characteristics of link-layer multicast over smart antennas for 802.11n WLANs in indoor environments via experiments. Section 6.4 introduces the system model, and formulates the problem into an optimization problem. Section 6.5 presents the optimal algorithm and Section 6.6 gives the on-line algorithm for practical deployment. Section 6.7 discusses the experimental results of the proposed algorithm. Finally, Section 6.8 concludes the chapter.

6.2 Related Works

There have been some works in the literature on reliable multicast in WLANs that explicitly acknowledge multicast frames for all clients. In [76], a broadcast medium window protocol was presented, where a multicast frame is transmitted to every client separately in a round-robin fashion. The transmission to each client is protected by RTS/CTS frames and acknowledged by an ACK frame. A client buffers all received and overheard frames. Upon receiving the RTS, the intended client replies the sequence number of unreceived frames in a CTS frame. The sender transmits a new frame if the client has already received the multicast frame by overhearing. However, the efficiency of this scheme is low as the sender contends for the wireless medium before transmitting to each client. A batch mode multicast MAC was proposed in [77], where the sender contends for the medium only once for each multicast frame. After obtaining the medium, a batch of RTS/CTSs is exchanged consecutively for all clients. Then the sender transmits the multicast frame. After that, the sender sends a request for ACK frame to each client sequentially and a client replies with an ACK frame. The control overhead is further reduced in [78] by piggybacking the acknowledge to the current multicast frame in CTS frames, rather than sending ACK frames.

There have also been some works in the literature on leader-based protocol (LBP) for multicast in WLANs. In [80], a leader was preselected to send ACK to multicast frames, to avoid collisions among multiple ACK frames. If a non-leader client fails to receive a frame, it sends a negative ACK frame to corrupt the ACK frame from the leader, to trigger a retransmission from the sender. It was shown through experiment that LBP outperforms the standard link-layer multicast if the leader is properly selected. However, as discussed earlier, LBP may lead to unnecessary retransmissions since non-leader clients do not know whether they have received an incorrect frame. In [82], the sequence number of an upcoming multicast frame is broadcast in a reliable manner ahead of the multicast frame to address such a problem. Another protocol was introduced in [83], where the client with the worst channel condition is selected as the "target" and multicast frames are translated into unicast frames to the target. These translated frames are further encoded with the forward error encoding (FEC) technique to increase redundancy, such that other clients can receive the frames with high probability by overhearing. A rate adaption algorithm was proposed in [84] for LBP to improve the efficiency of multicast transmissions.

Smart antennas have been adopted in some works to improve multicast performance in WLANs. In [87], a multicast scheme was proposed for WLANs with switched beamforming smart antennas, where the radiation patterns of antennas are prefixed. A multicast frame is first sent by a high-rate omni-directional transmission to cover clients with good channel conditions, then sent by one or more high-rate directional transmissions to cover the clients with poor channel conditions. For each directional transmission, only one beam is selected to be active. Multicast over switched beamforming smart antennas was further discussed in [88] and [89], where composite-beam patterns are generated to improve multicast performance by splitting transmit power to multiple individual beams either equally or asymmetrically. In [90], a multicast scheme was introduced for WLANs with dynamic beamforming smart antennas, where the radiation pattern of antennas can be dynamically adjusted. In the scheme, clients are first grouped according to the similarity of their channel gain. A beamforming vector is derived for each group such that the signal to noise ratio of the client with the worst channel gain is maximized.

Finally, frame aggregation and block ACK mechanisms of 802.11n have been applied in several link-layer multicast protocols. In [91], a sender sends a block ACK (BACK) request frame to each client sequentially after transmitting a number of multicast frames, and each client replies with a BACK frame. A mechanism named *groupcast with retires (GCR)* has been proposed in the IEEE 802.11aa standard [92] to provide robust audio video streaming in WLANs. GCR includes two retransmission policies for multicast: GCR unsolicited retries and GCR BACK. With GCR unsolicited retries, a sender retransmits a multicast frame one or more times to increase the receiving probability of clients. With GCR BACK, a sender sends a burst of multicast frames, then exchanges BACK request and BACK frames with all or a subset of multicast clients to determine the reception status of multicast frames, and retransmits failed multicast frames. Furthermore, in [94] and [95], frame aggregation mechanism is applied to LBP, where multiple multicast frames are aggregated into one frame before transmission. If a sender does not receive a BACK frame from the leader after transmitting an aggregated frame, it polls the leader and non-leader clients sequentially using BACK request frames. Nevertheless, these studies have not explored the benefits of smart antennas, and their scalability is still limited by the overhead of BACK frames. In the meanwhile, several rate adaptation algorithms were proposed in [46, 96] for unicast in 802.11n WLANs. However, the characteristics of MIMO data rates have not been studied for link-layer multicast in 802.11n WLANs.

6.3 Motivating Experiment

We start with an experiment on multicast in 802.11n WLANs with smart antennas, to reveal the factors that affect multicast performance. The experimental testbed consists of an 802.11n Zoneflex AP with smart antennas from Ruckus Wireless and six 802.11n laptops with omni-directional antennas as clients. The locations of the AP and clients are shown in Fig. 6.1. The experiment is conducted at 2.4GHz frequency band, at which the AP can transmit a multicast frame on 64 different antenna patterns. The transmitting power of the AP is fixed at $50mW$. The AP can support up to two spatial streams.

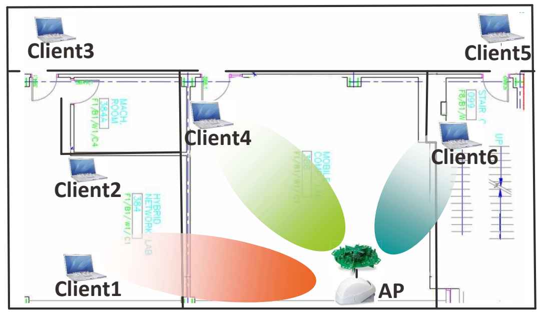


Figure 6.1: A multicast example in a testbed where the AP is equipped with smart antennas.

6.3.1 Limits of Multicast in 802.11n WLAN

In 802.11n WLANs, multicast frames can be transmitted at *legacy data rates* that are compatible with 802.11a/b/g. They can also be transmitted at *single-stream high throughput (HT)* data rates and *double-stream HT* data rates defined for 802.11n only. We have observed that several new features of 802.11n are not supported in link-layer multicast. First, multicast frames can only be transmitted on $20MHz$ channels, thus the high data rates from channel bonding mechanism (40MHz channel) are unavailable for multicast. In addition, short guard interval (SGI) among OFDM symbols cannot be used for multicast frames. As a result, the highest data rate that can be used for multicast in 802.11n WLANs supporting two spatial streams is $130Mbps$, which is much lower than the highest available data rate ($300Mbps$) for unicast frames. Moreover, the enhancement to MAC efficiency brought by the frame aggregation and block acknowledgment mechanisms is unavailable for multicast in 802.11n as well, as multicast frames are unacknowledged at the MAC layer. For a frame of length 1500bytes, the highest throughput for multicast in 802.11n is about $50Mbps$, while the highest throughput for unicast in 802.11n can be over $200Mbps$. Therefore, when the number of multicast clients is small, translating multicast frames to separate unicast frames to each client can be more efficient.

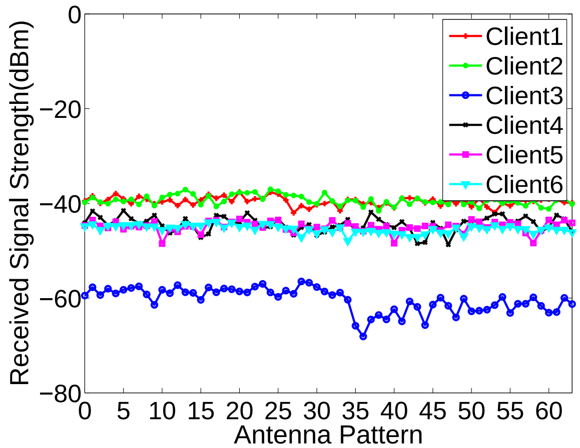


Figure 6.2: Received signal strength of Beacon frames transmitted on various antenna patterns.

6.3.2 Channel Gain of Smart Antennas

We now look at the channel gain of various antenna patterns, by measuring the received signal strengths of Beacon messages transmitted on these antenna patterns. This is equivalent to directly measuring the signal strengths of multicast frames transmitted on these antenna patterns due to the broadcast nature of wireless medium. The interval among Beacon messages is set to $100ms$ and the antenna pattern for Beacons is switched every 60 seconds. The signal strengths of all clients are plotted in Fig. 6.2, where each result is the average value of 600 runs. We can see that for each client, the difference between the highest signal strength and the lowest signal strength is more than $5dB$, which indicates that both the reliability and efficiency of multicast to this client can be boosted if the best antenna pattern for it is used to transmit the multicast frame. The reason is that both packet error rate and applicable data rate are highly related to the signal strength. More importantly, there is no obvious correlation between the signal strengths of any two clients, even if they are in the same direction from the angle of the AP. For example, client 3 and client 4 are in the same directions, but their signal strengths on the same antenna pattern are totally different from each other. Similarly, the signal strengths of client 5 and client 6 do not match with each other either. These observations reveal that clients cannot be simply divided into groups according to their lo-

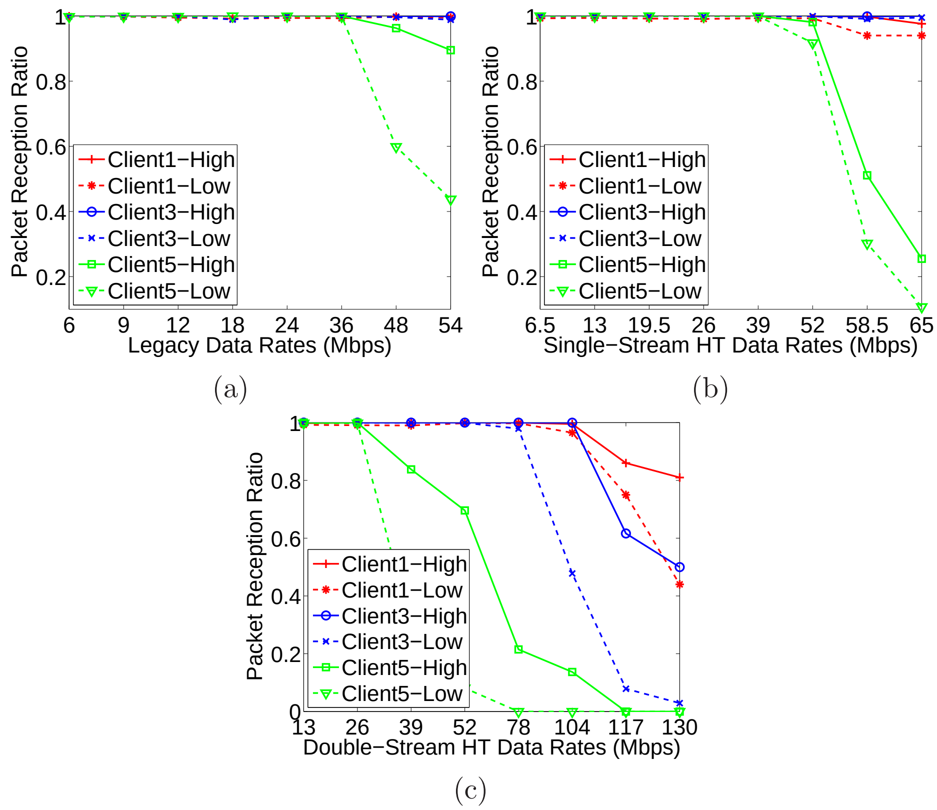


Figure 6.3: Packet reception ratio (PRR) of multicast frames transmitted at different multicast rates. (a) Legacy data rates. (b) Single-stream HT data rates. (c) Double-stream HT data rates.

cations, due to attenuation, reflection and multi-path effects of wireless signals in indoor environments.

6.3.3 PRR of Multicast Frames at Different Multicast Rates

We further examine the packet reception ratio of multicast frames transmitted on different antenna patterns and at various data rates. For each pair of antenna pattern and data rate, the AP transmits a batch of 3000 multicast frames and the length of each frame is 1500bytes. The PRRs of each pair of antenna pattern and data rate are measured at client 1, client 3 and client 5. For each data rate, we only present the highest PRR and the lowest PRR among the results from all antenna patterns. The experiment results are plotted in

Fig. 6.3, where the PRR of legacy data rates, single-stream HT data rates and double-stream data rates is plotted separately for clarity purpose. We can see that the gap between the highest PRR and the lowest PRR for the same client can be large at high data rates, validating the importance of selecting the best antenna pattern for a client. In addition, there is no obvious improvement in PRR by transmitting multicast frames at legacy data rates rather than at single-stream HT data rates. Furthermore, as shown in Fig. 6.3(c), the PRR of multicast at all clients drops dramatically when frames are transmitted at high double-stream data rates, indicating that double-stream data rates are more prone to be affected by interference and variance of channel conditions.

6.4 System Model and Problem Formulation

The above experiment shows that multicast clients cannot be simply categorized by their locations even with smart antennas in indoor environments. Furthermore, single-stream HT rates are more robust than double-stream HT rates for transmitting multicast frames in 802.11n WLANs. Based on these characteristics, we now describe the system model of multicast in 802.11n WLANs with smart antennas and then formulate the problem into an optimization problem.

6.4.1 System Model

We consider an 802.11n WLAN consisting of several APs equipped with smart antennas and a number of clients. To fully explore the benefits of smart antennas for multicast, we will focus on multicast in a single basic service set (BSS), which includes one AP and all clients associated to it. In this system model, we assume that interference from neighboring BSSs can be eliminated by assigning orthogonal channels to adjacent BSSs. In practical systems, the performance degradation caused by interference and collisions from nearby BSSs can be handled by the unicast translator procedure of our on-line algorithm to be presented in Section 6.6.

Suppose that there is a set of clients, C, in a BSS and all clients need to receive the multicast frames from the AP. The AP can transmit the multicast frame at a set of different data rates, denoted by R. We further assume that the smart antennas on the AP are pre-configured with a set of antenna patterns, denoted by P, and only one pattern can be selected to be active for a frame transmission. Note that an antenna pattern can be a composite beam rather than a single-lobe beam, to cope with rich indoor WLAN environments. In addition, the AP does not require channel state information (CSI) from clients to select the antenna pattern, thus all WLAN interfaces complying with the 802.11n standard can be used by the client to receive multicast frames. To provide satisfactory quality of service to upper-layer applications, we define a minimum threshold θ for the packet reception ratio (PRR) of multicast frames. For client c in C, as each antenna pattern may have a unique channel gain, the highest multicast rate that can be used with an antenna pattern without violating the PRR threshold maybe different from that of other antenna patterns. Thus for client c and antenna pattern p, we use a variable $r_{p,c} \in R$ to denote the highest data rate that can be used by antenna p such that the multicast PRR of client c is greater than θ.

6.4.2 Problem Formulation

As discussed earlier, if an omni-directional antenna is used, multicast frames have to be transmitted at a very low rate to ensure the reliability of transmissions to clients that have poor channel conditions. With smart antennas, the same multicast frame can be transmitted multiple times while the aggregated transmission time is still less than that of an omni-directional antenna. Each time an antenna pattern is selected to send the multicast frame to a subset of clients that have high channel gain from the selected antenna, thus the frame can be transmitted at a high data rate. We define a binary variable $x_{p,c}$ to indicate whether antenna pattern p is used to send the multicast frame to client c. If so, $x_{p,c} = 1$. Otherwise, it is zero.

For any client c, it must be covered by at least one antenna pattern to successfully receive multicast frames. This condition can be written as

$$\sum_{p \in P} x_{p,c} \geq 1, \forall c \in C$$

For any antenna pattern p, we further define a binary variable s_p to denote whether it is selected to cover at least one client. This definition can be expressed as

$$x_{p,c} \leq s_p, \forall c \in C; \forall p \in P$$

If an antenna pattern is selected, the multicast frame needs to be transmitted at a data rate that is low enough so that the PRR of all its covered clients is above θ.

Then the multicast data rate on antenna p should be

$$\min_{c \in C}\{r_{p,c} | x_{p,c} = 1\}$$

Suppose the length of the multicast frame is L. The aggregated transmission time for a multicast frame can be given by

$$\sum_{p \in P} \frac{s_p \cdot L}{\min_{c \in C}\{r_{p,c} | x_{p,c} = 1\}}$$

Accordingly, the problem of multicast over smart antennas in 802.11n WLANs can be formulated into the following optimization problem.

Minimize

$$\sum_{p \in P} \frac{s_p \cdot L}{\min_{c \in C}\{r_{p,c} | x_{p,c} = 1\}}$$

Subject to

$$\sum_{p \in P} x_{p,c} \geq 1, \qquad \forall c \in C \tag{6.1}$$

$$x_{p,c} \leq s_p, \ \forall c \in C \qquad \forall p \in P \tag{6.2}$$

The notations used in the formulation and the algorithms to be presented are listed in Table 6.1. In the formulation, Equation (6.1) ensures that every client is covered by at least one antenna pattern; Equation (6.2) guarantees that a multicast frame is transmitted on antenna p if it covers certain clients. Clearly, the problem is a mixed integer program, thus it is time-consuming to solve it by mathematical tools for a large number of clients or antenna patterns.

Table 6.1: List of notations used in link-layer multicast over smart antennas.

C	Set of multicast clients
P	Set of antenna patterns of the smart antenna
R	Set of physical rates used for multicast
L	Average length of multicast frames
θ	Minimum threshold for multicast PRR
$r_{p,c}$	Highest multicast rate for client c if covered by pattern p
$x_{p,c}$	Indicator whether client c is covered by pattern p
s_p	Indicator whether antenna pattern p is selected
M_C	Minimum transmission time for single-group multicast to C
T_C	Minimum transmission time for multi-group multicast to C
B	Batch size for packet reception ratio (PRR) reporting
M	Batch window for unicast translator of the on-line algorithm
$C_{p,r}$	Subset of clients that can receive frames transmitted on p at r
$w_{p,r}$	Scheduling weight for multicast on pattern p at rate r

6.4.3 NP-Hardness

We have the following lemma concerning the NP-hardness of the above mixed integer program.

Lemma 1. *The problem of multicast over smart antennas in 802.11n WLANs is NP-hard.*

Proof. We prove the lemma by reducing from the *bin packing* problem, which has been proved as NP-hard in [93]. The bin packing problem is to pack objects of different volumes into a finite number of bins, where each bin has the same volume V, and the objective is to minimize the number of bins used. Given any set of objects, we construct an auxiliary WLAN with smart antennas, such that the bin packing problem for these objects is the multicast problem in the auxiliary WLAN. We assume that the AP always transmits multicast frames at the same data rate, regardless of the antenna pattern used. In addition, every WLAN client associates with an object in the bin packing problem. For a subset of objects, if their summed volume is less than bin volume V, we check whether there exists an antenna pattern covering their associated clients. If not, we add a new pattern to the smart antennas to cover these clients. Then each antenna pattern corresponds to a bin in the bin packing problem. Recall that the objective of the multicast problem is to minimize the aggregated time to transmit a frame to all multicast clients. As the AP always transmits multicast frame at the same data rate, the transmission time of a multicast frame on each antenna pattern would be the same. Then the objective can be reached by minimizing the number of antenna patterns used to cover all clients. Clearly, the minimum number of bins used in the bin packing problem is equal to the number of antenna patterns used for multicast in the auxiliary WLAN. Thus, the problem of multicast over smart antennas in 802.11n WLANs is NP-hard. $\square$

6.5 Optimal Algorithm

To solve the above mixed integer program, we propose an optimal algorithm in this section. The algorithm derives the partition of clients, antenna pattern and data rate for each group in a bottom-up manner. We assume that the PRRs of various antenna patterns and data rates for all clients can be determined ahead through measurement. We also assume the channel condition for each client varies slowly such that the optimal results can be maintained by running the algorithm periodically.

For a set of clients, C, if the same multicast frame is transmitted only once on a specific antenna pattern, the frame should be sent at a data rate such that the PRR of all clients is above the threshold. We refer to this type of multicast as *single-group multicast* in the rest of the chapter. Let M_C denote the minimum transmission time for single-group multicast, which can be achieved by selecting an antenna pattern, such that the minimum data rate to reach all clients is maximized. Then M_C can be formally expressed as

$$M_C = \frac{L}{\max\limits_{p \in P} \min\limits_{c \in C} \{r_{p,c}\}}$$

where L is the average length of multicast frames.

Clients in C can also be partitioned into multiple disjoint groups and the same multicast frame is transmitted once for each group. In this case, the frame transmission time is the aggregated transmission time for all groups.

We refer to this type of multicast as *multi-group* multicast. Let T_C denote the minimum transmission time of multi-group transmission for set C. Note that M_C is a special case of T_C where all clients in C are in the same group. To determine T_C for set C, we first examine T_C and M_C for set C that has only a few clients.

Suppose there is only one client c_1 in C. Then the minimum transmission time M_C of single-group multicast is the same as the minimum transmission time T_C of multi-group multicast. The minimum transmission time of set $\{c_1\}$ is given by

$$T_{\{c_1\}} = M_{\{c_1\}} = \frac{L}{\max\limits_{p \in P}\{r_{p,c_1}\}} \tag{6.3}$$

In the case that there are two clients c_1 and c_2 in C, then the multicast frame can be transmitted to both clients by either single-group multicast, or by dividing the two clients into two groups and transmitting the frame via multi-group multicast. The minimum transmission time for set $\{c_1, c_2\}$ is the smaller value of single-group multicast and multi-group multicast, which can be expressed by

$$T_{\{c_1,c_2\}} = \min\{M_{\{c_1,c_2\}}, T_{\{c_1\}} + T_{\{c_2\}}\} \tag{6.4}$$

For any set C that has more than two clients, clients can be first partitioned into two subsets K and $C \setminus K$. Then the minimum transmission time T_C of set C is the smaller value of single-group multicast time M_C to all clients, and the minimum value of the aggregated multi-group transmission time T_K and $T_{C \setminus K}$ among all partition combinations. The minimum transmission time T_K for subset K can be further obtained by partitioning K into two smaller subsets K' and $K \setminus K'$. We can see that T_C can be obtained recursively with the termination condition that there are only one or two clients in a subset, whose minimum transmission time can be derived directly by using Equations (6.3) and (6.4). As shown in Section 6.3, there is no obvious correlation among the channel qualities of different clients for the same antenna pattern, or various antenna patterns for a specific client. Thus the minimum transmission time T_C can be obtained by the partitioning strategy. Accordingly, T_C for set C

can be expressed as follows

$$T_C = \min\{M_C, \min\{T_K + T_{C\backslash K} | \forall K \subset C\}\}$$

Then T_C can be determined in a bottom-up manner by taking advantage of this recursive property. First, the minimum transmission time for each client in C can be obtained. After that, the minimum transmission time of any two clients in C can be determined. The transmission time of any three clients in C can be obtained by using the minimum transmission time of a single client and any two clients. Eventually, T_C can be derived after the minimum transmission time of all subsets of C is obtained. The pseudo code of this optimal algorithm is given in Table 6.2.

It is nontrivial to determine T_C for set C, even if the transmission time of all subsets in C has been determined already. Assume that there are n clients in C. T_C can be obtained by partitioning any one, two, even one half of clients from C as one group, and all the remaining clients as the other group. The number of partitioning strategies to be examined could be as high as

$$\sum_{k=0}^{\lfloor n/2 \rfloor} \binom{n}{k} = \begin{cases} 2^{n-1}, \ n \text{ is odd} \\ 2^{n-1} - \frac{1}{2}\binom{n}{n/2}, \ n \text{ is even} \end{cases}$$

In case there are mobile clients or the channel conditions of several clients change rapidly, the partitioning strategies derived from this algorithm may become sub-optimal if the algorithm is not executed frequently. On the other hand, the overhead of PRR measurement and the computing load could be prohibitive when the algorithm is executed too often. Thus we will propose an on-line algorithm in the next section to determine the partitions of clients, antenna pattern and multicast rate of each partition more efficiently.

6.6 On-line Algorithm

In this section, we propose an on-line algorithm for multicast over smart antennas that can be deployed on off-the-shelf WLAN APs and clients with no need to modify the hardware or MAC layer protocol. As shown in Fig. 6.4,

Table 6.2: Optimal algorithm for multicast over smart antennas

Input:
  Set of multicast clients C;
  Set of antenna patterns P;
  Set of multicast rates R;
Output:
  Optimal multicast strategy;
Algorithm:
 for each antenna pattern p in P
  for each multicast rate r in R
   Send a batch of multicast frames on p at rate r;
   Measure the PRR at all clients;
  end for
 end for
 for each subset $K \subset C$
  if there is more than one client in K
   $M_K = \frac{L}{\max\limits_{p \in P} \min\limits_{k \in K} \{r_{p,k}\}}$;
   $T_K = \min\{M_K, \min\{T'_K + T_{K \setminus K'} | K' \subset K\}\}$;
   Store T_K;
  else
   Let k be the only client in K;
   $T_{\{k\}} = M_{\{k\}} = \frac{L}{\max\limits_{p \in P} \{r_{p,k}\}}$;
  end if
 end for

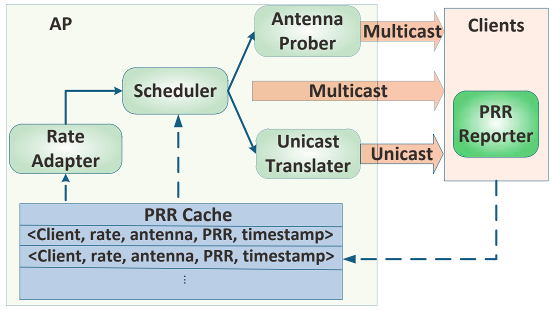


Figure 6.4: Framework of on-line multicast algorithm over smart antennas.

there are five procedures in the on-line algorithm: (1) PRR reporter to send the PRR of multicast frames from clients to the AP; (2) Antenna prober to measure channel qualities of unused antenna patterns in the middle of ongoing multi-group multicast; (3) Scheduler to partition clients into several groups, and to determine the antenna pattern and data rate for each group; (4) Rate adapter to adjust the data rate for each group dynamically according to the PRR reports from clients in the group; (5) Unicast translator to determine for which clients, under what conditions, to translate multicast frames into separate unicast frames.

There is also a PRR cache on the AP to store the PRR information reported from multicast clients. Each entry in the cache has five fields: client address, antenna pattern, data rate, PRR and the time that PRR report is received. The first three fields are used to identify which antenna pattern and at what data rate a batch of multicast frames are transmitted to the client. The timestamp field is used to find stale entries in the cache when cache replacement is necessary. The algorithm and its five procedures will be explained in detail next.

6.6.1 PRR Reporter

As discussed earlier, acknowledgment from clients is critical for reliable and efficient link-layer multicast in WLANs. However, the overhead could be overwhelming if every client acknowledges each multicast frame. Thus in this algorithm, clients send reports of PRR for a batch of multicast frames, rather than for every frame. Specifically, the PRR reporter procedure on clients records the sequence number of all received multicast frames. After transmitting a batch of multicast frames to all clients, the AP sends a *PRR request* frame to each client in a round-robin fashion, including the beginning sequence number and the ending sequence number of the last batch. After receiving a PRR request frame, the client sends a *PRR report* frame to indicate the reception information of multicast frames in the range specified in the PRR request frame. Note that both PRR request frames and PRR report frames are application-layer unicast packets, thus their reliability will be guaranteed automatically by the automatic repeat request mechanism of 802.11. In addition, there is no collision among PRR report frames since each client sends a PRR report frame only after receiving a PRR request frame to it. Furthermore, a client piggybacks the signal strength of the PRR request frame when sending the PRR report frame, such that the scheduler procedure can estimate the potential performance of translating multicast frames into unicast frames for that client.

The batch size for PRR reporting will affect multicast performance. We will determine the optimal batch size through experiments in the next section.

6.6.2 Antenna Prober

When an antenna pattern for a group can no longer provide satisfactory performance, we need to find a new antenna pattern to replace it. However, it is time-consuming to explore all patterns, especially when the number of antenna patterns is large. It is more efficient to probe the channel quality of unused antenna patterns simultaneously along with ongoing multicast transmissions.

Initially, there is no entry in the PRR cache. Thus the antenna prober needs to probe every antenna pattern for all clients to establish basic PRR information for the scheduling procedure. For each antenna pattern, a batch of multicast frames is transmitted to all clients at the basic data rate. Note that the same multicast frame is transmitted only once, as the clients have not been partitioned into groups yet. After an initial probing of all antenna patterns, the scheduler procedure can make scheduling decisions.

After initialization, the AP sends multi-group multicast according to the partitions derived from the scheduler procedure. The AP periodically transmits a batch of multicast frames as single-group multicast on the antenna pattern that has not been probed for the longest time, to update its latest channel quality. The probing frequency is dynamically adapted according to the load of multicast traffic, so as to avoid severely affecting the overall PRR of multicast transmissions. As the data rate for each group adapts dynamically according to channel conditions, the lowest multicast rate among all groups will be used as the data rate for probing batches.

6.6.3 Scheduler

The scheduler procedure partitions clients into multiple groups, and determines the antenna pattern and data rate for each group. The scheduler aims at maximizing the throughput of multicast clients, by minimizing the aggregated transmission time of a frame to all groups for multi-group multicast. We will use a greedy approach to partitioning clients, since the time complexity of exploring all partition combinations can be too high to determine the optimal partition timely.

For an antenna pattern $p \in P$ and a multicast rate $r \in R$, we define a subset $C_{p,r} \subset C$ of clients where the PRR of all clients in $C_{p,r}$ is above the threshold θ, based on information in the PRR cache. Intuitively, an antenna pattern p should be selected if there exists a non-empty subset $C_{p,r}$ where r is a relatively high data rate, e.g., 130Mbps, since the transmission time for clients in $C_{p,r}$ would be short. On the other hand, an antenna pattern p' could be selected as well in case that there are many clients in $C_{p',r'}$, even if the rate r' is relatively low. This is because that the number of transmissions for a multicast frame can be reduced as each antenna pattern covers more clients. Thus for a pair of antenna pattern p and multicast rate r, we define a weight $w_{p,r}$ to determine whether they should be used to transmit multicast frames to clients in $C_{p,r}$. Weight $w_{p,r}$ is defined as the time needed to transmit a frame of length L to clients in $C_{p,r}$, divided by the number of clients in $C_{p,r}$. Weight $w_{p,r}$ can be regarded as the average per-frame transmission time for each client in $C_{p,r}$. The definition can be expressed as

$$w_{p,r} = \frac{L}{r \cdot |C_{p,r}|}$$

Apparently, the lower $w_{p,r}$ is, the less time is needed to transmit a multicast frame to each client in $C_{p,r}$ on average. Thus in this scheduler procedure, pairs of antenna patterns and multicast rates are sorted in an ascending order of their weights. Each time the pair with the smallest weight is selected and all clients in the corresponding subset are added as a multicast group. After that, this group of clients is removed from the set of clients to be scheduled and the weights of all pairs are updated. The above steps are repeated until all clients are covered by an antenna pattern.

6.6.4 Rate Adapter

At the first time multi-group multicast frames are transmitted, the basic data rate is used since only the basic data rate has been used during the initialization of the antenna prober. The multicast data rate of each group should be adapted dynamically such that high data rates can be used as long as the channel condition allows. As revealed by the motivating experiments earlier, among legacy data rates, single-stream HT data rates and double-stream HT data rates, there is no obvious improvement in PRR if legacy data rates instead of single-stream HT data rates are used. In addition, double-stream HT data rates are more sensitive to interference and channel variance than single-stream HT data rates. Thus in this rate adapter procedure, only HT data rates of 802.11n will be used. Moreover, we will give high priority to single-stream HT data rates to enhance the robustness of multicast transmissions.

The rate adapter procedure adapts a data rate for each multicast group based on PRR reports from clients. A higher data rate will be explored for a group if the PRR of the last ten consecutive batches is above θ for that group. The AP sends an exploring batch for the group at single-stream data rate r_{single} which is the next rate higher than current rate $r_{current}$. If the PRR of the exploring batch is above the threshold, the $r_{current}$ of the group is updated to r_{single}. Otherwise, the AP sends an exploring batch at double-stream rate

r_{double} which is the next rate higher than $r_{current}$ but lower than r_{single}. In the case that there is no r'_{single} higher than $r_{current}$, the AP sends an exploring batch at data rate r'_{double} which is the next rate higher than $r_{current}$. Similarly, the $r_{current}$ of the group is updated to r'_{double} if the PRR of the exploring batch is above θ. Otherwise, the current rate $r_{current}$ will be kept for later multicast batches and the number of consecutive satisfactory multicasts for that group will be reset.

On the other hand, if the PRR of two consecutive batches for a group drops below θ, the AP sends an exploring batch for the group at the next single-stream rate which is lower than $r_{current}$. The rate will be further reduced if the PRR is still below the threshold. In the meanwhile, the scheduling procedure will evaluate other partition possibilities and update partitions if necessary.

6.6.5 Unicast Translator

It is more efficient to translate multicast frames into unicast frames for some clients under certain circumstances. First, as discussed in Section 6.3, translating multicast frames into unicast frames in 802.11n WLANs could be more efficient when the number of clients is small, since unicast frames can be transmitted at much higher data rates. Second, for clients that are affected by hidden terminals from other BSSs, their PRR can be consistently lower than the threshold regardless of the antenna pattern or multicast rate used, due to collisions caused by these hidden terminals. Third, for mobile clients it can be inefficient to frequently update the partition, antenna pattern or multicast rate as the channel conditions keep changing. Link-layer multicast frames can be translated into unicast frames easily by replacing the MAC address of the receiver in a frame. The key is to determine when the multicast frames should be translated for a client.

For a specific client, the data rate of unicast can be estimated according to channel qualities piggybacked in PRR report frames. Consequently, the transmission delay of unicast frames translated from multicast frames can be estimated as well. The scheduler can then determine whether it is more efficient to assign the client into a multicast group or to translate the frames into unicast for that client. Furthermore, we use the PRR report for each client in a batch window M to detect the existence of mobile terminals and clients affected by hidden terminals. Specifically, if the PRR of over one half batches of last M batches is lower than the threshold, we assume that the client is a mobile client or it is affected by hidden terminals, and thus translate multicast frames to it into unicast frames. The rationale behind this assumption is that temporary poor performance due to short-term channel variation can be coped with by the scheduler and the rate adapter through regrouping or rate adaptation. The translated unicast frames can be further protected from hidden terminals by enabling the RTS/CTS mechanism. Note that channel qualities of other antenna patterns for these clients will still be examined by the antenna prober. The scheduler stops translating multicast into unicast frames for a client if it is more efficient to put the client back into a multicast group without violating the PRR requirement.

Among all procedures of the on-line algorithm, the scheduler has the highest time complexity. It takes $O(|P| \cdot |R|)$ time to determine the subset and weight for each pair of antenna pattern and data rate. In practice, the number of unique antenna patterns and data rates in the PRR cache is quite limited. Thus the extra work of this on-line algorithm will not affect the normal operation of the AP.

6.7 Experimental Results

In this section, we evaluate the performance of the proposed on-line algorithm through experiments. We define a metric *multicast throughput* as the lowest amount of multicast data received per second by each client to evaluate the on-line algorithm, since it can reflect both the aggregated transmission time per frame, and the overhead of all procedures in the algorithm. We will first examine multicast throughput of the on-line algorithm under different number of clients and various PRR thresholds. We will also investigate the effectiveness of the algorithm when there are mobile clients or hidden terminals that affect some clients. Finally, we study the impact of batch size for PRR reporting on multicast throughput. We will compare the on-line algorithm with the strategy of translating multicast frames into unicast frames for all clients and the strategy of single-group multicast over an omni-directional antenna pattern.

The on-line algorithm is implemented in the testbed shown in Fig. 6.1. Besides the six clients, four more 802.11n clients with omni-directional antennas are placed randomly in the field. A traffic-generating program is deployed on the AP to generate multicast frames to the clients. The length of multicast frames is fixed to 1500 bytes. The data generating rate is set to a value that is higher than the data rate for all groups, such that the AP always has multicast frames to transmit. If not otherwise specified, the minimum PRR threshold is set to 0.9 and the batch size for PRR reporting is 30 frames. The batch window M to detect mobile clients and hidden terminals for the unicast translator procedure is set to 20 batches.

We first study multicast throughput of the on-line algorithm given different number of clients. The number of multicast clients varies from 1 to 10. The experiment results are plotted in Fig. 6.5(a), where on-line, unicast, and multicast-omni represent the on-line algorithm, translating multicast frames into unicast frames for all clients, and one-group multicast over an omni-directional antenna pattern, respectively. We can see that when the number of clients is less than 4, the unicast strategy can achieve a much higher throughput than the multicast-omni strategy. The reason is that unicast frames can be transmitted more efficiently due to frame aggregation and channel bonding mechanisms of 802.11n. In addition, we note that the throughput of the on-line algorithm is slightly lower than that of the unicast strategy, since the antenna prober transmits a batch of one-group multicast frames every few seconds even though the scheduler decides to translate multicast frames into unicast frames for all clients. As the number of clients grows beyond 5, the on-line algorithm outperforms the compared strategies, by taking advantage of both smart antennas and the inherited property of multicast in wireless medium. Moreover, the advantage of the on-line algorithm is more evident when the number of clients is high. When there are 10 clients in the field, the throughput of the on-line algorithm is 65% and 322% higher than that of the unicast and multicast-omni strategies, respectively.

We then examine multicast throughput of the on-line algorithm under different minimum PRR thresholds. The PRR threshold is varied from 0.4 to 0.9 in a step of 0.1. The results are given in Fig. 6.5(b), where the scenarios

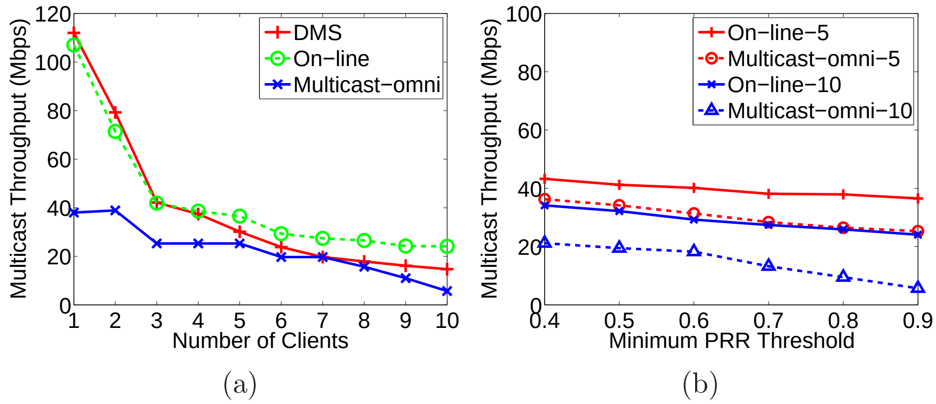


Figure 6.5: Multicast throughput of the on-line algorithm under different numbers of clients and various PRR thresholds. (a) Multicast throughput vs. Number of clients. (b) Multicast throughput vs. Minimum PRR threshold.

of 5 multicast clients and 10 clients are plotted separately. The unicast strategy is not presented in the figure as unicast transmissions are reliable due to retransmissions. It can be observed that as the PRR threshold increases, multicast throughput of both the on-line algorithm and multicast-omni strategy decreases. The reason is that the number of clients that an antenna pattern covers is small when the PRR threshold is high, resulting in high aggregated transmission time per frame. However, the on-line algorithm always outperforms the multicast-omni strategy in terms of multicast throughput, regardless of the PRR threshold or the number of clients. Furthermore, the advantage of the on-line algorithm is more remarkable when the minimum PRR threshold is relatively high, as the multicast-omni strategy has to transmit multicast frames at low data rates to ensure the PRR of all clients above the threshold.

Next, we investigate the performance of the on-line algorithm in the presence of mobile clients or hidden terminals. The number of clients is fixed at 10. Besides the unicast and multicast-omni strategies discussed earlier, we also compare the on-line algorithm with the strategy of disabling the unicast translator procedure of the on-line algorithm. To emulate mobile clients, we carry one or two clients and move in the field at approximately constant speed of 1m/s. Another laptop is carefully placed in the transmission range of client 3 but out of the carrier sense range of the AP to act as a hidden terminal. The experiment results are presented in Fig. 6.6(a), where m-1, m-2, h-1M, and h-2M represent the scenarios of one mobile client, two mobile clients, a hidden terminal with 1Mbps traffic rate and a hidden terminal with 2Mbps traffic rate, respectively. It can be noted that the on-line algorithm always leads to the highest multicast throughput in the presence of mobile clients or hidden terminals. Moreover, the throughput of the on-line algorithm with the unicast translator disabled is close to or slightly lower than that of the unicast strategy in the presence of a hidden terminal. This validates the effectiveness of the unicast translator when some clients are affected by intense interference from hidden terminals.

Finally, we evaluate the impact of batch size for PRR reporting on multicast throughput. The multicast throughput of the on-line algorithm is given in Fig. 6.6(b), where 3-clients, 5-clients and 10-clients stand for the scenarios that there are 3, 5, and 10 multicast clients. We can see that the optimal batch size for PRR reporting is related to the number of multicast clients. For the 3-clients and 5-clients scenarios the highest multicast throughput is achieved when the batch size is 20, while the optimal batch size for the 10-clients scenario is 30 frames. Multicast throughput drops for all scenarios when the batch size is smaller than 20, as a non-negligible percent of medium access time is occupied by the PRR request/report frames after each batch. On the other hand, multicast throughput also declines when the batch size grows beyond 40. This is because that there is a high probability that the PRR of a batch is lower than the threshold due to channel variance when the batch size is large. Then the rate adapter procedure is prone to reduce the multicast data rate for a group to boost the PRR of later transmissions. As a result, multicast throughput becomes lower.

6.8 Conclusions

In this chapter, we have studied the problem of multicast in 802.11n WLANs with smart antennas. Our objective is to minimize the transmission time of per multicast frame and thus maximize the multicast throughput, without sacrificing the PRR of clients. The method is to partition clients into multiple groups and transmit the same frame once for each group. An antenna pattern is selected for each group such that the frame can be transmitted at a high data

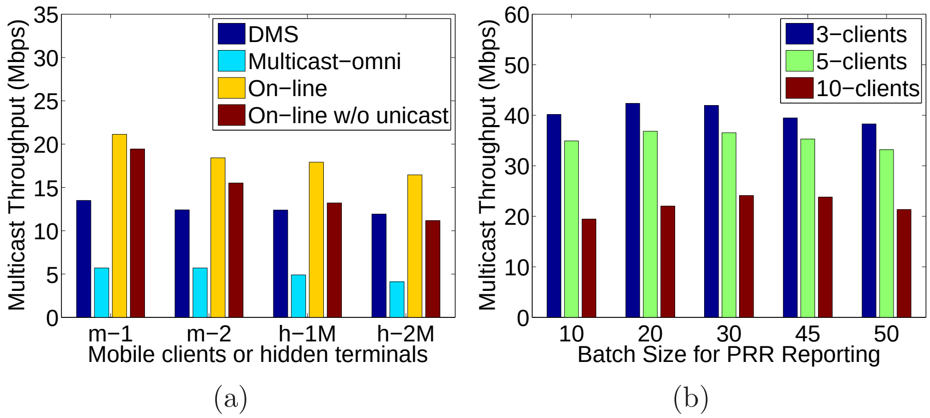


Figure 6.6: Multicast throughput of the on-line algorithm in the presence of mobile clients or hidden terminals, and the impact of batch size for PRR reporting. (a) Mobile clients and hidden terminals. (b) Impact of batch size.

rate. We formulated the problem into a mixed integer program and provided an algorithm to obtain optimal solutions. For dynamic network conditions, we have also provided an on-line algorithm that can be applied to deployed WLANs with no need to modify hardware or 802.11 MAC protocol. We have implemented the on-line algorithm on off-the-shelf WLAN products and conducted extensive experiments to evaluate its performance. The results shown that under various network scenarios, our proposed on-line algorithm can significantly improve multicast throughput compared with other strategies, while guaranteeing satisfactory PRR for all clients. In our future work, we will further evaluate the proposed algorithms in large-scale WLANs via simulations. In addition, we will study the joint problem of AP association and multicast over smart antennas in WLANs where APs are densely deployed, to enhance multicast performance and alleviate interference among nearby APs.

Chapter 7

Conclusions

This dissertation focuses its study on performance optimization in large-scale 802.11n wireless local area networks. A suite of resource allocation algorithms, protocol designs, and system implementation has been presented to identify and address several critical issues in this field. Specifically, first, an AP association algorithm has been proposed for 802.11n WLANs with heterogeneous clients. It ensures that the throughput of each client is proportional to its physical data rate, so as to improve network throughput. In addition, another simple yet effective AP association algorithm has been devised to boost network throughput, by associating different types of clients to different APs. Second, a distributed channel assignment algorithm has been presented for 802.11n WLANs with heterogeneous clients. In the algorithm, each AP updates its channel iteratively based on the estimated local network throughput, so as to maximize the network throughput. Moreover, another low-complexity channel assignment algorithm has also been introduced in the dissertation. This algorithm achieves high network throughput by minimizing the interference experienced by high-rate clients. Third, a cooperative retransmission protocol is designed to boost the efficiency of frame retransmissions in 802.11n WLANs. It takes advantage of the broadcast nature and spatial diversity of wireless medium. Each WLAN station dynamically selects the neighbor that has the best channel condition to help retransmit the failed sub-frames in an aggregated frame, so as to reduce the number of retransmissions and utilize higher data rates for retransmissions. Fourth, collision-free high-throughput client polling is considered for WLANs operating in the point coordination function mode. An algorithm is proposed to coordinate the client polling across BSSs, in order to eliminate collisions among nearby BSSs. It provisions high network capacity by employing an accurate interference model and reusing idle time slots. Finally, link-layer multicast over smart antennas is studied for 802.11n WLANs. As a specific antenna pattern of a smart antenna can greatly enhance the signal quality of a subset of clients, the multicast performance can be improved by transmitting each multicast frame several times at high data rates, while over different antenna patterns. Both optimal and on-line algorithms are presented to partition clients into subsets, choose the antenna pattern and data rate for each subset, so as to optimize link-layer multicast performance in WLANs.

To summarize, in this dissertation, we have carried out comprehensive studies on performance optimization in large-scale 802.11n WLANs. We have proposed efficient solutions to improve WLAN performance from both theoretical and systematic points of view, by combining algorithm and protocol design, mathematical modeling, theoretical analysis, simulation evaluation, and experiment validation techniques. The outcome of this research can be applicable to the widely deployed 802.11n WLANs, as well as the forthcoming 802.11ac WLANs. Our research would have a significant impact on both fundamental research on performance optimization in WLANs, and industrial development of WLAN products.

Bibliography

[1] IEEE 802.11n Standard, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Enhancements for Higher Throughput,", *IEEE*, Oct. 2009.

[2] M. Balazinska and P. Castro, "Characterizing Mobility and Network Usage in a Corporate Wireless Local-Area Network," *1st International Conference on Mobile Systems, Applications and Services*, pp. 303-316, May 2003.

[3] D. Schwab and R. Bunt, "Characterising the Use of a Campus Wireless Network," *IEEE INFOCOM*, pp. 862-870, Mar. 2004. vol. 7, no. 10, pp. 3748-3758, Oct. 2008.

[4] Y. Bejerano, S. Han and L. Li, "Fairness and Load Balancing in Wireless LANs Using Association Control," *IEEE/ACM Trans. Networking*, vol. 15, no. 3, pp. 560-573, June 2007.

[5] H. Gong, K. Nahm and J.W. Kim, "Distributed Fair Access Point Selection for Multi-Rate IEEE 802.11 WLANs," *5th IEEE Consumer Communications and Networking Conference*, pp. 528-532, Jan. 2008.

[6] O. Ercetin, "Association Games in IEEE 802.11 Wireless Local Area Networks," *IEEE Trans. Wireless Communications*, vol. 7, no. 12, pp. 5136-5143, Dec. 2008.

[7] W. Xu, C. Hua and A. Huang, "A Game Theoretical Approach for Load Balancing User Association in 802.11 Wireless Networks," *IEEE GLOBECOM*, pp. 1-5, Dec. 2010.

[8] P. BK and J. Kuri, "An Estimated Delay Based Association Policy for Web Browsing in a Multirate WLAN," *IEEE Trans. Network and Service Management*, pp. 1-13, June 2012.

[9] L. Du, Y. Bai and L. Chen, "Access Point Selection Strategy for Large-Scale Wireless Local Area Networks," *IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 2161-2166, Mar. 2007.

[10] S. Keranidis, T. Korakis, I. Koutsopoulos and L. Tassiulas, "Contention and Traffic Load-Aware Association in IEEE 802.11 WLANs: Algorithms and Implementation," *International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, pp. 334-341, May 2011.

[11] G.S. Kasbekar, P. Nuggehalli and J. Kuri, "Online Client-AP Association in WLANs," *4th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, pp. 1-8, Apr. 2006.

[12] P. Bahl, M. T. Hajiaghayi, K. Jain, S. V. Mirrokni, L. Qiu and A. Saberi, "Cell Breathing in Wireless LANs: Algorithms and Evaluation," *IEEE Trans. Mobile Computing*, vol. 6, no. 2, pp. 164-178, Feb. 2007.

[13] B. Kauffmann, F. Baccelli, A. Chaintreau, V. Mhatre, K. Papagiannaki and C. Diot, "Measurement-Based Self Organization of Interfering 802.11 Wireless Access Networks," *IEEE INFOCOM*, pp. 1451-1459, May 2007.

[14] W. Xu, C. Hua and A. Huang, "Channel Assignment and User Association Game in Dense 802.11 Wireless Networks," *IEEE ICC*, pp. 1-5, June 2011.

[15] T. Bonald, A. Ibrahim and J. Roberts, "The Impact of Association on the Capacity of WLANs," *7th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOPT)*, pp. 1-10, June 2009.

[16] C. Huang and P. Ramanathan, "Joint Association and Power Adaption in WiFi Community Networks," *45th Annual Conference on Information Sciences and Systems*, pp. 1-6, Mar. 2011.

[17] H. Ko, J. Shin, D. Kwak and C. Kim, "A Joint Approach to Bandwidth Allocation and AP-Client Association for WLANs," *35th IEEE Local Computer Networks*, pp. 576-581, Oct. 2010.

[18] L. Li, M. Pal and Y. R. Yang, "Proportional Fairness in Multi-Rate Wireless LANs," *IEEE INFOCOM*, pp. 1004-1012, Apr. 2008.

[19] S. C. Liew and Y. J. Zhang, "Proportional Fairness in Multi-Channel Multi-Rate Wireless Networks-Part I: The Case of Deterministic Channels with Application to AP Association Problem in Large-Scale WLAN,"

IEEE Trans. Wireless Communications, vol. 7, no. 9, pp. 3446-3456, Sep. 2008.

[20] K. Hung, B. Bensaou and R. Li, "Cross Layer Association Control for Throughput Optimization in Wireless LANs with Inter-AP Interference," *13th ACM MSWIM*, pp. 210-217, Oct. 2010.

[21] G. Bhanage, D. Vete, I. Seskar, and D. Raychaudhuri, "SplitAP: Leveraging Wireless Network Virtualization for Flexible Sharing of WLANs," *IEEE GLOBECOM*, pp.1-6, Dec. 2010.

[22] C. Chou, K. G. Shin and S. Shankar. N, "Contention-Based Airtime Usage Control in Multirate IEEE 802.11 Wireless LANs," *IEEE/ACM Trans. Networking*, vol. 14, no. 6, pp. 1179-1192, Dec. 2006.

[23] T. Joshi, A. Mukherjee, Y. Yoo and D. P. Agrawal, "Airtime Fairness for IEEE 802.11 Multirate Networks," *IEEE Trans. Mobile Computing*, vol. 7, no. 4, pp. 513-527, April 2008.

[24] T. Selvam and S. Srikanth, "Performance Study of IEEE 802.11n WLANs," *First International Communication Systems and Networks and Workshops (COMSNETS), 2009*, pp. 1-6, Jan. 2009.

[25] V. Shrivastava, S. Rayanchu, J. Yoon and S. Banerjee, "802.11n Under the Microscope," *8th ACM SIGCOMM Conference on Internet Measurement*, pp. 105-110, Oct. 2008.

[26] M. Laddomada, F. Mesiti, M. Mondin and F. Daneshgaran, "On the Throughput Performance of Multirate IEEE 802.11 Networks with Variable-Loaded Stations: Analysis, Modeling, and a Novel Proportional Fairness Criterion," *IEEE Trans. Wireless Communications*, vol. 9, no. 5, pp. 1594-1607, May 2010.

[27] R. Jain, D. Chiu and W. Hawe, "A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Systems," *DEC Research Report TR-301*, Sept. 1984.

[28] J. Kim, S. Kim, S. Choi and D. Qiao, "CARA: Collision-Aware Rate Adaptation for IEEE 802.11 WLANs," *IEEE INFOCOM*, pp. 1-11, April 2006.

[29] K. K. Leung and B. J. Kim, "Frequency Assignment for IEEE 802.11 Wireless Networks," *IEEE Vehicle Technology Conference*, pp. 1422-1426, Oct. 2003.

[30] M. Yu, H. Luo and K. K. Leung, "A Dynamic Radio Resource Management Technique for Multiple APs in WLANs," *IEEE Transactions on Wireless Communications*, vol. 5, no. 7, pp. 1910-1919, July 2006.

[31] A. Mishra, S. Banerjee and W. Arbaugh, "Weighted Coloring Based Channel Assignment for WLANs," *ACM SIGMOBILE*, vol. 9, no. 3, pp. 19-31, July 2005.

[32] Y. Liu, W. Wu, B. Wang, T. He, S. Yi and Y. Xia, "Measurement-Based Channel Management in WLANs," *IEEE WCNC*, pp. 1-6, April 2010.

[33] A. Mishra, V. Brik, S. Banerjee, A. Srinivasan and W. Arbaugh, "A Client-Driven Approach for Channel Management in Wireless LANs," *IEEE INFOCOM*, pp. 1-12, April 2006.

[34] J. K. Chen, G. de Veciana and T. S. Rappaport, "Improved Measurement-Based Frequency Allocation Algorithms for Wireless Networks," *IEEE GLOBECOM*, pp. 4790-4795, Nov. 2007.

[35] X. Luo, R. Iyengar and K. Kar, "Channel Assignment for Maximum Throughput in Multi-Channel Access Point Networks," *IEEE WCNC*, pp. 395-400, March 2007.

[36] M. Heusse, F. Rousseau, G. Berger-Sabbatel and A. Duda, "Performance Anomaly of 802.11b," *IEEE INFOCOM*, pp. 836-843, Apr. 2003.

[37] S. Chieochan, E. Hossain and J. Diamond, "Channel Assignment Schemes for Infrastructure-based 802.11 WLANs: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 12, no. 1, pp. 124-136, First Quarter 2010.

[38] A. Boukerche, S. Hong and T. Jacob, "A Distributed Algorithm for Dynamic Channel Allocation," *Mobile Networks and Applications*, vol. 7, no. 2, pp. 115-126, April 2002.

[39] A. Boukerche, "Handbook of Algorithms for Wireless Networking and Mobile Computing," *CRC Press*, Nov. 2005.

[40] M. Arslan, K. Pelechrinis, I. Broustis, S. Krishnamurthy, S. Addepalli and K. Papagiannaki, "Auto-Configuration of 802.11n WLANs," *ACM International Conference on Emerging Networking Experiments and Technologies (CoNext)*, no. 27, Nov. 2010.

[41] S. Fiehe, J. Riihijvi and P. Mahonen, "Experimental Study on Performance of IEEE 802.11n and Impact of Interferers on the 2.4 GHz ISM

Band," *ACM 6th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 47-51, June 2010.

[42] L. Deek, E. Garcia-Villegas, E. Belding, S. Lee and K. Almeroth, "The Impact of Channel Bonding on 802.11n Network management," *ACM International Conference on Emerging Networking Experiments and Technologies (CoNext)*, no. 11, Dec. 2011.

[43] Y. Lin and V.W.S. Wong, "WSN01-1: Frame Aggregation and Optimal Frame Size Adaptation for IEEE 802.11n WLANs," *IEEE GLOBECOM*, pp. 1-6, Nov. 2006.

[44] B. Kim, H. Hwang and D. Sung, "Effect of Frame Aggregation on the Throughput Performance of IEEE 802.11n," *IEEE WCNC*, pp. 1740-1744, March 2008.

[45] W. Liu, C. Huang and K. Feng, "Performance Analysis of Block Acknowledgement Mechanisms for Next Generation Wireless Networks," *IEEE WCNC*, pp. 1-6, April 2010.

[46] I. Pefkianakis, Y. Hu, S. H. Y. Wong, H. Yang and S. Lu, "MIMO rate adaptation in 802.11n wireless networks," *ACM MobiCom*, pp. 257-268, Sept. 2010.

[47] AirMagnet, "Impact of Legacy Devices on 802.11n Networks," *AirMagnet Whitepaper*, March, 2008.

[48] E. Weingartner, H. V. Lehn and K. Wehrle, "A Performance Comparison of Recent Network Simulators," *IEEE ICC*, pp. 1-5, June 2009.

[49] A. Sendonaris, E. Erkip and B. Aazhang, "User Cooperation Diversity– Part I: System description," *IEEE Trans. Comm.*, vol. 51, pp. 1927-38, 2003.

[50] J. N. Laneman, D. N. C. Tse and G. W. Wornell, "Cooperative Diversity in Wireless Networks: Efficient Protocols and Outage Behavior," *IEEE Trans. Information Theory*, vol. 50, pp. 3062-3080, 2004.

[51] H. Zhu and G. Cao, "rDCF: A Relay-Enabled Medium Access Control Protocol for Wireless Ad Hoc Networks," *IEEE Trans. Mobile Computing*, vol. 5 pp. 1201-1214, 2006.

[52] P. Liu, Z. Tao, S. Narayanan, T. Korakis and S. S. Panwar, "CoopMAC: A Cooperative MAC for Wireless LANs," *IEEE JSAC*, vol. 25, pp. 340-354, 2007.

[53] T. Guo and R. Carrasco, "CRBAR: Cooperative Relay-based Auto Rate MAC for Multirate Wireless Networks," *IEEE Trans. Wireless Communications*, vol. 8 pp. 5938-5947, 2009.

[54] L. Guo, X. Ding, H. Wang, Q. Li, S. Chen and X. Zhang, "Cooperative Relay Service in a Wireless LAN," *IEEE JSAC*, vol. 25, pp. 355-368, 2007.

[55] X. He and F. Y. Li, "An Automatic Cooperative Retransmission MAC protocol in Wireless Local Area Networks," *European Wireless Conference*, pp. 228-233, 2009.

[56] S. Moh and C. Yu, "A Cooperative Diversity-Based Robust MAC Protocol in Wireless Ad Hoc Networks," *IEEE Trans. Parallel and Distributed Systems*, pp. 353-363, 2011.

[57] F. A. Onat, D. Avidor, "Relay-Assisted ARQ in Wireless Ad-Hoc Networks," *IEEE WCNC*, pp. 500-505, 2008.

[58] G. N. Shirazi, P. Kong and C. K. Tham, "A Cooperative Retransmission Scheme in Wireless Networks with Imperfect Channel State Information," *IEEE WCNC*, pp. 1-6, 2009.

[59] M. Dianati, X. Ling, K. Naik and X. Shen, "A Node-Cooperative ARQ Scheme for Wireless Ad Hoc Networks," *IEEE Trans. Vehicular Technology*, vol. 55, pp. 1032-1044, 2006.

[60] T. Guo, R. Carrasco and W. Woo, "A Differentiated Cooperative MAC for QoS enhancement in Wireless LANs," *IEEE PIMRC*, pp. 1-5, 2008.

[61] X. He and F. Y. Li, "A Multi-Relay Cooperative Automatic Repeat Request Protocol in Wireless Networks," *IEEE ICC*, pp. 1-6, 2010.

[62] J.D. Morillo-Pozo, J. Garcia-Vidal, "A Low Coordination Overhead C-ARQ Protocol with Frame Combining," *IEEE PIMRC*, pp.1-5, 2007.

[63] Y. Qi, R. Hoshyar, M. A. Imran, R. Tafazolli, "H2-ARQ-Relaying: Spectrum and Energy Efficiency Perspectives," *IEEE JSAC*, pp. 1547-1558, 2011.

[64] M. Lu, P. Steenkiste, T. Chen, "CHARQ: Cooperative Hybrid ARQ for Wireless Video Streaming," *IEEE International Conference on Multimedia and Expo*, pp. 750-753, 2009.

[65] K. Lu, S. Fu and Y. Qian, "Increasing the Throughput of Wireless LANs Via Cooperative Retransmission," *IEEE GLOBECOM*, pp. 5231-5235, 2007.

[66] IEEE 802.11e Standard, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Amendment: Medium Access Method (MAC) Quality of Service Enhancements," *IEEE*, 2005.

[67] J. N. Al-Karaki and J. M. Chang, "EPCF: a Lightweight Multi-priority PCF for QoS Support in IEEE 802.11 Wireless LANs," *IEEE International Conference on Performance, Computing and Communications*, pp. 495-502, 2004.

[68] D. D. Vergados and D. J. Vergados, "Synchronization of Multiple Access Points in the IEEE 802.11 Point Coordination Function," *Vehicular Technology Conference IEEE 60th*, vol. 2, pp. 1073-1077, 2004.

[69] Y. Bejerano and R. S. Bhatia, "MiFi: A Framework for Fairness and QoS Assurance for Current IEEE 802.11 Networks with Multiple Access Points," *IEEE/ACM Transactions on Networking*, vol. 14, pp. 849-862, 2006.

[70] D. J. Vergados, D. D. Vergados and C. Douligeris, "DPS: An Architecture for VBR Scheduling in IEEE 802.11e HCCA Networks with Multiple Access Points," *Journal of Mobile Networks and Applications*, vol. 14, pp. 744-759, 2009.

[71] D. Dailey, "Uniqueness of Colorability and Colorability of Planar 4-regular Graphs are NP-complete," *Discrete Mathematics*, vol. 30, pp. 289-293, 1980.

[72] F. T. Leighton, "A Graph Coloring Algorithm for Large Scheduling Problem," *Journal of Research of the National Bureau of Standards*, pp. 489-503, 1979.

[73] D. Brèlaz, "New Methods to Color the Vertices of a Graph," *Communications of the Assoc. of Comput. Machinery*, vol. 22, pp 251-256, 1979. W. Klotz, "Graph Coloring Algorithms," *Mathematical Report*, pp. 1-9, 2002.

[74] R. M. Karp, "Reducibility Among Combinatorial Problems," *Complexity of Computer Computations*, pp. 85-103, 1972.

[75] C. Bron and J. Kerbosch, "Algorithm 457: Finding All Cliques of an Undirected Graph," *Communications of the ACM*, vol. 16, pp. 575-579, 1973.

[76] K. Tang And M. Gerla, "MAC Reliable Broadcast in Ad Hoc Networks," *IEEE MILCOM*, pp. 1008-1013 Aug. 2001.

[77] M. Sun, L. Huang, A. Arora And T. Lai, "Reliable MAC Layer Multicast in IEEE 802.11 Wireless Networks," *International Conference On Parallel Processing*, pp. 527-536, Aug. 2002.

[78] X. Wang, L. Wang, Y. Wang And D. Gu, "Reliable Multicast Mechanism in WLAN with Extended Implicit MAC Acknowledgment," *IEEE Vehicular Technology Conference*, pp. 2695-2699, May 2008.

[79] Y. Tanigawa, K. Yasukawa And K. Yamaoka, "Transparent Unicast Translation to Improve Quality of Multicast over Wireless LAN," *IEEE Consumer Communications and Networking Conference (CCNC)*, pp. 1-5, Jan. 2010.

[80] J. Kuri And S. K. Kasera, "Reliable Multicast in Multi-Access Wireless LANs," *IEEE INFOCOM*, pp. 760-767, March 1999.

[81] D. Dujovne And T. Turletti, "Multicast in 802.11 WLANs: An Experimental Study," *9th ACM International Symposium On Modeling Analysis and Simulation of Wireless and Mobile Systems*, pp. 130-138, Oct. 2006.

[82] C. Bao And W. Liao, "Performance Analysis of Reliable MAC-Layer Multicast for IEEE 802.11 Wireless LANs," *IEEE ICC* pp. 1378-1382 May 2005.

[83] R. Chandra, S. Karanth, T. Moscibroda, V. Navda, J. Padhye, R. Ramjee AndL. Ravindranath, "DirCast: A Practical and Efficient Wi-Fi Multicast System," *IEEE ICNP*, pp. 161-170, Oct. 2009.

[84] S. Choi, N. Choi, Y. Seok, T. Kwon And Y. Choi, "Leader-Based Rate Adaptive Multicasting for Wireless LANs," *IEEE GLOBECOM*, pp. 3656-3660, Nov. 2007.

[85] S. Bellofiore, C. A. Balanis, J. Foutz And A. S. Spanias, "Smart-Antenna Systems for Mobile Communication Networks. Part 1. Overview And Antenna Design," *IEEE Antennas and Propagation Magazine*, vol. 44, pp. 145-154, Jun. 2002.

[86] "Ruckus Wireless", *http://www.ruckuswireless.com.*

[87] S. Sen, J. Xiong, R. Ghosh And R. R. Choudhury, "Link Layer Multicasting with Smart Antennas: No Client Left Behind," *IEEE ICNP*, pp. 53-62, Oct. 2008.

[88] K. Sundaresan, K. Ramachandran And S. Rangarajan, "Optimal Beam Scheduling for Multicasting in Wireless Networks," *ACM MOBICOM*, pp. 205-216, Sep. 2009.

[89] H. Zhang, Y. Jiang, K. Sundaresan, S. Rangarajan And B. Zhao, "Wireless Data Multicasting with Switched Beamforming Antennas," *IEEE INFOCOM*, pp. 526-530, April 2011.

[90] E. Aryafar, A. Khojastepour, K. Sundaresan, S. Rangarajan And E. Knightly, "ADAM: An Adaptive Beamforming System for Multicasting in Wireless LANs," *IEEE INFOCOM*, March 2012.

[91] A. Lyakhov, V. Vishnevsky And M. Yakimov, "Multicast QoS Support in IEEE 802.11 WLANs," *IEEE MASS*, pp. 1-3, Oct. 2007.

[92] IEEE 802.11aa Amendment, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 2: MAC Enhancements for Robust Audio Video Streaming," *IEEE*, 2012.

[93] R. J. Fowler, M. S. Paterson And S. L. Tanimoto, "Optimal Packing and Covering in the Plane Are NP-Complete," *Information Processing Letter*, pp. 133-137, vol. 12, no. 3, June 1981.

[94] X. Wang. L. Wang, Y. Wang, Y. Zhang And A. Yamada, "Supporting MAC Layer Multicast in IEEE 802.11n: Issues And Solutions," *IEEE WCNC*, pp. 1-6, April 2009.

[95] W. Lim, D. Kim And Y. Suh, "Design of Efficient Multicast Protocol for IEEE 802.11n WLANs and Cross-Layer Optimization for Scalable Video Streaming," *Mobile Computing, IEEE Transactions on*, no. 99, April 2011.

[96] S. Lakshmanan, S. Sanadhya and R. Sivakumar, "On Link Rate Adaptation in 802.11n WLANs," *IEEE INFOCOM*, pp. 366-370, April 2011.

List of Publications

Journal Publications

- D. Gong, Y. Yang, "Link-Layer Multicast in Large-Scale 802.11n WLANs with Smart Antennas," under submission.

- D. Gong, M. Zhao and Y. Yang, "A Multi-Channel Cooperative MIMO MAC Protocol for Clustered Wireless Sensor Networks," under submission.

- M. Zhao, D. Gong and Y. Yang, "Cost Minimization for Mobile Data Gathering in Wireless Sensor Networks," under submission.

- D. Gong, Y. Yang, "Low-Latency SINR-based Data Gathering in Wireless Sensor Networks," to appear in *IEEE Transactions on Wireless Communications*, 2014.

- D. Gong, M. Zhao and Y. Yang, "Distributed Channel Assignment Algorithms for 802.11n WLANs with Heterogeneous Clients," *Journal of Parallel and Distributed Computing (JPDC)*, vol. 74, no. 5, pp. 2365-2379, May 2014.

- D. Gong, Y. Yang and Z. Pan, "Energy-Efficient Clustering in Lossy Wireless Sensor Networks, *Journal of Parallel and Distributed Computing (JPDC)*, vol. 73, no. 9, pp. 1323-1336, Sept. 2013.

- D. Gong and Y. Yang, "On-Line AP Association Algorithms for 802.11n WLANs with Heterogeneous Clients," to appear in *IEEE Transactions on Computers*, 2013.

Conference Publications

- D. Gong and Y. Yang, "Achieving Fair and Flexible Airtime Allocation in WLANs for Mobile Data Offloading," under submission.
- D. Gong, Y. Yang and H. Li, "An Efficient Cooperative Retransmission MAC Protocol for IEEE 802.11n Wireless LANs," *The 10th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE MASS)*, accepted, Oct. 2013.
- D. Gong, Y. Yang and H. Li, "Link-layer Multicast in Smart Antenna Based 802.11n Wireless LANs," *The 10th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE MASS)*, accepted, Oct. 2013.
- D. Gong and Y. Yang, "Distributed Algorithms for Joint Routing and Frame Aggregation in 802.11n Wireless Mesh Networks," *The 27th IEEE International Parallel & Distributed Processing Symposium (IEEE IPDPS)*, pp. 1122-1132, May 2013.
- D. Gong and Y. Yang, "Low-Latency SINR-Based Data Gathering in Wireless Sensor Networks," *The 32nd IEEE International Conference on Computer Communications (IEEE INFOCOM)*, pp. 1941-1949, April 2013.
- D. Gong, M. Zhao and Y. Yang, "Channel Assignment in Multi-Rate 802.11n WLANs", *IEEE Wireless Communications and Networking Conference (IEEE WCNC)*, pp. 392-397, April 2013.
- D. Gong and Y. Yang, "AP Association in 802.11n WLANs with Heterogeneous Clients", *The 31nd IEEE International Conference on Computer Communications (IEEE INFOCOM)*, pp. 1440-1448, March 2012.
- D. Gong, Y. Yang and H. Li, "High-Throughput Collision-Free Client Polling in Multi-AP WLANs", *IEEE Global Communications Conference (IEEE GLOBECOM)*, pp. 1-6, Nov. 2011.

- D. Gong, M. Zhao and Y. Yang, "Topology Control and Channel Assignment in Lossy Wireless Sensor Networks", *The 23rd International Teletraffic Congress (ITC 23)*, pp. 222-229, Sep. 2011.
- D. Gong, M. Zhao and Y. Yang, "A Multi-Channel Cooperative MIMO MAC Protocol for Wireless Sensor Networks", *The 7th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE MASS)*, pp. 11-20, Nov. 2010.
- M. Zhao, D. Gong and Y. Yang, "A Cost Minimization Algorithm for Mobile Data Gathering in Wireless Sensor Networks," *The 7th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE MASS)*, pp. 322-331, Nov. 2010.
- D. Gong, M. Zhao and Y. Yang, "Joint Channel Assignment and Space-Division Multiple Access Scheduling in Wireless Mesh Networks", *IEEE Wireless Communications and Networking Conference (IEEE WCNC)*, pp. 1-6, April 2010.
- Z. Pan, Y. Yang and D. Gong, "Distributed Clustering Algorithms for Lossy Wireless Sensor Networks," *The 9th International Symposium on Network Computing and Applications (NCA)*, pp. 36-43, July, 2010.