Access Point Association Coordination in Densely Deployed 802.11 Wireless Networks

by

Phillip Babatunde Oni

A thesis submitted to the
Department of Electrical and Computer Engineering
in conformity with the requirements for
the degree of Master of Applied Science

Queen's University
Kingston, Ontario, Canada
August 2015

Abstract

Dense deployment of wireless local area network (WLAN) access points (APs) is an important part of the next generation Wi-Fi and standardization (802.11ax) efforts are underway. Increasing demand for WLAN connectivity motivates such dense deployments, especially in geographical areas with large numbers of users, such as stadiums, large enterprises, multi-tenant buildings, and urban cities. Although densification of WLAN APs guarantees coverage, it is susceptible to increased interference and uncoordinated association of stations (STAs) to APs, which degrade network throughput. Therefore, to improve network throughput, algorithms are proposed in this thesis to optimally coordinate AP associations in the presence of interference.

In essence, coordination of APs in dense WLANs (DWLANs) is achieved through coordination of STAs' associations with APs. While existing approaches suggest tuning of APs' beacon powers or using transmit power control (TPC) for association control, here, the signal-to-interference-plus-noise ratio (SINRs) of STAs and the clear channel assessment (CCA) threshold of the 802.11 MAC protocol are employed. The proposed algorithms in this thesis enhance throughput and minimize coverage holes inherent in cell breathing and TPC techniques by not altering the transmit powers of APs, which determine cell coverage. Besides uncoordinated AP associations, unnecessary frequent transmission deferment is envisaged as another problem in DWLANs due to the clear channel assessment aspect of the carrier sensing multiple access collision avoidance (CSMA/CA) scheme in 802.11 standards and the short spatial reuse distance between co-channel APs. To address this problem in addition to AP association coordination, an algorithm is proposed for CCA threshold adjustment in each AP cell, such that CCA threshold used in one cell mitigates transmission deferment in neighboring cells.

Performance evaluation reveals that the proposed association optimization algorithms achieve significant gain in throughput when compared with the default strongest signal first (SSF) association scheme in the current 802.11 standard. Also, further gain in throughput is observed when the CCA threshold adjustment is combined with the optimized association. Results show that when STA-AP association is optimized and CCA threshold is adjusted in each cell, throughput improves. Finally, transmission delay and the number of packet re-transmissions due to collision and contention significantly decrease.

Acknowledgments

Firstly, thanks to the Almighty God, the author and finisher of my faith, for the gift of life, strength, and sound health to complete this work. My profound gratitude goes to my supervisor Prof. Steven D. Blostein for his guidance and support throughout the course of this work; I really appreciate his unrelenting feedback, patience, and supervision. Also, I would like to thank the administrative staffs whose guidance contributed to my academic success. Finally, my heartfelt appreciation goes to my family, especially my parents (Deacon James Oni and Mrs. Felicia Oni) and friends for their support and encouragements. Lastly, I appreciate my lab colleague, Ye Li for his encouragement and valuable discussion on the problem addressed in this thesis. God bless and keep you all.

Contents

List of Tables

List of Figures

List of Abbreviations

ACK	Acknowledgment
AP	Access Point
BEB	Binary Exponential Backoff
BER	Bit Error Rate
BSS	Basic Service Set
CCA	Clear Channel Assessment
CD	Collision Detection
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CSR	Carrier Sensing Range
CW	Contention Window
CTS	Clear to Send
DSC	Dynamic Sensitivity Control
DS	Distribution System
DL	Downlink
DWLAN	Dense Wireless Local Area Networks
ESS	Extended Service Set
ET	Experimental Topology

FEP	Frame Exchange Protocol
HEWSG	High Efficiency WLAN Study Group
HTTP	Hypertext Transfer Protocol
ISM	Industrial, Scientific and Medical Frequency Band
ITU	International Telecommunication Union
JFI	Jain's Fairness Index
LBT	Listen Before Talk
MAC	Media Access Control
NAV	Network Allocation Vector
NOMA	Nonorthogonal Multiple Access
OFDMA-VTS	Orthogonal Freq. Divi. Multiple Access with Variable Tone Spaces
OBSS	Overlapped Basic Service Set
PCS	Physical Carrier Sensing
PER	Packet Error Rate
PHY	Physical Layer
PLCP	Physical Layer Convergence Procedure
PMD	Physical Medium Dependent
RSSI	Received Signal Strength Indicator
RSS	Received Signal Strength
RTS	Request to Send
Rx	Receiver
RWMM	Random Waypoint Mobility Model
SE-FDM	Spectrally Efficient Frequency Division Multiplexing
SSF	Strongest Signal First

SINR	Signal-to-Interference Plus Noise Ratio
SISO	Single Input Single Output
STA	Station
TPC	Transmit Power Control
Tx	Transmitter
UL	Uplink
VCS	Virtual Carrier Sensing
VoIP	Voice Over IP
WiFi	Wireless Fidelity
WLAN	Wireless Local Area Network

Mathematical Notations

$\mathcal{N}$	Set of STAs
$\mathcal{A}$	Set of APs
N	$N = \lvert\mathcal{N}\rvert$, the number of STAs
$\mathcal{N}_n$	Set of simultaneously active STAs
$\mathcal{N}_n^i$	Set of STAs interfering with STA$_i$
M	$M = \lvert\mathcal{A}\rvert$, the number of APs
$\mathcal{E}$	Set of all edges connecting STAs to APs, $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{A}$
$\mathcal{M}$	$\mathcal{M} \subseteq \mathcal{E}$, set of all edges for perfect matching
x_{ij}	Binary variable indicating association of STA i to AP j
w_{ij}	The edge weight of STA i connected to AP j
d_{ij}	Distance between STA i and AP j
$P_{d_{ij}}$	Received power of STA i at distance d_{ij} from AP j
P_{tx}	Transmit power of all STAs
d_i^I	Interference range for desired signal of STA i
N_o	Background noise
γ_o	SINR threshold
I_n	Set of interfering sources to STA i

$\mathcal{I}^{ij}$	Magnitude of interference on desired signal of STA i
Γ	Network-wide CCA threshold
γ_{ij}	SINR of STA i through AP j
Υ_{ij}	Throughput of STA i through AP j
I_j	Maximum allowed interference at AP j
ω	Number of orthogonal channels
$\mathcal{J}$	Jain's fairness index

Chapter 1

Introduction

The growing need for wireless connectivity has prompted wide deployment of wireless local area networks (WLANs) for outdoor, indoor, and/or enterprise access. This increasing demand for low-cost wireless connectivity and the widespread of mobile devices with high speed WLAN connectivity necessitate the dense deployment of access points (APs), whereby 10s to 1000s of APs are closely deployed over a given area. Such dense deployment is also envisioned to play an important role in cellular-WiFi data offloading in the near future, especially in coverage areas with high concentrations of mobile users; for example, in stadiums during a sport event or in highly populated cities and multi-tenants apartment buildings. These trends raise the concern of balancing high traffic of densely distributed users among multiple APs and coordinating intra-cell/inter-cell interference such that throughput is enhanced while providing all users with network access.

Consequently, numerous studies e.g. [3], [4], [5], [6] [7] focus on WLAN capacity improvement and dense deployment of multiple WLAN APs. While high density of WLANs guarantees coverage, interference mitigation and association control techniques are paramount to achieve better throughput. Since stations (STAs) or users are not evenly distributed and STAs use the default strong signal first (SSF) association scheme whereby STAs associate with APs that offer strongest received signal strength (RSS), some APs experience congestion while others are lightly congested [6]; causing overall network throughput degradation. Therefore, association control methods are needed in dense networks to balance high traffic of STAs among APs. Thus far, some of the proposed techniques for capacity improvement include cell breathing [5], [6], MAC parameter tuning [8], AP selection [9], association control [3], [10], and clear channel assessment (CCA) adaptation [8], [11]. While these referenced works have achieved significant results in regular WLAN, new throughput-enhancing procedures are needed in dense WLAN (DWLAN).

The cell breathing technique in [5], [6] performs association control through reduction of an AP's beacon power, known as cell size dimensioning. Reducing beacon power reduces AP cell coverage and forces some stations (STAs) to either associate with nearby APs or go out of coverage. However, variation of beacon power may create coverage holes; a problem inherent in cell breathing. In this thesis, we address the AP association coordination problem by proposing algorithms for optimum assignment, to coordinate association of STAs to APs. Contrary to SSF, our proposed algorithms search for a set of STA-AP associations that enhances throughput over multiple APs in DWLANs. The goals of the association coordination algorithms can be summarized thusly: 1. matching of STAs to APs to prevent the problem of least-congested and overcrowded APs, thereby minimizing congestions at APs and maximizing throughput; 2. to contain severe contention inherent in dense deployment of IEEE 802.11 APs.

The channel access mechanism used in 802.11 networks is a contention-based method known as carrier sensing multiple access with collision avoidance (CSMA/CA), which basically facilitates access control and interference mitigation among contending stations [11], [12]. This carrier sensing technique has been exploited for capacity enhancement in recent works (e.g. [8], [11]), and it is governed by the CCA threshold. The CSMA/CA mechanism requires that a node senses the channel through CCA or physical carrier sensing (PCS) mechanism provided by the physical layer (PHY) before transmitting packets [12], [13], by sensing the current energy level in the channel. If the sensed energy is above a threshold known as the *CCA threshold*, transmission is deferred.

So, apart from the throughput degradation problem due to the inefficiency of SSF association schemes, the CSMA/CA scheme may cause the problem of persistent transmission deferment due to poor or inefficient CCA threshold setting and high traffic from densely distributed nodes in DWLAN. With poor CCA setting, nodes in one cell could defer transmissions to active transmissions in the adjacent cell due to the short spatial distance between co-channel APs in DWLANs, and consequently degrade overall network throughput. To address this CCA problem in addition to the association coordination, a CCA threshold calibration algorithm is proposed for CCA threshold tunning, such that CCA threshold used in one cell does not cause transmission deferment in neighboring cells.

The MAC/CSMA adaptation algorithms in [8], [11] advocate that MAC protocols and PHY parameters such as the CCA threshold ought to jointly adapt to network changes (interference, topology, and traffic) to improve WLAN capacity. Therefore, in addition to the association coordination, the impact of CCA threshold calibration on throughput enhancement is also explored. Following association coordination, we can obtain a set of STA-AP associations that improves throughput. Each AP and its associated STAs form a service area known as the basic service set (BSS) or *infrastructure* BSS (discussed in Chapter 2). CCA threshold adjustment for each AP's service area (BSS) is performed based on the cell-edge signal-to-interference plus noise ratio (SINR) using a proposed CCA threshold adjustment algorithm. That is, subsequent to performing optimum STA-AP association, an appropriate CCA threshold value is computed for each BSS.

1.1 Problem and Motivation

Interference will dominate the deployment of next generation WiFi as multiple APs are deployed densely to cover densely distributed users. In DWLAN, coverage is less of a design concern because of the dense or large deployment of APs (10s to 1000s) to cover all network area. However, effective interference mitigation becomes paramount to improve total network throughput. IEEE 802.11ax (discussed later) is an ongoing DWLAN standardization effort to address the throughput degradation problem in DWLAN. Underutilization of large number of APs in DWLAN is inevitable if STAs are permitted to use the default SSF association scheme in current IEEE 802.11 standard. So, new association schemes are required to improve global network throughput via efficient use of all the APs.

Although, different schemes (as discussed in Chapter 3) have been proposed in earlier studies to enhance WLAN throughput, most of these studies focus on regular WLAN, transmit power control (TPC), proportional fairness and AP selection. Techniques such as improved AP association technique and interference control [14] are required for throughput enhancement in DWLAN. Taking this as a motivation, new association technique is sought for in this thesis by proposing AP association coordination algorithms. Somehow, interference is also taken into account because the proposed STA-AP association algorithms require link SINRs to optimize STAs' associations with APs.

1.2 Contributions and Objectives

In Chapter 4, STA-AP association optimization algorithms are developed to efficiently coordinate STAs' associations with the APs in DWLAN environments. Three AP association algorithms are introduced for STA-AP association coordination in *quasi-static* and *dynamic* WLAN scenarios; these algorithms are motivated by the Kuhn-Munkres assignment algorithm. Our DWLAN is modeled as a weighted bipartite graph where the edge weights are the SINRs of nodes. To calibrate the CCA threshold for the PCS process in each BSS after applying the association algorithms, a CCA adaptation algorithm is also proposed in Chapter 4. The CCA threshold algorithm efficiently calibrate contention domain or sensing range for nodes in each BSS, in order to prevent perpetual intra-BSS/inter-BSS interference and collision. Therefore, the proposed algorithms in this thesis, are developed to meet the following objectives:

1. Increase efficient use of closely deployed APs (usually 100s - 1000s for dense coverage) such that no overloaded and lightly loaded APs.

2. Improve performance of WLAN in terms of uplink and downlink throughput by coordinating STAs' association with APs using the link SINRs.

3. Increase both downlink and uplink throughput via interference coordination through joint STA-AP association coordination and CCA threshold adjustment.

4. Increase the number of concurrent transmissions through efficient CCA threshold calibration, to further increase global network capacity.

The contribution is an improved association scheme using proposed algorithms that improve WLAN throughput by finding optimum association (or assignment) of STAs to APs in the network and adapts the CCA threshold to the resultant STAs-APs association. Throughout this thesis, the term *strongest signal first* (SSF) refers to the current association scheme in IEEE 802.11 standards, where a STA independently associates with an AP that offers strongest RSS. On the other hand, *optimal association* is the association obtained from our association optimization algorithms. This association optimization problem is formulated as a maximum weighted bipartite graph matching problem where the edge weights are link SINRs.

To the best of our knowledge, no prior work exists on the use of weighted bipartite matching for STA-AP association coordination to maximize throughput. Using such coordinated association, we show that throughput increases and an interference-minimizing CCA threshold can be determined for each AP channel or BSS. In our system model in Chapter 3, we will first assume the static situation that all STAs are saturated, i.e., always have packets to transmit and each STA can receive beacon(s) from multiple APs within its reception range. The former and the latter assumptions are consistent with [15], [16] and [9], [17] respectively, and reasonable for dense deployment scenarios. The second scenario of interest is a dynamic network scenario where users mobility within the network is allowed and users are joining and leaving the network.

In a nutshell, our study addresses the throughput enhancement in DWLAN using new association and CCA threshold adjustment schemes. First, the proposed AP association algorithms search for a set of STA-AP associations that improves throughput. Once this set of throughput enhancing STA-AP associations is found, the CCA threshold adjustment algorithm calibrates CCA threshold value for each BSS using the cell-edge SINR. Hence, the inefficiency of SSF association scheme and the fixed (nonadaptive) global CCA threshold setting in current IEEE 802.11 system motivate this study.

1.3 Organization of Thesis

In the next chapter, the WLAN protocols and mechanisms are discussed alongside related works on WLAN capacity improvement and dense deployments. In Chapter 3, we provide the system models and preliminary information used in our problem formulation. Chapter 4 contains new algorithms for association coordination and CCA threshold calibration in dense WiFi networks. In Chapter 5, we present the test scenarios used in investigating the performance of the proposed algorithms. The numerical results obtained from these scenarios are also presented in Chapter 5 with inferences and discussions. The thesis is concluded in Chapter 6 with summary of the study and discussion on future work.

Chapter 2

Background and Related Work

A thorough background on the problems addressed in this thesis is provided in this chapter. It contains related existing studies on the same subject of improving performance in WLAN. In Section 2.1, an overview of the wireless local area network (WLAN) is presented; encompassing the WiFi protocols and IEEE 802.11 standards with focus on the physical and the data link layer. Section 2.2 is the literature survey section, which covers existing related works that address the STA-AP association optimization, throughput maximization, multiple APs coordination, and the emerging dense WiFi deployments. Section 2.3 provides discussion on dense WLAN/WiFi deployment and the emerging standard promoting dense WLANs.

2.1 Overview of IEEE 802.11 and WLAN Operations

The protocols governing the operation of WLAN are defined in IEEE 802.11 standard, which remains the dominant standard. IEEE 802.11 standard constitutes a number of different specifications, that is, it is a suite of protocols or standards. The IEEE 802.11 standard defines the MAC layer and the PHY layer protocols. The MAC layer specifies the medium access procedures, which determine how and when a node can access the medium while the PHY layer coordinates transmission and reception of signals. These two layers are integral parts of the WLAN systems and they are thoroughly discussed in subsequent sections in this chapter. IEEE 802.11 standard is a family of standards for communication in the 2.4GHz Industrial Scientific Medical (ISM) frequency band, and the debut standard was released in 1997.

Since inception in 1997, the standard has been enhanced for improved performance through additional specifications. For example, the 1997 standard specifies 1 Mbps and 2 Mbps as the achievable bit rates while subsequent standard such as IEEE 802.11a and IEEE 802.11b specify higher bit rates. An overview of all IEEE 802.11 standards is available in [18]. IEEE 802.11ac is one of the important standards recently developed and published as a *de facto* standard in 2014. It promises a wider bandwidth up to 160MHz, more MIMO spatial streams (up to 8), high density modulation, multi-user MIMO, and throughput between 500 Mbps and 1Gbps by extending the air interface in 802.11n [18]. The most recent ongoing standard that motivates this research is the IEEE 802.11ax standard for dense WLAN and it is briefly discussed later in Section 2.3.1. The two layers of the IEEE 802.11 protocol stack relevant to the focus of this thesis, are discussed in Sections 2.1.2 and 2.1.4.

2.1.1 WLAN Architecture and Operating Modes

A general architecture of an IEEE 802.11 network is made up of several components, which are the STA, the AP, wireless medium, the BSS, the distribution system (DS), and the extended service set (ESS). STAs are basically components such mobile phones, laptops, etc that connect to the wireless medium, while an AP provides distribution services to its associated STAs. A BSS is the building block of IEEE 802.11

WLAN architecture, which consists of a set of STAs communicating with one another. When an AP is present in a BSS, it is referred to as an *infrastructure* BSS [13], as shown later in Figure 2.3, where an AP and its associated STAs form a BSS. ESS is a set of infrastructure BSS for the purpose of enabling mobility in WLANs. APs in an ESS communicate with one another to facilitate STAs roaming from one AP to another and forward traffic from one BSS to another within the same ESS.

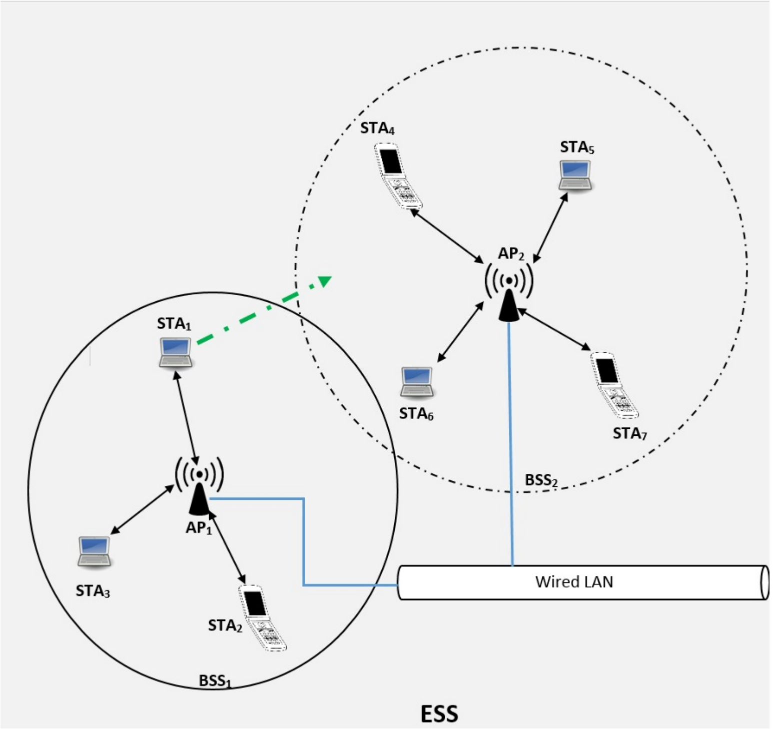


Figure 2.1: WLAN extended service set with two APs.

When an STA roams from one BSS to another, APs use the DS to forward data from one AP to another for the roaming STA. For example, in Figure 2.1, if STA_1 is moving from AP_1 to AP_2 within the same ESS, AP_1 uses DS to forward any data meant for STA_1 to AP_2, such that AP_2 forwards the data to STA_1. Wireless LANs can be implemented to support two main operating modes specified in the IEEE

802.11 standards; these modes are Ad Hoc (or independent) and infrastructure modes. In ad hoc or independent mode shown in Figure 2.2, the nodes can communicate directly with each other without the need for a central node (AP). For one node to communicate with the other, the transmitting STA establishes an ad hoc link with the receiver node before commencing transmission; nodes have direct link with one another for the duration of a communication session. In such independent node, there are more than one point of failures. Ad hoc mode has found applications in many areas today including military tactic missions for temporary wireless communication, home networks and other types of sensor networks.

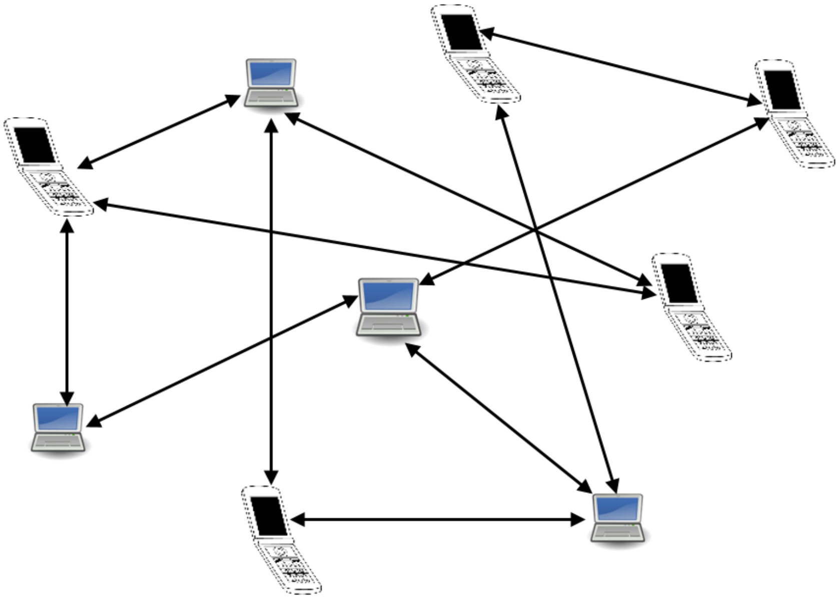


Figure 2.2: WLAN Ad Hoc (independent) mode.

Figure 2.3 shows a typical layout for an infrastructure-based WLAN where communication goes through a central point (AP). Contrary to the absence of an AP in the Ad Hoc mode, the infrastructure mode needs an AP through which STAs communicate with each other or transmit payloads. This implies that an STA needs a connection with the AP to communicate with other STAs or any other node connected to the wired LAN as shown in Figure 2.3. In thesis, our WLAN system model is an infrastructure-based WLAN where each AP and its associated STAs form a BSS, and all the BSSs belong to the same ESS.

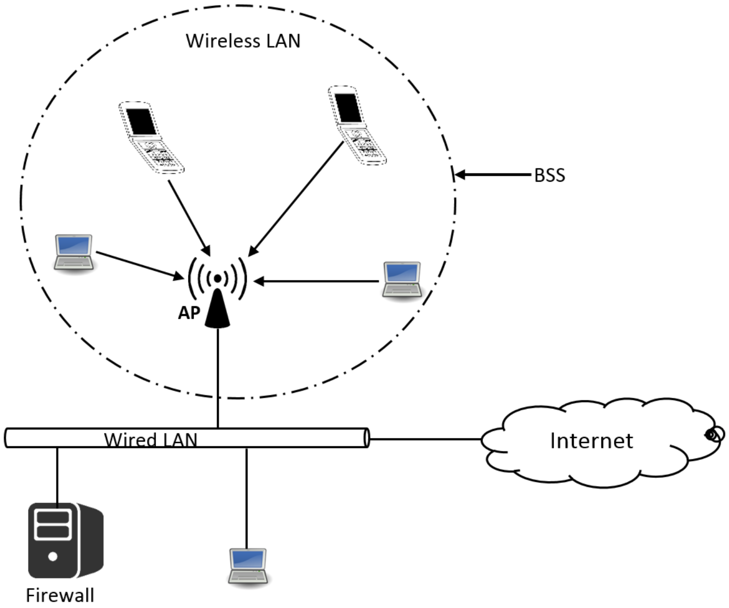


Figure 2.3: WLAN infrastructure mode.

2.1.2 The MAC Layer Protocol

The MAC layer specified in the IEEE 802.11 standard provides three (3) main functionalities to WLAN nodes. It is responsible for reliable delivery of data over the wireless media, fair access control, and data protection [13]. To ensure reliable data delivery over noisy channels, the IEEE 802.11 MAC layer uses a *frame exchange protocol* (FEP) to improve data reliability. This protocol is implemented in the IEEE 802.11 MAC so that the transmitter (Tx) is able to determine when a frame is successfully received by the receiver (Rx). However, there is an overhead cost associated with FEP because all STAs are mandated to decode the MAC header information in every received frame, and send an acknowledgment (ACK) frame to the Tx. Generally, the MAC FEP contains two frames, namely, the actual data frame and the ACK frame. The actual data frame is transmitted from Tx to Rx while the ACK frame is transmitted from the Rx to Tx to acknowledge the successful reception of the data frame. The overhead in MAC FEP becomes severe when ACK or data frame is in error.

In event that the Tx does not receive the ACK frame from the Rx, either due to errors in the data frame or corrupted ACK frame, the Tx will attempt to retransmit the original frame. Erroneous data frames or corrupted ACK frames can occur when the transmissions of other STAs interfere with the transmissions of the data and ACK frames. To prevent persistent frame retransmissions, no other STA is allowed to transmit while the data frame or ACK frame is being transmitted; this access control is done through the CSMA/CA mandating nodes to back off when a transmission is sensed on the channel during the physical carrier sensing (PCS) process. Retransmission of unacknowledged data frame reduces *error rate* of the medium, however, it causes unwanted increase in *bandwidth* consumption, which in turn reduces the overall network *throughput*.

In addition to data reliability, the *hidden terminal problem* is also addressed with the MAC FEP. It is overoptimistic to assume that every STA in a WLAN can communicate with other STAs in the network; this inability of nodes to communicate with all other nodes leads to the *hidden terminal problem* [13]. To explain the *hidden terminal problem* inherent in WLAN, Figure 2.4 illustrates an infrastructure-based WLAN where there is one AP, AP_1 with three (3) associated STAs, STA_1, STA_2, and STA_3. As a result of spatial distance, STA_1 can hear the transmision of STA_2, but not the transmission of STA_3. STA_2 can hear the transmissions of both STA_1 and STA_3 while STA_3 can only hear the transmission of STA_2 and not that of STA_1. Therefore, we have a *hidden terminal problem* because STA_1 will be unaware of an active transmission from STA_3 to AP_1, and vise versa; consequently, collisions may occur between the two transmissions.

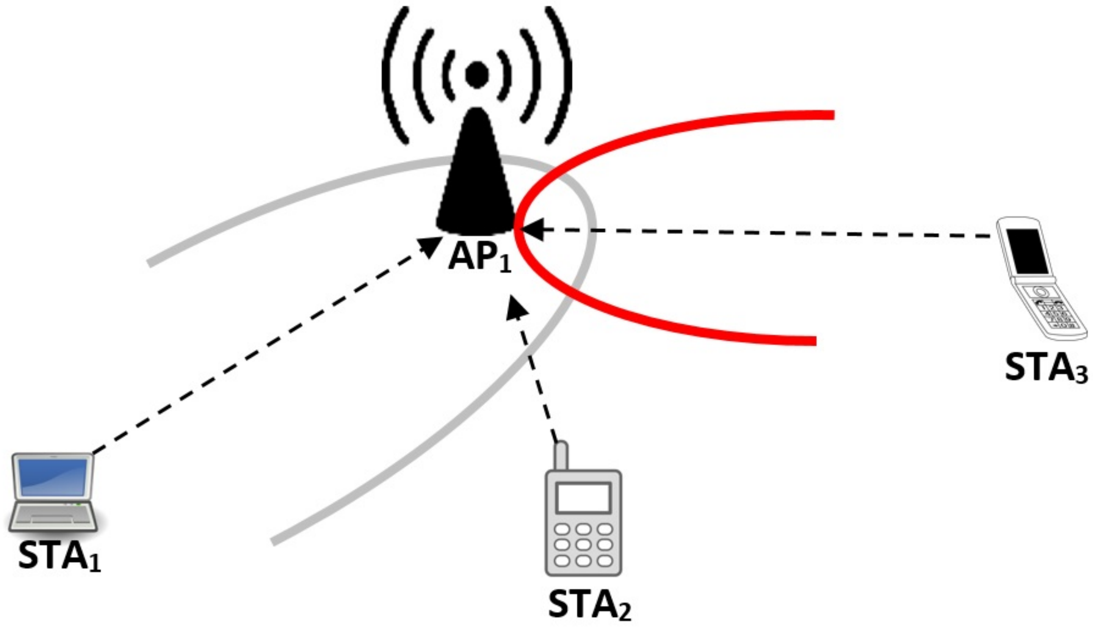


Figure 2.4: Example of the hidden terminal problem.

While the MAC FEP achieves data reliability through retransmission when ACK frame is not received, the *hidden terminal problem* is addressed differently. To solve the *hidden terminal problem*, two additional frames, namely, *clear-to-send* (CTS) frame and *request-to-send* (RTS) frame, are added to the IEEE 802.11 MAC FEP. In order to acquire the channel for transmission, the Tx sends a RTS frame to the Rx and the Rx responds back to the Tx with a CTS frame. The RTS and CTS frames are flags informing other STAs of an upcoming transmission and to defer their transmissions. To illustrate this with our example in Figure 2.4, let us assume that STA$_1$ intends to transmit a payload to AP$_1$. STA$_1$ first sends a RTS frame to AP$_1$, then AP$_1$ who can hear the transmissions of both STA$_1$ and STA$_3$ sends a CTS frame to STA$_1$. The CTS frame contains information, which signals to all STAs that there is an incoming transmission from STA$_1$, causing all other STAs to defer their transmissions. Figure 2.5 illustrates the scenario when the problem is mitigated using CTS/RTS frames.

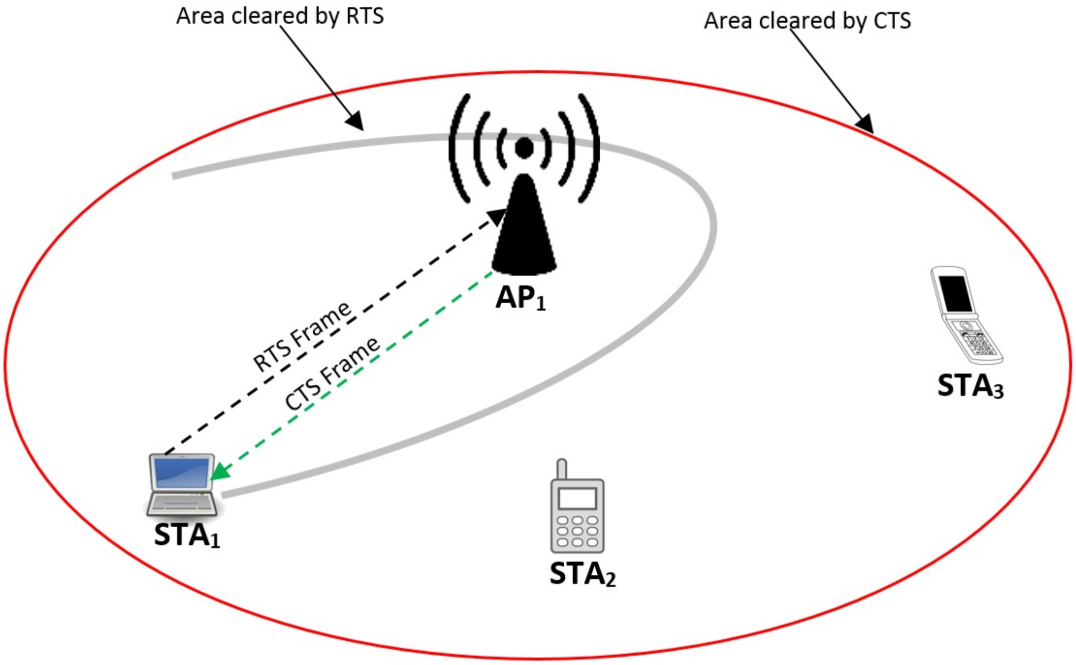


Figure 2.5: Hidden terminal problem addressed with CTS and RTS frames.

In essence, the RTS and CTS frames are used to inform all neighboring STAs of the transmission from the Tx (STA$_1$) to the Rx (AP$_1$), and consequently, all other STAs defer their transmissions. The RTS and CTS frames, as in the case of the

ACK frame, consume a significant amount of bandwidth and degrade overall network throughput. Data frame, RTS/CTS frames, and the ACK frame constitute the four integral frames of the MAC protocol [13]. Failure of any of these frames results in the retransmission of the failed frame. To mitigate the problem of throughput degradation due to the exchange of these frames, the CTS/RTS and ACK frames may be optionally disabled [13].

2.1.3 Access Mechanism in IEEE 802.11

Having discussed the data reliability function of the MAC protocol above, the access control function of the IEEE 802.11 MAC protocol is covered in this section. The MAC protocol used in both infrastructure-based and *ad hoc*-based WLANs, is a contention-based random access known as the CSMA/CA. This protocol does not provide collision detection (CD) and consequently, WLAN nodes employ random backoff methods known as binary exponential backoff to minimize collision probability. CSMA/CA as described in [13] is a "listen before talk" (LBT) access control method. In CSMA/CA, a STA listens to the medium to see if there is an active transmission. If there is a transmission occupying the medium, the STA backs off and defer its transmission.

The process of checking if the medium is busy or not, is the CSMA part of this access scheme, and it is coordinated using the physical carrier sensing (PCS) mechanism of the PHY; the PCS mechanism is discussed later in Section 2.1.5. Without the medium sensing process, a STA would commence transmission and corrupt the already active transmission on the channel as a result of collisions. The aim of CSMA/CA mechanism is to ensure correct transmission and reception of frames over the channel. In event that the medium is sensed to be busy, the listening STA defers its transmission for a period determined by the binary exponential backoff algorithm (BEB) [13]. The BEB algorithm generates a random waiting time uniformly distributed in a range known as the contention window (CW), and the CW varies from PHY to PHY. The listening STA defers its transmission and begins transmission after the backoff time elapses [12], [13], [19].

The collision avoidance (CA) portion of the CSMA/CA is implemented in IEEE MAC protocol using a time value known as the network allocation vector (NAV), which signals to a STA the amount of time left for the channel to become idle or available. The NAV value is carried in all frames during the channel busy period [13], [19]. NAV is a virtual carrier sensing (VCS) technique and when it is enabled, a STA avoids transmitting even though the PCS indicates that the channel is idle [20]. Combining the PCS with the NAV, the MAC takes care of the collision avoidance part of the CSMA/CA [12], [13], [19].

The CSMA/CA scheme uses time intervals to determine if the channel is busy or idle. For this purpose, IEEE 802.11 MAC specifies five (5) timing intervals [12], [19], which are the slot time, the short interframe space (SIFS), the priority interframe space (PIFS), the distributed interframe space (DIFS), and the extended interframe space (EIFS). The slot time and SIFS are the two basic time intervals, and they are determined by the PHY. The discussion of these time intervals are beyond the scope of this thesis, details can be found in [12], [13], and [19].

IEEE 802.11 MAC standard specifies two major configurations for medium access on the MAC protocol, namely, the point coordination function (PCF) and the distributed coordination function (DCF) [12], [13], [19] and [21]. These are two different access methods; the former is a centrally controlled access mechanism while the latter is distributed as its name implies. While DCF guarantees equal access and best effort services, PCF is suitable for real-time application/services. Although DCF is a contention-based scheme while PCF is contention-free, both methods can be combined to implement a hybrid access method. With the hybrid mechanism, a contention-free period is followed by a contention period; that is, DCF is used subsequent to PCF [21].

With PCF, there is a central node known as the access point (AP) coordinating access to the medium. Under the PCF access mechanism, a STA begins to transmit when permission is granted by the AP. When DCF is being used, access to the medium is coordinated using the carrier sense multiple access with collision avoidance (CSMA/CA). In this study, the DCF mechanism (i.e. we assume that all STAs on the WLAN are DCF-enabled) is considered in our WLAN implementation, whereby, STAs use the CSMA/CA method for contention. Therefore, it is necessary to discuss the DCF in more details.

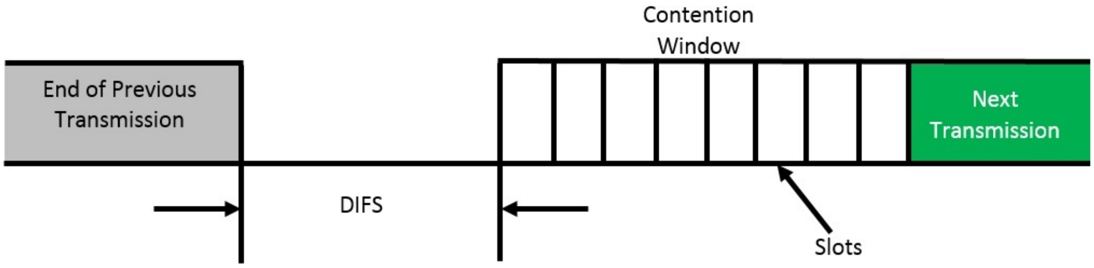


Figure 2.6: DCF timing in MAC access control scheme.

When a DCF-enabled STA wants to transmit, it sends a request to the MAC. On receipt of the request, the MAC checks if the channel is flagged idle or busy by both PCS and VCS. If both access control schemes flag the medium as idle (or free) for a time interval of DIFS, the MAC proceeds with the transmission. As shown in Figure 2.6, the MAC waits for DIFS time at the end of previous transmission and begins transmission of the next frame if the channel remains idle for DIFS time. If the previously transmitted frame is received in error or corrupted, the MAC checks with PCS and VCS to see if the channel is idle for EIFS before commencing the requested transmission. If either the PCS or VCS flags the medium as busy during the DIFS, the MAC chooses a backoff interval using the BEB algorithm while incrementing a retry counter. The *retry counter* depends on the length of the frame, and when the frame is in error or experiences collision on the medium, it indicates the number of times the frame may be retransmitted.

2.1.4 The Physical Layer (PHY)

The PHY is the bottommost layer of the Open System Interconnection (OSI) stack, and it is the interface between the MAC and wireless media. PHY performs three main functions in WLAN nodes. Firstly, it provides a frame exchange process between the MAC and PHY using the physical layer convergence procedure (PLCP) sublayer. Secondly, when the frame reaches the PHY through the PLCP, the PHY uses signal carrier and *spread spectrum* modulation to transmit data frames over the media using the physical medium dependent (PMD) sublayer. Thirdly, PHY communicates the channel status to the MAC using the PCS mechanism, which enables the MAC to determine if the medium is idle or busy [13], [19]. PHY performs the third function by sampling the energy on the channel using PCS (or CCA) process, if the energy is above a threshold known as the *CCA threshold*, it indicates to the MAC that there is an ongoing transmission on the channel. In this thesis, the discussion on PHY is of minimal interest. However, the PCS portion of PHY is significant to the problem addressed here and it is discussed in Section 2.1.5.

2.1.5 Physical Carrier Sensing in IEEE 802.11

PCS (also known as CCA) is an integral part of this study while it is assumed that VCS is disabled on all STAs. As a preamble to formulating the problem considered in this study, let us discuss the use of PCS in the contention process in IEEE 802.11 WLANs. The CSMA/CA is a contention based multiple access mechanism coordinating access to the medium.

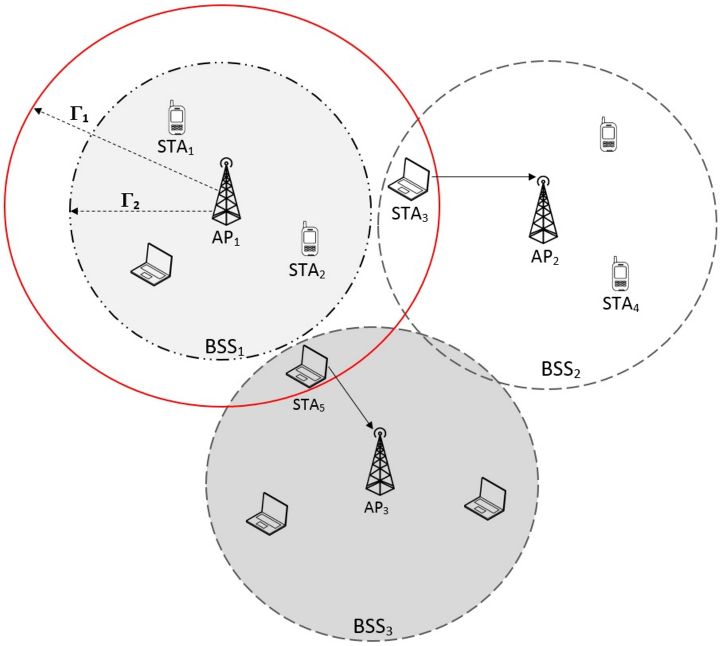


Figure 2.7: WLAN system model showing physical carrier sensing range.

In Figure 2.7, each dashed circle represents the coverage area of an AP (or a BSS). All STAs within the same coverage area of a BSS, compete to access the single channel. The major drawback of such a shared medium is that whenever an STA is transmitting, other STAs defer their transmissions and wait for the STA occupying the channel to finish its transmission.

STAs sense the channel using the PCS mechanism at the PHY because the CSMA/CA requires that each node senses the channel before commencing transmission. During this channel assessment period, the *CCA threshold* is used to determine if the channel is busy or idle. If the signal energy sensed during the CCA is above the CCA threshold, the PHY informs the MAC that the channel is busy. Consequently, the node performing the CCA defers its transmission until the channel becomes idle; the channel is idle when the energy sensed on the channel is below the CCA threshold. The deferment of transmissions due to a busy channel could severely impair performance in dense WLAN environments. To buttress this assertion, let us consider the impact of CCA on AP_1 in our system model of Figure 2.7.

Assuming the CCA threshold in BSS_1 to be Γ_1, the solid circle becomes the carrier sensing range of AP_1 and its associated STAs, which then covers STA_3 and STA_5 associated with AP_2 and AP_3 respectively. Consequently, in compliance with CSMA/CA procedures, AP_1 will always defer its transmission whenever STA_3 in BSS_2 or STA_5 in BSS_3 is transmitting or occupies the channel; assuming the three (3) APs are deployed on the same channel for spatial reuse. On the other hand, setting the CCA threshold to Γ_2 in BSS_1 alleviates this problem, and thus AP_1 assumes the signals coming from STA_3 and STA_5 as interferences rather than deferring its transmission. With Γ_2 as the CCA threshold value, the carrier sensing range (CSR) shrinks as depicted with the dashed circle covering BSS_1, hence, AP_1, STA_3 and STA_5 can transmit simultaneously in their respective BSS, thereby increasing the number of spatially separated concurrent transmissions.

In dense deployments, APs are deployed much more closely than the conventional 100m spatial reuse distance between co-channel APs. As illustrated with Figure 2.7, AP_1 will defer its transmission when STA_3 is transmitting if they are both operating on the same channel and the CCA threshold of AP_1 covers STA_3. Otherwise, AP_1 ignores and treats the signal coming from STA_3 as interference. The performance of WLANs degrades as the distance between two or more APs operating on non-orthogonal channels or in the same channel becomes shorter. Due to the limited number of orthogonal channels for high density (HD) IEEE 802.11 networks, co-channel interference from spatially separated APs on the same channel is inevitable. For efficient spatial reuse and network-wide performance enhancement, nodes in a BSS should not unnecessarily defer transmissions when nodes in neighboring BSSs are transmitting.

Prior to this study, this cell-shrinking technique has been achieved using transmit power control (TPC), mainly to prevent co-channel and/or inter-BSS collisions and increase spatial reuse. Cell breathing [5], [6] is an example of such cell-shrinking approach, which uses reduction of AP's beacon power. Using TPC to adjust CSR and improve spatial reuse, has the tendency of creating coverage holes and impairing proportional fairness among nodes. In this thesis, to avert this dilemma, we propose to maximize throughput using AP association optimization and CCA threshold adjustment. In other words, STA-AP association is first optimized, then the CCA threshold is calibrated such that transmission within one BSS does not restrict or delay transmission in the other BSSs on the same channel.

2.2 Related Works

Since our goal in this study is to improve throughput in DWLAN, it is ideal to describe WLAN capacity enhancement techniques in the literature. *Association control* is a prominent method in literature, and it simply advocates that the default strongest signal first (SSF) STA-AP association in IEEE 802.11 standards is not sufficient for better performance in WLAN; therefore, new association methods are desired. Association control can be used to maximize global network throughput and other objectives in WLAN [3], [22]. In [3], the authors leverage association control to obtain fair bandwidth allocation. They proposed efficient algorithms that jointly consider STA-AP association control and **max-min** fair bandwidth allocation; therein, association control is used for **max-min** with the objective fair bandwidth allocation.

Similarly in [23], STA-AP association is optimized with the objective of minimizing the effects of inter-network interference in WLAN deployments. The authors show how each network can optimize STA-AP associations such that the effects of inter-network interference are minimized. In their approach, an AP is selected from a set of APs that a STA can potentially associate with, to maximize the sum utility of the STA. Also, for throughput enhancement through association control, Karimi et. al [24] seek to obtain proportional fair AP association in densely deployed networks with an assumption that a set of networks share the same upstream provider. In their proposed solution for STA-AP association, multiple APs collaboratively serve the users provided the APs belong to the same upstream provider. Such collaborative system of association control helps mitigate interference with an overall objective of maximizing the weighted bandwidth shares of all STAs associated with an AP [24]; where the weights are costs of association.

The popularity and impact of association control are also obvious in [25] where the authors seek to maximize the overall network performance by optimally distributing STAs among APs using a centralized optimal association policy. Their result is compared with that of a related study in [26]. This association policy in [25] is a centralized approach to maximize network performance by optimally assigning STAs to APs. Although such centralized association policy has global knowledge of the network and is easy to implement, it requires certain radio measurements and transmission of the measurements to a central point, which consequently introduce additional network overhead. Therefore, using a centralized policy, performance enhancement is usually achieved with increased overhead cost.

Other existing studies on association control or STA-AP association optimization with a performance improvement objective include but are not limited to [22], [27], [1], [28], and [29]. Yen et. al [29] study STA-AP association control as AP selection games from a *game theoretic* viewpoint that jointly considers stability and fairness. Basically, their game theoretic framework maximizes the throughput utility of STAs, by allowing a STA to select and associate with an AP depending on the achievable throughput. Contrary to the game theoretic AP selection framework in [29], the authors in [28] formulate the AP selection in IEEE 802.11 mesh networks as a non-linear optimization problem to obtain a tradeoff between aggregate throughput and fairness. On one hand, the objective is to maximize overall throughput and on the other hand, to improve fairness; the optimization framework jointly considers throughput and fairness. In all of these works, the most prominent objective is to improve the capacity of IEEE 802.11 wireless networks; making throughput an essential performance metric.

Similarly in [27], using throughput as the performance metric, AP assignment algorithms are proposed to maximize UL-throughput objectives. Although, their primary objective is maximize overall users throughputs, the proposed *branch-and-bound* algorithm therein considers **max-min** fairness and proportional fairness in terms of throughput achieved by each STA. A special case of achieving proportional fairness through AP association control appears in [1], where the authors study proportional fairness in *multirate* WLANs using AP association. AP association is formulated as a nonlinear programming problem with the objective of maximizing user bandwidth utilities and then, an approximation optimization is obtained for proportional fairness. They show that fairness and AP association should be jointly considered for resource management in multirate WLANs; a related study to [1] can be found in [30].

Thus far, it is apparent that STA-AP association has been extensively exploited for capacity enhancement and proportional fairness in WLAN. For congestion relief and performance improvement in WLAN, a distributed association algorithm is proposed in [22]. This algorithm takes AP loads, RSS of APs at STAs, and interference into account when selecting an AP for an STA. As opposed to the SSF (or RSS-based) association, an algorithm that takes interference and AP load into account has potential to offer better throughput. This is because an STA using SSF association chooses AP that offers strongest signal. Associating with an AP that offers strongest signal does not guaranteed better throughput, especially when there is high degree of interference and load at the selected AP. Therefore, considering interference and AP load beforehand during association decision could enhance throughput.

Another mechanism for improving WLAN performance is MAC protocol/parameter tuning or adaptation as described in some existing studies (e.g. [31], [32], [33]).

In [31], the CSMA is made adaptive to interference by tuning the CCA threshold or/and transmit power depending on the type of interference. First, interference is categorized. Then, depending on the type of interference, a self-adapting CSMA mechanism is designed for either CCA threshold or transmit power. *Type-1* interference occurs when a weak interfering signal unnecessarily defer the transmission of the desired signal while *type-3* interference occurs due to the *hidden terminal* problem described in Section 2.1.2. For a *type-1* interference, the CCA threshold is adjusted and when *type-3* interference is detected, TPC is used [31].

The CCA threshold adaptation or tuning is relevant in this study and it has been described in Section 2.1.5 using a media contention scenario. Therefore, the scenario described in that section is similar to the CCA threshold tuning in [31] with the exception of the interference differentiation. Also, the authors in [32] show that CCA threshold adaptation proves effective in enhancing WLAN capacity. Their approach is to optimize the *spatial reuse* in DWLAN, and consequently gain a 190% increase in aggregate throughput. Taking this CCA threshold adaptation as a motivation in this thesis, CCA threshold adjustment is implemented subsequent to the STA-AP association optimization. In order words, STA-AP association is first optimized, then CCA threshold is adjusted based on the resultant STA-AP associations.

The PCS is enhanced with a tunable CCA threshold in [33], with the goal of avoiding interference in IEEE 802.11 mesh networks without using VCS. As discussed in Section 2.1.3, VCS is used to implement the CA portion of the CSMA/CA and STA keeps deferring its transmission as long as VCS flags the medium as “busy” even if the PCS flags are “idle,” until the VCS indicates that there is no transmission on the medium. To eliminate the need for VCS in mesh a topology, Zhu et. al [33] suggest that tuning the CCA threshold can be used to avoid collision or interference; consequently, the CA portion of the CSMA/CA is implemented without the VCS scheme. They analytically derive the optimal CCA threshold based on the network topology, reception power, and supported data rate. The primary objective of their approach is to permit simultaneous spatially separated transmissions while keeping UL SINR above a threshold.

Joint tuning of PCS, power, and rate has been explored in [34] to maximize spatial reuse while maintaining fairness among users. The proposed mechanism therein implements two tuning rules. First, for PCS tuning, an STA with short distance to the AP uses a high CCA threshold and a low transmit power while an STA with longer distance to the AP set a low CCA threshold and transmits at higher power. The drawback of such PCS tuning rule is that the fairness among STAs degrades; consequently, the second tuning rule addresses the fairness issue via time-sharing fairness. The time-sharing fairness stipulates that the time each STA occupies the channel is proportional to $\frac{1}{R_{\text{max}} N_{\text{cca}}}$, where R_{max} is the maximum supported data rate and N_{cca} is the number of STAs within the PCS range. Similarly, Fuemmeler et al. [20] design protocol to increase spatial reuse by tuning the PCS.

Most of the above works target regular WLAN/WiFi deployments while this thesis focuses on improving throughput in DWLAN deployments. In Section 2.3, we will examine and discuss the features and characteristics of DWLAN and propose throughput enhancement mechanisms in recent studies on DWLAN. In as much as widespread of studies considering throughput or capacity enhancement in IEEE 802.11 networks, our study in this thesis aims at improving throughput in DWLAN while we assume that existing mechanisms on guaranteeing fairness are sufficient or applicable in DWLAN; that is, we focus primarily on improving uplink and downlink throughputs in DWLAN through centralized AP association coordination and CCA threshold adjustment.

2.3 Dense WLAN (DWLAN) Deployment

Let us examine the characteristics and features of DWLAN in this section. The availability of high data rate in WLAN at low cost and the widespread WiFi enabled mobile devices such as tablets, smart phones, laptops, gaming consoles, etc. led to an increase in the number of APs deployed per area in places like multi-tenant residential areas, enterprise or business offices, and indoor/outdoor hotspots in hotels, airports, and cafés [14]. Consequently, as more APs are closely deployed and the number of mobile STAs contending for the medium increases, we have dense WiFi networks. Although, DWLAN guarantees coverage, achievable data rate decreases significantly as a result of increased overall interference. Therefore, novel techniques are needed for capacity enhancement in DWLAN in order to guarantee better per-node throughput [14]. Figure 2.8 illustrates a regular WLAN while with increase in number of APs, Figure 2.9 illustrate a DWLAN.

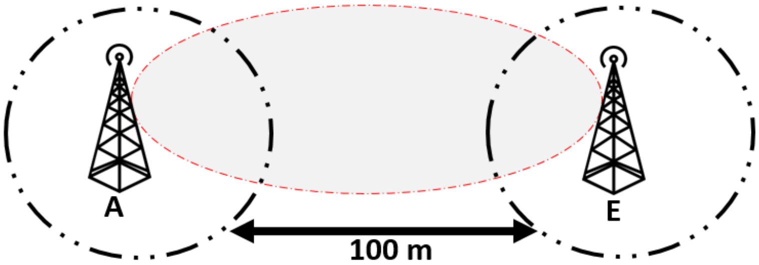


Figure 2.8: Regular WLAN deployment with sufficiently spatial reuse.

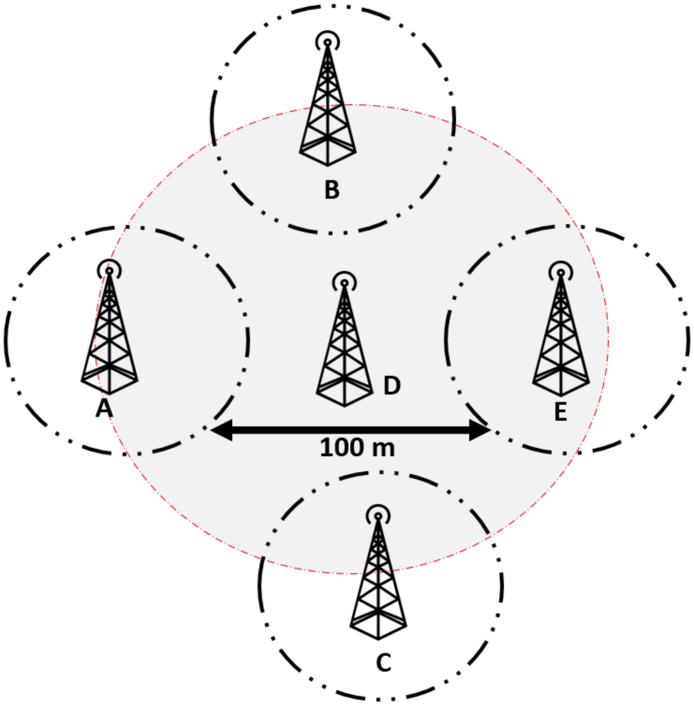


Figure 2.9: DWLAN deployment with large interference area.

2.3.1 IEEE 802.11ax DWLAN Standard

Standardization of IEEE 802.11ax for DWLAN has been initiated by the IEEE High Efficiency WLAN Study Group (HEWSG) with the goal of improving per-node throughput of DWLAN in the presence of sources of interference. Approaches of the IEEE 802.11ax Task Group to enhance node throughput in DWLAN are congestion control, interference and frame conflicts mitigation. Reducing the size of control frames (minimizing overhead) and increasing the data frame size aim to resolve the congestion problem in DWLAN by preventing nodes from occupying the channel for a longer period of time. The hidden terminal problem is responsible for frame conflicts in hidden STAs, and frame conflicts can be reduced by modifying the channel access scheme and overlapped BSS (OBSS) management. To mitigate interference inherent in DWLAN, HEW SG suggests that access by cell-edge STAs with low SINR should be restricted [14].

2.3.2 Experimental Review of DWLAN Deployment

Figure 2.10 shows the observed channel occupancy of IEEE 802.11n Wi-Fi networks in an indoor communications laboratory at Queen's University where there are several cubicals and approximately 37 AP signals are observed using **LinSSID** on a Wi-Fi enabled Linux client. Figure 2.10 depicts the measured RSS levels in 2.4GHz band. The measurements from our laboratory observation in Figure 2.10 are shown in Figure 2.12. We can see that the signal strength received by our testbed STA varies from channel to channel, and similar RSS levels are observed on some AP channels.

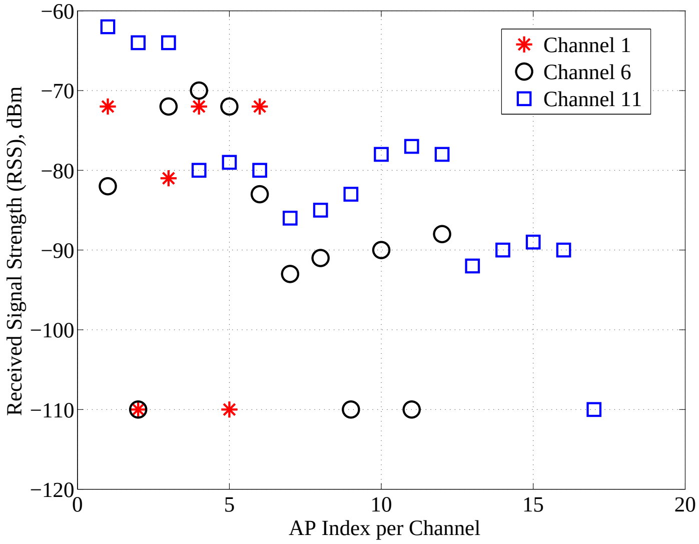


Figure 2.10: RSS (dBm) of beacon signals received by an STA from 37 APs in a campus dense WiFi environment: signals captured using LinSSID.

For example, the STA has similar RSS levels of AP signals from AP_1. Therefore, it is observed that an STA can receive multiple signals from multiple APs in DWLAN. Similarly, a Cisco's illustration of dense WiFi network in stadium is shown in Figure 2.11, where APs are tightly deployed to serve a large number of users during sport events or concerts whereby users need data access.

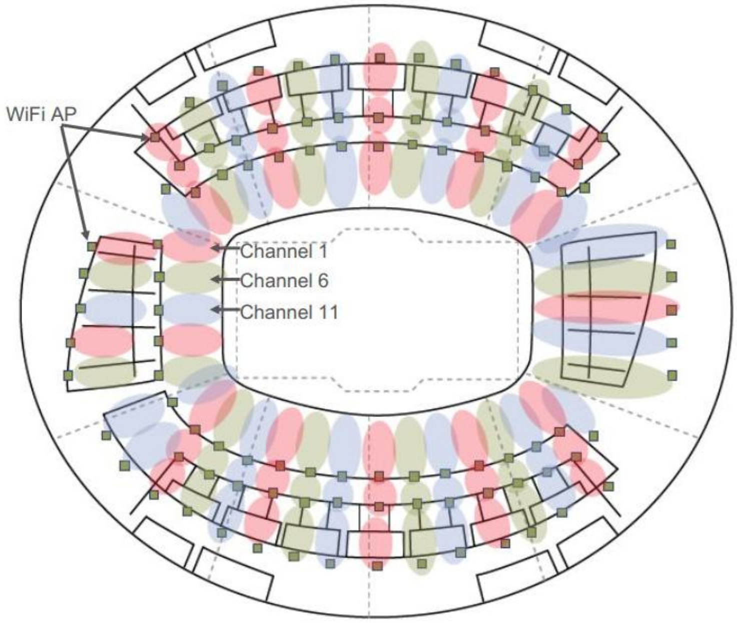
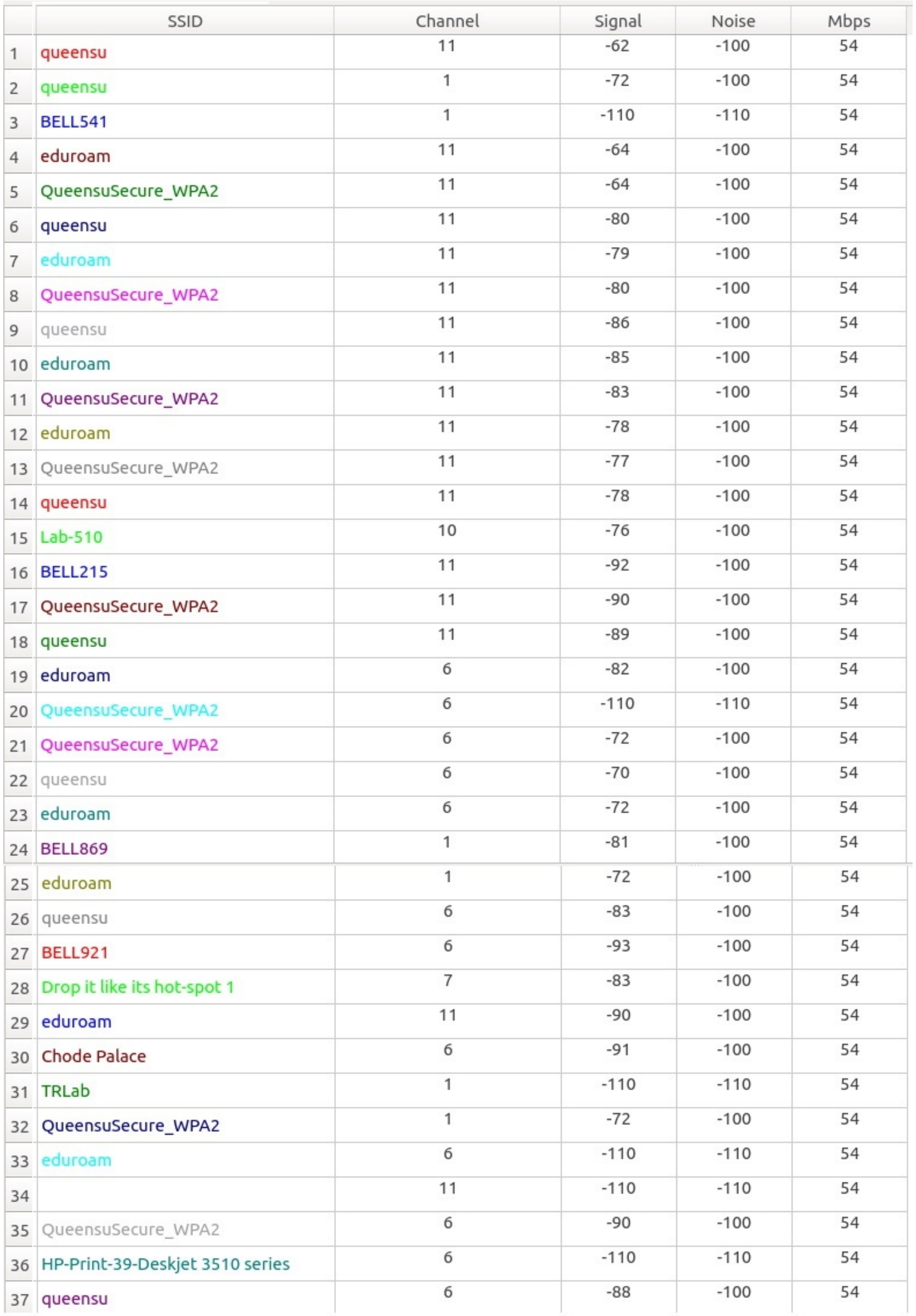


Figure 2.11: Cisco stadium dense WiFi deployment.

From our testbed observation data in Figure 2.12 and plotted in Figure 2.10, it is apparent that each STA in densely deployed APs setting can find multiple APs (5 - 37) within its reception or sensing range; this observation is in consensus with [17], [9]. Our testbed STA received signal RSS of -64dBm from AP_4 (with SSID: eduroam) and AP_5 (with SSID: QueensuSecure_WPA2) on *channel 11* as indexed in Figure 2.12. Similarly, it has RSS levels of -72dBm from AP_{21}, AP_{23} on *channel 6*, and AP_{25} on *channel 1*. In such dense scenario, the default SSF association becomes inefficient



	SSID	Channel	Signal	Noise	Mbps
1	queensu	11	-62	-100	54
2	queensu	1	-72	-100	54
3	BELL541	1	-110	-110	54
4	eduroam	11	-64	-100	54
5	QueensuSecure_WPA2	11	-64	-100	54
6	queensu	11	-80	-100	54
7	eduroam	11	-79	-100	54
8	QueensuSecure_WPA2	11	-80	-100	54
9	queensu	11	-86	-100	54
10	eduroam	11	-85	-100	54
11	QueensuSecure_WPA2	11	-83	-100	54
12	eduroam	11	-78	-100	54
13	QueensuSecure_WPA2	11	-77	-100	54
14	queensu	11	-78	-100	54
15	Lab-510	10	-76	-100	54
16	BELL215	11	-92	-100	54
17	QueensuSecure_WPA2	11	-90	-100	54
18	queensu	11	-89	-100	54
19	eduroam	6	-82	-100	54
20	QueensuSecure_WPA2	6	-110	-110	54
21	QueensuSecure_WPA2	6	-72	-100	54
22	queensu	6	-70	-100	54
23	eduroam	6	-72	-100	54
24	BELL869	1	-81	-100	54
25	eduroam	1	-72	-100	54
26	queensu	6	-83	-100	54
27	BELL921	6	-93	-100	54
28	Drop it like its hot-spot 1	7	-83	-100	54
29	eduroam	11	-90	-100	54
30	Chode Palace	6	-91	-100	54
31	TRLab	1	-110	-110	54
32	QueensuSecure_WPA2	1	-72	-100	54
33	eduroam	6	-110	-110	54
34		11	-110	-110	54
35	QueensuSecure_WPA2	6	-90	-100	54
36	HP-Print-39-Deskjet 3510 series	6	-110	-110	54
37	queensu	6	-88	-100	54

Figure 2.12: Dense Wi-Fi Testbed Analysis.

to improve throughput in the presence of interference. Hence, in this thesis, we will leverage the fact that each STA can see multiple AP within sensing range to coordinate AP associations; that is, seek a set of AP associations that reduces interference and consequently improves throughput in DWLAN.

2.3.3 Throughput Enhancement in DWLAN

Shin et al. [14] discuss three possible techniques to improve throughput in DWLAN. These techniques are physical layer improvements, elevating spectral efficiency, and controlling overall interference levels. In effort to implement the technique of exploiting cellular technology, Qualcomm Inc. and Huawei Technologies Co. ltd. announced a version of the Long Term Evolution-Advanced (LTE-A) technology that will operate on frequency around the 5GHz unlicensed frequency band (LTE-U). The primary objective of LTE-U is to offload some data traffic from the congested cellular networks to the less congested unlicensed frequency band. Intuitively, data traffic of DWLAN nodes can be offloaded to cellular networks in order to minimize congestion on DWLAN [14].

Another method to improve DWLAN performance is the elevation of *spectral efficiency* [14]. Improved spectral efficiency can be achieved in WLAN by adopting some techniques from the cellular technologies. For example, nonorthogonal multiple access (NOMA), spectrally efficient frequency division multiplexing(SE-FDM), and orthogonal frequency division multiple access with variable tone spaces (OFDMA-VTS) in 5G cellular networks can be used in DWLAN to improve spectral efficiency; these spectral efficiency enhancement techniques are thoroughly discussed in [14].

To increase throughput in DWLAN in the presence of interference, HEWSG suggested dynamic sensitivity control (DSC) and TPC as solutions [14]. The function of DSC is to change the CCA node sensitivity depending on the magnitude of the measured interference level. When the measured interference level by an STA is close to the beacon signal level from the associated AP, the STA defers its transmission in order to reduce interference, and proceeds with its transmission when lower interference is measured. Using TPC, the transmit power is controlled at the transmitter in order to control the power level at the receiver. This helps minimize interference as opposed to using fixed power at all transmitting nodes. TPC leverages the distance between the transmitter and the receiver to control transmit power. If the distance between an STA and its associated AP is short, a lower transmit power is used in order to take advantage of small path loss while reducing intra-BSS interference. Using lower transmit power at a shorter distance to the AP also enhance spatial reuse by allowing APs and STAs in adjacent cells to transmit concurrently on the same channel.

In [14], the authors compare per-node throughput of DWLAN with and without interference control. To improve performance in DWLAN, our approach is to coordinate multiple APs is to coordinate their STA-AP associations via a central controller; in essence, we are revisiting joint association control and CCA threshold adjustment as possible method to improve throughput in DWLAN. We can liken our approach to the method of controlling overall interference levels suggested in [14] because the proposed method in this thesis takes the SINR of nodes into account.

2.4 Chapter Summary

In summary, this chapter provides appropriate background on the IEEE 802.11 standard for WLAN, the common architectures (ad hoc and infrastructure-based) for WLAN deployments, the MAC and PHY protocols including the channel access mechanisms (PCS and NAV) and configurations (DCF and PCF), and discusses an example scenario on PCS operation and CCA threshold adjustment. Existing related studies on performance enhancement in WLAN have been discussed whereby the proposed methods focus on increasing spatial reuse (or concurrent transmissions) and proportional fairness through AP association coordination or/and CCA threshold adaptation for better performance in terms of throughputs. A testbed to describe the multi-AP signals scenario in DWLAN environment has been used to provide an example of a DWLAN. Lastly, to enhance throughputs in DWLAN, the proposed schemes (DSC and TPC) by HEWSG are discussed.

Chapter 3

Problem Formulation and System Models

Existing studies on multiple APs coordination through different methods such as association control, CCA adaptation, AP selection, cell breathing, and MAC parameter tuning have been reviewed in Chapter 2. This chapter describes the WLAN system models that are used throughout this thesis. It also presents the model assumptions used in formulating the association optimization problem as a weighted bipartite graph matching. The proposed algorithms in Chapter 4 are based on the system models and background information in this chapter.

3.1 Preliminaries

A bipartite graph matching is used to find the optimum association (or matching) of STAs to APs. Figure 1 shows an example of a bipartite graph, where the left vertices are related to the right vertices with appropriate edge weights. Figure 1 represents a WLAN modeled as a bipartite graph $G = (\mathcal{N}, \mathcal{A}, \mathcal{E})$ in which $\mathcal{N}$ is a non-empty set of STAs on the left, $\mathcal{A}$ is the set of APs, and $\mathcal{E}$ is the set of edge weights such that $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{A}$. With known edge weights, total number of STAs $N = |\mathcal{N}|$ and $M = |\mathcal{A}|$ APs, each STA $i \in \mathcal{N}$ can be matched to exactly one AP.

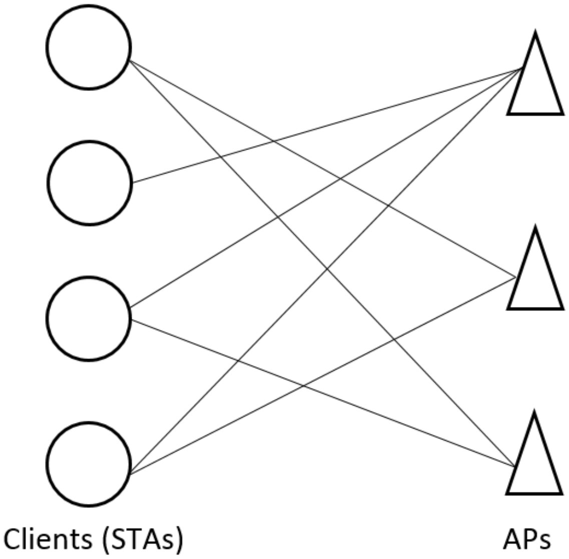


Figure 3.1: Bipartite graph representation of wireless LAN.

The problem of finding a throughput maximal match can be cast in terms of a classical combinatorial optimization problem [35], which has been previously solved using the Kuhn-Munkres algorithm [36]. Relevant notations for the system model are summarized in Table 3.1. A match $\mathcal{M}$ is a subset of edges $\mathcal{M} \subseteq \mathcal{E}$ connecting each vertex in $\mathcal{N} \cup \mathcal{A}$ such that each vertex in $\mathcal{A}$ is an endpoint of at least one edge in $\mathcal{M}$. In essence, a possible association occurs if each STA vertex is connected to exactly one AP vertex in $\mathcal{A}$. We assume that each AP vertex is connected to at least one STA vertex in $\mathcal{N}$. The WLAN is modeled as a bipartite graph with clients (STAs) on the left associated with APs on the right as depicted in Figure 1.

The edge from STA i to AP j is weighted by w_{ij} as illustrated in Figure 2, representing the SINR of STA i through AP j. The weight varies as the number of APs within the STA's reception range changes and depends on the path loss model and users mobility. Bipartite graph matching has been applied in different domains

Table 3.1: List of Important Symbols

Symbols	
$\mathcal{N}$	Set of STAs
$\mathcal{A}$	Set of APs
$\mathcal{N}_a$	Set of simultaneously active STAs
N	$N = \lvert\mathcal{N}\rvert$, the number of STAs
M	$M = \lvert\mathcal{A}\rvert$, the number of APs
$\mathcal{E}$	Set of all edges connecting STAs to APs, $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{A}$
$\mathcal{M}$	$\mathcal{M} \subseteq \mathcal{E}$, set of all edges for perfect matching
x_{ij}	Binary variable indicating association of STA i to AP j
w_{ij}	The edge weight of STA i connected to AP j
d_{ij}	Distance between STA i and AP j
$P_{d_{ij}}$	Received power of STA i at distance d_{ij} from AP j
P_{tx}	Transmit power of all STAs
d_i^I	Interference range for desired signal of STA$_i$
N_o	Background noise
γ_o	SINR threshold
$\mathcal{I}^{ij}$	Magnitude of interference on desired signal of STA$_i$
Γ	Network-wide CCA threshold
γ_{ij}	SINR of STA$_i$ through AP$_j$
Υ_{ij}	Throughput of STA$_i$ through AP$_j$
I_j	Maximum allowed interference at AP$_j$

[36], [37], [35]. Here, the goal is to obtain an optimal match (or association) that maximizes network-wide throughput; the weighted bipartite graph matching is solved using Kuhn-Munkres algorithm. Therefore, the association algorithms proposed in this thesis use Kuhn-Munkres algorithm at some point to find optimal assignment of STAs to APs. The matching of the STAs to the APs is performed such that there are no isolated STA-vertices (that is, all STAs have network access and no coverage holes). With the WLAN shown in Figure 2, $G = (\mathcal{N} \cup \mathcal{A}, \mathcal{E})$ is the bipartite graph for the network with set of edges $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{A}$, w_{ij} representing SINRs.

The semi-matching approach, which is a relaxation of the maximum bipartite matching problem [35] is considered. A semi-matching is a set of edges $\mathcal{M} \subseteq \mathcal{E}$ such that each vertex in $\mathcal{A}$ is an endpoint of at least one edge in $\mathcal{M}$, that is, each AP is an endpoint of at least one STA. Starting with a set of STA-AP **SSF** associations on a given WLAN, the proposed association optimization algorithms find an improved set of associations that maximizes network throughput. The **SSF** association supported in today's 802.11 networks allows a STA to independently associate with an AP that offers the strongest RSS. Association optimization has potential impact in DWLANs where STAs may receive multiple strong signals from more than one AP, resulting in a set of possible AP associations for each STA. Therefore, the quest is to coordinate STAs' associations with AP such that global throughput increases.

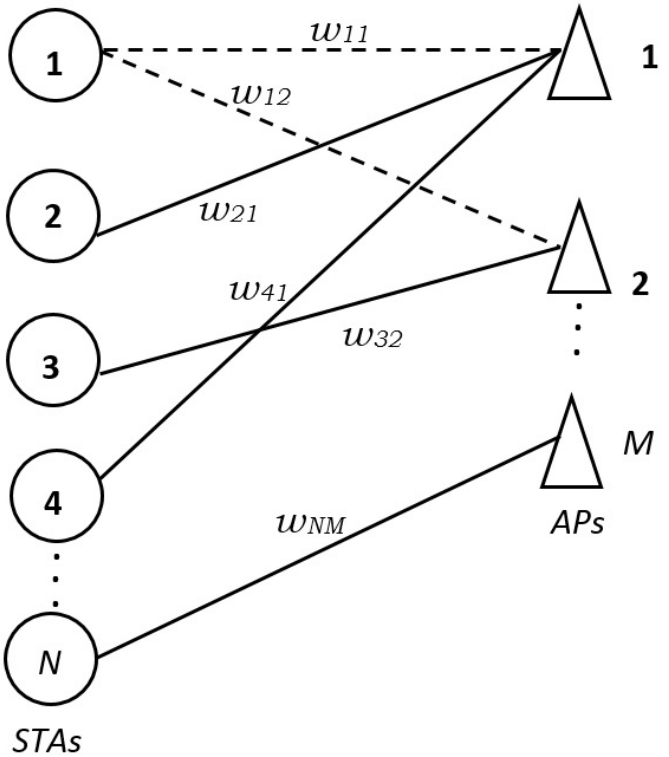


Figure 3.2: Wireless LANs modeled as a weighted bipartite graph.

3.2 System and Network Models

3.2.1 Uplink System Model

In the network model in Figure 3.2, a steady-state scenario is considered whereby all STAs are already present in the network and STAs are not leaving or/and joining the network dynamically and the channel gains are normalized to unity. The former assumption corresponds to a *quasi-static* mobility pattern [3] of users, in which STAs stay within the physical location of the network for a relatively long time. Prior to AP-STA association optimization, it is assumed that each STA has independently associated with an AP using the SSF association. In this SSF association, the throughput at some APs may be degraded due to the degree of contention among STAs. The maximum traffic an AP can support decreases as more active STAs are associated with an AP and contention for the shared-medium becomes severe [4].

For example in Figure 3.2, the degree of contention at AP_2 is lower than that at both AP_1 and AP_M. As a result, the throughput at AP_2 could be higher by virtue of the shared medium by the fewer contending STAs. To improve network throughput, the proposed association schemes match the STAs to APs by considering the edge weights w_{ij} (or SINRs) and the CCA sensing threshold, Γ. To perform the association optimization, a set of throughput maximizing edge weights connecting STAs to APs needs to be estimated, and the SINR of each STA should at least be equal to the CCA sensing threshold. The received signal strength indicator (RSSI) in the IEEE 802.11 standard is used to measure the RSS of an STA [12], and is used to estimate the edge weight w_{ij}. For example, the minimum RSS in a 20 MHz channel is -65 dB at 54 Mbit/s using 64-QAM [13].

For each AP that an STA detects within the carrier sensing range, there is a

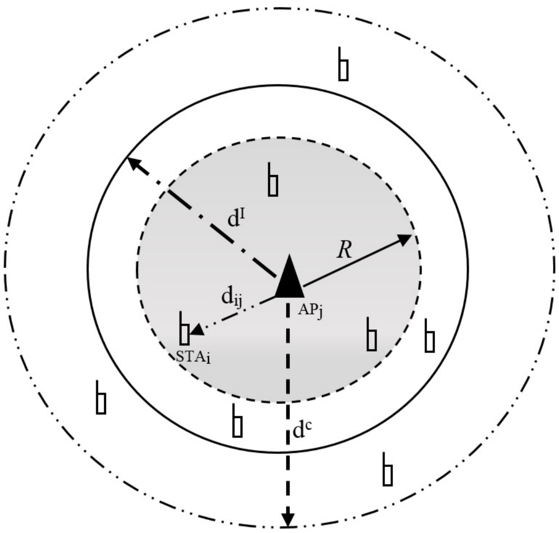


Figure 3.3: AP deployment showing CCA sensing and interference ranges.

corresponding weight representing the SINR. For instance in Figure 3.2, STA$_1$ detects two (2) APs {1,M}, resulting in weight vector $\mathbf{w}_1 = [w_{11}, w_{1M}]$ for STA$_1$ indicating that there are two possible associations for STA$_1$ and it will be matched to the AP that maximizes its throughput. Therefore, with the proposed association technique, STA$_1$ will be matched to an AP by choosing w_{ij} from $\{\mathbf{w}_i, j = 1, 2, \ldots, M\}$ such that the SINR of STA$_1$ is not below the CCA threshold. Each STA i is spatially separated from AP j, and using a path loss model, the received power of STA i through AP j subject to attenuation from propagation is given by:

$$P_{d_{ij}} = P_{tx} - PL_{d_o} + \delta \log_{10}\left(\frac{d_{ij}}{d_o}\right) \qquad (dBm) \tag{3.1}$$

where P_{tx} is the transmit power (all STAs are assumed to transmit at the same power) and $P_{d_{ij}}$ is the received power at distance d_{ij} separating STA i and AP j, PL_{d_o} is the path loss at reference distance $d_o = 1$m and the path loss exponent is denoted as δ.

One type of DWLAN considered here assumes appropriate frequency planning and consists of a large number of APs placed apart from one another. This type of model reflects the deployment patterns found in apartment building, airports, multi-office enterprises, and even stadiums. The coverage area of an AP is depicted in Figure 3.3, where the interference range, d_i^I of the i^{th} STA's signal is defined as [7]:

$$d_i^I = d_{ij} \left(\frac{1}{\frac{1}{\gamma_o} - \left(\frac{d_{ij}}{d_o}\right)^{\delta} \frac{N_o}{P_{d_{ij}}}} \right)^{\frac{1}{\delta}}, \qquad 1 \leq i \leq N, 1 \leq j \leq M, \tag{3.2}$$

where N_o is the background noise, γ_o is the SINR threshold defined for different data rates in 802.11 networks (see Table 3.2 for SINR thresholds in 802.11a/g WLAN), and each γ_o depends on the modulation and coding techniques being used. Any node transmitting within d_i^I interferes with STA$_i$'s uplink signal at receiver AP.

Table 3.2: SINR requirements for different data rates in 802.11a/g WLAN [1], [2].

Rate (Mbps), R_k	54	48	36	24	18	12	9	6
SINR, γ_o (dB)	24.6	24	18.8	17	10.8	9	7.8	6
Minimum sensitivity, θ (dBm)	-65	-66	-70	-74	-77	-79	-81	-82

To determine if a transmitter would potentially interfere with the desired signal, an illustration with Figure 3.4 is appropriate. Let us assume that STA$_2$ in Figure 3.4 is a transmitter at distance d_k^* from AP$_1$ receiver and our desired signal is transmitted from STA$_1$ to AP$_1$. Using Equation (3.2), we can determine the interference range, d_1^I for STA$_1$'s signal as illustrated in Figure 3.4. If $d_k^* \leq d_1^I$ (i.e STA$_2$ is within the interference range of STA$_1$) and STA$_2$ begins transmission at the same time with STA$_1$, then signal coming from STA$_2$ will interfere with STA$_1$'s signal at the AP$_1$ receiver; hence, STA$_2$ is a potential interferer. However, since STA$_3$ is outside the interference range of STA$_1$, it is not regarded as a potential interference source.

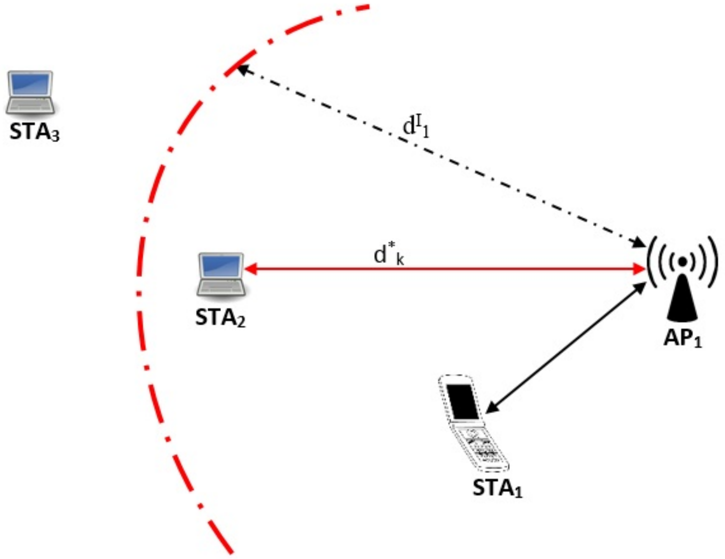


Figure 3.4: Interference illustration.

Due to the CSMA/CA protocol, only a subset of STAs can transmit concurrently in the uplink for a given transmission time. Out of these, $\mathcal{N}_i \subseteq \mathcal{N}$ might be interfering with the desired signal from STA$_i$ at the AP receiver. Let k represent an index of an STA in $\mathcal{N}_i$, i.e., STA$_k$ is at distance d_k from the AP receiver such that $d_k \leq d^I_i$. The total interference on STA$_i$'s signal received at AP$_j$ can be written as:

$$\mathcal{I}^{ij} = \sum_{k \in \mathcal{N}_i, k \neq i} P_{d_{kj}} \qquad 1 \leq i \leq N, 1 \leq j \leq M, \tag{3.3}$$

where $P_{d_{kj}}$ is the signal power of interference received at AP$_j$ from each interference source k in $\mathcal{N}_i$ transmitting simultaneously with STA$_i$ and affecting STA$_i$'s signal at the AP. $P_{d_{kj}}$ is estimated using the path loss model in (3.1).The achievable throughput of a STA in 802.11 networks depends on the SINR of the STA, and Table 3.2 presents

SINR requirements for different bit rates. Constrained by the total interference in (3.3), background noise, N_o, and path loss model in (3.1), the achievable SINR of STA i through AP j is given as:

$$\gamma_{ij} = \frac{P_{d_{ij}}}{N_o + \mathcal{I}^{ij}}, \quad 1 \leq i \leq N, 1 \leq j \leq M. \tag{3.4}$$

Therefore, for successful reception at certain data rates and modulation schemes, the received SINR defined in (3.4) must be at least equal to γ_o defined in Table 3.2. Using CSMA, an STA or AP would be able to sense the channel and detect any active transmission within the carrier sensing distance, $d^c = d_{ij}\left(1 + \gamma_o^{\frac{1}{\delta}}\right)$. The transmission range, R, of an AP is defined in terms of the lowest received power, P_R from a STA at the edge of the cell as $R = d_o\left(\frac{P_{d_o}}{P_R}\right)^{\frac{1}{\delta}}$, which corresponds to the cell radius of each AP. The carrier sensing scheme in WiFi mandates stations to sense the channel before transmitting. If the channel is busy, they back off and defer transmission. The energy level sensed by a STA consists of both the interference and noise levels. Therefore, for the STA to commence transmission, the interference level under negligible noise should satisfy CCA threshold, Γ:

$$\mathcal{I}^{ij} \leq \Gamma, \qquad 1 \leq i \leq N, 1 \leq j \leq M. \tag{3.5}$$

The constraint in (3.5) ensures that the SINR of STA i through AP j is at least equal to the network-wide CCA threshold, Γ. Since WLAN uses CSMA, if the interference exceeds Γ, any node attempting to access the medium will receive the "busy" signal and backoff during the carrier sensing period. Therefore, to ensure that an AP can support an STA on its channel, STA i will only be matched to AP j with

SINR satisfying the constraint in (3.5). With the known set of interfering sources, the throughput of STA i through AP j can be upper bounded based on Shannon capacity equation as:

$$\Upsilon_{ij} = \log_2 \left(1 + \gamma_{ij}\right), \qquad 1 \leq i \leq N, 1 \leq j \leq M, \tag{3.6}$$

where γ_{ij} corresponds to SINR weight w_{ij} of STA$_i$. Maximum weighted bipartite matching algorithm [36], [37], [35] guarantees that the association algorithm selects edges that maximize overall network throughput. While different algorithms may be used for bipartite graph matching, it has been shown that the Hungarian (Kuhn-Munkres) algorithm provides optimal matching for a bipartite graph matching problem [35].

3.2.2 Downlink System Model

In the downlink of a WLAN, APs transmit data to their respective associated STAs. In this section, the system model for the downlink (AP-to-STA) is presented. It is of interest to analyze the achievable throughput in the downlink because it is envisaged that the majority of dense Wi-Fi traffic will occur in the downlink. From our system model in Section 3.2.1, the set of APs is denoted as $\mathcal{A}$. In this section, we will model transmissions from these APs to their respective associated STAs in the downlink. The goal is to estimate the capacity for downlink transmissions.

The number of available orthogonal channels in Wi-Fi networks depends on the IEEE 802.11 standard being supported. For instance, IEEE 802.11b/g standard supports 3 non-overlapping channels from the 14 available channels while IEEE 802.11a provides 8 non-overlapping channels. The most recent *de facto* standard, IEEE 802.11ac has two non-overlapping channels for 80MHz and one 160MHz non-overlapping channel. Therefore, for any supported standard, let each channel from the set of available orthogonal channels be denoted as ω. For this downlink model, we will assume that all APs transmit at the same power P_a (mW) and let d_{ji} is the distance between the transmitting AP and the receiving STA. Consequently, using the same path loss model in Equation 3.1, the RSS at STA$_i$ from APj is estimated as follows:

$$P_{d_{ji}} = P_a - PL_{d_o} + \delta \log_{10}\left(\frac{d_{ji}}{d_o}\right) \qquad (dBm)\,. \tag{3.7}$$

Similar to the contention in the uplink scenario, CSMA/CA is also used in the downlink for MAC operation. Let us denote the set of APs operating on the same channel ω as $\mathcal{A}^{\omega}$. Therefore, due to CSMA/CA only a subset of $\mathcal{A}^{\omega}$ can transmit simultaneously on channel ω. For example, in our experimental tested in Section 2.3.2, there are 15 APs operating on *channel 11*. So, with CSMA/CA protocol, only a subset of these APs operating on *channel 11* can be concurrently active depending on the contention domain of each AP, which is determined by the CCA threshold.

Let $\mathcal{A}_a^{\omega}$ represent the set of active APs in $\mathcal{A}^{\omega}$ on channel ω. By virtue of the CCA threshold, a subset of the active APs in $\mathcal{A}_a^{\omega}$ will be in the contention domain of AP$_j$. Therefore, all active APs in the contention domain of AP$_j$ form a set of co-channel APs with AP$_j$ and this set is denoted as $\mathcal{A}_{aj}^{\omega}$. This implies that AP$_j$ will contend for the medium with other co-channel APs in $\mathcal{A}_{aj}^{\omega}$, and if any of the APs in $\mathcal{A}_{aj}^{\omega}$ is transmitting on the channel, other APs remain idle in the downlink. This scenario is mathematically expressed as:

$$\mathcal{A}_j^{\omega} := \{m \in \mathcal{A}_a^{\omega}, m \neq j | \lambda h_{jm} > \Gamma, \mathcal{A}_{aj}^{\omega} \subseteq \mathcal{A}_a^{\omega} \subseteq \mathcal{A}^{\omega} \subseteq \mathcal{A}\}, j \in \mathcal{A}, \omega \in \mathcal{C} \tag{3.8}$$

where λ is the total energy sensed on channel ω during the PCS, $\mathcal{C}$ is the set of channels in any supported or chosen IEEE 802.11 standard and h_{jm} is the channel gain between AP$_j$ and AP$_m$. Interference coordination in the downlink of DWLAN is essential because deploying more APs may not translate to an increase in capacity, but will inevitably increase each AP's contention domain.

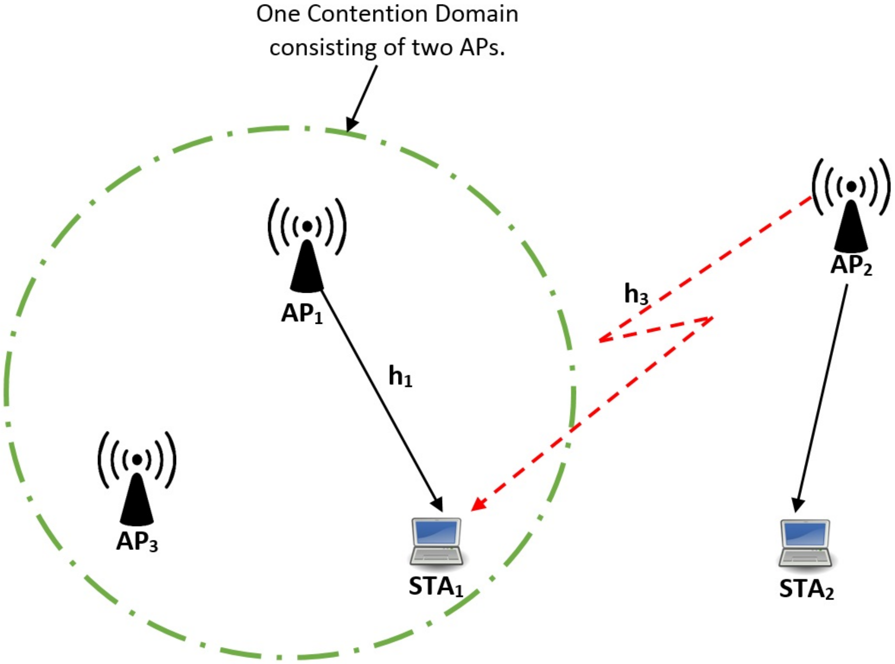


Figure 3.5: Downlink interference at receiver STA.

Figure 3.5 illustrates an interference scenario in the downlink. In this figure, AP$_2$ is outside the contention domain of AP$_1$ therefore, signal coming from AP$_2$ will interfere with downlink transmission of AP$_1$ at the receiver of STA$_1$ depending on the separating distance and the magnitude of channel gain h_3. From this scenario, the total interference at the receiving STA$_1$ in the downlink can be estimated depending on the number of APs transmitting outside the contention domain of AP$_1$ and whose signals can be heard by STA$_1$. Let $\mathcal{A}^I$ represents set of APs interfering with the downlink signal of AP$_j$ at STA$_i$ receiver, the total interference received at STA$_1$ from all interfering APs in $\mathcal{A}^I$ is estimated as:

$$\mathcal{I}^{ji} = \sum_{z \in \mathcal{A}^I, i \in \mathcal{N}, j \in \mathcal{A} | j \neq z, \mathcal{A}^I \subset \mathcal{A}} P_{zi} h_{zi}, \tag{3.9}$$

where P_{zi} is the received signal power at STA$_i$ from the z^{th} interfering AP, which is estimated using the path loss model in Equation (3.1) using distance d_{zi} separating the interfering AP$_z$ and the receiving STA$_i$, and h_{zi} represents the channel gain of the link between STA$_i$ receiver and the interfering AP$_z$. Here, we assumed no fading and the channel gains are normalized to unity. As a result of interfering signal power received at STA$_i$ from APs outside AP$_j$'s contention domain, the SINR γ_{ji} of link between AP$_j$ and STA$_i$ in the downlink becomes:

$$\gamma_{ji} = \frac{P_{d_{ji}}}{\mathcal{I}^{ji} + N_o}, \qquad 1 \leq i \leq N, 1 \leq j \leq M, \tag{3.10}$$

where $P_{d_{ji}}$ is the received signal power subject to path loss from AP$_j$ to STA$_i$ over a spatial distance, d_{ji}. Therefore, using Shannon's capacity equation, the achievable downlink throughput from AP$_j$ to STA$_i$ becomes:

$$\Upsilon_{ji} = \log\left(1 + \gamma_{ji}\right), \qquad 1 \leq i \leq N, 1 \leq j \leq M. \tag{3.11}$$

3.3 Chapter Summary

The system model is described in this chapter, whereby DWLAN is modeled as a weighted bipartite graph with STA nodes and AP nodes on left and right respectively. The weight of each edge connecting a STA to an AP is the SINR of the link between the STA and the target AP. The *path loss model* is computed using Equation (3.1) while the carrier sensing range, transmission range, and interference distance have been explicitly defined. Finally, the throughputs of STA-to-AP (uplink) and AP-to-STA (downlink) links are evaluated using the well known Shannon's formula.

Chapter 4

Throughput Enhancement in Dense WLAN

In this chapter, we present the proposed algorithms for association coordination and CCA threshold calibration to enhance throughput in DWLANs. One method to coordinate multiple APs in dense deployments is the use of an association optimization algorithm. Following STA-AP association coordination, an attempt is made to determine appropriate CCA threshold for each BSS from using the offline and the dynamic (or online) algorithms in Section 4.1.1 and Section 4.1.2 respectively. As discussed earlier, CCA threshold is one of the MAC protocol parameters that can be tuned or calibrated in DWLANs to improve throughput. Therefore, a CCA calibration algorithm is proposed in Section 4.2 to calibrate the CCA threshold values for each BSS subsequent to STA-AP association coordination.

4.1 AP Association Coordination in DWLAN

AP association coordination is considered for both *quasi-static* and dynamic network scenarios. We present an offline algorithm in Section 4.1.1, which assumes a DWLAN with quasi-static network condition whereby users are not leaving and/or joining the network. In Section 4.1.2, a dynamic DWLAN is considered while the association algorithms for two dynamic scenarios are presented in Sections 4.1.2.1 and 4.1.2.2.

4.1.1 Offline AP Association Algorithm

We assume that an initial bipartite graph is obtained from SSF associations as employed in current 802.11 standards. STAs first associate with APs using SSF, then the bipartite graph of this set of SSF associations is used as the input graph to the association algorithm. The association optimization algorithm is presented as **Algorithm 1**, and assumed to be executed at a central controller with network-wide information about all STAs and APs on the network. With this algorithm, it is shown that there exist a set $\{w_{ij}|1 \leq i \leq N, 1 \leq j \leq M\}$ from weighted bipartite matching that maximizes network throughput, that is, we

$$\begin{aligned}
&\text{maximize} && \sum_{j}^{M}\sum_{i}^{N}\Upsilon_{ij}x_{ij} && \text{(4.1a)}\\
&\text{subject to} && \sum_{j=1}^{M} x_{ij} = 1, \forall i \in \mathcal{N} && \text{(4.1b)}\\
& && \sum_{j=1}^{M} \mathcal{I}^{ij}x_{ij} \leq \Gamma, \forall i \in \mathcal{N} && \text{(4.1c)}\\
& && x_{ij} \in \{0,1\}, 1 \leq i \leq N, 1 \leq j \leq M, && \text{(4.1d)}
\end{aligned}$$

where the variable x_{ij} in constraint 4.1(b) ensures that each STA associates with one and only one AP; $x_{ij} = 1$ if STA$_i$ is associated with AP$_j$, and $x_{ij} = 0$ otherwise. At the initialization stage, **Algorithm 1** collects the SINRs of all STAs to construct an $N \times M$ weight matrix, $\mathbf{W}$, representing the uplink SINRs of N STAs through M APs on the WLAN. The association optimization begins in *Step 2* of **Algorithm 1** by performing the weighted bipartite matching using **Algorithm 2** with $G = (\mathcal{N}, \mathcal{A}, \mathcal{E})$

as the input weighted graph.

In **Algorithm 2**, without using slack variables, φ in Equation (4.2) can be computed in $O\left(N^2\right)$ time, but using slack variables yields efficient computation with complexity $O\left(N\right)$ in search for an augmenting path [38], [39], [40]. With a centralized execution of this algorithm, optimal association is achievable with polynomial complexity $O\left(N^3\right)$ on a WLAN with N STAs. Efficient implementation of the Hungarian algorithm has been extensively discussed in the literature, e.g., [38] and [39] provide thorough details. When an optimum matching, $\mathcal{M}$ is obtained in *Step 2(c)* of **Algorithm 1**, the selected edge weight matrix maximizes the overall WLAN uplink throughput.

Algorithm 1 : Optimal Association

```
Input : $G = (\mathcal{N}, \mathcal{A}, \mathcal{E})$, $\gamma_{ij}$, $\Gamma$
Output : Match $\mathcal{M}$ of $N$-STAs to $M$-APs, $\Upsilon_{ij}$, $\forall i \in \mathcal{N}$, $j \in \mathcal{M}$.
1. Initialization:
    a. Set given SSF association as initial graph, $G$
    b. Compute throughput, $\Upsilon_{ij}$ of SSF association.
for ($i \in \mathcal{N}$ & $j \in \mathcal{A}$ & $x_{ij} == 1$) do
        c. Estimate SINR, $\gamma_{ij}$ of STA$_i$ through AP$_j$
        d. Set $w_{ij} \leftarrow \gamma_{ij}$ and construct $N \times M = |\mathcal{N}| \times |\mathcal{A}|$ weight matrix $\mathbf{W}$:
                $\mathbf{W}(i,j) = \{w_{ij} \in \mathbb{Z} : \forall i \in \mathcal{N}, j \in \mathcal{A}\}$
end for
2. Begin Association Optimization:
    a. Begin with an empty match, $\mathcal{M} \leftarrow \emptyset$
    b. Call Algorithm 2: perform $N = |\mathcal{N}|$ stages.
    c. Output matching, $\mathcal{M}$ and throughput $\Upsilon_{ij}$
```

Algorithm 2 : Hungarian Algorithm [40]

1. Declare dual variables α_i and β_j for set $\mathcal{N}$ of STAs and set $\mathcal{A}$ of APs respectively as follows: $\forall i \in \mathcal{N}, \alpha_i = 0$ and $\forall j \in \mathcal{A}, \beta_j = \min_i (w_{ij})$.

2. Select an unmatched node $i \in \mathcal{N}$ and set it as the root STA$_i*$ of a Hungarian tree.

3. Set the unmatched nodes as the root of the growing Hungarian tree.

4. Create two sets for each covered or reachable nodes $i \in \mathcal{N}$ and $j \in \mathcal{A}$ in the Hungarian tree: $\mathcal{N}^* = i \in \mathcal{N}$ and $\mathcal{A}^* = j \in \mathcal{A}$. If an augmenting path is found, Go to 6. If no augmenting path and the Hungarian tree keeps growing, Go to 5 and modify slack variables.

5. Modify the dual variables α_i and β_j using Equation (4.2):

$$\varphi = \frac{1}{2} \min_{i \in \mathcal{N}, j \notin \mathcal{A}} (w_{ij} - \alpha_i - \beta_j) \tag{4.2}$$

$$\alpha_i \leftarrow \begin{cases} \alpha_i + \varphi & i \in \mathcal{N} \\ \alpha_i - \varphi & i \notin \mathcal{N} \end{cases}$$

$$\beta_i \leftarrow \begin{cases} \beta_j - \varphi & j \in \mathcal{A} \\ \beta_j + \varphi & j \notin \mathcal{A} \end{cases}$$

6. Flip the matched and unmatched edges along an augmenting path to augment the current matching, $\mathcal{M}_N = (\mathcal{M}_{N-1} - P) \cup (P - \mathcal{M}_{N-1})$, where $\mathcal{M}_{N-1}$ is the previous matching at $N - 1$ stage and P is the set of edges in the augmenting path.

4.1.2 Online AP Association Algorithm

In this section, we propose an online version of **Algorithm 1** to optimize STA-AP association in dynamic network scenarios where STAs are joining and leaving the network as opposed to the quasi-static scenario in the previous section. The optimal association algorithm in Section 4.1.1 solves the assignment problem in $O\left(N^3\right)$ using the Hungarian algorithm and it assumes the existence of a-priori information about the edge weights, w_{ij}, which are estimated using RSSI. In wireless networks, new users dynamically join and exit the network, and move within the network. We therefore seek to obtain an optimal association that considers network changes, and update the optimal association when there is a change in the network. The default strong signal first (SSF) association method specified in IEEE 802.11 standard allows a new user entering the network to select the AP with the strongest received signal strength (as explained earlier). In dense WLAN, SSF association could cause the problem of least-congested and overloaded APs, which degrades the network-wide throughput.

In this section, two dynamic scenarios will be considered, first, a scenario whereby new users are joining while some existing users are leaving the network. The second scenario considers user mobility within the network, a situation whereby certain users change their physical locations. This mobility is modeled using the *Random Waypoint Mobility Model* (RWMM) [41], [1]. Ideally, we would like to obtain STA-AP association for new entrants such that the throughputs of already associated STAs are not degraded as a result of changes in the network. Our online algorithm for STA-AP association is inspired by [40], [42] and throughout this section, we will use the terminologies and descriptions therein. Given an initial optimal association, $\mathcal{M}$ from Section 4.1.1, we want to repair the assignment of STAs to APs when edge weights change due to mobility, addition of new users, or exit of users.

4.1.2.1 New STAs Joining the Network

When the network receives new STAs, new pairs of vertices for these new entrants are added to the existing weighted matched bipartite graph, G with optimal matching, $\mathcal{M}$. In other words, G is the graph of the optimal association obtained in Section 4.1.1 with an optimal matching $\mathcal{M}$. When new nodes are added to G, we have an extended graph $G^{'} = \left(\mathcal{N}^{'}, \mathcal{A}^{'}, \mathcal{E}^{'}\right)$ with weight matrix $\mathbf{W}^{'}$ of size $N + 1 \times N + 1$. Therefore, the goal is to determine the maximum weighted matching of the extended graph $G^{'}$ that gives a new optimal association that includes the new STAs joining the network without affecting the throughput of already associated STAs. The algorithm outlined in **Algorithm 3** provides the steps to obtain optimal association for the extended WLAN.

The arrival of new STAs does not change the edge weights of existing associated STAs, but the idea of **Algorithm 3** is to efficiently admit the new users by finding admissible edges for them. Following the initialization of matching $\mathcal{M}$ in Step (1a), $N + 1 \times N + 1$ weight matrix $\mathbf{W}^{'}$ is constructed for the extended graph $G^{'}$ consisting of new weights w_{ij} for the new entrants. In Step (1c), feasible values are determined for the new dual variables α_{N+1} and β_{N+1} while the new optimal association is obtained in Step (2) using the conventional Hungarian algorithm routine outlined in **Algorithm 2**. The initialization phase is an $O\left(N\right)$ operation while the Hungarian routine has $O\left(N^2\right)$ complexity; thus, **Algorithm 3** has $O\left(N^3\right)$ complexity (please see [42] for proof of this complexity) using the data structure specified in [38] for implementation.

Algorithm 3 : Incremental Association Optimization

```
Input : $G^{'} = \left(\mathcal{N}^{'}, \mathcal{A}^{'}, \mathcal{E}^{'}\right)$, $\gamma_{ij}$, $\Gamma$, $\mathcal{M}$
Output : Maximum Matching $\mathcal{M}^{'}$, $\Upsilon^{'}_{ij}$, $1 \leq i \leq N$, $1 \leq j \leq M$.
1. Initialization:
    a. $\mathcal{M}^{'} \leftarrow \mathcal{M}$
    b. Construct $N+1 \times N+1$ weight matrix $\mathbf{W}^{'}$:
        $w^{'}_{ij} \leftarrow \gamma_{ij}$, $\mathbf{W}^{'} = \{w^{'}_{ij} | \forall i \in \mathcal{N}^{'}, j \in \mathcal{A}^{'}\}$
    c. Set dual variables $\alpha_{N+1}$ and $\beta_{N+1}$:
        $\beta_{N+1} = \max\{\max_{1\leq i\leq N}\left(w^{'}_{i(N+1)} - \alpha_i\right), w^{'}_{(N+1),(N+1)}\}$
        $\alpha_{N+1} = \max_{1\leq j\leq N+1}\left(w^{'}_{(N+1)j} - \beta_j\right)$
2. Begin Association Optimization:
    a. Begin with $\mathcal{M}^{'}$
    b. Call Algorithm 2: perform $N = |\mathcal{N}^{'}|$ stages.
    c. Output matching, $\mathcal{M}^{'}$ and throughput $\Upsilon^{'}_{ij}$
```

4.1.2.2 Mobility: Dynamic SINRs

User mobility and arrival of new users result in certain dynamics within the network; that is, some users change their physical locations within the networks. Similarly, when new STAs join the network, there is a possibility of increased interference caused to other users, and consequently, the SINRs (corresponding to edge weights) of some STA-AP links decrease. In situations like these, we can repair the optimal association $\mathcal{M}$ to obtain a new optimal association $\mathcal{M}^{*}$. Therefore, in this section, we are concerned with repairing the optimal association $\mathcal{M}$ when any of the link SINRs (edge weights) changes either due to user's mobility or increased interference on the link.

A change in the edge weight matrix **W** can occur either in the i^{th} column or j^{th} row; resulting in change in the through-maximizing SINR. A new optimal association $\mathcal{M}^*$ is performed using the steps outlined in **Algorithm 4**.

Algorithm 4 : Dynamic Association Optimization

```
Input : $G = (\mathcal{N}, \mathcal{A}, \mathcal{E})$, $\gamma_{ij}$, $\mathcal{M}$, $\mathbf{W}$
Output : Optimal matching $\mathcal{M}^*$, $\Upsilon_{ij}$, $1 \leq i \leq N$, $1 \leq j \leq M$.
1. Initialization:
    $\mathcal{M}^* \leftarrow \mathcal{M}$
  if row $j$ of $\mathbf{W}$ changed:
      Set $\mathcal{M}^*$ = $\mathcal{M}^*$ - $\mathcal{E}(i,j)$ //Remove the changed or affected edge
      Set $\alpha_i^* = \max_j (w_{ij} - \beta_j)$
  else if column $i$ of $\mathbf{W}$ changed:
      Set $\mathcal{M}^* = \mathcal{M}^*$ - $\mathcal{E}(i,j)$ //Remove the changed or affected edge
      Set $\beta_j^* = \max_i (w_{ij} - \alpha_i)$
2. Begin Association Optimization:
  //Starting with updated $\mathcal{M}^*$ from the initialization phase.
  a. Begin with match, $\mathcal{M}^*$
  b. Call Algorithm 2:  perform $N^* = |\mathcal{N}^*|$ stages.
  c. Output matching, $\mathcal{M}^*$ and throughput $\Upsilon_{ij}$
```

Staring off with the optimal matching, $\mathcal{M}$ from Section 4.1.1, if there is a change in the j^{th} row, the affected edge weight becomes unmatched and it is removed from $\mathcal{M}^*$. Removing the affected edge from the matching decreases the cardinality of $\mathcal{M}^*$ by one (i.e. $|\mathcal{M}^*| = |\mathcal{M}^*| - 1$).A similar step is taken when the change occurs in the i^{th} column of **W**. Eliminating the affected edge permits us to seek another admissible edge for the affected node by computing new feasible values for the previous dual variables α_i and β_j to obtain new dual variables α_i^* and β_j^*. Therefore, letting $\mathcal{N}^*$ be the set of nodes with changed edge weights, which are now unmatched after the initialization phase, we can obtain a complete matching, $\mathcal{M}^*$ by performing $N^* = |\mathcal{N}^*|$ stages of the Hungarian routine outlined in **Algorithm 2**. The initialization phase of **Algorithm 4** has $O(N)$ complexity. If there are N^* affected nodes, N^* stages of **Algorithm 2** are executed, there is a resultant computational complexity of $O(N^*N^2)$. We remark that using the incremental algorithm to update AP associations does not change the optimality of the resultant matching but reduces the computational process of optimizing AP associations from scratch.

4.2 Intra-BSS CCA Threshold Calibration

After optimal association $\mathcal{M}$, each AP forms a contention domain known as the basic service set (BSS) depicted in Figure 4.1; a BSS consists of an AP and its associated STAs. By virtue of the WLAN contention mechanism, STAs in BSS_1 must defer to other STAs' transmissions within the same BSS to prevent the *hidden terminal* problem. A CCA threshold is determined for each BSS such that all STAs in the same BSS have an identical CCA threshold setting for sensing and contention in the uplink. This reduces the degree of contention and packet loss due to minimized collision. The CCA threshold for each $\text{BSS}_j, 1 \leq j \leq M$ is determined using **Algorithm 5**, where the cell-edge SINR ϕ in the BSS_j of AP_j is used to compute the maximum allowed interference, which determines the CCA threshold, Γ_j for BSS_j.

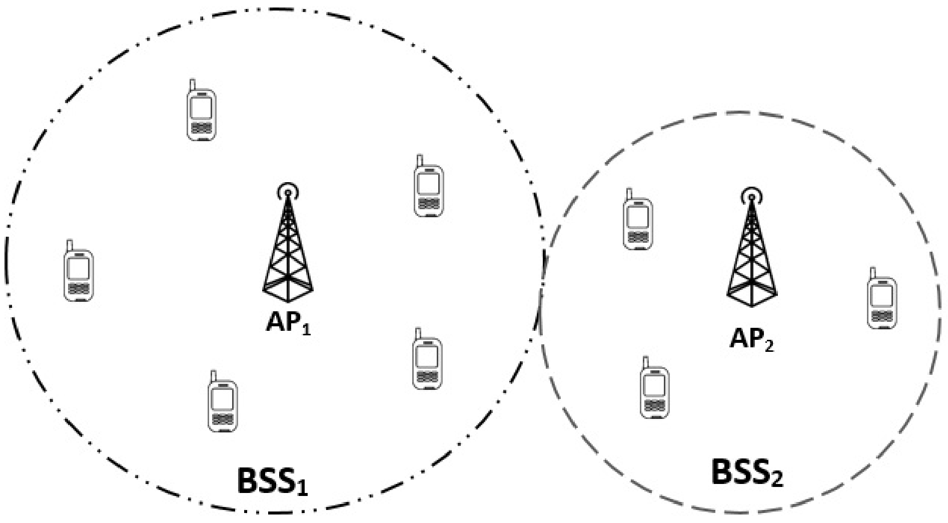


Figure 4.1: WLAN basic service set of each AP.

Algorithm 5 : CCA Adjustment

```
CCA setting for each BSS$_j$, $1 \leq j \leq M$ in the resultant matching, $\mathcal{M}$:
1. Obtain SINRs of STAs associated with AP$_j$ in BSS$_j$.
2. Let $\phi$ be the lowest SINR in BSS$_j$.
3. Let $\varrho$ be the received power of STA with the lowest SINR $\phi$ in BSS$_j$
4. Set $\gamma_o$ to correspond to the SINR with the highest data rate in Table 3.2 such that $\gamma_o \leq \phi$
5. Calibrate $\Gamma_j$ to support $R_k$ for STA with SINR $\phi$ in BSS$_j$:
       $\Gamma_j = \Big(\frac{\varrho}{\gamma_o} - N_o\Big), 1 \leq j \leq M.$
6. AP$_j$ broadcasts new CCA setting $\Gamma_j$ to all STAs in BSS$_j$
7. All STAs in BSS$_j$ set CCA sensitivity to $\Gamma_j$.
```

This reduces the hidden terminal problem and ensures that the user at the cell-edge, i.e., having the lowest SINR, can be supported while preventing interference at the AP's receiver through the MAC layer contention avoidance mechanisms (CSMA/CA). Therefore, in each BSS, **Algorithm 5** performs an adjustment of the CCA threshold for each supported data rate. For $1 \leq j \leq M$, AP$_j$ sends this new CCA threshold, Γ_j to all STAs in its BSS.

As an alternative to using **Algorithm 5**, the CCA threshold adjustment can be performed more frequently than optimizing AP associations. In fact, CCA threshold adjustment can be done dynamically and periodically every slot time. This enables nodes to choose an appropriate CCA threshold based on estimating the interference level at the beginning of the current slot time. However, unlike the CCA threshold adjustment in **Algorithm 5** that is centralized at the AP in a BSS, a slot-time-based CCA threshold adjustment process would be distributed.

4.3 Chapter Summary

To maximize or enhance throughput in DWLAN, algorithms have been proposed for association coordination and CCA adjustment. Since Kuhn-Munkres algorithm achieves optimal assignment, the AP association algorithms in this chapter might guarantee optimal association. The offline AP association algorithm assumes *a priori* information about the link SINRs and that the link SINRs remain unchanged overtime. On the other hand, the online AP association algorithm considers network dynamics when link SINRs change due to users leaving and/or joining the network or users mobility. To further improve throughput achieve from optimizing AP associations, an algorithm is proposed for calibrating the CCA threshold when the set of STA-AP associations in the network changes. Next, we implement and test these algorithms in Chapter 5 to quantify the performance gains in DWLAN test scenarios.

Chapter 5

Performance Evaluation

The performance metrics and the numerical results are discussed in this chapter. First, the system parameters and assumptions made in simulating and analyzing the proposed algorithms are discussed. In Sections 5.1, 5.2 and 5.3, the simulation setup, system scenarios, and the performance metrics used for evaluation are presented, respectively. Section 5.4 presents the numerical results for the uplink performance in a *quasi-static* scenario. The downlink throughput performance in a *quasi-static* scenario are presented in Section 5.5. Results are presented in Sections 5.6 and 5.7 for CCA threshold adjustment and fairness performance, respectively, while Section 5.8 compares performances in two scenarios of interest. Section 5.9 presents performance in a dynamic scenario and the chapter is concluded in Section 5.10.

5.1 Simulation Setup and Parameters

For the simulation, we use MATLAB 8.2 to generate random network nodes on a specified network area and simulate the CSMA/CA protocol described earlier in Sections 2.1.2 and 2.1.3. The locations of the Tx-Rx pairs are determined by estimating the Euclidean distances between the transmitters and the receivers on an area of 200m

× 200m where APs and STAs are randomly positioned. The PHY and MAC models are based on the PHY and MAC parameters specified in the IEEE 802.11 standards for IEEE 802.11b network as listed in Table 5.1. For transmission in the downlink and the uplink, APs transmit with power of 100mW (20 dBm) and all STAs transmit with a uniform power of 15.85 mW (12 dBm) respectively on a 2.4GHz band based on the IEEE 802.11b protocol. For the entire channel model, we consider only the single input single output (SISO) case whereby each AP and STA has single transmit-receive 2.4 GHz antenna.

Table 5.1: Simulation parameters for IEEE 802.11b WLAN

Simulated Network	
WiFi Protocol	802.11b
Simulation network area	200m × 200m
Number of STAs $N = \|\mathcal{N}\|$	300
Number of APs $M = \|\mathcal{A}\|$	30
Simulation time	200 sec
PHY Parameters	
Number of Antenna	1 (2.4GHz-antenna)
Frequency Band	2.4 GHz
STAs Transmit power	15.85mW
APs Transmit Power	100mW
Background Noise, N_o	-90dBm
Path loss exponent, δ	3.4 (ITU rush hour)
Receiver sensitivity	- 65dBm
MAC Parameters	
Slot time	20μs
SIFS	10μs
DIFS	50μs
Packet Size	1460 bytes
Min. CW	32
Max. CW	1024
Global CCA threshold, Γ	- 60dBm
PCS	Enabled
VCS	Disabled
RTS/CTS	Disabled

To model the channel between a transmitter-receiver pair, we ignore fading (i.e. normalize channel gains to unity) and use a simple path loss model in Equation (3.1) with the path loss exponent set as $\delta = 3.4$ based on the ITU recommendation for "rush hour" propagation model [43] to estimate the received power at the receiver. From our testbed measurement in Figure 2.12 under Section 2.3.2, the background noise in the observed campus dense Wi-Fi environment is -100 dBm, but to estimate link SINR in our model, the noise power N_o is set to -90 dBm while the receiver sensitivity is -65dBm. The MATLAB 8.2 code for the topology setup and PHY (channel) model is presented in Appendix A while the generated Wi-Fi network layouts are depicted and discussed later under Section 5.2.

Subsequent to the PHY modeling, the same MATLAB version is used to simulate the CSMA/CA algorithm. The flowchart and the MATLAB code for the CSMA/CA algorithm are provided in Appendix B and Appendix C, respectively. Recapping from Section 2.1.3, the CSMA/CA uses time intervals (Slot time, SIFS, and DIFS) to determine if the channel is busy or idle. To account for these time intervals in our simulation, we define the overall simulation time as 200 seconds and set the slot time, SIFS, and DIFS to 20μs, 10μs, and 50μs respectively as specified in the 802.11b standard. Setting a discrete simulation time is important not only because the CSMA/CA is a time-based protocol, it is also useful is measuring transmission time delay. To determine active APs and STAs, the packet arrival rates at the STAs and APs is randomized in MATLAB environment using exponential distribution with parameter $\lambda = 1/\text{Slot time}$ (1 packet per slot time).

While the time duration of a packet depends on the modulation type and number of carriers, to avoid the simulation process from becoming overly complex, we assume that all nodes transmit a constant symmetric packet size of length 1460 bytes per slot time. For the channel assessment or sensing part of the CSMA/CA, the PCS is enabled with the global CCA threshold, Γ predefined as -60dBm, while VCS and RTS/CTS frames are disabled. From the discussion in Section 2.1.2, while it is an option to use RTS/CTS frames and VCS, disabling them reduces frame exchange overhead. Therefore, as illustrated in Section 2.1.5, when packets randomly arrive at any node's buffer, the node senses the channel to see that the energy level does not exceed -60dBm. If the sensed energy on the channel is below this threshold, the node proceeds through the stages of transmitting its packet as mandated by the CSMA/CA protocol. The minimum and maximum contention window in 802.11b are 32 and 1024 respectively.

The flowchart for the CSMA/CA protocol is depicted in Appendix B and the MATLAB implementation code is provided in Appendix C. Other simulation environments for CSMA/CA protocol include OPNET, NS-2, NS-3, and other off-the-shelf simulators. However, MATLAB is used to develop our custom simulation because of the flexibility it offers in terms of top-down approach to writing the required code and taking total control. This avoids the complexity of implementing a module for the optimal matching algorithm on top of OPNET, NS-2 and NS-3. Without the need to implement a customized optimal AP association algorithm, these third party simulators would have also been suitable for simulating Wi-Fi transmissions.

5.2 Simulated System Scenarios

Our system setup for simulation is in two facets. The first setup imitates a Wi-Fi environment with randomly deployed APs while the second scenario is an AP deployment that follows a regular pattern. We generate random distances on a 200 × 200 m grid for the STAs locations on the network and other random distances representing AP locations on the same grid to emulate DWLANs. The WLAN consists of APs that are randomly deployed over an area of 200 m × 200 m, and each AP is spatially separated from one another. The STAs are randomly distributed at different locations resulting in an experimental topologies shown in Figures 5.1 and 5.2. The distance d_{ij} separating each STA i from each AP j and the corresponding received signal strength in dBm are measured.

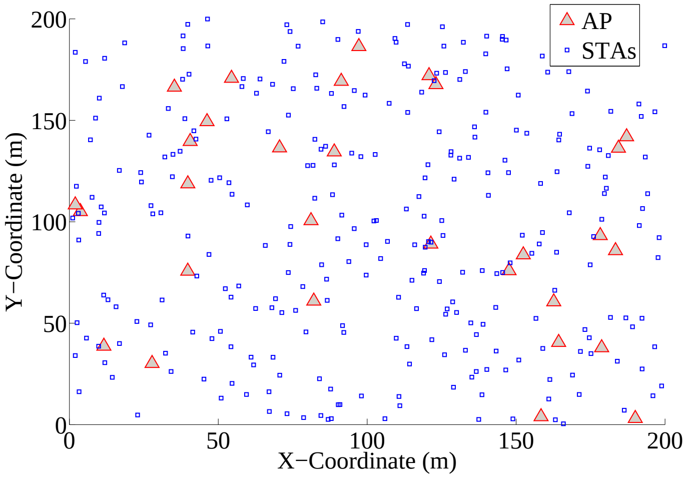


Figure 5.1: Experimental Topology 1 (ET-1): Random AP deployment.

The random deployment of STAs and APs in Figure 5.1 is referred to as experimental topology 1 (ET-1) and it corresponds to the type of WLAN deployments currently found in multi-tenant apartment buildings, multi-office enterprise buildings, and campus locations where APs are not usually deployed in regular patterns as in cellular networks. The second experimental topology (ET-2) in Figure 5.2 emulates a conference hall or an event auditorium with a large number of WiFi enabled devices. APs are uniformly spaced 20m apart on the left, center, and right parts of the

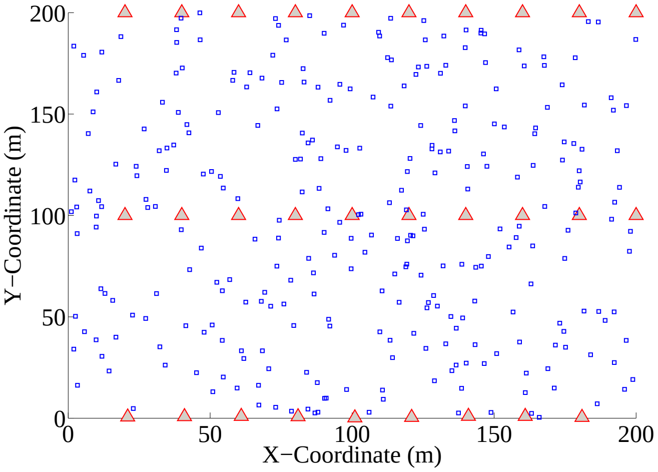


Figure 5.2: Experimental Topology 2 (ET-2): Regular AP deployment

auditorium roof-top. In both ET-1 and ET-2, it is assumed that sufficient frequency planning has been performed and APs transmit on the available orthogonal channels to avoid co-channel interference.

In both the random (ET-1) and the regular (ET-2) topologies, there are 300 STAs associated with 30 APs. Prior to using **Algorithm 1** from Section 4.1.1 for association optimization, it is assumed that all STAs independently associate with APs using the SSF association. Starting with this SSF association, **Algorithms 1** and **2** perform optimum association of STAs to APs using weighted bipartite matching. Recall that, AP association optimization is motivated by the fact that there is high contention and suboptimal association that degrades throughput in a high density WLAN [24]. With the proposed optimal association algorithm, the majority of the STAs obtain a strong signal when matched with an AP other than the one chosen using SSF association; this maximum weighted bipartite matching algorithm becomes efficient in dense 802.11 networks where STAs receive sufficient signal strength from multiple APs; for example, the minimum RSS in a 20MHz channel is -65dB at 54Mbps

(64-QAM).

5.3 Performance Metrics

The simulation results are presented and discussed using four (4) performance metrics, which are throughput, packet transmission delay, number of packet re-transmissions, and fairness. The uplink and downlink throughputs are estimated using Shannon capacity in Equations (3.6) and (3.11), respectively, to benchmark the achievable throughput performance in SSF scheme against the proposed AP associations scheme. Also, due to the unavoidable contention in WLAN access protocol (discussed in Section 2.1.3), throughput can either get better or worse over a given period of time even if the number of associated STAs remains constant in a *quasi-static* network. Therefore, in subsequent sections, we measure throughput performance of these AP association schemes over a specific simulation time to reveal how well the algorithms perform in time. In the same vein, the throughput obtained using fixed CCA threshold is compared to the throughput when the CCA threshold is adjusted per BSS using **Algorithm 5** in Section 4.2. This reveals the performance improvement of per-BSS CCA adjustment after the AP association is optimized.

As described in Sections 2.1.2 and 2.1.3, the inevitable contention and collision inherent in CSMA/CA protocol degrade performance. Therefore, to have a broader view of performance improvement in the presence of CSMA/CA access protocol, we further compare performance in terms *transmission delay* incurred over time. This gives a more complete picture of the effect of CSMA/CA protocol on AP association. In a nutshell, *transmission delay* occurs due to the length of back-off time and congestion experienced by transmitting nodes while waiting for the channel to become idle;

this delay is measured per BSS over a specified simulation time when SSF association is used compared to when the proposed association scheme is in effect.

Similarly, recapping from Sections 2.1.2, 2.1.3, and 2.1.5 under Chapter 2, collision occurs when two or more nodes simultaneously attempt to access the medium. When this happens, existing packets on the channel get corrupted leading to a re-transmission of the affected packets. While payload packets can get corrupted due to collision, ACK frames from a receiver are also susceptible to collisions when a transmitter's signal arrives on the channel while ACK data is being transmitted. The packet re-transmission rate depends on the rate of packet loss due to collision or unacknowledged packet receipt (a situation when Tx doesn't receive ACK for the transmitted packets) mandating the Tx to re-transmit a packet or a block of packets. Hence, for a specified simulation time, we measure the number of packet re-transmissions in SSF scheme and our proposed scheme for comparison.

Subsequently, the simulation time will be referred to as *network operational time*, which helps keep track of performance over a given time in both *quasi-static* and *dynamic* scenarios; it is most important in measuring transmission delay and packet retransmission incurred for a specified time. The last performance metric is the Jain's fairness index (JFI), which has been widely used (e.g. in [1] [44]) to quantify performance in terms of fairness. Therefore, in this thesis, with the known throughput Υ_{ij} between STA$_i$ and AP$_j$, the throughput-based fairness of the SSF association scheme is compared with that of the optimal association using the following well-known fairness index equation [1] [44]:

$$\mathcal{J} = \frac{\left(\sum_{i=1}^{N} \Upsilon_{ij}\right)^2}{N\left(\sum_{i=1}^{N} {\Upsilon_{ij}}^2\right)} \qquad \text{where} \qquad \Upsilon_{ij} \geq 0. \tag{5.1}$$

5.4 Uplink Performance Enhancement

In this section, the uplink throughput performances in ET-1 and ET-2 are presented for the quasi-static network and dynamic scenarios. Subsequently, transmission delay and the number of retransmissions per BSS in ET-1 and ET-2 are presented for the two scenarios of interest.

5.4.1 Quasi-Static Scenario in ET-1 Network

The performance improvement of the proposed optimal association algorithm is measured using the rate determined by Eq. (3.6). Comparing the performance of optimal association with that of SSF association in ET-1, Figure 5.3 presents the cumulative distribution of STAs' uplink throughput and reveals significant performance improvement of the proposed optimal association over the SSF association, where at least 90% of the STAs experience a significant increase in throughput when their associations with APs are coordinated using **Algorithm 1**. Examining the 10% level of STAs in Figure 5.3, about 54% (2.6 to 4 Mbps) performance improvement in scenario ET-1 is obtained using the AP association scheme over SSF association currently supported in 802.11 standards.

The achievable throughput per STA depends on the degree of contention among STAs associated with an AP and because the amount of traffic an AP can support depends on the number of associated STAs. Therefore, Figure 5.4 depicts the average uplink throughput per AP over the number of associated STAs. It can inferred from this figure that the proposed optimal association scheme enhances the average UL throughput at majority of the BSSs when compared with the SSF Association. Although, for some APs the performance of the optimal association scheme seems

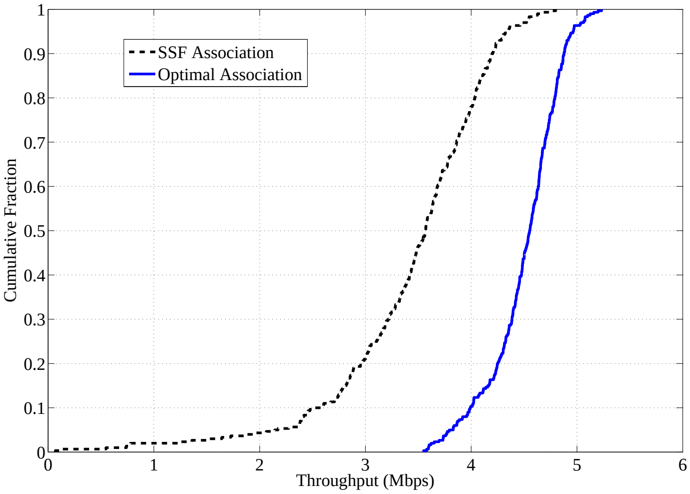


Figure 5.3: Cumulative distribution of throughputs of all STAs in the random experimental topology (ET-1) using SSF association versus optimal association.

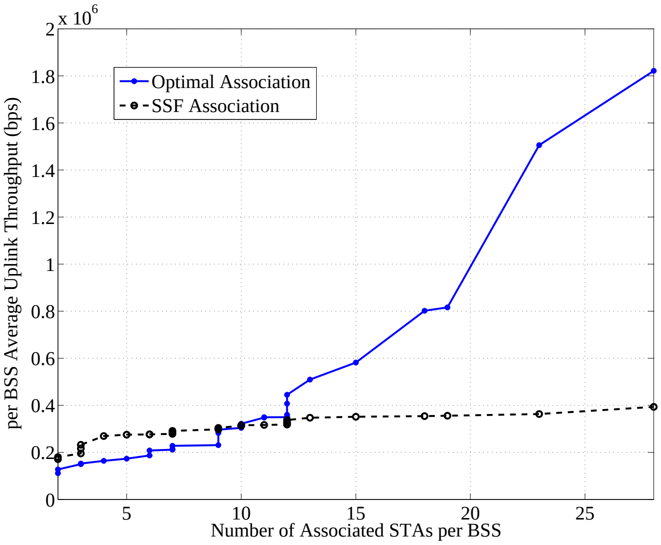


Figure 5.4: Average uplink throughput per AP versus number of associated STAs in ET-1.

insignificant as seen in Figure 5.4 where the SSF association scheme performs better at the first nine (9) APs with 5 to 10 associated STAs. Regardless of this observation, the optimal association scheme achieves better average uplink throughput for 70% of the BSSs, which is significant and would increase the overall network throughput.

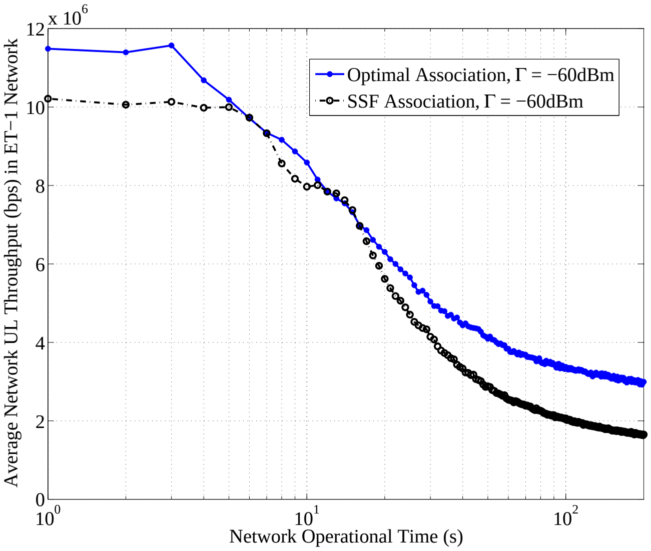


Figure 5.5: Average network uplink throughput over network operational time of 200 seconds in ET-1 network with fixed CCA threshold of −60dBm.

Throughput enhancement over a network operational time of 200 seconds is presented in Figure 5.5. Figure 5.5 presents the average network-wide UL throughput achieved using both association scheme with a globally fixed CCA threshold of −60dBm. Examining the average UL throughput for 90% of the uptime, the optimal association scheme yields approximately 50% throughput gain over SSF association. For the majority of the simulation time, with the optimal association scheme, the average network UL throughput remains greater than the average UL throughput from the SSF association scheme.

To further compare these two association algorithms in the ET-1 network, average delay, the number of packet retransmissions are two other important performance metrics. The average transmission delay in ET-1 is observed and Figure 5.6 presents

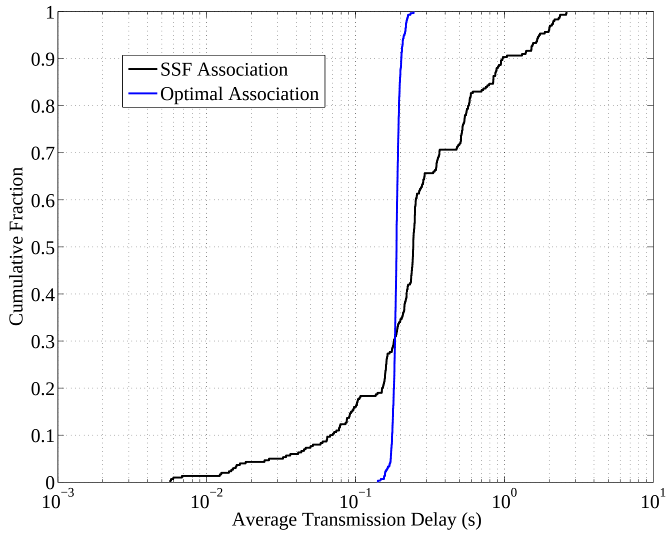


Figure 5.6: Cumulative distribution of per STA packet transmission delay (s) in ET-1 network with 200 seconds operational time and 300 STAs.

the cumulative distribution of per STA average packet transmission delay experienced during the overall network uptime. Overall, optimal association outperforms the default SSF scheme as shown in Figure 5.6, which reveals that 70% of users will experience a higher delay using SSF association while the proposed optimal association minimizes the delay in SSF association by nearly 50% for the majority of users.

In the same vein, Figure 5.7 shows the overall network transmission delay in both UL and DL of ET-1 averaged over network operational time. Obviously, the proposed optimal association reduces packet transmission delay more than the default IEEE 802.11 SSF association. Examining the average delay incurred from both association schemes between 10s and 100s of network operational time in Figure 5.7, we can conclude that the proposed optimal association significantly reduces delay; overall, packet transmission delay reduces by almost 80% especially between 10 to

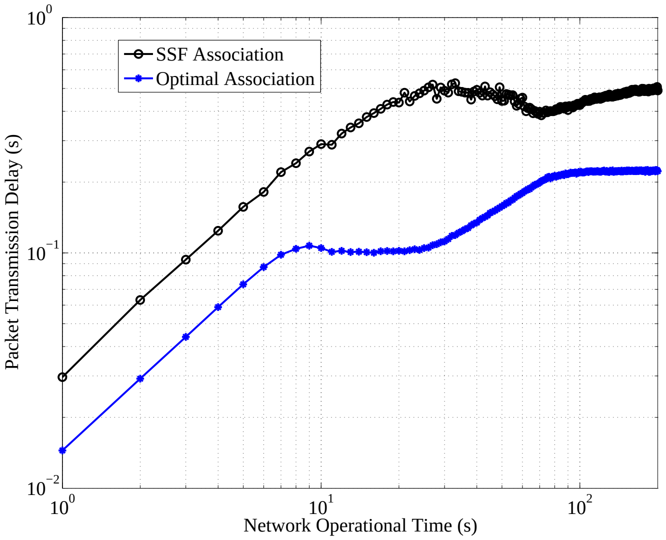


Figure 5.7: The average packet transmission delay (s) in ET-1 network with 300 STAs over 200 seconds operational time.

100s. As shown in the figure, with optimal association, the maximum transmission delay experienced by nodes is 0.22s while SSF association yields a delay of 0.5s; this performance gain is significant.

Due to collision, contention, and PCS (or CCA) induced back-off, packet retransmissions are inevitable in WLAN. Figure 5.8 depicts the average number of packet retransmission in ET-1 versus the corresponding average delay. We can see that there is a significant increase in the average number of packet retransmission encountered from using SSF association scheme when compared with the optimal association. Observing the maximum delay of 0.5s in Figure 5.8, the proposed optimal association method minimizes approximately 33.3% of the number of packet retransmissions incurred with SSF scheme. Minimizing packet retransmission rate has advantages, which include improved network reliability in terms of reduced packet error rates, efficient bandwidth usage, and energy efficiency of battery powered STAs.

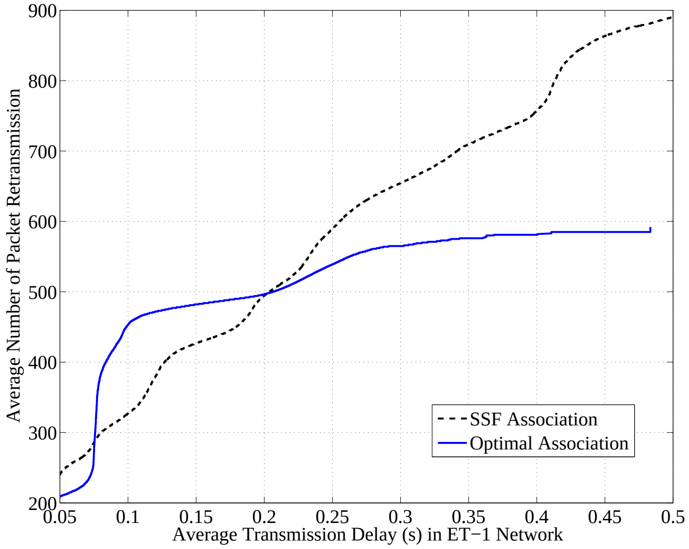


Figure 5.8: Average packet transmission delay (s) versus total number of packet retransmissions in ET-1 network with 200 seconds operational time, 30 APs and 300 contending STAs.

5.4.2 Quasi-Static Scenario in ET-2 Network

Applying the optimal association scheme in **Algorithm 1** to coordinate AP associations in ET-2 network enhances STA throughputs. This is apparent from Figure 5.9 where at least 90% of the STAs is guaranteed increase in throughput. From Figure 5.9 showing STA throughputs in ET-2, a more modest improvement of approximately 15% is observed at the $10th$ percentile of the cumulative distribution.

The degree of contention among STAs associated with an AP affects each STA's achievable uplink throughput. In Figure 5.10, the average uplink throughput per AP as a function of the number of associated STAs is presented. For majority of the APs in ET-2, the proposed optimal association scheme enhances the average UL throughput at each BSS when compared with the SSF Association. However, as the number of STAs associated with an AP increases, the performance of the

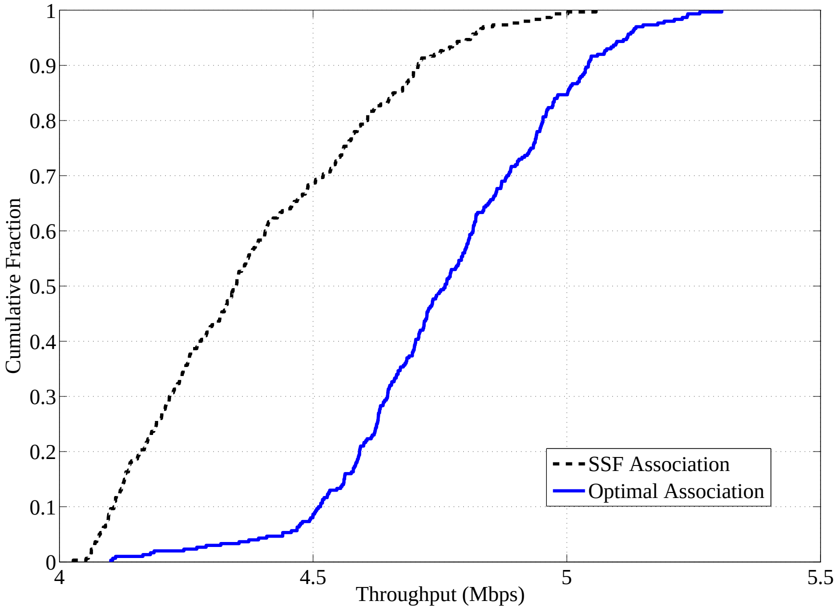


Figure 5.9: Cumulative distribution of throughputs of all STAs in the regular experimental topology (ET-2) using SSF association versus optimal association.

optimal association scheme becomes insignificant as seen in Figure 5.10 where the SSF association scheme outperforms the proposed scheme at three (3) APs with 17, 21, and 24 associated STAs. Regardless of this observation, our optimal association scheme achieves better average uplink throughput for 90% of the APs, which is a significant enhancement.

Figure 5.11 depicts throughput enhancement over network operational time. During the simulation, the network uptime is set to 200 seconds and the achievable network-wide UL throughput over this time is observed. Figure 5.11 presents the average network-wide UL throughput where the CCA threshold is globally fixed as -60dBm for each association scheme. Observing the average network UL throughput over 50% of the uptime, it is apparent that the optimal association scheme significantly outperforms SSF association. Similarly, for a longer uptime (approximately

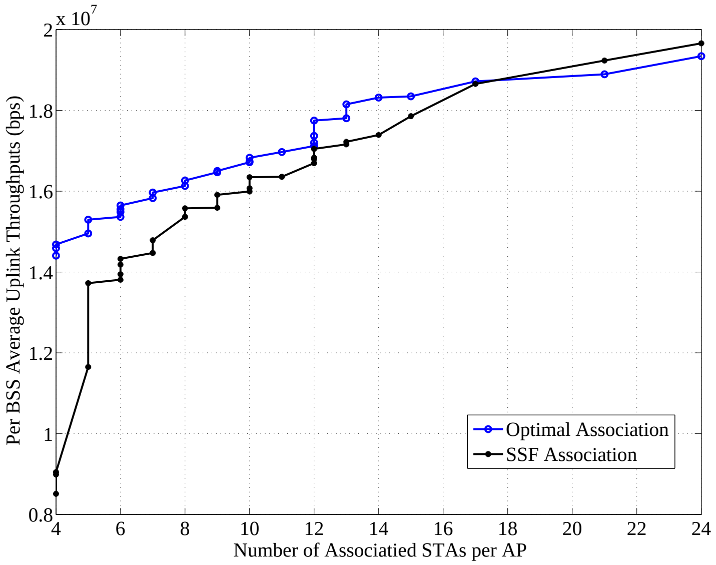


Figure 5.10: Average uplink throughput per AP versus number of associated STAs in ET-2.

90%) after using optimal association scheme, the average network UL throughput remains better than the average UL throughput from SSF scheme. As obvious in Figure 5.11, it is worth mentioning that observing the network for longer time might result in both association schemes achieving equal average UL throughput at a given time, t; not all the time though.

Next, we quantify performance in terms of average delay, the number of packet retransmissions incurred over the network uptime, and the proportional fairness in each BSS. In the ET-2 network, the average transmission delay is observed and Figure 5.12 depicts the cumulative distribution of per STA average packet transmission delay experienced during an uptime of 200 seconds. Over this simulation time, optimal association outperforms the default SSF scheme because Figure 5.12 reveals that 95% of the time, STAs will experience a higher delay using SSF association while less delay is guaranteed with the proposed optimal association.

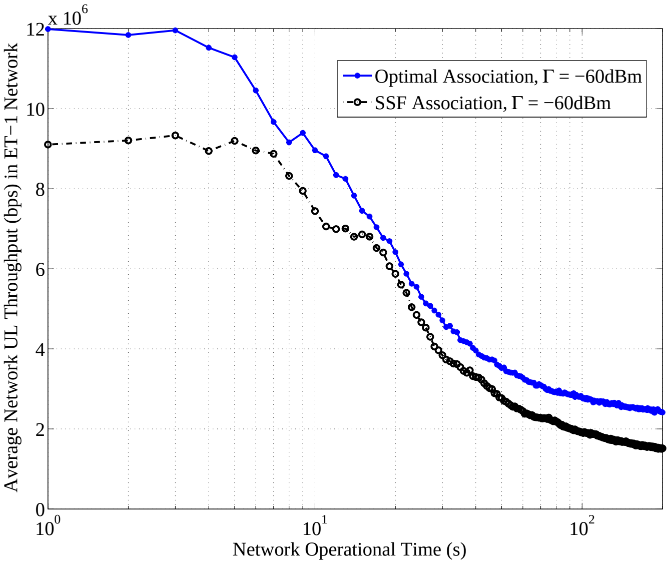


Figure 5.11: Average network uplink throughput over network operational time in ET-2 network with fixed CCA threshold of −60dBm.

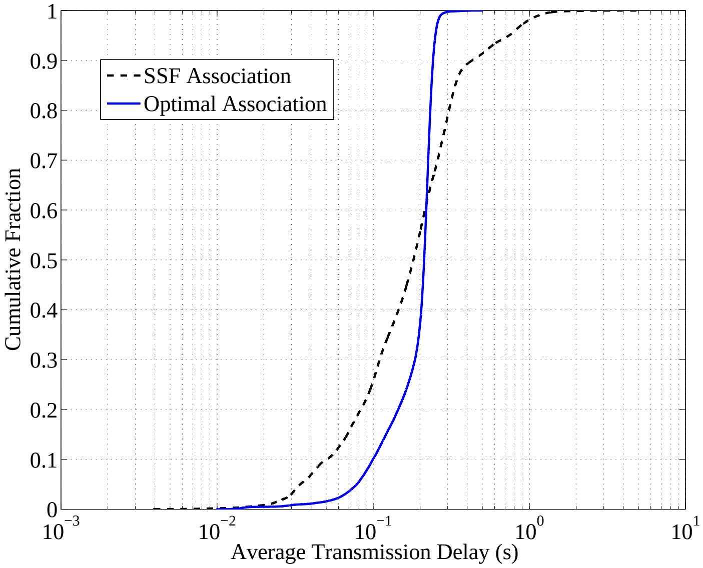


Figure 5.12: Cumulative distribution of per STA packet transmission delay (s) in ET-2 network with 200 seconds operational time and 300 STAs.

Similarly, in Figure 5.13 depicts the overall network transmission delay in both UL and DL of ET-2 averaged over network operational time; for each second of time, the average overall network transmission delay is measured and apparently, the proposed optimal association minimizes delay better than the IEEE 802.11 default SSF association. Observing the average delay incurred from using each association scheme after 10s of network operational time in Figure 5.13, optimal association minimizes approximately 20% of the delay incurred with SSF association.

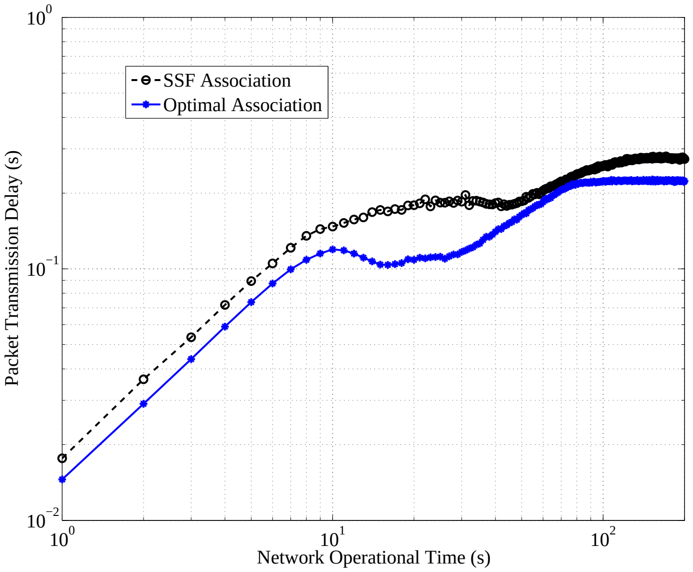


Figure 5.13: The average packet transmission delay (s) in ET-2 network with 300 STAs over 200 seconds operational time.

Observing the number of packet retransmissions due to collision, contention, and PCS (or CCA) induced back-off in Figure 5.14, the average number of packet retransmission in ET-2 versus the corresponding average delay is presented. From this figure, the average number of packet retransmission encountered from using SSF association scheme grows exponentially with increase in transmission delay while the situation is more stable with the optimal association. Taking the maximum average delay of 0.5s in Figure 5.14 as a reference point, the proposed optimal association method minimizes approximately 36.2% of the number of packet retransmissions incurred with SSF scheme; this is a significant performance improvement, which guarantees network reliability in terms of reduced packet error rates, efficient bandwidth usage, and energy efficiency of battery powered STAs.

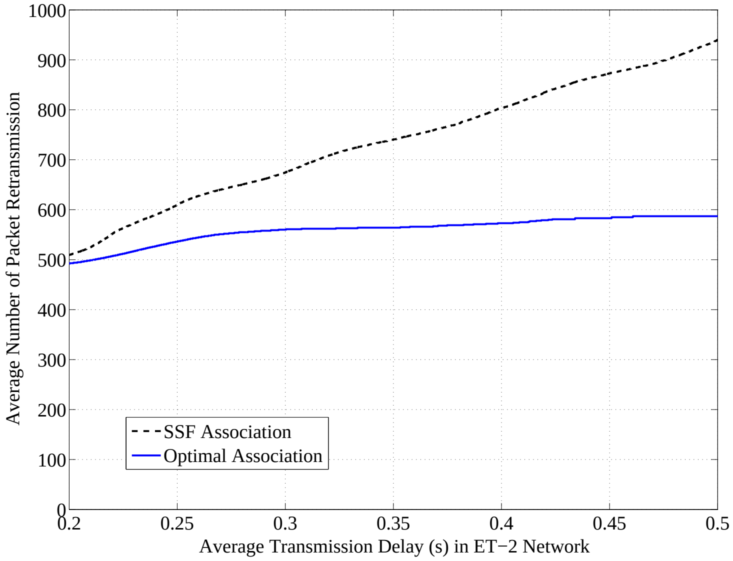


Figure 5.14: Average packet transmission delay (s) versus total number of packet retransmissions in ET-2 network with 200 seconds operational time, 30 APs and 300 contending STAs.

5.5 Downlink Throughput Enhancement

Enhancing throughput in the downlink is also important because it is envisaged that most of the future Wi-Fi traffic would be in the downlink. Therefore, the impact of using the proposed AP association scheme in the downlink performance is presented in this section for the two network topologies. In the ET-1 network, APs randomly transmit buffered packets to their respective associated STAs in the DL. Contention and PCS induced delay also impacts performance in the DL because APs, just like the STAs, also contend for the channel. Figure 5.15 shows the cumulative distribution of the overall DL throughput between each AP and its STAs in ET-1 network; 30 APs transmitting DL traffic to 300 STAs.

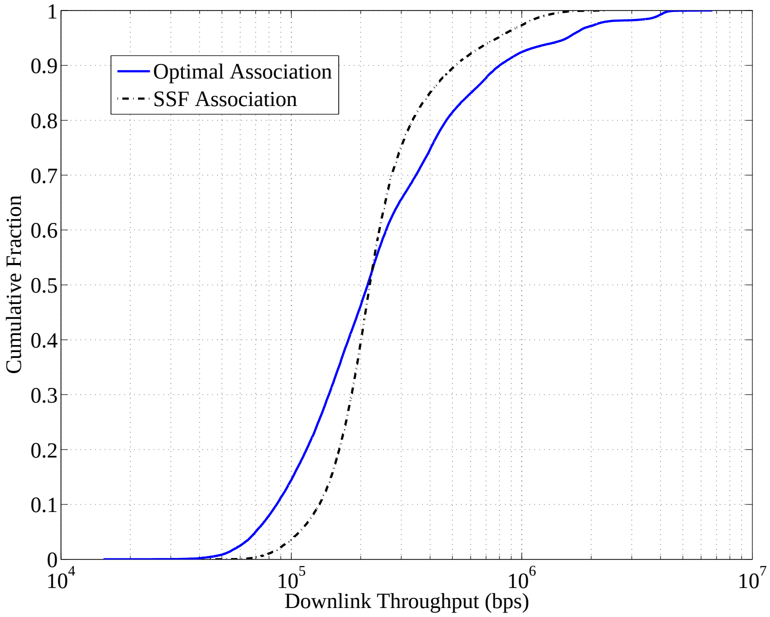


Figure 5.15: Cumulative distribution of downlink throughputs in all BSSs in ET-1 network with 30APs and 300 STAs using SSF association versus Optimal association.

From Figure 5.15, the DL throughput gain doubles at the 75th percentile when the association scheme is changed from SSF association to optimal association. Although no gain is achieved at the 50th percentile in Figure 5.15, 90% of the DL transmissions carried out after optimal association, occur at throughputs 3 times higher than the achievable throughputs from SSF association; this is a significant gain. In the same vein, Figure 5.16 shows the average DL throughput per BSS in ET-1 where it is obvious that the optimal association yields average throughput that is twice the average DL throughput from using SSF association in each BSS.

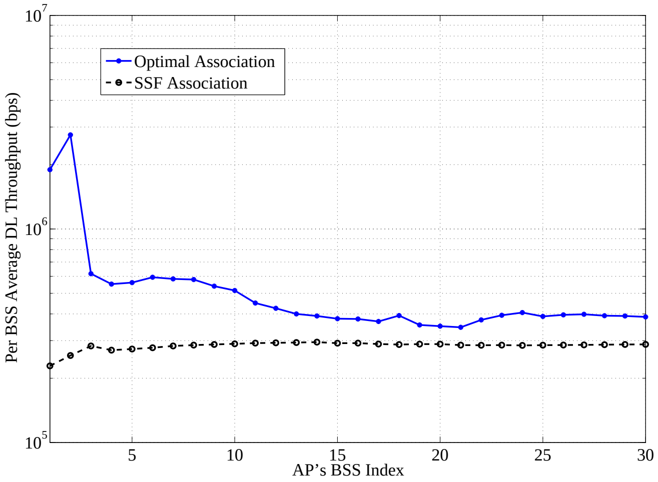


Figure 5.16: Average downlink throughputs per BSS in ET-1 network with 30 BSSs and 300 STAs using SSF association versus Optimal association.

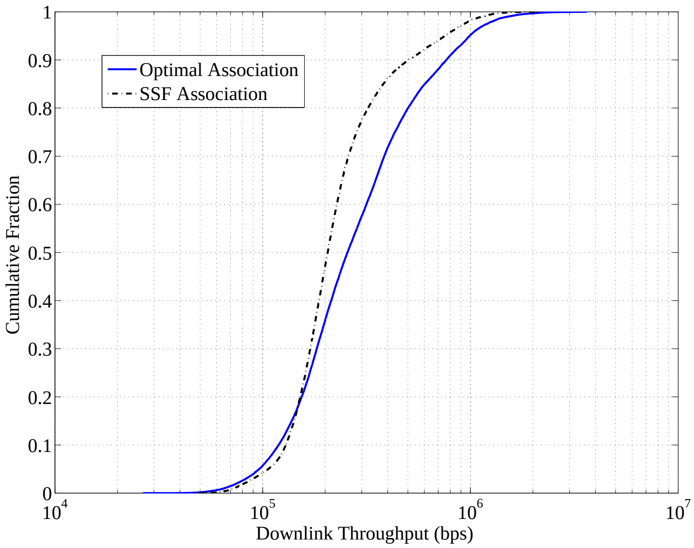


Figure 5.17: Cumulative distribution of downlink throughputs in ET-2 with 30 APs serving 300 STAs using SSF Association versus Optimal Association.

Figures 5.17 and 5.18 show the cumulative distribution of the overall DL throughputs and the average DL throughput per BSS respectively in ET-2 network. Figure 5.17 shows an increasing gain in throughput from the 20th percentile to the 50th percentile and beyond 75th percentile; this observation shows the significant performance of optimal association over the SSF scheme. Observing the 90th percentile in the same figure, the throughput gain in the DL using the proposed scheme is 4 times the throughput from using SSF scheme. Similarly in Figure 5.18, with the optimal association, there is approximately 26.8% gain in the average DL throughput per BSS in ET-2, which is a noticeable significant gain for using the optimal scheme over the default IEEE 802.11 SSF scheme.

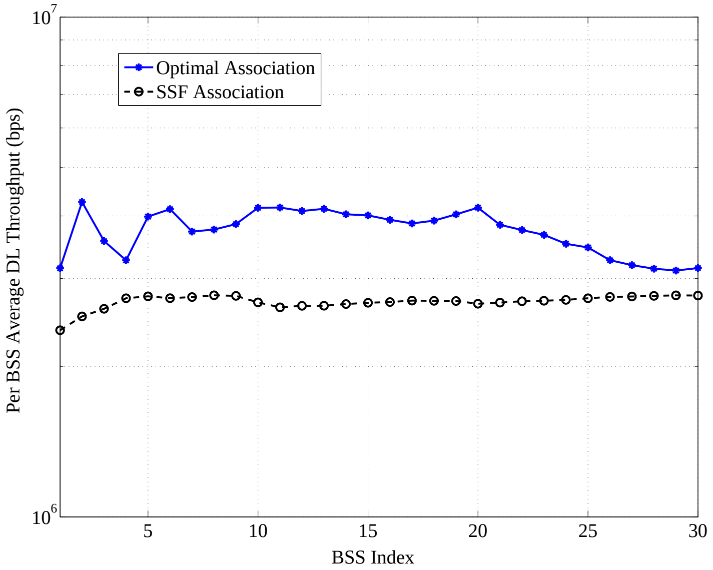


Figure 5.18: Average downlink throughputs per BSS in ET-2 network with 30 BSSs and 300 STAs using SSF association versus Optimal association.

5.6 Performance of CCA Threshold Adjustment

Next, we investigate the performance of CCA adjustment. Figure 5.19 depicts the performance of CCA adjustment based on **Algorithm** 5 from Section 4.2 over a global fixed CCA setting in ET-1. Fig. 5.19 shows a further improvement in throughput when optimal association is combined with CCA calibration. Looking at the 10th percentile of STAs, it is apparent that adjusting the CCA threshold in each BSS enhances STA throughput from 3.8 to 6 Mbps, a 58% improvement.

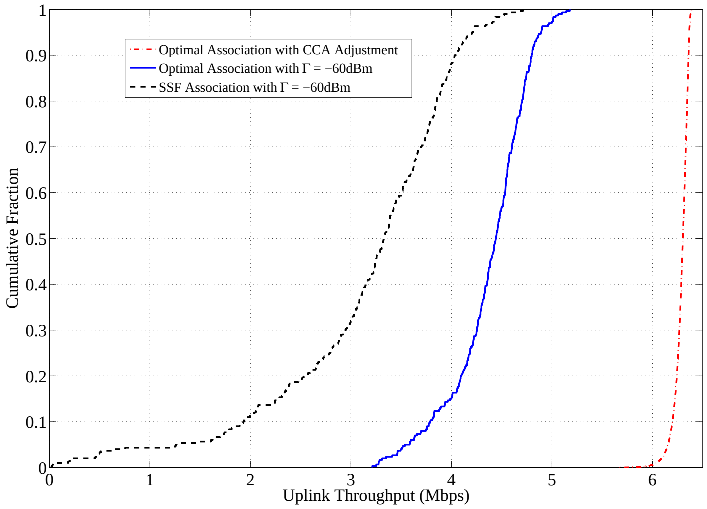


Figure 5.19: Throughput gain from adjusting the CCA threshold per BSS based on cell-edge SINR over fixed CCA threshold, $\Gamma = -60$dBm in ET-1 network with 300 STAs and 30 APs.

In the case of ET-2, when the CCA adjustment is performed in addition to association optimization, the throughput at the 10th percentile of Figure 5.20, improves from 4.4 Mbps to 6.3 Mbps, which is 43% improvement. Hence, the effect of combining AP association optimization with CCA yields approximately 130% improvement in throughput in ET-1 and 66% improvement in ET-2.

Lastly on CCA threshold adjustment, in Figure 5.21, the cell edge throughput

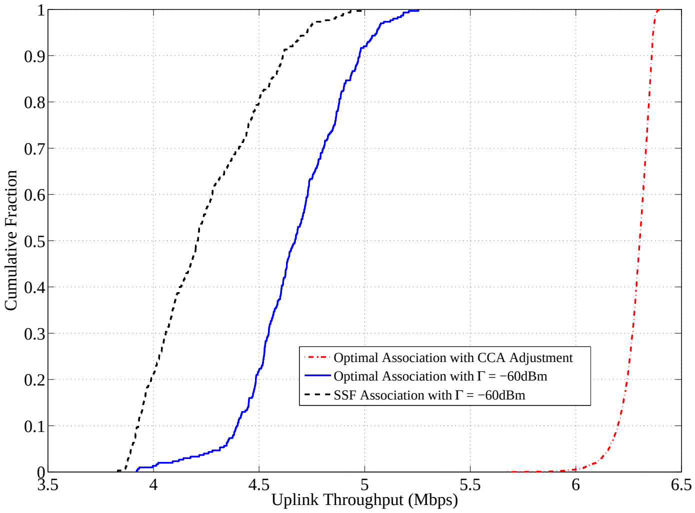


Figure 5.20: Throughput gain from adjusting the CCA threshold per BSS based on cell-edge SINR over fixed CCA threshold, $\Gamma = -60$dBm in ET-2 network with 300 STAs and 30 APs.

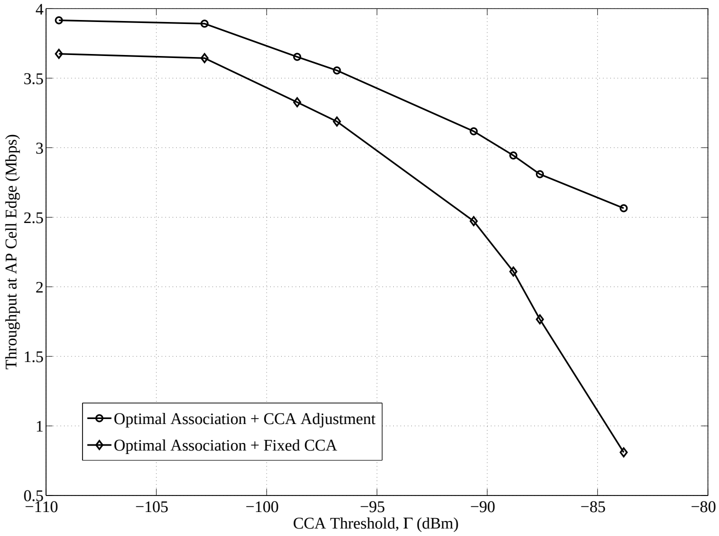


Figure 5.21: Effect of CCA threshold adjustment on cell-edge STAs: performance of per BSS CCA threshold adjustment over global CCA of -60dBm.

increases significantly when CCA threshold is calibrated using **Algorithm 3**. In Figure 5.21, we observed throughputs at the cell-edges of seven selected STAs, each located at the cell-edge of different APs (or BSSs). We first observe their throughputs when the CCA threshold is fixed as −60dBm for all the BSSs. Thereafter, the CCA threshold is adjusted per BSS using the SINR of STA at the cell-edge. The horizontal axis in Figure 5.21 shows the CCA threshold determined for each BSS, and the entire plot compares the cell-edge throughputs in fixed CCA case with per BSS CCA adjustment under optimal association scheme.

5.7 Throughput-based Fairness

Thus far, performance benchmarking has been based on throughput, degree of delay, and the number of packet retransmissions. The next performance metric of interest is the Jain's Fairness Index, which is determined according to Equation 5.1.

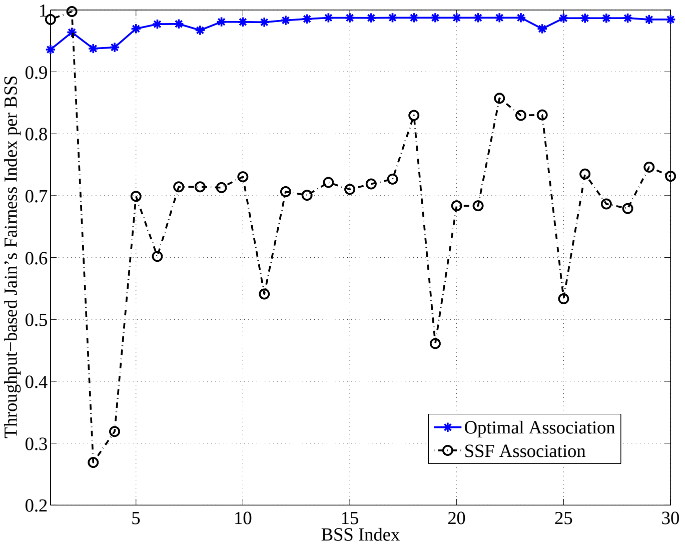


Figure 5.22: Throughput-based Jain's Fairness Index per BSS in ET-1 Network.

In Figures 5.22 and 5.23, the Jain's Fairness Index of our proposed optimal association scheme in **Algorithm 1** is plotted against the Jain's Fairness Index of the

SSF association scheme for ET-1 and ET-2 networks respectively. The Jain's Fairness Index $\mathcal{J}$ returns values between 0 and 1 (i.e. $\mathcal{J} \in [0, 1]$) and a better fairness is achieved as $\mathcal{J}$ increases towards 1. Clearly from Figures 5.22 and 5.23, the optimal association scheme results in fairness among users while enhancing performance at the same time. Although, the primary objective of optimizing AP association is to maximize throughput, it turns out that a by-product of the performance-enhancing optimal association is a solution with high fairness compared with the SSF association scheme; when fairness is explicitly computed.

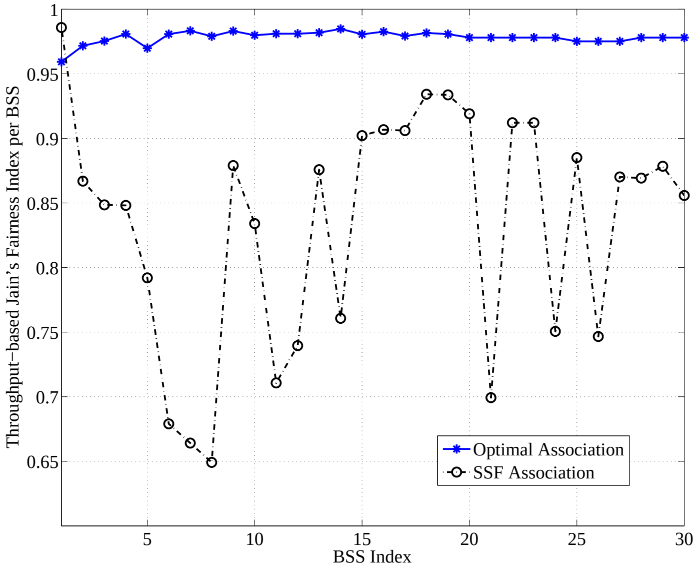


Figure 5.23: Throughput-based Jain's Fairness Index per BSS in ET-2 Network.

In Figure 5.22, the worst case fairness of 0.27 is recorded under SSF association in ET-1 network while optimal association scheme balances throughput maximization with proportional fairness with an average fairness of 0.96 across all BSSs in ET-1. Similar situation occurs in ET-2 network where SSF scheme achieves the lowest fairness of 0.65 while the optimal association guarantees fairness above 0.95.

5.8 Performance Comparison: ET-1 versus ET-2

In this section, the overall performance in ET-1 (random APs deployment) is compared with that of the ET-2 (regular APs deployment). First, optimizing AP associations offers more significant improvement in random AP deployment than the regular grid deployment. We remark that it is not overly optimistic to recommend random AP deployment for dense Wi-Fi networks. First, random AP deployment like the simulated ET-1 network in this thesis, is commonplace in most Wi-Fi environments because users are usually located in random locations within the network; take multi-tenant building and campus networks for example: here users are located in both indoors and outdoors of many buildings.

Secondly, it is evident from this study that optimizing AP associations in ET-1 network results in higher throughput gain than in a network like ET-2 (with regular AP placement). In ET-1 network, 90% of the users experience at least 20.6% increase in throughput as a result of optimizing AP association while in ET-2, 7.8% gain in throughput is observed for 90% of the users. This buttresses the need for AP association optimization scheme (motivating the one proposed in this thesis) in Wi-Fi networks with random AP placements because the impact of our proposed association scheme is more significant in ET-1 than in ET-2 network. Having said this, the proposed AP association scheme in this thesis is useful in both random and regular AP deployments because significant performance improvement is achieved in both networks when AP association is optimized rather than using the SSF scheme.

Although, better aggregate throughput is observed in regular grid AP deployment (ET-2), overall, combining AP association optimization with CCA threshold adjustment in ET-1 yields 130% improvement in users throughput while 66% improvement is observed in ET-2. Based on this observation, we remark that a greater performance gain occurs in the random APs deployment scenario (ET-1) than in regular grid AP deployment. This is likely due to the fact that in a regular topology, APs already cover most STAs locations with sufficient SINRs and STAs are associated with APs offering the best rate.

5.9 Dynamic Network Scenario

The goal of the dynamic **Algorithm 3** and **4** from Section 4.1.2 is to update the optimal association obtained in **Algorithm 1** when the network condition or structure changes due to new user arrivals, existing users' departure, or users mobility within the network. Therefore, in this section, we will examine the performance of repairing the AP associations in DWLANs when new STAs join the network or/and SINRs of associated STAs changed. For the simulations, users are incrementally added to the network and allowed to move randomly within the network based on RWMM. Therefore, rather than relying on SSF association to capture these changes, the dynamic optimal association algorithms in **Algorithm 3** and **Algorithm 4** find new set of optimal associations for the new entrants and existing users.

For this dynamic scenario analysis, the random network topology depicted in ET-1 is used. This choice is motivated by the fact that in practice, APs are normally deployed in a random fashion and not in the regular deployment found in cellular networks. ET-1 best describes the network scenario in most Wi-Fi locations such as apartment buildings, airports, campuses, hot-spots, and even medium to large enterprise buildings. For the analysis here, we focus primarily on throughput and the transmission delay as performance metrics. The impact of user mobility and new users joining the network is depicted in Figures 5.24 and 5.25, respectively. After every time slot (20 seconds) of the network uptime, new users are added incrementally to the network (ET-1). Once these new users join the network, they move randomly within the network at random speed according to RWMM. With these dynamics (new users and mobility), the degree of interference and contention change, and consequently, the SINRs of both new and existing users change.

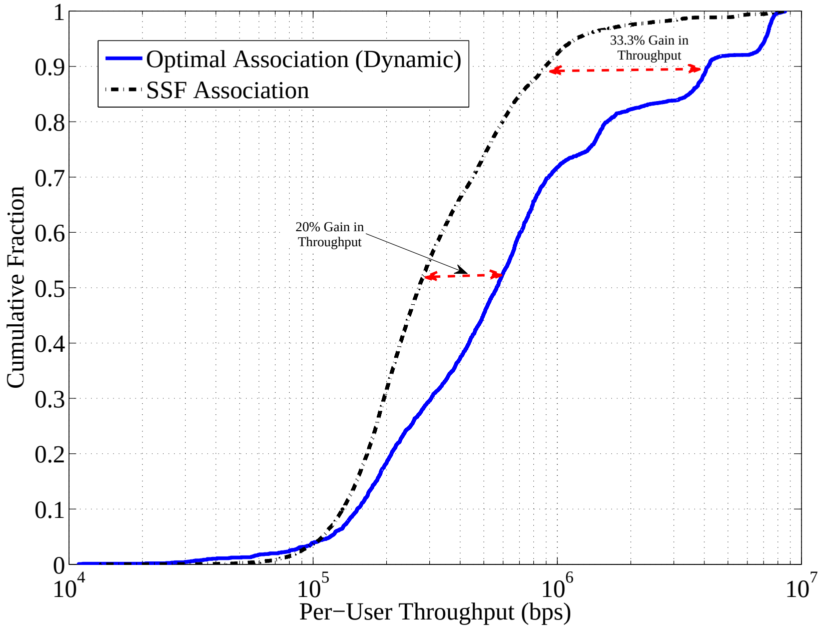


Figure 5.24: Cumulative distribution of users throughputs in ET-1 Network using SSF association against combining the dynamic optimal association algorithms.

Figure 5.24 presents the improvement in user throughputs, where the most noticeable improvements are 20% and 33.3% gains at the 50 percentile and 90 percentile of the distribution respectively. The throughput distribution in Figure 5.24 contains the throughputs of users that are incrementally added to ET-1 network of 30 APs, starting from 50 to 300 STAs. As shown in Figure 5.25, the average network throughput

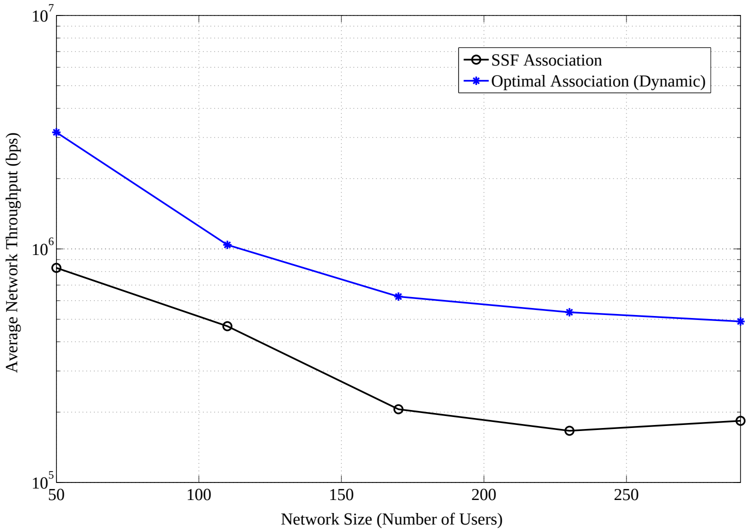


Figure 5.25: Average network throughput per network size in ET-1 Network using SSF association versus the dynamic optimal association with global CCA threshold of -60dBm.

relative to the network size is plotted. The combined dynamic optimal association algorithms offers improvement, which is twice the throughput in SSF association. From Figure 5.25, as the network size grows, the average network throughput decreases; this is due to the increase in contention among users (more users means higher degree of contention). Even though throughput declines with growing number of STAs, it can be enhanced using the proposed optimal scheme as obvious in Figure 5.25, where for example, 57.1% gain is achieved when the number of users increased from 110 to 170.

When the number of users reaches 300 in ET-1 network from incremental arrival of users starting from 50 users, the per user throughput is observed after the 5th time of repairing AP associations using the dynamic algorithms. The user throughput is sorted from minimum to the maximum and plotted against each user's index on the horizontal axis as shown in Figure 5.26. The per user throughput for all the 300

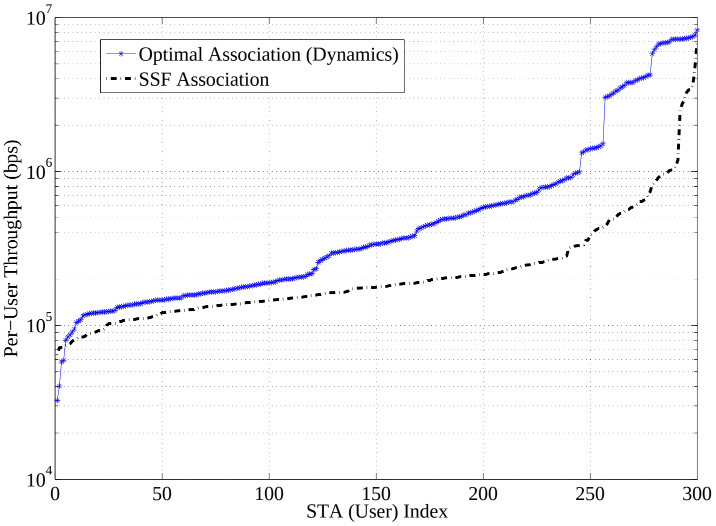


Figure 5.26: Per-user throughput for the maximum network size of 300 users in ET-1 Network of 30 APs using SSF association against dynamic optimal association with global CCA threshold of -60dBm.

users is plotted in Figure 5.26 where the dynamic optimal association outperforms the SSF scheme, by achieving higher throughput for at least 95% if the users under the dynamic situation.

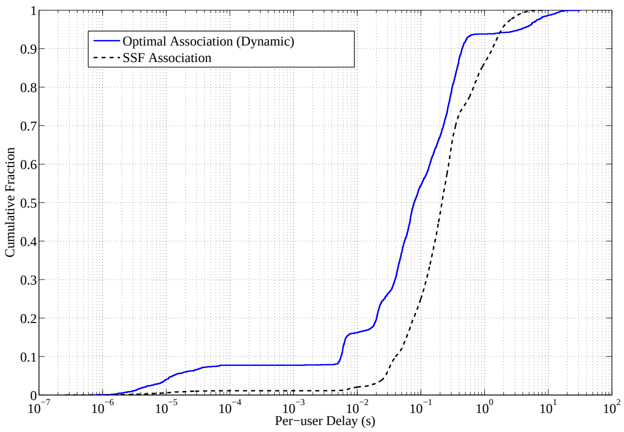


Figure 5.27: Cumulative distribution of transmission delay experienced by users in ET-1 Network using SSF association against the dynamic optimal association algorithms.

Finally, for the dynamic scenario, Figure 5.27 depicts the cumulative distribution of transmission delays experienced by all STAs as the network size grows from 50 users to 300 users. From Figure 5.27, the maximum delay in SSF association is 20s while the maximum recorded transmission delay in the dynamic optimal association is 6s. This signifies that the worst-case delay in the SSF-based network can be reduced by 70% using the proposed dynamic optimal association algorithms in a dynamic network. Another way to interpret Figure 5.27 is that, we can guarantee 90% of the users a transmission delay below 1s by using the proposed dynamic algorithms to dynamically coordinate AP associations.

5.10 Chapter Summary

The results obtained from simulations are analyzed and discussed in this chapter. Two different types of generic networks, named, ET-1 and ET-2 in this thesis are used to benchmark the performance of the proposed AP associations scheme in **Algorithm 1** over the default SSF association in IEEE 802.11 standards for Wi-Fi. While **Algorithm 1** is suitable for *quasi-static* Wi-Fi networks, **Algorithm 3** and **Algorithm 4** are proposed for dynamic network environments. The three (3) optimal association algorithms yield performance improvement in the simulated DWLAN over SSF association in both *quasi-static* and dynamic network scenarios. Also, the performance of calibrating the CCA threshold using **Algorithm 5** based on the STA at cell-edge of an AP has been discussed. The results show that combining AP associations coordination with CCA threshold calibration yields a further increase in throughput. Overall, the proposed AP association scheme provides significant increase in throughput and guarantees higher fairness over the SSF scheme.

Chapter 6

Summary and Conclusions

6.1 Summary

The main advantage of dense AP deployment is that, it significantly increase network coverage. However, such dense network inevitably causes significant increase in interference, which degrades overall network throughput. To mitigate the effect of interference in DWLAN, performance enhancement can be addressed using different approaches to maximize SINRs of nodes. One approach that motivates this study is the optimization or coordination of STAs-APs associations to improve performance. Another method would be interference coordination, which has been explored by the authors in [45] for performance enhancement in dense wireless networks. In Chapter 1, the motivation to explore new AP associations scheme to improve performance in DWLAN is discussed. Similarly, the effect of PCS or CCA process on overall network throughput necessitates the optimization of the CCA threshold that governs PCS in CSMA/CA. Therefore, in this thesis, calibrating efficient CCA threshold per BSS is secondary to the objective of optimizing AP associations.

Prior to formulating the AP association problem, an overview of WLAN technology and standard is discussed in Chapter 2 under the background section to provide an insight into the wireless technology being studied. Chapter 2 also provides an avenue to understand different architecture in WLAN, the CSMA/CA protocol, the PHY layer, and the CCA/PCS process; understanding all these protocols and processes helps in identifying where the problem lies and the approaches to mitigate the effects of the problem. Section 2.2 in Chapter 2 surveys existing WLAN capacity or performance enhancement schemes. The most prominent schemes in literature include AP association optimization/coordination [1, 3, 22–28, 30], TPC [31], MAC protocol/parameter tuning or adaptation [31–33], and a hybrid scheme, for example, the joint tuning of PCS, power, and rate in [34]; these studies demonstrate how AP association can be exploited for performance enhancement in DWLANs.

The WLAN system model and problem formulation are presented in Chapter 3, where the STA-to-AP association problem is formulated as a bipartite graph matching problem. Literally, the AP association problem is formulated as an optimization problem, which is solved using the proposed algorithms in Chapter 4, which are motivated by the Kuhn-Munkres algorithm for assignment problem. In order to achieve this objective of developing new AP association scheme, three algorithms are proposed in Chapter 4. These algorithms are based on the Kuhn-Munkres algorithm highlighted in Algorithm 2 and they are developed to match STAs to APs in DWLANs such that no isolated STAs or idle APs; Algorithm 1 takes care of AP association in *quasi-static* network while Algorithms 4 and 3 are dynamic versions and perform optimal re-association when there is a change in the network due to dynamics such as user mobility and new users arrival.

Chapter 5 presents the performance analysis where four (4) major performance metrics, namely, throughput, transmission delay, number of packets retransmission, and Jain's fairness index are employed for benchmarking the performance of the proposed scheme over the existing SSF scheme specified in IEEE 802.11 standard. The performance analysis is based on two simulated network scenarios, named ET-1 and ET-2 imitating random and regular AP deployment respectively, and each network scenario consists of 300 STAs and 30 APs.

6.2 Conclusion

The possibility of improving the performance in dense WLANs through STA-AP association optimization and physical carrier sensing adjustment has been explored in this study. Thus far, association optimization and CCA threshold adjustment algorithms have been proposed to improve throughput. The proposed association optimization algorithm provides an improved performance in terms of throughput when compared with the existing SSF association in current 802.11 networks. It requires available radio measurements fed to a central controller either via the air interface or Ethernet backbone. The radio measurement is already available in 802.11 nodes based on the 802.11k mechanism. While this transmission of measurements adds to the network overhead, STAs' throughputs improve and the overhead is only incurred for a small fraction of the time.

The AP association algorithm in Section 4.1.1 is referred to as *optimal association* and the *optimality* claim is based on the fact the origin algorithm (Kuhn-Munkres) is optimal for solving assignment problem; this optimality is not proven in this thesis, but interested reader is referred to [38] and [39] for further reading. For the simulation setups, two network scenarios are employed, namely, random APs deployment referred to as ET-1 and a regular APs deployment referred to as ET-2 in this thesis. The network with randomly deployed APs is commonplace and it imitates the Wi-Fi APs placement found in multi-tenant apartment buildings, airport hot-spots, and enterprise network environment. The simulation results from the ET-1 network with 300 STAs and 30 APs show that in a network with randomly deployed APs, a minimum 54% gain in the uplink throughput is achieved when the network switches from SSF association scheme to the optimal association. Hence, the proposed optimal association scheme outperforms the SSF association currently supported in IEEE 802.11 systems. Similarly goal of improving users throughput is achieved in ET-2 where majority of the STAs transmit at better throughput when using the optimal scheme.

To further quantify performance of both association schemes, the *average transmission delay* experienced by each STA is measured. Unsurprisingly, SSF scheme yields more transmission delay than the optimal scheme in both ET-1 and ET-2. Higher transmission delay degrades performance and it is not desirable in Wi-Fi networks especially for delay intolerant applications such voice over IP (VoIP) and Video streaming. Therefore, the optimal scheme suggested in this thesis has the capability to minimize delay better than the SSF scheme. Also, the number of *packet re-transmission* due to delay, contention, and collision grows exponentially when SSF association is employed. However, when the AP association switches from SSF scheme to optimal scheme, transmission delay decreases and consequently, the number of packet re-transmission is minimized across the network.

Additionally, the fairness of the proposed optimal scheme is compared with that of the SSF scheme using what is referred to as the *throughput-based fairness* in this thesis, in order to investigate if users can achieve equal throughput. This fairness is measured per BSS (or AP) using the Jain's Fairness Index equation, which measures fairness on a scale of [0, 1] and a higher value or a value approaching 1 means better fairness. In both ET-1 and ET-2, the optimal scheme offers better throughput-based fairness than the SSF scheme. In fact, the average fairness of the optimal scheme is above 0.9 in both ET-1 and ET-2 while the average fairness of the SSF scheme is 0.7 is ET-1 and 0.8 in ET-2. From both end users and service providers viewpoints, an algorithm that offers improved fairness is desirable for equal network access and increased revenue respectively. Overall, the proposed optimal scheme outperforms the SSF scheme in terms of throughput, fairness, transmission delay, and the number of packet re-transmission; the performance gains are significant.

6.3 Future Work

First, the proposed algorithms in this thesis assume there is a central controller that has global information about the STAs and APs on the network and these algorithms are executed at this central controller. In lieu of the need for a controller, we are exploring in future work, the possibility of developing a distributed version where APs use local neighbor-to-neighbor information for association. Efficient AP association scheme boosts performance and guarantees fairness. Despite that algorithms have been proposed for AP association in this study, there is still room for more studies on efficient AP association schemes that will minimize interference, increase spatial reuse, maximize SINR and throughputs, improve spectral efficiency, ensure proportional fairness. Therefore, there are more parameters for AP association problem formulation or optimization, especially in the emerging DWLANs, to find optimal AP separating distance for densification, optimal CCA threshold for PCS, and optimal set of STA-AP associations to maximize throughputs and other performance metrics.

Secondly, this current study considers a network with symmetric traffic with the assumption that all users and APs transmit identical packet in the UL and DL respectively. In a future study, the performance of these algorithms would be investigated using different types of packetized traffic such as hypertext transfer protocol (HTTP) and VoIP traffics. Using different types of traffics helps to benchmark performance in WLAN environments with asymmetric requests from users because the best effort packets tolerate more delay than multimedia packets (such as VoIP or Video data) and multimedia data requires more bandwidth than the best effort traffic. This will clarify the achievable throughputs under different traffic conditions.

This thesis investigates both the UL and DL throughput, transmission delay in the UL, quasi-static and dynamic WLAN scenario whereby STAs are leaving and joining the network (dynamic users distribution using a mobility model), but does not document performance related to packet loss or packets in error. It is interesting to investigate packet error rate (PER) or bit error rate (BER) in an extended study, in order to have a larger picture of performance. Nevertheless, this study shows that by optimizing STA-AP association and adjusting carrier sensing parameters (CCA threshold), an increase in throughput can be achieved. Overall, the numerical results show that coordinating users' association with APs using the proposed algorithms yields an increase in throughput. Finally, adjusting the CCA threshold to the lowest SINR will further improve STAs' throughputs including those at the cell edges.

Bibliography

[1] L. Wei, W. Shengling, C. Yong, C. Xuizhen, X. Ran, A. A. Mznah, and A. Abdullah. AP association for proportional fairness in multirate WLANS. *IEEE/ACM Trans. on Networking*, 22(1), February 2014.

[2] G. Aditya and K. Sachin. Strider: automatic rate adaptation and collision handling. In *Proc., IEEE SIGCOMM'11*, Toronto, Canada, August 2011.

[3] B. Yigal, H. Seung-Jae, and L. Li(Erran). Fairness and load balancing in wireless LANS using association control. IEEE/ACM Trans. on Networking, 15(3), June 2007.

[4] E. G. Villegas, R. V. Ferre, and J. P. Aspas. Load balancing in WLANS through IEEE 802.11k mechanisms. In *Proc., 11th IEEE Symp. on Computers and Comm.*, Pula-Cagliari, Sardinia, Italy, 2006.

[5] B. Yigal and H. Seung-Jae. Cell breathing techniques for load balancing in wireless LANS. In Proc., IEEE Infocom, Barcelona, Spain, June 2006.

[6] B. Yigal and H. Seung-Jae. Cell breathing techniques for load balancing in wireless LANS. IEEE Trans. on Mobile Computing, 8(6), June 2009.

[7] Z. Jing, G. Xingang, L. L. Yang, W. S. Conner, R. Sumit, and H. M. Mousumi. Adapting physical carrier sensing to maximize spatial reuse in 802.11 mesh networks. ACM Journal, Wireless Communications and Mobile Computing - Special Issue: Emerging WLAN Applications and Technologies archive, 4(8):933 – 946, December 2004.

[8] V. P. Mhatre and K. Papagiannaki. Optimal design of high density 802.11 WLANS. In *ACM CoNEXT*, Lisboa, Portugal, 2006.

[9] Z. Yanfeng, N. Zhisheng, Z. Qian, T.Bo, Z. Zhi, and Z. Jing. A multi-AP architecture for high-density WLANS: protocol design and experimental evaluation. In *Proc., IEEE SECON*, CA, U.S.A, 2008.

[10] A. George, C. W. Pradeep, F. Carlo, and T. Leandros. Optimizing client association for load balancing and fairness in millimeter-wave wireless networks. IEEE/ACM Trans. on Networking, 2014.

[11] Z. Jing, M. Benjamin, G. Xingang, and L. York. Adaptive csma for scalable network capacity in high-density WLAN: a hardware prototyping approach. In *Proc., IEEE Infocom*, Barcelona, Spain, 2006.

[12] IEEE/ANSI. Wireless LAN MAC and PHY layer specifications. Technical Report IEEE Std 802-11-1999, Institute of Electrical and Electronics Engineers, 1999.

[13] B. O'Hara. *IEEE 802.11 handbook: a designer companion.* IEEE Standards Information Network, IEEE Press,, N.Y., USA, 2005.

[14] K. Shin, I. Park, J. Hong, D. Har, and D. Cho. Per-node throughput enhancement in wi-fi densenets. *IEEE Comm. Magazine*, 53(1):118 – 125, January 2015.

[15] V. L. Hai and S. Taka. Collision probability in saturated IEEE 802.11 networks. In *Proc., Australian Telecommunication Networks and Applications Conference*, Australia, 2006.

[16] K. Ponnusamy and A. Krishnan. Saturation throughput analysis of IEEE 802.11b wireless local area networks under high interference considering capture effects. *International Journal of Computer Science and Information Security*, 7(1), 2010.

[17] A. Aditya, J. Glenn, S. Srinivasan, and S. Peter. Self-management in chaotic wireless deployments. In *Proc., MobiCom'05*, Germany, 2005.

[18] S. Banerji and R. S. Chowdhury. On ieee 802.11: wireless LAN technology. *Int'l Journal of Mobile Net. Comm. & Telematics*, 3(4), 2013.

[19] IEEE 802.11 WG. Part 11: Wireless LAN MAC medium access control (MAC) and physical layer (PHY)specifications. Technical report, Institute of Electrical and Electronics Engineers, August 1999.

[20] J A. Fuemmeler, N. H. Vaidya, and V. V. Veeravalli. Selecting transmit powers and carrier sense thresholds in CSMA protocols for wireless Ad Hoc networks. In *Proc., ACM Intl. Wireless Internet Conf.(WICON'06)*, Boston, MA, USA, August 2006.

[21] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda. Performance anomaly of 802.11b. In *Proc., IEEE INFOCOM'03*, San Francisco, CA, March 2003.

[22] O. Ekici and A. Yongacoglu. A novel association algorithm for congestion relief in IEEE 802.11 WLANS. In *Proc., ACM IWCMC'06*, Vancouver, British Columbia, Canada., July 2006.

[23] B. Akash, S. Michael, S. Ivan, R. Jennifer, and R. Dipankar. Network cooperation for client-AP association optimization. In *Proc., 10th International Symp. on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*, Germany, 2012.

[24] O. B. Karimi, J. Liu, and J. Rexford. Optimal collaborative access point association in wireless networks. In *Proc., IEEE INFOCOM'14*, Toronto, Canada, 2014.

[25] M. Abusubaih and A. Wolisz. An optimal station association policy for multi-rate IEEE 802.11 wireless LANs. In *Proc., ACM MSWiM'07*, Crete Island, Greece, October 2007.

[26] A. Kumar and V. Kumar. Optimal association of stations and aps in an IEEE 802.11 WLAN. In *Proc., National Communications Conference (NCC)*, Bangalore, India, January 2005.

[27] I. Koukoutsidis and V. A. Siris. Access point assignment algorithms in WLANS based on throughput objectives. In *Proc., IEEE Symp. on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks and Workshops (WiOPT 2008)*, Berlin, Germany, April 2008.

[28] H. Gong, K. Nahm, and J. W. Kim. Access point selection tradeoff for IEEE 802.11 wireless mesh network. In *Proc., IEEE Consumer Communications and Networking Conference (CCNC 2007)*, Las Vegas, NV, USA, January 2007.

[29] L. Yen, J. Li, and C. Lin. Stability and fairness of AP selection games in IEEE 802.11 access networks. *IEEE Trans. on Vehicular Tech.*, 60(3):1150 – 1160, March 2011.

[30] L. Li, M. Pal, and Y. R. Yang. Proportional fairness in multi-rate wireless LANs. In *Proc., IEEE INFOCOM'08*, Phoenix, USA, March 2008.

[31] J. Zhu, X. Guo, S. Roy, and K. Papagiannaki. CSMA self-adaptation based on interference differentiation. In *Proc., IEEE GLOBECOM'07*, Washington, DC, USA, November 2007.

[32] I. Jamil, L. Cariou, and J. Hélard. Improving the capacity of future IEEE 802.11 high efficiency WLANs. In *Proc., IEEE Intl. Conf. on Telecom.*, Lisbon, Portugal, May 2014.

[33] J. Zhu, X. Guo, L. L. Yang, W. S. Conner, S. Roy, and M. M. Hazra. Adapting physical carrier sensing to maximize spatial reuse in 802.11 mesh networks. *Wireless Communications and Mobile Computing - Special Issue: Emerging WLAN Apllications and Technologies archive*, 4(8):933 – 946, December 2004.

[34] Z. Zhou, Y. Zhu, Z. Niu, and J. Zhu. Joint tuning of physical carrier sensing, power and rate in High-Density WLAN. In *Proc., IEEE Asia-Pacific Conf. on Communications*, Bangkok, October 2007.

[35] N. J. A. Harvey, R. E. Ladner, L. Lovasz, and T. Tamir. Semi-matchings for bipartite graphs and load balancing. *Journal of Algorithms*, 59(1):53 – 78, April 2006.

[36] R. E. Burkard and E. Cela. *Handbook of combinatorial optimization, supplement volume A*, chapter Linear assignment problems and extensions. Kluwer Academic Publishers, 1999.

[37] J. C. Setubal. Sequential and parallel experimental results with bipartite matching algorithms. Technical Report IC-96-09, Institute of Computing, State University of Campinas, Brazil, 1996.

[38] Lawler E. L. *Combinatorial Optimization: Networks and Matroids.* Holt, Reinehart and Winston, New York, N.Y., USA, 1976.

[39] Papadimitriou C. H. and Steiglitz K. *Combinatorial Optimization: Algorithms and Complexity.* Dover Publications, USA, 1998.

[40] G. A. Mills-Tettey, A. Stentz, and M. B. Dias. The dynamic hungarian algorithm for the assignment problem with changing costs. Technical report, Robotics Institute, School of Computer Science, Carnegie Mellon University, July 2007.

[41] C. Bettstetter, H. Hartenstein, and X. Perez-Costa. Stochastic properties of the random waypoint mobility model. *Wireless Networks*, 10(5):555–657, 2004.

[42] I. H. Toroslu and G. Üçoluk. Incremental assignment problem. *Information Sciences*, 117(6):15231529, 2007.

[43] ITU-R Recommendation. Propagation data and prediction methods for the planning of indoor radiocommunication systems and radio local area networks in the

frequency range 900 mhz to 100 ghz. Technical Report ITU-R P.1238-4, International Telecommunication Union, 2012.

[44] L. Wei, C. Yong, C. Xuizhen, A. A. Mznah, and A. Abdullah. Achieving proportional fairness via AP power control in multi-rate WLANS. *IEEE Trans. on Wireless Comm.*, 10(11), February 2011.

[45] B. Soret, K. I. Pedersen, N. T. K Jørgensen, and V. Fernádez-López. Interference coordination for dense wireless networks. *IEEE Comm. Magazine*, 53(1):102 – 109, January 2015.

[46] Z. Hongqiang and F. Yuguang. Physical carrier sensing and spatial reuse in multirate and multihop wireless Ad hoc networks. In *Proc., IEEE Infocom*, Barcelona, Spain, 2006.

[47] S. Boyd and L. Vandenberghe. *Convex optimization.* Cambridge Univ. Press, New York, N.Y., USA, 2004.

[48] 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. on Mobile Computing*, 2(2):164 – 177, February 2007.

[49] A. Duda. Understanding the performance of 802.11 networks. In *Proc., IEEE 19th International Symp. on Personal, Indoor and Mobile Radio Communications*, Cannes, France, 2008.

[50] J. Deng, P. K. Varshney, and Z. J. Haas. A new backoff algorithm for the IEEE 802.11 distributed coordination function. In *Proc., IEEE Sixth International Conference on Fuzzy Systems and Knowledge Discovery*, Tianjin, China, 2009.

[51] M. N. Krishnan and A. Zakhor. Throughput improvement in 802.11 WLANS using collision probability estimates in link adaptation. In *Proc., IEEE Wireless Communications and Networking Conference (WCNC)*, Sydney, Australia, 2010.

[52] M. N. Krishnan, S. Pollin, and A. Zakhor. Local estimation of probabilities of direct and staggered collisions in 802.11 WLANS. In *Proc., IEEE Global Telecommunications Conference*, Honolulu, HI, 2009.

Appendix A

MATLAB Code: Network Layout and PHY Model

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%                 Author: Phillip B. Oni                              %%%
%%%              Dept. of Elec. & Comp. Engr, Queen's University        %%%
%%%          Simulated Network Nodes and PHY model (Summer/FAll 2014)   %%%
%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all
clc
%=========================Parameters=======================================
%Path Loss exponent
alpha = 3.4; %ITU rush hour
% %Noise
Noise = -90; %dB
noiL = 10^(Noise/10); %Noise Linear
%Number of APs and STAs
nAP = 30; nSTA = 300;
%Shadowing Variance
```

```
shdVar = 0.1;
%Experimental Area
AreaY = 200;
AreaX = 200;
%Lambda
freq = 2.4e9;  % IEEE 802.11b
vel = 3e8;
lambda = vel/freq;
%Gains
Gt = 1; Gr = 1;
%Reference Distance
d0 = 1;
%STAs Transmit Power
pwTX_STA = 15.85; % 15.85mW, 12dBm
%txPwr = 10;
txPwr = 10*log10(pwTX_STA); % dBm
%AP Transmit Power
pwTX_AP = 100; % 15.85mW, 12dBm
%txPwr = 10;
txPwr_AP = 10*log10(pwTX_AP); % dBm
%SNR Thresholds
snr_o = 24.6; %dB for Maximum receiver sensitivity
snr_oL = 10^(snr_o/10);
%================================Network Layout=============================
%-------------------------AP Locations
sepD = 20; %AP-to-Ap separation distance
%----------------Deploy APs on Lower part
APLocXL = rand(1,nAP);
APLocYL = rand(1,nAP);
```

```
mm = 1;
for aa = 1:nAP
        if aa ~= 1
            mm = aa - 1;
        end
        APLocXL(aa) = APLocXL(mm) + sepD;
end
%--------------Deploy APs on Upper part
APLocXU = rand(1,nAP)*0;
APLocYU = (rand(1,nAP)*0) + 80;
mn = 1;
for ab = 1:nAP
        if ab > 1
            mn = ab - 1;
        end
        APLocXU(ab) = APLocXU(mn) + sepD;
 end
%-------------Deploy APs on Middle part
APLocXM = rand(1,nAP)*0;
APLocYM = (rand(1,nAP)*0) + 40;
n = 1;
for am = 1:nAP
        if am > 1
            n = am - 1;
        end
        APLocXM(am) = APLocXM(n) + sepD;
end
APX = [APLocXL, APLocXU, APLocXM];
APY = [APLocYL, APLocYU, APLocYM];
```

```
%Total Number of AP
tAP = length(APX);
%-------------------------STAs Locations
%Distance of STAs
STAdis = zeros(1, length(nSTA));
STALocX = rand(1,nSTA)*AreaX;
STALocY = rand(1,nSTA)*AreaY;
stax = STALocX;
stay = STALocY;
%------------------------AP-STAs or Tx-Rx---------Distances
disSTAAP = zeros(nSTA, tAP);
for ie = 1:nSTA
    for ee = 1:tAP
        disSTAAP(ie, ee) = sqrt((APX(ee) - stax(ie))^2 + (APY(ee) - stay(ie))^2);
    end
end
%-----------------Random AP placement-------------------
tAP = nAP;
APdis = zeros(1, length(nAP));
APLocX = rand(1,nAP)*AreaX;
APLocY = rand(1,nAP)*AreaY;
apx = APLocX;
apy = APLocY;
%---------------------Network Layout Figures-----------------
apx = [APLocXL, APLocXU, APLocXM];
apy = [APLocYL, APLocYU, APLocYM];

plot(APLocXL, APLocYL, 'r^');
plot(APLocXU, APLocYU, 'r^');
plot(APLocXM, APLocYM, 'r^');
%----------------------------SSF association distance-----------
DinM = zeros(1, length(nSTA));
pwrRXDinMRan = zeros(1, length(nSTA));
for ip = 1:nSTA
DinM(ip) = min(disSTAAP(ip,:)); %Strongest signal distance
%Receiver Power at SSF distance
pwrRXDinMRan(ip) = txPwr - netPathLoss(alpha,lambda,Gt,Gr,DinM(ip));
end
%------------------AP-STA Association based on Received Signal strength-----
APSTA_Assoc = zeros (nSTA, tAP);
for tm = 1:nSTA
   for ma = 1:tAP
       if disSTAAP(tm, ma) == DinM(tm)
           % 1 indicates association between STA i and AP j
           APSTA_Assoc(tm, ma) = 1;
       else
           %No Association
          APSTA_Assoc(tm, ma) = 0;
       end
   end
end
%----------------------Assign 1 or 0 to Association variable----------------
SSF_xval = zeros(nSTA, nAP);%Association variable x
for to = 1:nSTA
    for ot = 1:nAP
        if disSTAAP(to, ot) == STA_minDis(to)
            SSF_xval(to, ot) = 1; % There is an association
        else
```

```
            SSF_xval(to, ot) = 0; %No Association
        end
    end
end
%--------------------Carrier Sensing and Interference Ranges----------------
CS_dis = zeros(1, length(nSTA));
Inf_dis = zeros(1, length(nSTA));
Noise = zeros(1,length(nSTA));
for cs = 1:nSTA
        %The STA can detect active transmission within this range
        CS_dis (cs) = DinM(cs) * (1 + (snr_oL ^(1/alpha)));
        %Other transmission within this range interfer desired signal
        Noise(cs) = pwrRXDinMRan(cs)+30/ snr_oL;
        %Interference range
        Inf_dis(cs) = CS_dis(cs) - DinM(cs);
end
%---------------------Number of STAs per AP----------------------------------
%---------------------Number of STAs associated to an AP j------------------
STAwithAP = zeros(1,nAP);
SSF_RSS_Assoc  = zeros(nSTA, nAP);
for qk = 1:nSTA
    for qki = 1:nAP
            %Check if association Exists
            if((SSF_xval(qk,qki)==1))
                STAwithAP (qki) = STAwithAP (qki) + 1;
                SSF_RSS_Assoc(qk, qki) = STAAPsrvcdPwr(qk, qki);
            else
                 STAwithAP(qki)  = STAwithAP (qki) + 0;
            end
    end
end

%==================================SINR Estimation=========================
%Interference
%Minimum Interference Distance at 56 Mbps
STA_Inf_dis = zeros(1,nSTA);
snr_oL = 10^(snr_o/10);
max_Inf = zeros(1,nSTA);
for lk = 1:nSTA
    %Compute Interference range for each STA
    STA_Inf_dis(lk) = STA_minDis(lk).*(((1./(1/snr_oL - ...
        ((STA_minDis(lk)).^(alpha)).*(No_L./SSFrvcdPwr(lk))))).^(1/alpha));
    %Compute Maximum interference for each STA based on interference range
    max_Inf(lk) = ULpwTx - netPathLoss(alpha, lambda, Gt, ...
    Gr, STA_Inf_dis(lk));
end

%Uplink SINRs
UL_SINR = zeros(1,nSTA);
RxPower_STA = zeros(1, nSTA);
for im = 1:nSTA
    RxPower_STA(im) = 10.^(SSFrvcdPwr(im)./10);
    %SINR under maximum system sensitivity
    UL_SINR(im) = 10*log10(RxPower_STA(im)./No_L + 10.^(max_Inf(im)./10));
end
%%
%==============================Per STA PHY Rate Estimation=================
STA_PhyRate_SFF = zeros(1,nSTA);
```

```
for iu = 1:nSTA
    %Phy Rate of each STA
    STA_PhyRate_SFF(iu) = log2(1+10^(UL_SINR(iu)./10));
end
%Phy Rate of each STA connected AP
STAPhyRate = zeros(nAP, nSTA);
SSFxVal = SSF_xval';
for op = 1:nAP
    for ip = 1:nSTA
        %Store Phy Rate for each STA
        if((SSFxVal(op, ip)==1))
            STAPhyRate(op, ip) = STA_PhyRate_SFF(ip);
        else
            STAPhyRate(op, ip) = 0;
        end
    end
end

%==========================================================================
%                           Optimal Association Begins
%==========================================================================
Optimal_Assoc = zeros(nAP,nSTA);
%RSS of STA-to-AP
Opt_RcvdPwr = zeros(nAP,nSTA);
%Initial RSS values from SSF Associations
%This matrix it edge weight matrix
Init_SSFRSS = STAAPsrvcdPwr;
%Begin Association Optimization using Hungarian Algorithm function
%For maximization problem, we add -1 to the weight matrix
```

```
%Brute Force
MC1 = -1*Init_SSFRSS(1:30,1:30);
    MA1 = OptimalMatching1(MC1);
MC2 = -1*Init_SSFRSS(31:60,1:30);
    MA2 = OptimalMatching1(MC2);
MC3 = -1*Init_SSFRSS(61:90,1:30);
    MA3 = OptimalMatching1(MC3);
MC4= -1*Init_SSFRSS(91:120,1:30);
    MA4 = OptimalMatching1(MC4);
MC5 = -1*Init_SSFRSS(121:150,1:30);
    MA5 = OptimalMatching1(MC5);
MC6 = -1*Init_SSFRSS(151:180,1:30);
    MA6 = OptimalMatching1(MC6);
MC7 = -1*Init_SSFRSS(181:210,1:30);
    MA7 = OptimalMatching1(MC7);
MC8 = -1*Init_SSFRSS(211:240,1:30);
    MA8 = OptimalMatching1(MC8);
MC9 = -1*Init_SSFRSS(241:270,1:30);
    MA9 = OptimalMatching1(MC9);
MC10 = -1*Init_SSFRSS(271:300,1:30);
    MA10 = OptimalMatching1(MC10);
    %x values for associations
Optimal_Match = [MA1; MA2; MA3; MA4; MA5; MA6; MA7; MA8; MA9; MA10];
%Optimal Match Received Power
Optimal_RSS = zeros(nSTA, nAP);
%Vector RSS Power
Opt_RSS = zeros(1,nSTA);
Opt_dis = zeros(1,nSTA);
for uo = 1:nSTA
```

```
    for ou = 1:nAP
        if(Optimal_Match(uo, ou)==1)
            Optimal_RSS(uo, ou)=Init_SSFRSS(uo,ou);
            Opt_RSS(uo) = Init_SSFRSS(uo,ou);
            %Get Distance of association
            Opt_dis(uo) = disSTAAP(uo, ou);
        else
            Optimal_RSS(uo, ou)=0;
        end
    end
end
%%
%===========================SINR for Optimal===============================
%Interference
STA_Inf_dis_OP = zeros(1, nSTA);
max_Inf_OP = zeros(1, nSTA);
for kk = 1:nSTA
    %Compute Interference range for each STA
    STA_Inf_dis_OP(kk) = Opt_dis(kk).*(((1./(1/snr_oL - ...
        ((Opt_dis(kk)).^(alpha)).*(No_L./Opt_RSS(kk))))).^(1/alpha));
    %Compute Maximum interference for each STA based on interference range
    max_Inf_OP(kk) = ULpwTx - netPathLoss(alpha, lambda, Gt, ...
    Gr, STA_Inf_dis_OP(kk));
end

%SINR
RxPower_STA_Op = zeros(1, nSTA); %Received power linear
UL_SINR_Op = zeros(1, nSTA);
for mm = 1:nSTA
```

```
    RxPower_STA_Op(mm) = 10.^(Opt_RSS(mm)./10);
    %SINR under maximum system sensitivity
    UL_SINR_Op(mm) = 10*log10(RxPower_STA_Op(mm)./No_L + 10.^(max_Inf_OP(mm)./10));
end

%%
%========================Per STA PHY Rate Estimation=======================
STA_PhyRate_Op = zeros(1,nSTA);
for iiu = 1:nSTA
    %Phy Rate of each STA
    STA_PhyRate_Op(iiu) = log2(1+10^(UL_SINR_Op(iiu)./10));
end
%Phy Rate of each STA connected AP
STAPhyRateOp = zeros(nAP, nSTA);
AssxValOp = Optimal_Match';
for opp = 1:nAP
    for ipp = 1:nSTA
        %Store Phy Rate for each STA
        if((AssxValOp(opp, ipp)==1))
            STAPhyRateOp(opp, ipp) = STA_PhyRate_Op(ipp);
        else
            STAPhyRateOp(opp, ipp) = 0;
        end
    end
end


%%================================Path Loss Function=======================
function  PLatd = netPathLoss(alpha,lambda, Gt, Gr, d)
```

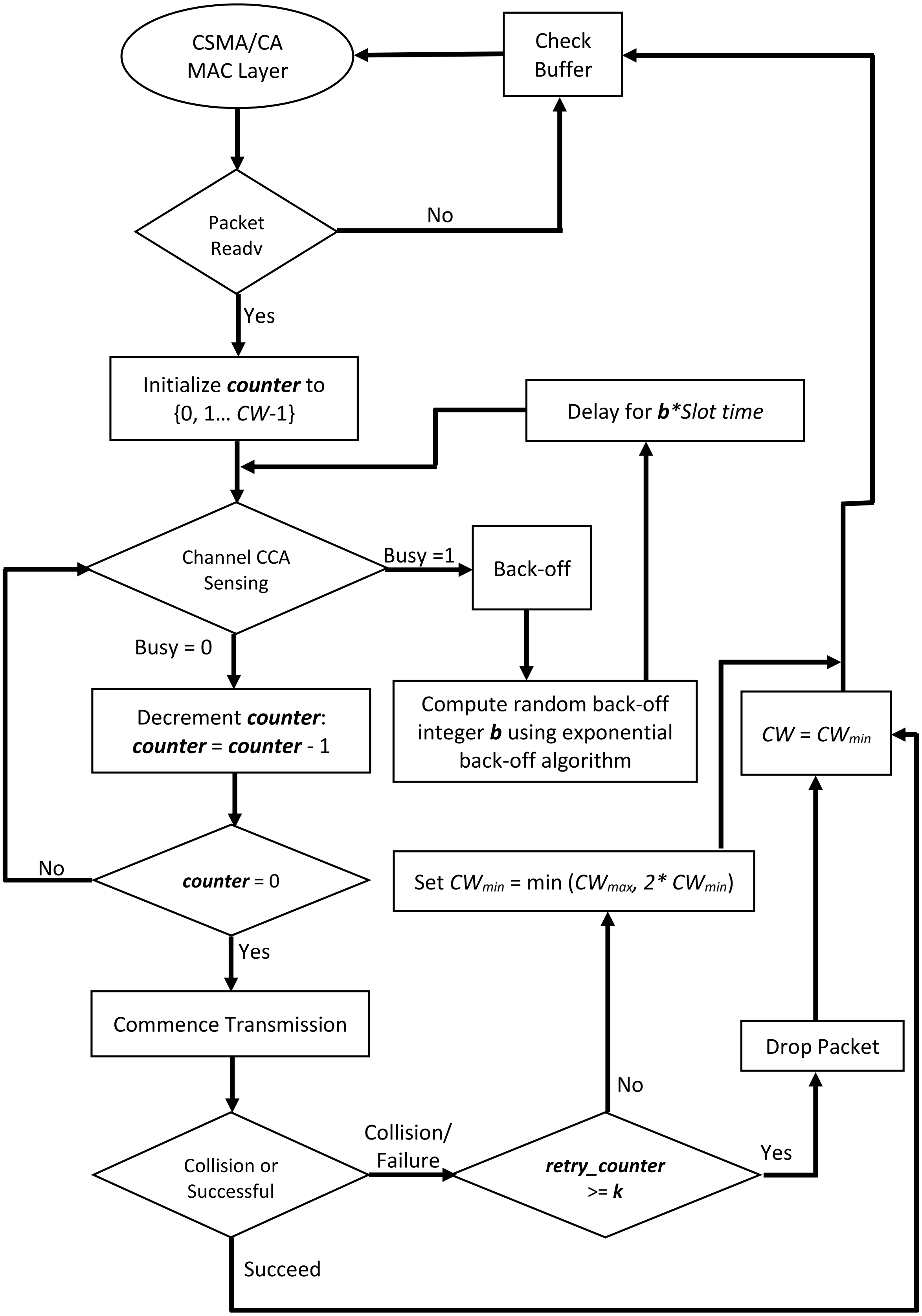
```
    do = 1;
    %path loss at reference d = 1m
    pLref = - 10*log10((Gt*Gr*(lambda^2))/(((4*pi)^2)*(do)^2));
    %Path Loss at a Given distance (dB)
    PLatd = pLref + (10*alpha*log10(d));
return;


%========Carrier Sensing Function==========================================
function busy = CarrierSensing(ccthres, totalintPwr, stapwr, Noise)
    %totalintPwr is the total interference power
        %ccthres is the predefined CCA threshold, totalintPwr is the total
        %interference, and stapwr is the power of STA sensing the channel
    if (totalintPwr + stapwr + Noise) > ccthres
        busy = 1;
    else
        busy = 0;
    end
return;
```

Appendix B

Simple Flowchart for CSMA/CA Algorithm

The CSMA/CA protocol is discussed in, but for implementation purpose, the flowchart used in developing the MATLAB code in Appendix C is presented below. Most of the parameters and terminologies are discussed in Sections 2.1.2 and 2.1.3.


CSMA/CA
MAC Layer
Check
Buffer
Packet
Readv
No
Yes
Initialize counter to
{0, 1… CW-1}
Delay for b*Slot time
Channel CCA
Sensing
Busy =1
Back-off
Busy = 0
Decrement counter:
counter = counter - 1
Compute random back-off
integer b using exponential
back-off algorithm
CW = CW_min
No
counter = 0
Set CW_min = min (CW_max, 2* CW_min)
Yes
Commence Transmission
Drop Packet
No
Collision/
Failure
Yes
Collision or
Successful
retry_counter
>= k
Succeed

Appendix C

MATLAB Code: CSMA/CA

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%                 Author: Phillip B. Oni                              %%%
%%%              Dept. of Elec. & Comp. Engr, Queen's University        %%%
%%%          CSMA/CA (Summer/FAll 2014)   %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %SSF_Tx_Delay = zeros(nAP, NSTA_num, Total_SIM_Tim);
Total_SIM_Time = 200; %Maximum Simulation Time
%Variable to Store Transmission Delay over time
SSFTxD_nSTA = zeros(nSTA, Total_SIM_Time);
%Variable to Store Number of Packet Retransmission over time
SSFReTx_nSTA = zeros(nSTA, Total_SIM_Time);
%Variable to Store Transmission Delay over time per AP
SSF_Tx_Delay = zeros( nSTA, Total_SIM_Time, nAP);
%Variable to Store Number of Packet Retransmission over time
SSF_ReTx_Rate = zeros( nSTA, Total_SIM_Time, nAP);
%Optimal Case
SSF_Tx_Delay_OP = zeros( nSTA, Total_SIM_Time, nAP);
```

```
SSF_ReTx_Rate_OP = zeros( nSTA, Total_SIM_Time, nAP);

STA_Thru_allAP_OP = zeros( nSTA, Total_SIM_Time, nAP);

    %Flag for Idle or Busy Channel

    Ch_Status_Flag = false;

    %Flag for active Transmission

    Txm_Comm = false;

    %Flag for Collision

    Tx_STA_Collision = false;

    %Total Deferring Time

    STA_Tx_Defer_Time = 0;

    %Uplink Throughout

    UL_STA_Thruput = 0;

    %Downlink Throughput

    DL_Thruput = 0;

    %All STAs Throughputs

    STA_ULThruput = zeros(nSTA, nAP);

for iik = 1:nAP

      N_STA = STAwithAP(iik);

    for mki = 1:nSTA

       if (APtoSTA(iik, mki) == 1)

           PHY_Rate = (STAPhyRate(iik, mki)).*10^(6);

           Status1 = ['========================STA', ...

               num2str(mki), ' is Associated with AP', num2str(iik),...

             ' ===============Transmitting at ',num2str(PHY_Rate),' bps'];

           disp(Status1)

%CCA Threshold

CCAth = -60; %dB

CCAL = 10^(CCAth/10); %linear

%Bit Error rate to determine when packet is in error
```

```
BER = 10^(-6);
%Sharing Transmission Time among users
Tx_Shared_Time = zeros(1, N_STA+1);
%Counter for Number of Transmissions
Tx_Cnter = 0;
%Buffer Size
Buff_Size = 6000;
%Packets in Buffer due to Delay
STA_Delay_Buff = zeros(1, Buff_Size);
%Keep track of delay encountered
Delay_at_Tx_AP = 0;
%Packet Retransmission rate
Re_Tx_Pkt_Rate = 0;
Pkt_STA_Rxsmit = zeros(1, N_STA+1);
%Throughput over Simulatiion time
STA_ULThruput_Time = zeros(1, Total_SIM_Time);
%Packet re-Transmission over Simulation Time
Packet_ResendTotalRate = zeros(1, Total_SIM_Time);
%Transmission Delay
Tx_Delay = zeros(1, Total_SIM_Time);
                %======================Packet Arrival======================
                Pkt_Arrive_Time_atQue = 1:N_STA;
                %Packet Length
                Pkt_Size_Length = zeros(1,N_STA+1);
                %STAs having Packet ready
                STA_having_Pkt = zeros(1, N_STA+1);
                %Contention Window for STAs
                Min_CWin = 1:N_STA+1;
                %Backoff Timer for all STAs
                STA_Backoff_Time = 1:N_STA+1;
                %==================Packet Buffer Size for each STA=========
                Tx_STA_Buffer = zeros(N_STA+1, Buff_Size);
                %STAs in Collision
                STAs_Colission = zeros(1, N_STA+2);
%============================CSMA/CA Parameters============
%Slot time in 802.11b
Slot_Time = 20*10^(-6);
%Dividing Simulation time into slot time
Actual_total_Time = Total_SIM_Time/Slot_Time;
%Short interframe space
SIFS = 0.5*Slot_Time; %SIFS is half of Slot time
%Distributed interframe space
DIFS = 2.5*Slot_Time;
%Arrival Time
Pkt_Avrg_Arrival_Time = 10;
%Packet Size
Pkt_Avrg_STA_PktLen = 1460*8; %byte to bits;
%Buffer Size
STA_Buff_Size = zeros(1,N_STA+1);
%AP Buffer Threshold
%Random Arrival of STAs
for ii = 1:N_STA
    Pkt_Arrive_Time_atQue(ii) =  exponentialDist(Pkt_AvrgArrival_Time);
    Pkt_Size_Length(ii) = exponentialDist(Pkt_Avrg_STA_PktLen);
end
%==========================Contention Window===============
for ii = 1:N_STA+1
    %Contention Window
    Min_CWin(ii) = 32;
    STA_Backoff_Time(ii) = 1600;
end
%================================Transmission==============
for tim = 1:Actual_total_Time
for jj = 1:N_STA
    %Period for ACK transmission
    ACK_Tx = 14*8/(PHY_Rate*Slot_Time);
    %Set the buffer threshold
        Buff_Thresh = 8*10^6/(PHY_Rate*Slot_Time);
    if tim == Pkt_Arrive_Time_atQue(jj)
        %Checking Buffer is less than Buffer threshold
        if STA_Buff_Size(jj) < Buff_Thresh - Pkt_Size_Length(jj)
       Tx_STA_Buffer = PushStore(Tx_STA_Buffer, jj, Pkt_Size_Length(jj));
            STA_Buff_Size(jj) = STA_Buff_Size(jj) + Pkt_Size_Length(jj);
            STA_having_Pkt(jj) = 1;
            if STA_Backoff_Time(jj) == 1600;
                %Randomly Select STAs for Backoff
                STA_Backoff_Time(jj) = RandomSTA(Min_CWin(jj)) ;
            end
        end
        %Generate Packet Arrival Rate based on Exponential Distribution
    Pkt_Arrive_Time_atQue(jj) = exponentialDist(Pkt_AvrgArrival_Time)...
        +Pkt_Size_Length(jj)+tim;
    Pkt_Size_Length(jj) = exponentialDist(Pkt_Avrg_STA_PktLen);
    end
end

for jj = 1:N_STA+1
```

```
    %Check kif jjth STA has packet and check if channel is free or busy
    if STA_having_Pkt(jj)==1 && Ch_Status_Flag == false
       if STA_Backoff_Time(jj) == 0
           STAs_Colission = Collision_Among_STAs(STAs_Colission, jj);
           if Pkt_STA_Rxsmit(jj) == 0
               Tx_Shared_Time(jj)= tim+DIFS;
           end
           Txm_Comm = true;
       else
           STA_Backoff_Time(jj) = STA_Backoff_Time(jj) - 1;
       end
    end
end

if Txm_Comm == true
    Ch_Status_Flag = true;
    n = STAs_Colission(1);
    if n == 1
        STA_Tx_Defer_Time = floor(tim + SIFS + DIFS + ACK_Tx +...
            Tx_STA_Buffer(STAs_Colission(2), 2));
        Tx_STA_Collision = false;
    else
         STA_Tx_Defer_Time = floor(tim + DIFS + ...
             MaxSize(Tx_STA_Buffer,STAs_Colission));
        Tx_STA_Collision =  true;
    end
    Txm_Comm = false;
end
%Check if channel is busy
```

```
if tim == STA_Tx_Defer_Time && Ch_Status_Flag == true
    if Tx_STA_Collision == false
        n = STAs_Colission(2);
        %Estimate Packet Error Rate
        PER = 1 - (1-BER)^(Tx_STA_Buffer(n, 2)*PHY_Rate*Slot_Time);
        %ACK in Error
        Err_ACK = 1 - (1 - BER)^(ACK_Tx*PHY_Rate*Slot_Time);
        if rand>=PER && rand>=Err_ACK
            %Packet is not Loss
            Pkt_Loss = 0;
        else
            %Packet is Loss
            Pkt_Loss = 1;
        end
    end
    %Flags for Transmission Status
     STA_Txr_Flag(tx_ACK>0)=1;
     STA_Txr_Flag(tx_Data>0)=2;
     STA_Txr_Flag(tx_RTS>0)=3;
     STA_Txr_Flag(tx_cts>0)=4;
    STA_Txr_Flag(tx_ACK>0)=5;
     STA_Txr_Flag(tx_Idle>0)=6;
     STA_Txr_Flag(tx_Have>0)=7;
     STA_Txr_Flag(tx_waiting>0)=8;
     STA_Txr_Flag(tx_backoff>0)=9;
     %------------------------ACK Frame must be transmitted---------------
      for flg = 1:n
          if(STA_Txr_Flag(flg)<=9 && STA_Txr_Flag(flg)>=5 && ...
                  currentFrameLen(flg)<=2)
        %Disable RTS and CTS Frames Transmission
        tx_RTS = zeros(1,n);
        %Output values
        if(STA_Txr_Flag(flg)==5)
            %Number of Successful Transmissions
            txm_Count = txm_Count + 1;
        end
        if(STA_Txr_Flag(flg)==6)
            %Number of Collisions
            col_Count = col_Count + 1;
        end
        if(STA_Txr_Flag(flg)==7)
            %Total Failed Transmissions
            unreach_Count = unreach_Count + 1;
        end
        Timer(CurrFrame_dest(flg))=ACK_Size;
        ACK_FrameDest(CurrFrame_dest(flg))=flg;
        STA_Txr_Flag(CurrFrame_dest(flg))=6;
        STA_Txr_Flag(flg)=8;
    end
end
       if mod(tim,1/Slot_Time)==0
              timer = tim*Slot_Time;
              timer = round(timer);
          %Throughput over SImulation Time
          STA_ULThruput_Time(timer) = UL_STA_Thruput;
          %Total Delay over Simulation Time
          Tx_Delay(timer) = Delay_at_Tx_AP*Slot_Time/Tx_Cnter;
          %Count Number of Packet Retransmission
          Packet_ResendTotalRate(timer) = Re_Tx_Pkt_Rate;
              %Reset for Next transmission
              UL_STA_Thruput = 0;
              Re_Tx_Pkt_Rate = 0;
              Delay_at_Tx_AP =0;
              Tx_Cnter=0;
      end
         %Check for collision or packet loss
      if Tx_STA_Collision == false && Pkt_Loss == 0
          %Determine STAs with loss packet or collision
          n = STAs_Colission(2);
          %Put packets in buffer
                      STA_Buff_Size(n) =  STA_Buff_Size(n) ...
                          - Tx_STA_Buffer(n,2);
      if n == N_STA + 1
          UL_STA_Thruput = UL_STA_Thruput + Tx_STA_Buffer(n,2)...
              *Slot_Time*PHY_Rate;
          %Increment the transmission counter
          Tx_Cnter = Tx_Cnter + 1;
          %Measure Delay
          Delay_at_Tx_AP = tim - STA_Delay_Buff(2);
          STA_Delay_Buff = DelayBuffPop(STA_Delay_Buff);
      else
          if STA_Buff_Size(N_STA+1) < Buff_Thresh - Tx_STA_Buffer(n,2);
              if STA_Backoff_Time(N_STA+1) == 1600;
                  STA_Backoff_Time(N_STA+1) =   RandomSTA(Min_CWin(N_STA+1));
              end
              Tx_STA_Buffer = PushStore(Tx_STA_Buffer,N_STA+1,...
                  Tx_STA_Buffer(n,2));
```

```
          STA_Buff_Size(N_STA+1) = STA_Buff_Size(N_STA+1)+...
              Tx_STA_Buffer(n,2);
          STA_having_Pkt(jj) = 1;
          z = Tx_Shared_Time(n);
          STA_Delay_Buff = DelayBuffPush(STA_Delay_Buff,z);
          Pkt_STA_Rxsmit(n) = 0;
      end

  end
Tx_STA_Buffer = PopRetrieve(Tx_STA_Buffer,n);
      Min_CWin(n) = 32;
      kk = Tx_STA_Buffer(n,1);
      if kk == 0
          STA_having_Pkt(n) = 0;
          STA_Backoff_Time(n) = 1600;
      else
          STA_Backoff_Time(n) = RandomSTA(Min_CWin(n));
      end
  else
      Re_Tx_Pkt_Rate = Re_Tx_Pkt_Rate+1;
      n = STAs_Colission(1);
      for ik = 1:n
          ki = STAs_Colission(ik+1);
          Min_CWin(ik) = CWIncrease(Min_CWin, ki);
          STA_Backoff_Time(ki) = RandomSTA(Min_CWin(ki));
          Pkt_STA_Rxsmit(ki)=1;
      end

  end
```

```
            STAs_Colission = zeros(1,N_STA+2);
            STA_Tx_Defer_Time = 0;
            Ch_Status_Flag = false;
            Tx_STA_Collision = false;
end
end
        SSFTxD_nSTA(mki,:) = Tx_Delay;
        SSFReTx_nSTA(mki,:) = Packet_ResendTotalRate;
        STA_Thru_allAP_OP(mki,:)    = STA_ULThruput_Time;
       else
           Status0 = ['=======================No Association between AP '...
               , num2str(iik),' and STA ',num2str(mki),' ============='];
           disp(Status0)
       end
    end
    SSF_Tx_Delay(:, :, iik) = SSFTxD_nSTA;
    SSF_ReTx_Rate(:,:,iik) = SSFReTx_nSTA;
end

%----Function for Delay Buffer POP-----------------------------------------
function y = DelayBuffPop(DelayBuff)
STA_Delay_Buff(:) = [STA_Delay_Buff(1), STA_Delay_Buff(3:Buff_Size), 0];
STA_Delay_Buff(1) = STA_Delay_Buff(1) - 1;
y = STA_Delay_Buff;
%-------------Function for Delay Buffer Push-------------------------------
function y = DelayBuffPush(DelayBuff, ST)
i = STA_Delay_Buff(1);
    if i < 6000
        STA_Delay_Buff(i+2) = z;
        STA_Delay_Buff(1) = STA_Delay_Buff(1)+1;
    end
y = STA_Delay_Buff;
%----------------Collision and Buffer Length
%Maximum Collision and Buffer Length
function y = MaxSize(PktBuff, CollSTAs)
    max= 0;
    for i=1:CollSTAs(1)
        if PktBuff(CollSTAs(i+1),2) > max
            max = PktBuff(CollSTAs(i+1),2);
        end
    end
y = max;
%--------------------Randomized Contention Window--------------------------
function  yy = RandomSTA(CWin)
 yy = 0;
 while yy == 0
     yy = floor(rand*Min_CWin);
 end
 %---------------------Exponetial Distrbituon for Packet arrival-----------
function STAs = exponentialDist(N_STAs)
    STAs = 0;
    while STAs == 0
        STAs = rand(1);
        STAs = floor((-N_STAs)*log(1-STAs));
    end
%========Carrier Sensing Function=========================================
function busy = CarrierSensing(CCAL, totalintPwr, stapwr, Noise)
    %totalintPwr is the total interference power
```

```
        %ccthres is the predefined CCA threshold, totalintPwr is the total
        %interference, and stapwr is the power of STA sensing the channel
    if (totalintPwr + stapwr + Noise) > CCAL
        busy = 1;
    else
        busy = 0;
    end
return;
%---------------------------Fairness Index Computation---------------------
Thru_av_OP = zeros(nAP, nSTA);
Thru_Sq = zeros(nAP, nSTA);
for ii = 1:nAP
    for jj = 1:nSTA
        Thru_av_OP(ii, jj) = (mean(STA_Thru_allAP_OP(jj,:,ii)));
        Thru_Sq(ii, jj) = (Thru_av_OP(ii, jj)).^2;
    end
end
Thru_DOWN_OP = zeros(1,nAP);
Thru_UP_OP = zeros(1,nAP);
Fainress_J = zeros(1, nAP);
N_value = 10;
for ij = 1:nAP
    if ij == 1
        NonZe = nonzeros(Thru_Sq(ij,:));
        NonZe2 = nonzeros(Thru_av_OP(ij,:));
         Thru_DOWN_OP(ij) = N_value*(sum(NonZe(:)));
         Thru_UP_OP(ij) = (sum(NonZe2(:)))^2;
         Fainress_J(ij) = Thru_UP_OP(ij)./Thru_DOWN_OP(ij);
    else
```

```
        NonZe = nonzeros(Thru_Sq(ij,:));
        NonZe2 = nonzeros(Thru_av_OP(ij,:));
         Thru_DOWN_OP(ij) = N_value*(sum(NonZe(1:10)));
         Thru_UP_OP(ij) = (sum(NonZe2(1:10)))^2;
         Fainress_J(ij) = Thru_UP_OP(ij)./Thru_DOWN_OP(ij);
    end
end

%Data Process
perBSS_ReTx = zeros(1, nAP);
perBSS_DelayTx = zeros(1, nAP);
for ii = 1:nAP
    perBSS_ReTx(ii)= mean(nonzeros(SSF_ReTx_Rate(:,:,ii)));
    perBSS_DelayTx (ii) = mean(nonzeros(SSF_Tx_Delay(:,:,ii)));
end
figure(1)
plot(perBSS_ReTx);
figure(2)
plot(perBSS_DelayTx);
% CDFplot of PHY Rate
figure(1);
ssfRate = cdfplot(STA_PhyRate_SFF);
set(ssfRate,'LineStyle','-')
hold on;
optrate = cdfplot(STA_PhyRate_Op);
set(optrate,'LineStyle','--')
xlabel('Uplink Throughput (Mbps)');
ylabel('Cumulative Fraction');
```

```
%CDFplot of Simulated CSMA Rates
figure(2);
ssfp = cdfplot(CDPAll_UL_Thruput_SSF(:)');
set(ssfp,'LineStyle','--');
hold on;
optp = cdfplot(CDPAll_UL_Thruput_Opt(:)');
set(optp,'LineStyle','-');
xlabel('Uplink Throughput (Mbps)');
ylabel('Cumulative Fraction');

%Plot of CSMA Simulated Rates Over time
figure(3);
xt = 1:Total_SIM_Time;
plot(CDPAll_UL_Thruput_SSF(:)','-b');
hold on;
plot(CDPAll_UL_Thruput_Opt(:)','-k')
%axis([0,Total_SIM_Time,0,1.8*10^6]);
xlabel('Simulation Time (s)');
ylabel('Uplink Throughput (Mbps)');
%%
%Network structure
figure(4);
hold on;
plot(APLocX, APLocY, 'r^');
plot(stax, stay, 'bs');
```

Appendix D

MATLAB Code: Optimal Matching

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%                 Author: Phillip B. Oni                              %%%
%%%              Dept. of Elec. & Comp. Engr, Queen's University        %%%
%%%          Optimal Assignment (Summer/FAll 2014)                      %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function opmatch = OptimalMatching1(edWeight_Matrix)
%This function is based on the Kuhn-Munkres Optimal Assignment Algorithm
%           Pseucode Available at:
%   http://csclab.murraystate.edu/bob.pilgrim/445/munkres.html
%The function returns optimal match in a logical matrix
%Set Optimal Match Variable to Logical matrix (0s and 1s) of size edge
%weight
opmatch = logical(size(edWeight_Matrix));
%Logical comparison of elements in the edge weight
edWeight_Matrix(edWeight_Matrix~=edWeight_Matrix)=Inf;
%Check for nonzero elements
Real_edWeight = edWeight_Matrix < Inf;
```

```
nonZ_Col = any(Real_edWeight);%Returns Nonzero Columns
nonZ_Row = any(Real_edWeight, 2);
T_Col = size(nonZ_Col);
T_Row = size(nonZ_Row);
N_elem = max(T_Row, T_Col);
if ~N_elem
    return
end
    Opt_Weight = zeros(N_elem);
    Opt_Weight(1:T_Row, 1:T_Col) = edWeight_Matrix(nonZ_Row, nonZ_Col);
    %Subtracting row minimum using elem-by-elem binary arithmetic operation
        Opt_Weight = bsxfun(@minus, Opt_Weight, max(Opt_Weight,[],2));
        %Find locations of Zeros
        loc_Zero = ~Opt_Weight;
        %Starred Zeros
        marked_Zeros = false(loc_Zero);
        while  any(loc_Zero(:))
            [i, j] = find(loc_Zero, 1);
            marked_Zeros(i, j) = 1;
            loc_Zero(i, :) = 0;
            loc_Zero(:,j) = 0;
        end
   while true
       %Cover all Colums with Starred Zero to obtain maximum matching
       Zeros_Col = flase(N_elem);
       visited_Col = any(marked_Zeros);
       %Find unreached columns
       if ~any(~visited_Col)
           break
```

```
end
visited_Row = false(n,1);
%Search for unmarked zeros
while true
    loc_Zero(:) = false;
 loc_Zero(~visited_Row, ~visited_Col) = ~Opt_Weight(~visited_Row...
        , ~visited_Col);
    %Finding Augmenting Path and Unmarked Zeros
    KM_Step6 = true; %Step 6 of the Kuhn Munkres Algorithm
    while any(any(loc_Zero(~visited_Row, ~visited_Col)))
        %Unmarked colums and rows
        [ii, jj] = find(loc_Zero, 1);
        Zeros_Col(ii, jj) = true;
        mkrd_Z = marked_Zeros(ii,:);
        if ~any(mkrd_Z)
            KM_Step6 = false;
            break;
        end
        visited_Row(ii) = true;
        visited_Col(mkrd_Z) = false;
        loc_Zero(ii,:) = false;
 loc_Zero(~visited_Row, mkrd_Z) = ~Opt_Weight(visited_Row, mkrd_Z);
    end
    if KM_Step6 == true
        new_Weight = Opt_Weight(~visited_Row, ~visited_Col);
        min_elem = min(min(new_Weight));
        if min_elem == inf
            return
        end
```

```
          Opt_Weight(visited_Row, visited_Col)= Opt_Weight(visited_Row,...
                   visited_Col)+min_elem;
             Opt_Weight(~visited_Row, ~visited_Col)= new_Weight - min_elem;
           else
               break
           end
       end
       row_Z = marked_Zeros(:,jj);
       marked_Zeros(ii,jj) = true;
       while any(row_Z)
           marked_Zeros(row_Z,jj) = false;
           jj = Zeros_Col(row_Z, :);
           ii = row_Z;
           row_Z = marked_Zeros(:,jj);
           marked_Zeros(ii, jj) = true;
       end
   end
opmatch(T_Row, T_Col) = marked_Zeros(1:T_Row, 1:T_Col);


function opmatching_dyn = OptimalMatchingDyn(edgWeight)
    %Initializing size of matching matrix
    opmatching_dyn = zeros(size(edgWeight));
    %Find nodes with outgoing edges
    nRow = sum(~isinf(edgWeight), 1);
    nCol = sum(~isinf(edgWeight),2);
    %Identify isolated vertices
    iso_Row = find(nRow~=0);
    iso_Col = find(nCol~=0);
```

```
Weigh_Size = max(length(iso_Row),length(iso_Col));
reduce_Weight = zeros(Weigh_Size);
reduce_Weight(1:length(iso_Row),1:length(iso_Col))= edgWeight(iso_Row...
    ,iso_Col);
%Hungarian routine
marker = true;
KM_Step = 1;

while marker
    switch KM_Step
        case 1
            %Step 1:
            size_m = length(reduce_Weight);
            for ij = 1:size_m
                %Find minimum edge weight in a row
                Row_min = min(reduce_Weight(ij,:));
                %Subtract row minimum from each row
                reduce_Weight(ij,:) = reduce_Weight(ij,:)-Row_min;
                KM_Step = 2;
            end
        case 2
            %Find zero element in reduce_Weight
            size_m = length(reduce_Weight);
            identfier = zeros(1,size_m);
            visited_row = zeros(length(reduce_Weight),1);
            visited_col = zeros(length(reduce_Weight),1);
            for ik = 1:size_m
                for ki = 1:size_m
```

```
                    if reduce_Weight(ik, ki)==0 && visited_row(ik)==0 ...
                        && visited_col(ki)==0
                    identfier(ik,ki) = 1;
                    visited_row(ik)=1;
                    visited_col(ki) = 1;
                    end
                end
            end
            %Reset visited nodes variables
            visited_row = zeros(length(reduce_Weight),1);
            visited_col = zeros(length(reduce_Weight),1);
            KM_Step = 3;
        case 3
            %Find covered nodes and marked with zero
            %If all nodes are covered, optimal matching is found
            visited_col = sum(identfier, 1);
            if sum(visited_col) == length(reduce_Weight)
                %Optimal matching is reached
                KM_Step = 7;
            else
                KM_Step = 4;
            end
        case 4
             kk= 1; jj = 1;
            size_m = length(reduce_Weight);
            flag4 = true;
            stop = true;
            while flag4
                %Find the uncovered zeros
```

```
while stop
  if reduce_Weight(kk,jj)==0 && visited_row(kk)==0 && ...
            visited_col(jj) == 0
        row = kk;
        col = jj;
        stop = false;
  end
  jj = jj+1;
  if jj > size_m; jj=1;
      kk = kk+1;
  end

  if kk > size_m;
      stop = false;
  end
end

%No augmenting path find go step 6
if row == 0
    KM_Step = 6;
    flag4 = false;
    row = 0;
    col = 0;
else
    %Prime the uncovered zeros
    reduce_Weight(row, col) = 2;
    if sum(find(reduce_Weight(row,:)==1))~=0
        visited_col(row)=1;
        Z_col = find(reduce_Weight(row,:)==1);
```

```
                            col_vis(Z_col) = 0;
                        else
                            KM_Step = 5;
                            flag4 = false;
                            row = 0;
                            col = 0;
                        end
                    end
                end
            case 5
                flag5 = true;
                z = 1;
                while flag5
                    rw = find(reduce_Weight(:,col(z))==1);
                    if rw > 0
                        z = z+1;
                        %Mark the row with starred zero
                        row(z,1) = rw;
                        col(z,1) = col(z-1);
                    else
                        flag5 = 0;
                    end

                    if flag5 == true
                        cl = find(reduce_Weight(row(z),:)==2);
                        z = z+1;
                        row(z,1) = row(z-1);
                        col(z,1)=cl;
                    end
                end
                %Remove all starred zeros
                for z = 1:length(row)
                    if reduce_Weight(row(z),col(z))==1
                        reduce_Weight(row(z),col(z))=0;
                    else
                        reduce_Weight(row(z),col(z))=1;
                    end
                end

                visited_row = zeros(length(visited_row));
                visited_col = zeros(length(visited_col));
                KM_Step =3;
            case 6
                r_covrd = find(visited_row ==0);
                c_covrd = find(visited_col ==0);
                min_weight = min(min(reduce_Weight(r_covrd,c_covrd)));
                reduce_Weight(find(r_covrd==1),:) = reduce_Weight(find( ...
                    r_covrd==1),:)+min_weight;
                reduce_Weight(:,find(c_covrd==0)) = reduce_Weight(:,find( ...
                    c_covrd ==0)) - min_weight;
                KM_Step = 4;
        end
    end
opmatching_dyn(iso_Row, iso_Col) = reduce_Weight(1:length(iso_Row),1:length ...
    (iso_Col));
```

Appendix E

MATLAB Code: Dynamic Scenario

```
%Dynamic Network Scenario
nSTA_varying = 50:60:300;
nAP = 30;
%Change Network Size an add new users every 10 seconds
Net_Up_Time = 20:20:100;
%Experimental Area
AreaY = 200;
AreaX = 200;
%Noise Level
No = -90; %dBm
%Linear scale dBm to Watt
No_L = 0.001*10^(No/10);
%Shadowing Variance
shdVar = 0.1;
%-----------------Random AP Location---------------------------------------
    tAP = nAP;
    APdis = zeros(1, length(nAP));
    APLocX = rand(1,nAP)*AreaX;
    APLocY = rand(1,nAP)*AreaY;
    apx = APLocX;
    apy = APLocY;
    freq = 2.4e9;
    vel = 3e8;
    lambda = vel/freq;
%Gains
Gt = 1; Gr = 1;
%Reference Distance
d0 = 1;
%Transmit Power
pwTX = 15.85; % 15.85mW, 12dBm
%txPwr = 10;
txPwr = 10*log10(pwTX); % dBm
alpha = 3.4; % ITU Rush hour
%Each time
for jt = 1:length(Net_Up_Time)
    Total_SIM_Time = Net_Up_Time(jt);
    %for ii = 1:length(nSTA_varying)
        nSTA = nSTA_varying(jt);
        UL_SINR = zeros(1, nSTA);
        STA_minDis = zeros(1, nSTA);
        SSF_RxPwr = zeros(1, nSTA);
        %SF_xval = zeros(1, nSTA);
        SSF_xval = zeros(nSTA, nAP);
        Status1 = ['=====Network Changed===', num2str(nSTA),...
            ' STAs Now on the Network===='];
        disp(Status1)
```

```
  %-----------------------Distance of STAs---------------------------
  STAdis = zeros(1, length(nSTA));
  STALocX = rand(1,nSTA)*AreaX;
  STALocY = rand(1,nSTA)*AreaY;
  stax = STALocX;
  stay = STALocY;

  %DistaAP-STAs distances
  dsbtwSTA_AP = zeros(nSTA, nAP);
  RxbtwSTA_AP = zeros(nSTA, nAP);
  STAwithAP = zeros(1, nAP);
for ij = 1:nSTA
  for im = 1:nAP
      dsbtwSTA_AP(ij, im) =sqrt((apx(im) - stax(ij))^2 + (apy(im) ...
      - stay(ij))^2);

      RxbtwSTA_AP(ij, im)= txPwr - netPathLoss(alpha, lambda, Gt, ...
      Gr, dsbtwSTA_AP(ij, im));
  end
end

  %SSF Association
 for ik = 1:nSTA
  STA_minDis(ik) = min(dsbtwSTA_AP(ik,:));
  SSF_RxPwr(ik) = max(RxbtwSTA_AP(ik,:));
 end

  %SSF Association Variable
```

```
        for ok = 1:nSTA
            for im = 1:nAP
                if  dsbtwSTA_AP(ok, im) == STA_minDis(ok);
                    SSF_xval(ok, im)= 1;
                    %Number of STAs per AP
                    STAwithAP(im)  = STAwithAP(im)+1;
                else
                    SSF_xval(ok, im)= 0;
                    STAwithAP(im)  = STAwithAP(im)+0;
                end
            end
        end%End of SSF Association variable
Optimal Association Begins
%===========================================================================
Optimal_Assoc = zeros(nAP,nSTA);
%Initial RSS values from SSF Associations
%This matrix it edge weight matrix
%Init_SSFRSS = %                           Optimal Association Begins
%===========================================================================
Optimal_Assoc = zeros(nAP,nSTA);
%This matrix it edge weight matrix
Init_SSFRSS = RxbtwSTA_AP;
%Begin Association Optimization using Hungarian Algorithm function
%For maximization problem, we add -1 to the weight matrix
%Init_SSFRSS = [Init_SSFRSS DummyAP_col];
%Call the Hungarian routine
Optimal_Match =OptimalMatchingDyn((-1*Init_SSFRSS));
%Optimal Match Received Power
Optimal_RSS = zeros(nAP, nSTA);
```

```
%Vector RSS Power
Opt_RSS = zeros(1,nSTA);
Opt_dis = zeros(1,nSTA);
%InitSSFRSS = Init_SSFRSS';
%dis_Opt = dSTAAP
for uo = 1:nSTA
    for ou = 1:nAP
        if(Optimal_Match(uo, ou)==1)
            Optimal_RSS(uo, ou)=Init_SSFRSS(uo,ou);
            Opt_RSS(uo) = Init_SSFRSS(uo,ou);
            %Get Distance of association
            %Opt_dis(uo) = dsbtwSTA_AP(uo, ou);
        else
            Optimal_RSS(uo, ou)=0;
            Opt_RSS(uo) = max(Init_SSFRSS(uo,:));
        end
    end
end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%SINR Estimation
%Minimum Interference Distance using 56 Mbps
STA_Inf_dis = zeros(1,nSTA);
snr_o = 24.6; %SINR for 56Mbps
snr_oL = 10^(snr_o/10);
max_Inf = zeros(1,nSTA);
for lk = 1:nSTA
    %Compute Interference range for each STA
    STA_Inf_dis(lk) = STA_minDis(lk).*(((1./(1/snr_oL - ...
        ((STA_minDis(lk)).^(alpha)).*(No_L./Opt_RSS(lk))))).^(1/alpha));
    %Compute Maximum interference for each STA based on interference range
    max_Inf(lk) = txPwr - netPathLoss(alpha, lambda, Gt, ...
    Gr, STA_Inf_dis(lk));
end
%Uplink SINRs
RxPower_STA = zeros(1, nSTA);
for im = 1:nSTA
    RxPower_STA(im) = 10.^(Opt_RSS(im)./10);
    %SINR under maximum system sensitivity
    UL_SINR(im) = 10*log10(RxPower_STA(im)./No_L + 10.^(max_Inf(im)./10));
end
%==============================Per STA PHY Rate Estimation=================
STA_PhyRate = zeros(1,nSTA);
for iu = 1:nSTA
    %Phy Rate of each STA
    STA_PhyRate(iu) = log2(1+10^(UL_SINR(iu)./10));
end
%Phy Rate of each STA connected AP
STAPhyRate = zeros(nAP, nSTA);
SSFxVal = Optimal_Match';
for op = 1:nAP
    for ip = 1:nSTA
        %Store Phy Rate for each STA
        if((SSFxVal(op, ip)==1))
            STAPhyRate(op, ip) = STA_PhyRate(ip);
        else
            STAPhyRate(op, ip) = 0;
        end
    end
```

```
end
%CSMA/CA Transmissions Begins======================================
SSFTxD_nSTA = zeros(nSTA, Total_SIM_Time);
SSFReTx_nSTA = zeros(nSTA, Total_SIM_Time);
STAUL= zeros(nSTA, Total_SIM_Time);
SSF_Tx_Delay_OP = zeros(nSTA, Total_SIM_Time, nAP);
SSF_ReTx_Rate_OP = zeros(nSTA, Total_SIM_Time, nAP);
STA_Thru_allAP_OP = zeros(nSTA, Total_SIM_Time, nAP);
    %Average Throughput per time
    Thru_perAVNetSize_OP = zeros(1, nAP);
    Thru_UL_OP = zeros(nSTA, Total_SIM_Time, nAP, length(Net_Up_Time));
    Delay_OP =zeros(nSTA, Total_SIM_Time, nAP, length(Net_Up_Time));
    Pkt_RxM = zeros(nSTA, Total_SIM_Time, nAP, length(Net_Up_Time));
    Delay_AvgNetSize_OP = zeros(1, nAP);
%Total Uplink Throughput per Network Size
Thru_Total_OP = zeros(1, length(nSTA_varying));
Delay_Total_OP = zeros(1, length(nSTA_varying));
%Association
%APtoSTA_OPop = Optimal_Match';
for iik = 1:nAP
  N_STA = STAwithAP(iik);
for mki = 1:nSTA
   if (SSFxVal(iik, mki) == 1)
       PHY_Rate = (STAPhyRate(iik, mki)).*10^(6);
       Status1 = ['========================STA', num2str(mki)...
   , ' is Associated with AP', num2str(iik),' =======Transmitting at ',...
   num2str(PHY_Rate),' bps'];
       disp(Status1)
                        %%
```

```
        %CCA Threshold
        CCAth = -60; %dB
        CCAL = 10^(CCAth/10); %linear
        %Bit Error rate
BER = 10^(-6);
%Number of STAs
%N = 5;
%Delay
Delay = 0;
%Packet Retransmission rate
STA_Retransmit_Packet_Rate = 0;
%Counter for Number of Transmissions
Tx_Cnter = 0;
Buff_Size = 6000;
Total_STA_Pkt_Retransmit = zeros(1, N_STA+1);
%Sharing Transmission Time among users
Tx_Shared_Time = zeros(1, N_STA+1);
%Packets in Buffer due to Delay
STA_Delay_Buff = zeros(1, Buff_Size);
%Flag for idle or Busy Channel
Ch_Status_Flag = false;
%Flag for active transmissionn
Txm_Comm = false;
%Flag for Collision
    Tx_STA_Collision = false;
    %Total Deferring Time
    Tx_DeferTime = 0;
    %Uplink Throughout
    STA_UL_Thruput = 0;
```

```
        %Downlink Throughput
        %DL_Thruput = 0;
        %All STAs Throughputs
        %STA_ULThruput = zeros(nSTA, nAP);
        %Throughput over Simulatiion time
        STA_ULThruput_Time = zeros(1, Total_SIM_Time);
        %Packet re-Transmission over Simulation Time
        Packet_ResendTotalRate = zeros(1, Total_SIM_Time);
        %Transmission Delay
        Tx_Delay = zeros(1, Total_SIM_Time);
                %======================Packet Arrival Time=================
                %%
                Pkt_Arrive_Time_atQue = 1:N_STA;
                %Packet Length
                STA_Pkt_Vector = zeros(1,N_STA+1);
                %STAs having Packet ready
                STA_having_Pkt = zeros(1, N_STA+1);
                %Contention Window for STAs
                Min_STA_CWin = 1:N_STA+1;
                %Backoff Timer for all STAs
                STA_Backoff_Time = 1:N_STA+1;
                %==================Packet Buffer Size for each STA=========
                Tx_STA_Buffer = zeros(N_STA+1, Buff_Size);
                %STAs in Collision
                STAs_Colission = zeros(1, N_STA+2);
%============================CSMA/CA Parameters============================
%Slot time in 802.11b
Slot_Time = 20*10^(-6);
%Dividing Simulation time into slot time
```

```
Actual_total_Time = Total_SIM_Time/Slot_Time;
%Short interframe space
SIFS = 0.5*Slot_Time; %SIFS is half of Slot time
%Distributed interframe space
DIFS = 2.5*Slot_Time;
%Period for ACK transmission
%Average Arrival Time
Pkt_Avrg_Arrival_Time = 10;
%Average Packet Length
Pkt_Avrg_STA_PktLen = 1460*8; %byte to bits;
%Buffer Size
Buff_Size = zeros(1,N_STA+1);
%AP Buffer Threshold
%Random Arrival of STAs
for ii = 1:N_STA
    Pkt_Arrive_Time_atQue(ii) =  exponentialDist(Pkt_Avrg_Arrival_Time);
    STA_Pkt_Vector(ii) = exponentialDist(Pkt_Avrg_STA_PktLen);
end
%==========================Contention Window===============================
    for ii = 1:N_STA+1
        Min_STA_CWin(ii) = 32; %Contention Window
        STA_Backoff_Time(ii) = 1600;
    end
    %%
  %================================Transmission=============================
for tim = 1:total_Time
    for jj = 1:N_STA
        %Total time to Return ACK  for transmission
        ACK_Tx = 14*8/(PHY_Rate*Slot_Time);
    %Set the buffer threshold
        Buff_Thresh = 8*10^6/(PHY_Rate*Slot_Time);
        %Check packet in Queue
    if tim == Pkt_Arrive_Time_atQue(jj)
        if Buff_Size(jj) < Buff_Thresh - STA_Pkt_Vector(jj)
            Tx_STA_Buffer = PushStore(Tx_STA_Buffer, jj, ...
                STA_Pkt_Vector(jj));
            Buff_Size(jj) = Buff_Size(jj) + STA_Pkt_Vector(jj);
            STA_having_Pkt(jj) = 1;
            if STA_Backoff_Time(jj) == 1600;
                %Randomly Select STAs for Backoff
                STA_Backoff_Time(jj) = RandomSTA(Min_STA_CWin(jj)) ;
            end
        end
     %Generate Packet Arrival Rate based on Exponential Distribution
Pkt_Arrive_Time_atQue(jj) = exponentialDist(Pkt_Avrg_Arrival_Time)+...
        STA_Pkt_Vector(jj)+tim;
    STA_Pkt_Vector(jj) = exponentialDist(Pkt_Avrg_STA_PktLen);
    end
end

    for jj = 1:N_STA + 1
        %Check if a STA has packet and Sense the channel
        if STA_having_Pkt(jj)==1 && Ch_Status_Flag == false
           if STA_Backoff_Time(jj) == 0
               STAs_Colission = Collision_Among_STAs(STAs_Colission, jj);
               if Total_STA_Pkt_Retransmit(jj) == 0
                   Tx_Shared_Time(jj)= tim+DIFS;
               end
```

```
                   Txm_Comm = true;
               else
                   STA_Backoff_Time(jj) = STA_Backoff_Time(jj) - 1;
               end
            end
        end

    if Txm_Comm == true
        Ch_Status_Flag = true;
        n = STAs_Colission(1);
        if n == 1
            Tx_DeferTime = floor(tim + SIFS + DIFS + ACK_Tx ...
                + Tx_STA_Buffer(STAs_Colission(2), 2));
            %No Collision
            Tx_STA_Collision = false;
        else
             Tx_DeferTime = floor(tim + DIFS + MaxSize(Tx_STA_Buffer,...
                 STAs_Colission));
             %Collision
            Tx_STA_Collision =  true;
        end
        Txm_Comm = false;
    end

        if tim == Tx_DeferTime && Ch_Status_Flag == true
            if Tx_STA_Collision == false
                n = STAs_Colission(2);
                %Estimate Packet Error Rate
                PER = 1 - (1-BER)^(Tx_STA_Buffer(n, 2)*PHY_Rate*Slot_Time);
```

```
           %ACK in Error
           Err_ACK = 1 - (1 - BER)^(ACK_Tx*PHY_Rate*Slot_Time);
           if rand>=PER && rand>=Err_ACK
               %Packet is not Loss
               Pkt_Loss = 0;
           else
               %Packet is Loss
               Pkt_Loss = 1;
           end
       end

       %Flags for Transmission Status
        STA_Txr_Flag(tx_ACK>0)=1;
        STA_Txr_Flag(tx_Data>0)=2;
        STA_Txr_Flag(tx_RTS>0)=3;
        STA_Txr_Flag(tx_cts>0)=4;
       STA_Txr_Flag(tx_ACK>0)=5;
        STA_Txr_Flag(tx_Idle>0)=6;
        STA_Txr_Flag(tx_Have>0)=7;
        STA_Txr_Flag(tx_waiting>0)=8;
        STA_Txr_Flag(tx_backoff>0)=9;
%------------------------ACK Frame must be transmitted---------------
 for flg = 1:n
     if(STA_Txr_Flag(flg)<=9 && STA_Txr_Flag(flg)>=5 && ...
             currentFrameLen(flg)<=2)
         %Disable RTS and CTS Frames Transmission
         tx_RTS = zeros(1,n);
         %Output values
         if(STA_Txr_Flag(flg)==5)
```

```
            %Number of Successful Transmissions
            txm_Count = txm_Count + 1;
        end
        if(STA_Txr_Flag(flg)==6)
            %Number of Collisions
            col_Count = col_Count + 1;
        end
        if(STA_Txr_Flag(flg)==7)
            %Total Failed Transmissions
            unreach_Count = unreach_Count + 1;
        end
            Timer(CurrFrame_dest(flg))=ACK_Size;
            ACK_FrameDest(CurrFrame_dest(flg))=flg;
            STA_Txr_Flag(CurrFrame_dest(flg))=6;
            STA_Txr_Flag(flg)=8;
    end
end
           if mod(tim,1/Slot_Time)==0
              timer = tim*Slot_Time;
              timer = round(timer);
              %Throughput over SImulation Time
              STA_ULThruput_Time(timer) = STA_UL_Thruput;
              %Total Delay over Simulation Time
              Tx_Delay(timer) = Delay*Slot_Time/Tx_Cnter;
         Packet_ResendTotalRate(timer) = STA_Retransmit_Packet_Rate;
                  %Reset for another transmission
                  STA_UL_Thruput = 0;
                  %Count number of Retransmit
                  STA_Retransmit_Packet_Rate = 0;
```

```
                Delay =0;
                Tx_Cnter=0;
         end
%Check if still no collision and packet loss
if Tx_STA_Collision == false && Pkt_Loss == 0
    n = STAs_Colission(2);
    %Check size of buffer
    Buff_Size(n) =  Buff_Size(n) - Tx_STA_Buffer(n,2);
    if n == N_STA + 1
        %Compute throughput at this point
STA_UL_Thruput = STA_UL_Thruput + Tx_STA_Buffer(n,2)*Slot_Time*...
            PHY_Rate;
        %Increment the transmission counter
        Tx_Cnter = Tx_Cnter+1;
        Delay = tim - STA_Delay_Buff(2);
        STA_Delay_Buff = DelayBuffPop(STA_Delay_Buff);
    else
        if Buff_Size(N_STA+1) < Buff_Thresh - Tx_STA_Buffer(n,2);
            if STA_Backoff_Time(N_STA+1) == 1600;
                %Randomized Backoff
    STA_Backoff_Time(N_STA+1) = RandomSTA(Min_STA_CWin(N_STA+1));
            end
      Tx_STA_Buffer = PushStore(Tx_STA_Buffer,N_STA+1,...
                Tx_STA_Buffer(n,2));
            Buff_Size(N_STA+1) = Buff_Size(N_STA+1)+Tx_STA_Buffer(n,2);
            STA_having_Pkt(jj) = 1;
            ST = Tx_Shared_Time(n);
            STA_Delay_Buff = DelayBuffPush(STA_Delay_Buff,ST);
            Total_STA_Pkt_Retransmit(n) = 0;
```

```
       end

end
               Tx_STA_Buffer = PopRetrieve(Tx_STA_Buffer,n);
               Min_STA_CWin(n) = 32;
               kk = Tx_STA_Buffer(n,1);
               if kk == 0
                   STA_having_Pkt(n) = 0;
                   STA_Backoff_Time(n) = 1600;
               else
                   STA_Backoff_Time(n) = RandomSTA(Min_STA_CWin(n));
               end
           else
               %Increment the number of retransmitted packets
               STA_Retransmit_Packet_Rate = ...
                   STA_Retransmit_Packet_Rate+1;
               n = STAs_Colission(1);
               for ikk = 1:n
                   ki = STAs_Colission(ikk+1);
                   Min_STA_CWin(ikk) = CWIncrease(Min_STA_CWin, ki);
                   STA_Backoff_Time(ki) = RandomSTA(Min_STA_CWin(ki));
                   Total_STA_Pkt_Retransmit(ki)=1;
               end

           end
           STAs_Colission = zeros(1,N_STA+2);
           Tx_DeferTime = 0;
           Ch_Status_Flag = false;
           Tx_STA_Collision = false;
```

```
                end
               end
        SSFTxD_nSTA(mki,:) = Tx_Delay;
        SSFReTx_nSTA(mki,:) = Packet_ResendTotalRate;
        STAUL(mki,:) = STA_ULThruput_Time;
       else
           Status0 = ['===================No Association between AP ', ...
               num2str(iik),' and STA ',num2str(mki),' ============='];
           disp(Status0)
       end
    end

    SSF_Tx_Delay_OP(:, :, iik) = SSFTxD_nSTA;
    SSF_ReTx_Rate_OP(:,:,iik) = SSFReTx_nSTA;
    STA_Thru_allAP_OP(:,:, iik) = STAUL;
   %Average uplink throughput per Time and Network size
   %Thru_perAVNetSize_OP(iik) = mean(nonzeros(STAUL));%Average over STAs
   %Delay_AvgNetSize_OP(iik) = mean(nonzeros(SSFTxD_nSTA));%Average
end
Thru_UL_OP(:,:,:,jt) = STA_Thru_allAP_OP;%Average over AP
Delay_OP(:,:,:,jt) = SSF_Tx_Delay_OP;%Average over AP
Pkt_RxM(:,:,:,jt) = SSF_ReTx_Rate_OP;

%Thru_Total_OP(jt) = sum(nonzeros(STA_Thru_allAP_OP));
%Delay_Total_OP(jt) = sum(nonzeros(SSF_Tx_Delay_OP));
%end
end
AvgThru = zeros(1, length(nSTA_varying));
```

```matlab
for iy = 1:length(nSTA_varying)
    AvgThru(iy) = sum((Thru_UL_OP(:,:,:,:)));
end
Opt1 = nonzeros(Thru_UL_OP(:,:,:,:));
Ssf = nonzeros(Thru_UL(:,:,:,:));
OptD = nonzeros(Delay_OP(:,:,:,:));
SsfD = nonzeros(Delay_SSF(:,:,:,:));
cdfplot(OptD); hold on; cdfplot(SsfD);
OptRx = nonzeros(Pkt_RxM(:,:,:,:));
SsfDRx = nonzeros(PktRxM_SSF(:,:,:,:));
plot(sort(OptRx(300:500))); hold on; plot(sort(SsfDRx(300:500)),'--k');
sizep = 0;
ThrupPerSize_OP = zeros(9300, length(Net_Up_Time));
ThrupPerSize_OP(:, 5) = Opt1(37201:46500);
for uui = 1:length(Net_Up_Time)
    sumThr_OP(uui) = sum(ThrupPerSize_OP(:,uui))/10^6;
    AvThrupPerSize_OP(uui) = sum(ThrupPerSize_OP(:,uui))/nSTA_varying(uui);
end
plot(nSTA_varying, sort(AvThrupPerSize_OP)/10^2)
plot(sort(ThrupPerSize_OP(1:300,5)))
for uui = 1:length(Net_Up_Time)
    sumThr_SSF(uui) = sum(ThrupPerSize_OP(1:9300,uui));
    AvThrupPerSize_SSF(uui) = sum(ThrupPerSize_SSF(1:9300,uui))...
        /nSTA_varying(uui);
end
plot(nSTA_varying, (AvThrupPerSize_SSF)/10^2,'--k'); hold on;
plot(nSTA_varying, (AvThrupPerSize_OP)/10^2)
plot(sort(ThrupPerSize_OP(1:300,5)),'-b'); hold on;
plot(sort(ThrupPerSize_SSF(1:300,5)),'--k');
```

```
ThrupPerSize_SSF = zeros(88540, length(Net_Up_Time));
ThrupPerSize_SSF(:, 5) = Ssf(354161:end);
%----Function for Delay Buffer POP-----------------------------------------
function y = DelayBuffPop(DelayBuff)
STA_Delay_Buff(:) = [STA_Delay_Buff(1), STA_Delay_Buff(3:5001), 0];
STA_Delay_Buff(1) = STA_Delay_Buff(1) - 1;
y = STA_Delay_Buff;
%-------------Function for Delay Buffer Push-------------------------------
function y = DelayBuffPush(DelayBuff, ST)
i = STA_Delay_Buff(1);
    if i < 5000
        STA_Delay_Buff(i+2) = ST;
        STA_Delay_Buff(1) = STA_Delay_Buff(1)+1;
    end
y = STA_Delay_Buff;
%========Carrier Sensing Function=========================================
function busy = CarrierSensing(CCAL, totalintPwr, stapwr, Noise)
    %totalintPwr is the total interference power
        %ccthres is the predefined CCA threshold, totalintPwr is the total
        %interference, and stapwr is the power of STA sensing the channel
    if (totalintPwr + stapwr + Noise) > CCAL
        busy = 1;
    else
        busy = 0;
    end
return;
%----------------Collision and Buffer Length
%Maximum Collision and Buffer Length
function y = MaxSize(PktBuff, CollSTAs)
```

```
    max= 0;
    for i=1:CollSTAs(1)
        if PktBuff(CollSTAs(i+1),2) > max
            max = PktBuff(CollSTAs(i+1),2);
        end
    end
y = max;
%--------------------Randomized Contention Window--------------------------
function  zy = RandomSTA(Min_STA_CWin)
 zy = 0;
 while zy == 0
     zy = floor(rand*Min_STA_CWin);
 end
 %---------------------Exponetial Distrbituon for Packet arrival-----------
 function STAs = exponentialDist(N_STAs)
    STAs = 0;
    while STAs == 0
        STAs = rand(1);
        STAs = floor((-N_STAs)*log(1-STAs));
    end
```