OFDMA BASED DEVICE-TO-DEVICE COMMUNICATION PROTOCOLS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

YUNUS CAN GÜLTEKİN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

SEPTEMBER 2015

Approval of the thesis:

OFDMA BASED DEVICE-TO-DEVICE COMMUNICATION PROTOCOLS

submitted by **YUNUS CAN GÜLTEKİN** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver
Dean, Graduate School of **Natural and Applied Sciences** ________________

Prof. Dr. Gönül Turhan Sayan
Head of Department, **Electrical and Electronics Engineering** ________________

Prof. Dr. Ali Özgür Yılmaz
Supervisor, **Electrical and Electronics Eng Dept,METU** ________________

Examining Committee Members:

Prof. Dr.Yalçın Tanık
Electrical and Electronics Engineering Dept., METU ________________

Prof. Dr. Ali Özgür Yılmaz
Electrical and Electronics Engineering Dept., METU ________________

Prof. Dr. Buyurman Baykal
Electrical and Electronics Engineering Dept., METU ________________

Assoc. Prof. Dr. Cüneyt Fehmi Bazlamaçcı
Electrical and Electronics Engineering Dept., METU ________________

Prof. Dr. Ezhan Karaşan
Electrical and Electronics Engineering Dept., Bilkent University ________________

Date: 09.09.2015

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: YUNUS CAN GÜLTEKİN

Signature :

ABSTRACT

OFDMA BASED DEVICE-TO-DEVICE COMMUNICATION PROTOCOLS

Gültekin, Yunus Can
M.S., Department of Electrical and Electronics Engineering
Supervisor : Prof. Dr. Ali Özgür Yılmaz

September 2015, 94 pages

This thesis considers the physical and media access control layer (PHY/MAC) architectures of a device-to-device (D2D) communication protocol for ad hoc networks. We propose an orthogonal frequency-division multiple access (OFDMA) based distributed D2D solution for frequency hopping networks operating with small packet sizes. The main contributions to the literature are adding an average received power estimation method to achieve interference-awareness, and implementing a time slotted connection scheduler to attain efficient spatial reuse. Design aspects are discussed for both peer discovery and connection scheduling stages. Simulation results are presented to demonstrate the potential of the proposed structure from the throughput perspective in networks where a frequency hopping strategy is implemented, and the packet sizes are relatively small as in Bluetooth.

Keywords: Ad hoc networks, D2D communications, device discovery, distributed scheduling, OFDMA.

ÖZ

OFDMA TABANLI CİHAZDAN CİHAZA HABERLEŞME PROTOKOLLERİ

Gültekin, Yunus Can
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. Ali Özgür Yılmaz

Eylül 2015, 94 sayfa

Bu çalışmada, ad hoc ağlar için cihazdan cihaza (D2D) haberleşme protokollerinin fiziksel ve ortam erişim kotrolü katmanları ele alınmıştır. Frekans atlamalı ve küçük paket boyutları ile çalışan ağlar için dikgen sıklık bölümlemeli çoklu erişim (OFDMA) yöntemi temel alınarak dağıtık bir çözüm önerilmiştir. Girişim farkındalığı sağlamak için önerilen bir ortalama alınan güç bilgisi kestirimi yöntemi ve etkin bir uzamsal yeniden kullanım elde etmek için önerilen zaman dilimli bir bağlantı planlayıcısı literatüre yapılan başlıca katkılardır. Özellikle akran keşfi ve bağlantı planlama evreleri için tasarıma bakış açıları tartışılmıştır. Önerilen yapının Bluetooth gibi frekans atlamalı ve küçük paket boyutlu ağlarda veri hacmi artışı açısından kullanım olanaklarını göstermek için benzetim sonuçları sunulmuştur.

Anahtar Kelimeler: Ad hoc ağlar, D2D haberleşme, akran keşfi, dağıtık bağlantı planlama, OFDMA.

To my family

ACKNOWLEDGMENTS

I would like to thank my supervisor, Prof. Dr. Ali Özgür Yılmaz, for giving me the opportunity to work on this thesis and for his valuable guidance. I was unbelievably lucky to have him as my mentor.

I owe to The Scientific and Technological Research Council of Turkey (TÜBİTAK) for their financial support during my two years of post graduate studies.

I would like to thank Aselsan A.Ş. for giving me financial support covering one year of my study and more importantly, giving me the chance to work with Ali Bulut Üçüncü, Alptekin Yılmaz, and Dr. Tuğcan Aktaş.

It was a pleasure to have both regular and technical conversations with Dr. Gökhan Muzaffer Güvensen, Samet Gelincik, Pınar Şen, and Seçil Özdemir from our Communications Research Laboratory.

Finally, I would like to thank my friends and housemates for their support, and my family above all. I wouldn't be the person who I am right now without the endless encouragement and love of Nevriye and Şeref Gültekin.

TABLE OF CONTENTS

CHAPTERS

APPENDICES

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

1G	First Generation
2G	Second Generation
3G	Third Generation
4G	Fourth Generation
5G	Fifth Generation
ACI	Adjacent Channel Interference
ACK	Acknowledgement
AP	Access Point
AWGN	Additive White Gaussian Noise
BLE	Bluetooth Low Energy
BS	Base Station
CDF	Cumulative Distribution Function
CDMA	Code Division Multiple Access
CFO	Carrier Frequency Offset
CID	Connection Identity
COST	Cooperation in the Field of Scientific and Technical Research
CP	Cyclic Prefix
CQI	Channel Quality Indicator
CRC	Cyclic Redundancy Check
D2D	Device-to-Device
DFT	Discrete Fourier Transform
DHCP	Dynamic Host Configuration Protocol
DVB	Digital Video Broadcasting
ENB	Evolved Node B
FDM	Frequency Division Multiplexing
FHS	Frequency Hoping Synchronization
FHSS	Frequency-Hopping Spread Spectrum
GSM	Global System for Mobile Communications
IEEE	Institute of Electrical and Electronics Engineers
IDFT	Inverse Discrete Fourier Transform
ISI	Intersymbol Interference
ISM	Industrial, Scientific and Medical

LAN	Local Area Network
LTE-A	Long Term Evolution Advanced
M2M	Machine-to-Machine
MAC	Media Access Control
MAP	Maximum A Posteriori Estimation
MIMO	Multiple-Input and Multiple-Output
NFC	Near Field Communications
OFDM	Orthogonal Frequency Division Multiplexing
OFDMA	Orthogonal Frequency Division Multiple Access
P2P	Peer-to-Peer
PAN	Personal Area Network
PCS	Paging and Connection Scheduling
PDRID	Peer Discovery Resource Identity
PHY	Physical Layer
QAM	Quadrature Amplitude Modulation
QOS	Quality of Service
QPSK	Quadrature Phase Shift Keying
SINR	Signal-to-Interference-Plus-Noise Ratio
SIR	Signal-to-Interference Ratio
SNR	Signal-to-Noise Ratio
TDD	Time Division Duplexing
UE	User Equipment
WLAN	Wireless Local Area Network

CHAPTER 1

INTRODUCTION

Mobile telecommunications have evolved perpetually starting from late seventies. First generation (1G) of wireless telephony technology used analog telecommunications standards and signals. The main purpose of 1G was to transmit voice, and this was achieved by only modulating it to higher frequencies. Second generation (2G) cellular systems were first offered to include advantages of digital communications in voice transmission such as spectrum efficiency. After the Global System for Mobile Communications (GSM) standard had widespread area of utilization, 2G systems introduced data service to mobile communications. Third generation (3G) of mobile systems were introduced in the early twenty first century to satisfy the demand for data rate and internet access which increased rapidly with the popularization of the smart-phones. Finally, fourth generation (4G) cellular networks are developed and introduced around 2010 to provide broadband internet access for advanced mobile applications like cloud computing, high-definition mobile TV, IP telephony etc.

In [1], the definition of fifth generation (5G) of cellular networks is fundamentally built upon the term *new use cases*. 5G is expected to enable far more and diverse applications which cannot possibly be supported by an earlier generation such as online gaming, intelligent transport systems, unmanned aerial vehicle control, public safety, e-Health, context and proximity-aware advertising etc. Consequently, fundamental challenges of 5G research arises from this variety of applications that are anticipated to be used. Unlike voice, text, or video transmission and internet access for social networking, the primary need of the applications of the new age is not only data rate. 5G researchers prefer to depict the demands of new use cases of 5G in a multi-dimensional space in which all of the dimensions represent another system characteristic to be satisfied such as rate, reliability, cost, latency, security, energy efficiency etc. To give examples, some expectations from a 5G network are:

- Enabling very high data rates along with instantaneous connectivity,
- Provide broadband access in extremely crowded environments such as shopping malls, stadiums etc. (i.e., massive multiple access),
- Delivering user specific service experience with context and proximity-aware applications,
- Supply reliable and amazingly real-time connections (i.e., very strict latency constraints)
- Handling many large number of devices of different types efficiently (i.e., Internet of Things, IoT).

Satisfying these demands in a single framework is obviously a challenging work. On the other hand, there are also new tools and enabling technologies which are being developed or investigated in the literature [2], [3]. Among them, the major technologies which are anticipated excitedly are millimeter wave (mm-wave) communications [4] and massive multiple-input multiple-output (MIMO) systems [5].

The aforementioned enabling technologies are mainly investigated for the intensive data rate increase in future cellular systems. On the other hand, the massive increase in the number of devices connected to the network is also an important problem. Together with the rapid development in the electronics production technology, the number of devices which are able to connect to the web has increased immensely. In addition to smart phones, tablets and laptops, telecommunication engineers expect that electricity meters, household devices, smart health care chips, any kind of sensors, watches and many other electronic devices will be connected to each other and share information. In fact, as an example, electricity distribution companies have already begun to measure the consumption of traffic lights and public lighting posts remotely using GSM in Turkey. This remarkable advance in the communicating devices has led to a significant densification of the wireless networks. It is anticipated that there will be 25 billion devices connected by 2020. In addition, the wireless traffic will be a few thousand times higher than today's. In the light of these predictions, we can easily foresee that along with the problem of satisfying the demand for data rate and latency, another focus should be solving this problem in an immensely dense network.

Before offering a solution to this problem, we should gain more insight about the cellular structure. Throughout the advancement course of the cellular systems, BSs have always been used to control the communications. Basically, all transmissions in a cellular network must go through the BS regardless. This structure fits the traditional cellular services with low data rates such as messaging and voice calling. Besides, these types of communication services are established between distant users (separated by more than few kilometres) which cannot communicate with a point-to-point connection due to the inherent nature of the cellular communication system. Moreover, coupled with the increasing density of mobile users especially in urban areas, cell areas have shrunk in recent years. Microcells, picocells, and femtocells are introduced to cover indoor areas such as malls, offices, and houses, respectively. Cheap, small and low-power cellular base stations are designed and deployed in metropolitan areas.

On the other hand, mobile connections in 5G networks will be much more diverse and intense from the application end so that establishing each connection through a BS will be nearly impossible. In addition, many types of applications will not need to be connected to an infrastructure continuously such as sensor monitoring, advertising, content distribution, gaming etc. These types of connections where the range of communication is relatively small can be appropriate for point-to-point transmissions. Therefore, D2D communication strategies which is the main focus of this thesis work can be implemented underlaying cellular networks to take the burden off the infrastructure or independently to form an environment specific network which is connected to the web through a local area network (LAN) wireless access point (AP).

Finally, the aforementioned types of applications will have data packets of small size. For example, it is reasonable to expect that a few dozens of bytes of packets will be enough to transmit data fusion of the information about the congestion of a road acquired form cars, advertisement messages which are sent to people in a shopping mall, the electricity consumption information collected over a smart grid system etc. In such networks, distributed connection scheduling should be carefully designed to maximize network throughput considering the transmission efficiency and quality of service (QoS) supplied to maximum number of users. Furthermore, such networks can be designed to operate with a frequency hopping strategy to minimize the detrimental effects which can be caused due to the massive number of various wireless networks working in unlicensed band.

This thesis work includes a literature review of D2D communication aspects and examples. In addition, an OFDMA based implementation of a D2D communication protocol is designed and simulated. Furthermore, we propose some modifications to make a D2D protocol suitable for networks where a frequency hopping strategy is implemented, and the packet sizes are relatively small. As an example, one can imagine a 1 km x 1 km site where the energy consumption of illumination and climatization structures, and basic household devices is to be minimized. In such a network, there will be several thousands of devices distributed over the area which have to share their limited information with each other or collect them in a central node collaboratively in a wireless fashion. This kind of sensor monitoring problem among many others can be solved by our approach.

The layout of the thesis is as follows: In Chapter 2 we will give the definition, aspects, and examples of D2D communication systems. Then in Chapter 3, peer discovery phase of a D2D communication protocol will be examined, and an OFDMA based approach will be simulated. Finally, we will investigate the connection establishment procedure of such a system, explain our approach which is developed for the aforementioned type of networks, and present simulation results to illustrate the performance of our approach in Chapter 4. In Chapter 5 we will state our conclusions briefly.

CHAPTER 2

DEVICE-TO-DEVICE COMMUNICATIONS

2.1 Definition

The methods which allow devices to establish a direct communication without taking any help from an infrastructure of BSs or APs are called as D2D communications in general [6]. D2D strategies are designed as an add-on mode for Long Term Evolution Advanced (LTE-A) [7] networks in which user equipments (UEs) exchange information over a point-to-point link by utilizing cellular spectrum rather than through eNB (i.e., base station). The main purpose was to increase the spectral efficiency by using D2D strategies as an underlay to cellular networks. On the other hand it is expected that D2D communications will be an inherent part of 5G networks [8].

In [6], D2D communications are categorized into three types which are:

- Peer-to-peer (P2P) communications which consist of point-to-point links and data exchange,
- Cooperative communication where UEs act as relays to increase coverage, and take advantage of cooperative diversity [9],
- Multi-hop communications which includes routing and network coding to transmit data to a BS or to a user which cannot be communicated directly.

Even though D2D communications are implemented to improve spectral efficiency in cellular networks, they lead to an extra interference since they use the same spectrum. Therefore effective interference management and interference coordination should be utilized to satisfy performance needs as addressed in [10]. Furthermore, efficient resource allocation strategies must be put to use to increase spectral efficiency by avoiding inter-connection interference [11] whether the D2D strategy is underlaying a cellular network or implemented independently. We will present our resource allocation approach which is designed to maximize network throughput in Chapter 4.

2.2 Taxonomy

D2D communications can be classified [6] by their degree of independence from the network as follows:

- Network-Controlled D2D in which the communication setup, resource allocation, interference management are coordinated by the BS. Although the use of a centralized node improves performances of these stages, as the number of D2D links in the system increases the control signalling overhead becomes abundant and decreases spectrum efficiency,
- Network-Assisted D2D in which the communications are organized in a distributed fashion by devices, but limited information supplied by the cellular network is used to improve resource management,
- Self-Organized D2D in which devices orchestrate all of the communication control procedures themselves which avoids redundant device-to-BS communication while fully distributed approaches may cause some errors.

The D2D protocol under investigation in thesis work is designed for a self-organized D2D network. Moreover, D2D communications can be classified [12] by the spectrum that they utilize as:

- Inband D2D where both cellular and D2D communications use the same cellular spectrum. Since we have a control over the cellular spectrum by using BSs, researchers believe that interference can be managed easily if D2D also uses the same spectrum,

- Outband D2D where unlicensed spectrum is utilized to realize D2D communications. The reason of using this approach to avoid the interference between D2D and cellular connections. To use unlicensed spectrum, another interface should be used such as Wi-Fi Direct [13], ZigBee [14], Bluetooth [15], or FlashLinQ [16].

The D2D protocol under investigation in thesis work is designed as an outband D2D system. Furthermore, inband D2D communications can be classified again by the spectrum that they utilize as:

- Underlay Inband D2D in which cellular and D2D users share the same spectrum,
- Overlay Inband D2D where D2D users have their dedicated frequency bands separated from cellular users.

The focus of this thesis work which is outband D2D can be further categorized into two as:

- Controlled Outband D2D where the interface which deals with the unlicensed spectrum is controlled by the cellular infrastructure. In this configuration, controlling functions are provided by BS while communications are is left to the second interface,
- Autonomous Outband D2D where all of the stages of the D2D communications are completed by the D2D interface.

The protocol which is designed and simulated in this thesis work is for an autonomous outband self-organized D2D network with absolutely no help form an infrastructure.

2.3 Examples of D2D Protocols

Potential use-cases of D2D communications are sorted in [17] as multicasting [18], P2P communications [19], video dissemination [20], machine-to-machine (M2M) communications [21], cellular offloading [22], sensor networks [23] etc. Moreover, the first attempt to implement D2D communications in a cellular network is FlashLinQ [16] which is offered by Qualcomm.

Wi-Fi Direct and Bluetooth are familiar D2D technologies where the communication takes place in the unlicensed industrial, scientific and medical (ISM) band. In this band, unexpected interference exists and cannot be controlled. To overcome this issue, FlashLinQ uses the time division duplexing-OFDMA (TDD-OFDMA) as in the case of LTE-A [7]. A brief summary of existing D2D protocols are given in the following subsections.

2.3.1 Wi-Fi Direct

Wi-Fi is a wireless networking technology [13] which enables devices to network in the 2.4 GHz or 5 GHz ISM bands. All wireless local area network (WLAN) products which are based on the Institute of Electrical and Electronics Engineers' (IEEE) 802.11 standards are called commonly as Wi-Fi.

In general, Wi-Fi divides the bandwidth into fourteen 22 MHz-wide channels which are separated by 5 MHz. Consequently, adjacent channels are overlapping. When a connection is established, Wi-Fi signals uses five of these channels which adds up to 25 MHz-wide bandwidth. To avoid interference, closely located networks should use non-overlapping sets of channel. By the same manner in a cellular network, frequency reuse is exploited by spatially distinct networks.

In conventional Wi-Fi networks, there are wireless APs to control the communication. These APs mainly function as a router between users. Nearly all of the Wi-Fi networks are formed in a mode called the infrastructure mode. In such a formation, APs are central nodes where other devices are connected to it, and all communications go through the AP. On the other hand, Wi-Fi Direct is a Wi-Fi standard which allow devices to establish connections without any help from an AP (i.e., P2P). In fact, Wi-Fi Direct devices decide which one will act as an AP during the connection establishment signalling as will be explained in Section 4.3.

To discover devices in close range, P2P Discovery phase which is equivalent to the inquiry process of Bluetooth is implemented. This control signalling is explained in Section 3.2. For the connection establishment, the Group Formation phase which is equivalent to the page process of Bluetooth is implemented. This formation procedure is explained in Section 4.3.

2.3.2 Bluetooth

Bluetooth is a wireless data transfer technique which works over short distances (i.e., up to 100 m), employs frequency-hopping spread spectrum (FHSS), and forms personal area networks (PAN) using the 2.4 GHz ISM radio band [15]. Classic Bluetooth (i.e., release 1.0) partitions data into packets and sends them in one of the 79 channels having a 1 MHz of bandwidth while the latest release of Bluetooth (i.e., v4, Low Energy) uses 40 channels having 2 MHz of bandwidth between 2402 MHz and 2480 MHz. Achievable data rates for the technology scaled from 1 Mbit/s up to 24 Mbit/s through the versions using different the modulation types. Transmit powers are reduced to 10 mW from 1 W through those versions.

Bluetooth works in a packet-based fashion by employing master-slave structure. These master-slave groups are called piconets and they are the lowest level groupings in a Bluetooth network. A master device can communicate with up to seven slaves in a piconet for classic Bluetooth. In the latest release, there is no limitation on the number of slaves in a piconet. All of the devices in a piconet use the same clock which is the master's. Moreover, the hopping sequence is also determined by the master node by using a very long pseudo-random sequence. Packet transmissions are based on master's clock, and by using this clock, slots are defined to have two clock cycle durations which equals 625 μs. Masters transmit their packets in odd numbered slots and vice versa. The communication packets can be 1, 3, or 5 slots long.

Multiple piconets in a Bluetooth network can form a scatternet. In wireless communications literature, a network is called as multi-hop if two users are able to communicate by using another user as a relay in the absence of a point-to-point connection. Through a scatternet, a multihop network structure can be formed and two piconets can communicate by the help of a shared node which is in both of the piconets.

For the discovery of neighboring devices, Bluetooth uses a procedure called the inquiry process. This device discovery method will be explained in Section 3.3. Moreover, Bluetooth also implements a page process for communication establishment which will be investigated in Section 4.4.

2.3.3 FlashLinQ

In [16], Qualcomm proposed a new D2D protocol for P2P ad hoc networks in the 2.4 GHz ISM band. Unlike the traditional P2P solutions such as Bluetooth and Wi-Fi, FlashLinQ works synchronously where this synchronization is provided from cellular networks. Moreover, FlashLinQ uses an OFDMA structure to take the parallel channel access advantage of it. This is a novel approach, and eases both device discovery and resource allocation algorithm designs. Consequently, this is one of the reasons we put more focus on FlashLinQ among existing D2D strategies.

The peer discovery approach of FlashLinQ which will be explained in Section 3.4 allows users to transmit their discovery information on their subcarriers. Using the orthogonality of subcarriers in an OFDMA structure, this allows devices to discover multiple nodes (i.e., a few thousand [16]) in a 1-km range. This long device discovery range is obtained by using rateless codes which enable discovery at low SNRs. Furthermore, since users do not use the same frequencies to send their discovery messages, the amount of interference is limited.

The distributed scheduling algorithm of FlashLinq also leverages the parallel channel access feature of OFDMA. As to be explained in Section 4.5, transmitter and receiver pairs transmit energy-level based single tone signals in the subcarriers assigned to their link after the pairing phase. By using a priority list and a signal-to-interference-plus-noise ratio (SINR) threshold, peers decide to yield transmission or not for the sake of high priority users by using those signals in a distributed fashion.

FlashLinQ is tested by both simulations and practical implementation, and it is shown to be superior over Wi-Fi from link rates, network throughput, and latency perspectives. We believe it is a prospering contribution to the D2D communication literature since it includes synchronous signalling, OFDMA structure, robust discovery, and relative long range of operability. On the other hand, we will propose our approach which took inspiration from FlashLinQ and is more suitable for networks in which a frequency hopping strategy is implemented and the packets are of relatively small size.

2.4 Motivation of the Thesis Work

In Wi-Fi Direct, communication control packet sizes are variable between 6 and 257 bytes [13], where the payload is said to be an Ethernet frame which can be of length between 42 and 1500 bytes [24]. Moreover, in Bluetooth [15], the size of data packets are said to be 1, 3, or 5 time slots or equivalently up to 2790 bits. In addition, some communication control packets are assigned to be of size 144 or 240 bits.

Although packet sizes are very small especially in the latest release of Bluetooth, we expect that these sizes will further decrease in the future for sensor networks of various applications, intelligent transport systems, proximity-aware advertising etc. In such D2D networks, distributed scheduling algorithms will be of significant importance from the efficiency and network throughput perspectives. In FlashLinQ, peer discovery takes 2% of the time. Moreover, FlashLinQ completes resource allocation for data transmissions (i.e., scheduling) every 2 ms, and data transmissions occur in a time of around 2 ms [16]. A rough calculation with the parameters given in [16] reveals that the amount of data which can be sent during a data segment is approximately 10000 bits. Furthermore, users are scheduled for a whole data segment.

On the other hand, it is very inefficient to schedule users to a data transmission period where 10 kbits can be sent if the packets that users have are very small in comparison. As a solution, we propose in Section 4.6 a time slotted data transmission and to schedule users in time slots. This will keep the transmission efficiency the same while exploiting the spatial reuse for multiple time slots separately rather than for a whole data segment.

In addition, FlashLinQ uses single tone energy-level based signals to accomplish interference management. Users extract the expected SINR information which they will face during data transmission, and use this information along with a priority list for yielding decisions which is explained in Section 4.5. However, a frequency hopping pattern is implemented in networks under consideration in this thesis work. The one-slot, single-tone, energy-level based probing of the channel conditions will not be enough to represent the channel quality between peers in such a case. Furthermore, in cases where the frequency selectivity of the physical medium is high, this approach can lead to erroneous decisions such that in the specific channel that the energy-level based signal is transmitted, there can be a deep fade. But since during the data transmission whole of the bandwidth will be used for orthogonal frequency division multiplexing (OFDM), frequency diversity can be enough for reliable communication. To eliminate these effects, we propose a received power estimation procedure in Appendix A and average this information in multiple peer discovery phases. By this long-term averaging of the received power from a user, we plan to determine the link quality between users effectively.

In Chapter 3, our contributions to the discovery stage will be explained and simulated. The performance of the received power estimation procedure will be of primary importance. In Chapter 4, our time slotted approach for the distributed scheduling will be explained and simulated. The performance of our algorithm for different payload amounts, different transmission efficiencies, and different path loss characteristics will be the most important results of this thesis work.

CHAPTER 3

PEER DISCOVERY

In this chapter, after motivating the peer discovery design of this thesis work, we will introduce the peer discovery mechanisms of some well-known D2D systems in Sections 3.2, 3.3, and 3.4. Then we will provide a detailed explanation of the OFDMA based peer discovery design, describe the discovery structure that we offer, and investigate its performance via simulations in Sections 3.5 and 3.6 respectively.

3.1 Motivation

In all of the commercialized cellular networks, UEs only establish communication with the BS and there is no direct connection between UEs [25]. Those BSs not only fulfill the relaying task between two end users or control the resource allocation in their cells, but also save mobile users and their limited batteries from the burden of searching for neighboring users, finding a route to reach their peers, and any other complex control procedure. Therefore, to set a connection up immediately if necessary, UEs periodically exchange signalling with the BS in their cells.

In a D2D system where there is no centralized node like a BS or some other infrastructure, peer discovery is necessary before devices arrange a D2D connection and start direct communication [12]. In the peer discovery phase, devices listen to the environment to find peers in the neighborhood. Every two P2P devices which identify each other are called D2D candidates. Once two devices discover each other, they can try to establish a connection in the following phases which are paging and connection scheduling, and will be described in Chapter 4.

Since peer discovery is the prerequisite for the connection establishment in wireless ad hoc and cellular networks, in every such network, devices use a type of device discovery phase to identify nodes in their vicinity and prepare to establish a connection. The most popular D2D technologies are the Bluetooth [15] and the Wi-Fi Direct [13]. In Bluetooth, there is the inquiry process to allow nodes to identify other devices. The inquiry process is a typical example of peer discovery and will be explained in Section 3.3. In Wi-Fi Direct, there is the device discovery phase which is equivalent to the Bluetooth's inquiry process and will be explained in Section 3.2. In addition, Qualcomm launched a new D2D technology which is called FlashLinQ [16] having an OFDMA based peer discovery which is investigated in detail in Sections 3.4 and 3.5.

This thesis work is inspired by FlashLinQ and offers some modifications to its protocol and frame structure to make it more suitable for networks where a frequency hopping strategy is employed as it will be clarified in this chapter, and to make it more efficient in networks with packets of small size as it will be introduced in Chapter 4.

3.2 Wi-Fi Direct's Device Discovery

The P2P discovery phase of the Wi-Fi Direct enables P2P devices to rapidly detect each other and prepare to establish a link [13]. This discovery procedure includes four principal components:

- Device Discovery where two P2P devices meet in a common channel and swap their identity information (e.g., name, type, ID),
- Service Discovery which is optional and enables P2P devices to find out upper-layer services before constituting a connection,
- Group Formation which is utilized to arrange a new P2P group and appoint one device as the group owner,
- P2P Invitation which is used to invite a P2P device to join present P2P group.

The device discovery of Wi-Fi Direct corresponds to peer discovery phase in our proposed D2D protocol and is a good example of a successfully implemented D2D discovery protocol.

Device discovery helps two P2P equipment find each other, meet in one of the channels of Wi-Fi, and transfer device information such as device type, name, and identity. For in-band discovery, discovery is achieved by implementing two phases (*Scan Phase* and *Find Phase*) of signalling and two states (*Listen State* and *Search State*) for devices to be in.

Device Discovery makes use of two types of frames to transfer device information: Probe request and probe response. These frames may include the addresses and types of the sending device, requested device type, information for timing synchronization etc. Two devices that exchanged probe request and response frames are said to be discovered by each other.

A P2P device which is not a part of any P2P group may use the listen state to be detectable. In the listen state, P2P device resides on a given listen channel which is selected from a pre-determined set (e.g., Channels 1, 6, and 11 in the 2.4 GHz ISM band). In this state, P2P devices only respond to probe request frames. Devices are only allowed to be in this state in the find phase. In the find phase, devices will stay in the listen state periodically and be available.

In the scan phase, P2P devices uses the scanning process defined in IEEE Standard 802.11-2012 [26], and listen to the environment by scanning all of the supported channels, collect information about nearby devices, and spot the finest probable potential channel to form a P2P group. The P2P device in the scan phase will not answer to a probe request frame.

In the find phase which is used to assure that two concurrently searching P2P devices meet on a common channel to set a communication, P2P devices cycle between two states: Listen and search. P2P devices in the listen state wait on a fixed channel and scan for the probe request frames to keep themselves discoverable. On the other hand, P2P devices in the search state transmit probe request frames on a set of channels. In summary, P2P devices in the find phase alternate between listen and search states which corresponds to waiting for or transmitting probe request frames respectively, and they try to converge on the same channel to identify each other by exchanging device information. To guarantee this convergence, devices randomize the time spent in each of the states. Moreover, the time to converge is minimized by keeping the number of channels small. An example timing diagram of a discovery process in Wi-Fi Direct can be examined in Figure 3.1.

In Figure 3.1, discovery process of two devices is illustrated. Firstly, both devices start with the scan phase where they listen to all of the channels of Wi-Fi to see whether there are nodes in their vicinity or not. After that, devices progress to the find phase where they periodically listen for the so called social channels for probe request frames in the listen state and send probe request frames in the search state. P2P device 1, sends a probe request frame in the channel 1 in his search state but doesn't get a response since device 2 listens for channel 6 for that time. But when device 1 changes its search channel to channel 6 and sends a probe request frame, device 2 responds and both devices discover each other.

3.3 Bluetooth's Device Discovery

In Bluetooth, inquiry process is implemented for peer discovery. It is important to note that the maximum frequency hopping rate of the devices during this process (also during the page process) is 3200 hops/s. Referring to [27], during inquiry process there are three substates for devices to be in, which are:

- Inquiry Substate which master devices switch to when they plan to discover new devices within range,
- Inquiry Scan Substate which slave devices go into when they plan to be discoverable and listen to the inquiry messages sent by inquiring devices,
- Inquiry Response Substate which slave devices enter to respond to the inquiry messages.

A device which needs to be discovered switches to the inquiry scan substate to be discoverable. Beforehand, this device can be in the standby state to save power by

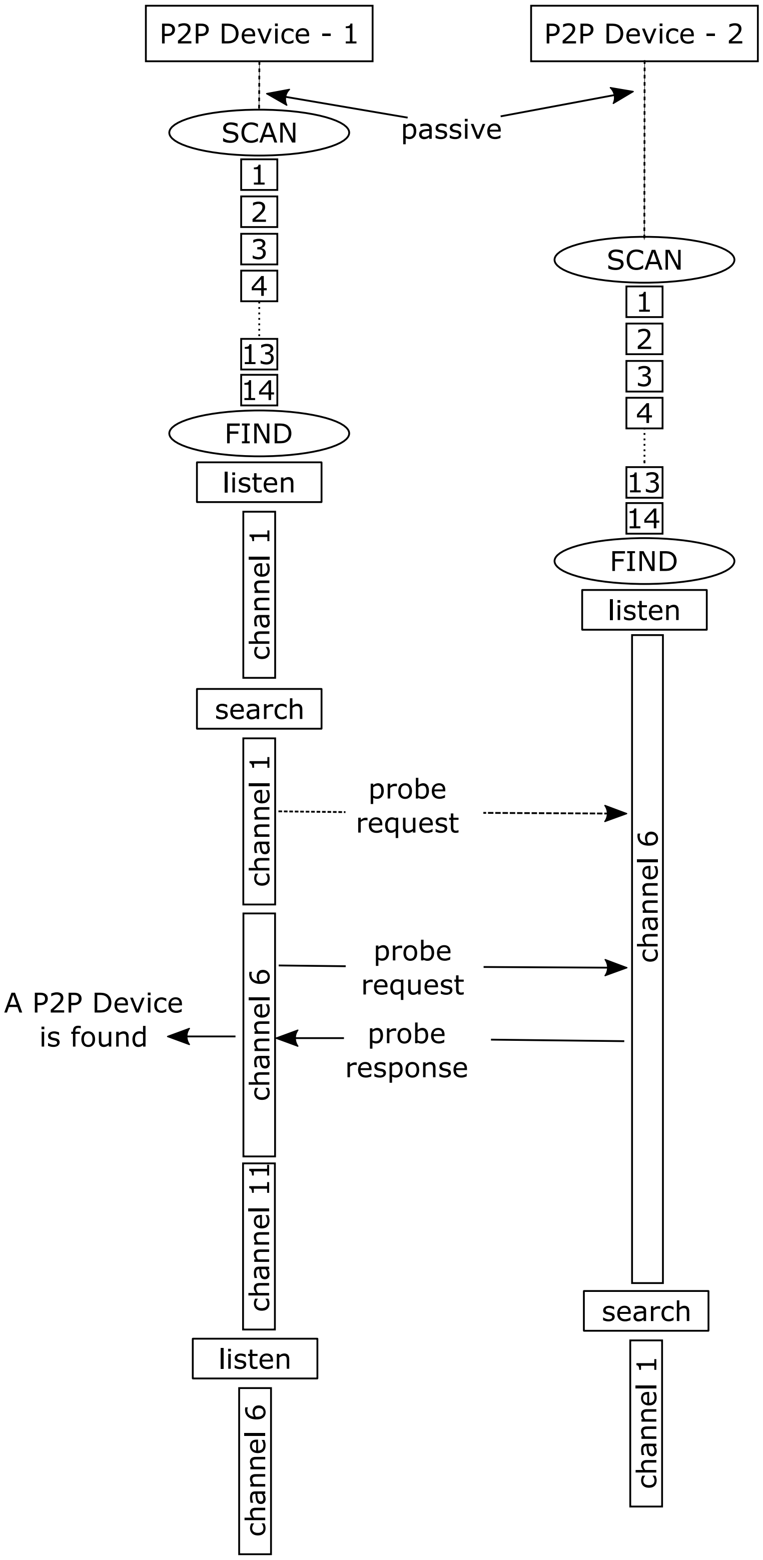


Figure 3.1: Example Wi-Fi Direct in-band device discovery procedure

being idle or can be in the connection state and be a part of another piconet. In the inquiry scan substate, devices (inquiry scanners) listen to the 32 inquiry scan channels among the 79 1 MHz-wide transmission channels used by Bluetooth protocol in the 2.4 GHz ISM band. Inquiry scanners change the inquiry scan channel at a very slow rate. They change the frequency that they listen in every 1.28 seconds and scan for that channel for 11.25 ms only. This slow hopping is designed to ensure maximize probability of correct reception of inquiry messages. During the scanning of length 11.25 ms, inquiry scanners wait for inquiry messages (inquiry requests) sent by inquiring devices. When they get a request, they go into the inquiry response substate, wait for 625 μs, and they send a response on the same channel. Inquiry responses include device addresses, the information about the clock phases, and the information about cyclic redundancy check (CRC). This information exchange is fundamentally used for synchronization between the master device and slave devices, and is enough to set a connection. Before inquiry scanners return to the inquiry scan substate, they wait for a random amount of time between 0 and 640 ms to avoid contention which can be caused by multiple inquiry responses in a close range. In addition to this random back-off, since the hopping pattern for the inquiry scan frequencies of different inquiry scanners are determined from their free-running clocks, the probability of contention is very small.

A device which needs to discover new devices in the neighborhood enters to the inquiry substate to send inquiry messages. In this substate, devices (inquirers) hop through all possible 32 inquiry scan channel frequencies in a randomized fashion at a higher rate than the regular transmission hopping. In each regular time slot of length 625 μs (which corresponds to a hopping rate of 1600 hops/s), inquirers send two inquiry messages on different frequencies in the beginning and in the middle of the time slot [28]. In the following time slot, they listen for the responses in those channels. This method doubles the frequency hopping rate of inquiring devices in the inquiry substate since they change the frequency every 312.5 μs (which corresponds to a hopping rate of 3200 hops/s). Inquiring devices use the information in the inquiry responses to page and to synchronize, and don't acknowledge these response messages.

The communication in the piconet between the master device and its slaves stops when the master device switches to the inquiry substate to discover new devices [29]. In addition, transmitting in all of the inquiry scan frequencies for a relatively long time creates a high amount of interference to the devices in the close range. To keep the transmission efficiency (e.g., the fraction of time that is used for data transmission) high, and to keep the interference created on the inquiry scan channels low, the time that is spent for the inquiry process should be as short as possible as a design rule. An example inquiry process timing diagram is given in Figure 3.2.

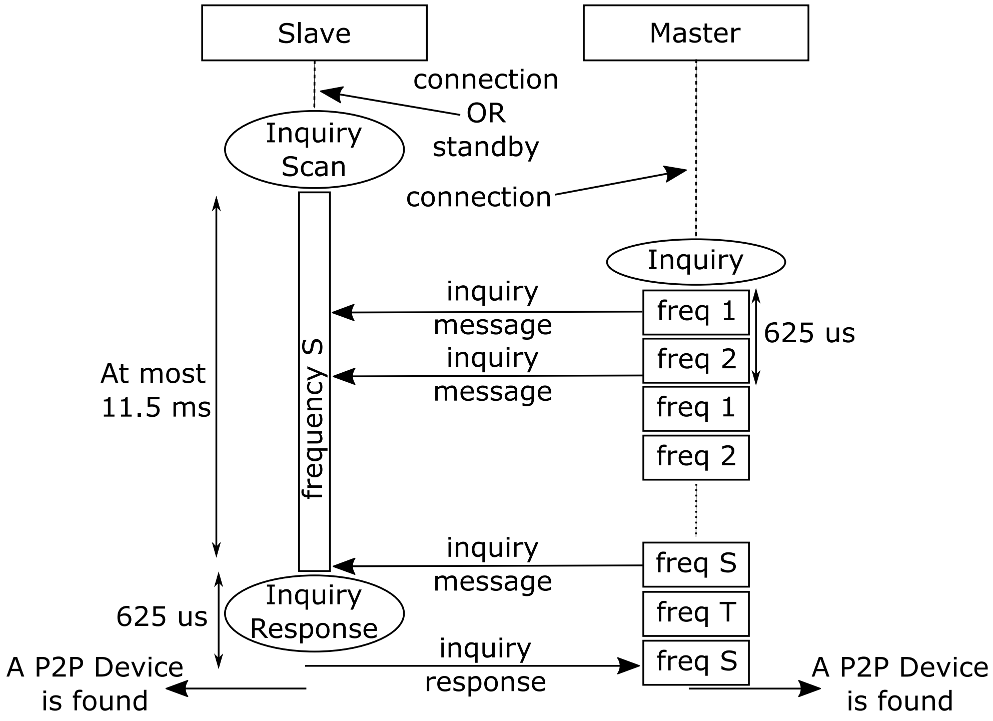


Figure 3.2: Example Bluetooth in-band device discovery procedure

In Figure 3.2, a device which has no established connection decides to be discoverable and goes into to the inquiry scan substate. Then a master device which is already in a piconet decides to discover new devices and switches to the inquiry substate. The potential slave device listens an inquiry scan channel for at most 11.25 ms. On the other hand the master device sends two inquiry messages on different channels in a time slot and listens to the answers consecutively which doubles its hopping rate. When the master device sends an inquiry message in the potential slave's inquiry scan channel (e.g., frequency S), the slave device sends an inquiry response in the same channel 625 μs later. These two devices have now discovered each other and will synchronize and form a connection during the page process.

3.4 FlashLinQ's Device Discovery

As we mentioned in Chapter 2, FlashLinQ operates in a synchronous manner unlike traditional ad hoc networks (e.g., Wi-Fi based ones, Bluetooth). Along with the OFDMA based structure, this synchronism enables designing a very simple and efficient peer discovery mechanism for FlashLinQ. In [30], it is outlined that FlashLinQ has a reservation based peer discovery method where all devices need to be globally synchronized. The timing structure of the device discovery phase of the FlashLinQ can be seen in Figure 3.3.

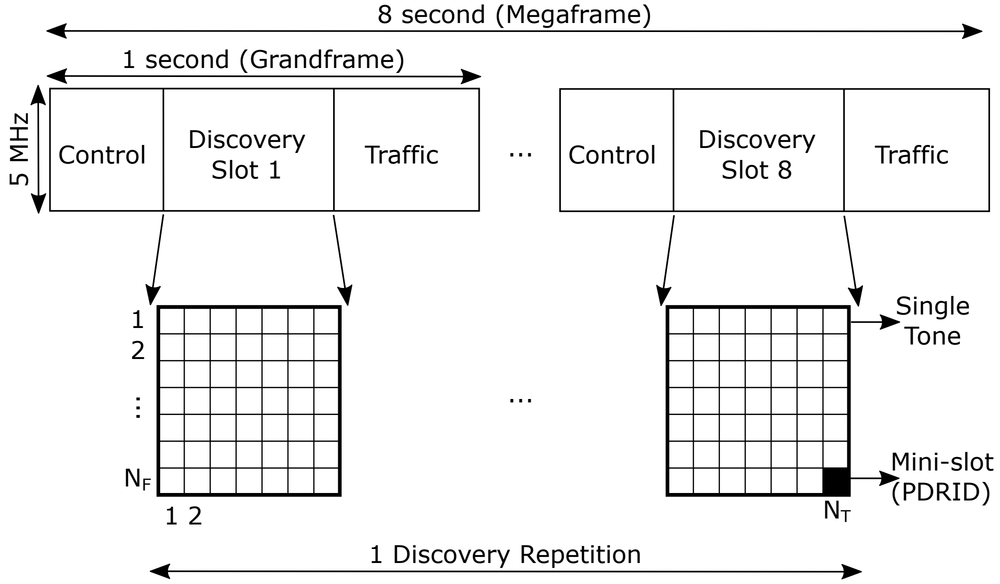


Figure 3.3: Peer discovery timing structure of FlashLinQ

In Figure 3.3, the time interval labelled as Control is the part where devices synchronize to each other. In [16], this synchronization is claimed to be achieved through currently installed code division multiple access (CDMA) based systems. It is indicated that the digital video broadcasting (DVB-H) in Europe or any other system which uses the frequency band under investigation can be utilized to accomplish the synchronization task.

Following the Control segment, a FlashLinQ grandframe continues with one of the discovery slots of duration 20 ms. Discovery repetition period of FlashLinQ is 8 seconds. Each repetition is divided into 8 slots, and these slots are distributed over 8 grandframes. In a discovery repetition, time and frequency resources are divided into N_T time slots and N_F single tone parallel channels respectively. Therefore, a peer discovery repetition is a time-frequency grid which consists of $N_F \times N_T$ elements (e.g., time-frequency slots) which are called mini-slots. As an illustration, the rectangle which is filled in black in Figure 3.3 is a mini-slot. These mini-slots are used to transmit discovery information and matched injectively to PDRIDs of devices. Devices transmit their discovery information in the mini-slots (e.g., single tone) matched to their PDRID, and listen to all of the channels for the remaining time slots of the peer discovery phase. For the case of FlashLinQ, N_T is set at 64 and N_F is set at 56. A mini-slot consists of 70 consecutive OFDM symbols and the data rate for the discovery is 70 bits per user every 8 seconds by using a convolutional code of rate 1/2 and quadrature phase shift keying (QPSK) modulation. Those bits represents the presence of the device and extracted from the unique identity of the device.

In a FlashLinQ system, when a user joins the network, the device should first find a proper PDRID to promote its existence to the neighboring devices. Since there is no centralized node in the network, devices select their own PDRIDs. To do this, there are two methods [31]:

- Random Selection

  In this approach, any D2D user trying to join the network selects one of the N_F x N_T PDRIDs randomly. In this kind of approach, even if there are unused PDRIDs in the network, there will be PDRIDs which are used by multiple users, and this will lead to an unnecessary interference in the peer discovery transmission phase.

- Greedy Selection

  In the greedy algorithm, users will first listen to the channels when joining the network, and select the PDRID where the received power is minimum (e.g., the highest probability of slot's being empty). This approach will ensure that the unused PDRIDs will probably be selected first. Moreover if all of the PDRIDs are selected, users will again will select the PDRID with the lowest signal power which implies the device using it probably is spatially distant and the amount of interference created to each other will be low.

One of the major problems that can be encountered in the PDRID assignment procedure is the case where multiple users try to join the network simultaneously and select the same PDRID. This case occurs more in the greedy assignment approach than the random assignment approach. In greedy assignment, if users that are closely located try to join the network at close-by time instants, the probability of receiving the lowest amount of power in the same mini-slot, and consequently the probability of selecting the same PDRID is high. This can lead to a congestion in the mini-slot, and to a discovery message collision. To prevent this problem, the greedy algorithm can be implemented in such a way that a mini-slot is randomly selected from the set of mini-slots which carry energy within the lower 5 of all mini-slots as suggested in [31].

Furthermore, due to the mobility of the users, devices having the same PDRIDs may come closer and increase the amount of interference that they cause on each other during peer discovery block repetition. To abstain from this problem, users can sometimes listen in their mini-slots rather than transmitting discovery information, and if there is another user which uses the same mini-slot and causes a dangerous amount of interference, users can restart the PDRID selection algorithm.

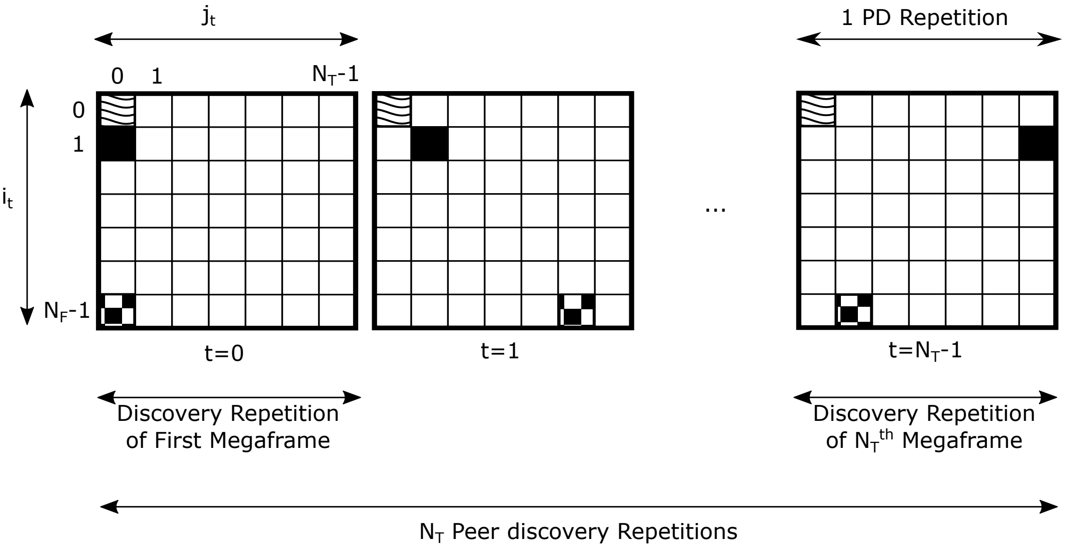
After selecting a PDRID, the devices transmit their discovery information in the mini-slots (e.g., single tone) assigned to their PDRID. This use of mini-slots to transmit discovery information brings out two issues to be solved on PHY layer when considered in conjunction with the fundamentals of frequency division multiplexing (FDM) based design. Those issues are [31]:

- Problems caused by the half-duplex nature of the transmission,
- Adjacent-channel interference (ACI) caused by extraneous power from an adjacent channel.

In a FlashLinQ network where the devices are half-duplex, two or more users may transmit their discovery information in the same time slot and may never hear each other if their PDRID does not change.

In addition, when spatially distant users utilize adjacent frequency channels for peer discovery, a D2D node may receive very different amounts of power in these channels due to the distance dependent nature of path loss characteristics. In such a case, the unintended spread of the transmitted spectrum out of the channel due to the limited capability of hardware or the adjacent-channel leakage due to the steepness of the pulse shape in frequency domain may suppress the power in the adjacent-channel if the path loss values differ significantly. This problem leads to the case where the receiving node cannot decode the discovery messages correctly even the signal in that channel is received.

To overcome those two problems, FlashLinQ allocates the time-frequency resources in a special manner [31], and changes the location of mini-slots which are assigned to users for every discovery repetition. This resource assignment method is illustrated in Figure 3.4:



j_t
1 PD Repetition
0
1
N_T-1
0
1
i_t
N_F-1
...
t=0
t=1
t=N_T-1
Discovery Repetition of First Megaframe
Discovery Repetition of N_T^{th} Megaframe
N_T Peer discovery Repetitions

Figure 3.4: Discovery resource locations for different discovery repetitions

In Figure 3.4, a complete discovery repetition which consists of $N_F \times N_T$ mini-slots repeats itself N_T times in the long run. As in Figure 3.3, these repetitions have a period of 8 seconds, and scattered over 8 grandframes of a megaframe. OFDMA is implemented for the frequency division, and every mini-slot which is filled the same way (e.g., PDRIDs) represents the resources that are used by a single device in repeating discovery blocks. As an example, a user having the PDRID corresponding to the slots which are filled with black will use the first channel in the first time slot for the first discovery repetition. In the second repetition, the aforementioned user will use the first channel in the second time slot. For all of the PDRIDs, the location of the peer discovery resource changes in each block. In [31], the time and frequency hopping solutions for the PDRID resource locations are formulated. The time and frequency indexes of a PDRID resources in different blocks can be written as:

$$i(t) = I, j(t) = J \otimes (I \oplus t), \tag{3.1}$$

where t is the index of the discovery repetition block $(t = 0, ..., N_T - 1)$, (i, j) are the indexes of a PDRID resource element in a discovery block, $\otimes$ and $\oplus$ denote multiplication and addition in Galois field $GF(N_T)$, and (I, J) is the canonical representation of the total $N_F \times N_T$ PDRIDs. In addition, N_T is selected such that $N_T > N_F$, and N_T is a power of a prime number. In Figure 3.4, frequency hopping is included for the sake of simplicity.

As a consequence of the equation (3.1) a theorem is given in [31] stating that two distinct mini-slots occupy the same time slot at most once for t from 0 to $N_T - 1$. By using this theorem, they claim that:

- Even if a user transmits in the same time slot with $N_F - 1$ other users in a discovery repetition, it is ensured that in the following $N_T - 1$ repetitions they will transmit in different time slots and be able to hear and discover each other by the help from the shift of resource element. As a result, the effect of the half-duplex problem is minimized.
- Even if spatially distant users utilize adjacent channels in a discovery repetition, it is ensured that in the following repetitions they will transmit in different mini-

slots and not cause that much interference on each other. As a result, the effect of ACI problem is minimized.

3.5 Proposed Modifications to FlashLinQ

As we explained in Section 3.4, FlashLinQ repeats its discovery phase in every megaframe and discovery repetitions are partitioned into 8 slots, and distributed over 8 grandframes. Every user transmits its discovery message in a single time slot (e.g., mini-slot) corresponding to its PDRID. As a result, users broadcast their presence once in a megaframe. This transmission is done in the time slot assigned to them which is in one of those discovery slots in one of the grandframes. This strategy can be clearly understood in Figure 3.3. Furthermore, they implemented the hopping pattern for the discovery resource locations in different discovery repetitions which is formulated in (3.1). Using this methodology, the locations of time-frequency slots where users transmit discovery messages are different in every discovery repetition which is illustrated in Figure 3.4. Consequently, ACI and half-duplex problems are avoided.

Our design approach to the discovery OFDMA grid is different than FlashLinQ's. In our proposal, a discovery repetition consists of M block as in FlashLinQ. But rather than completing discovery message transmission in a single time slot, users transmit their discovery messages in multiple time slots of short duration in our proposal. The time slot durations in FlashLinQ are 2.5 ms [16]. On the other hand, we are implementing a frequency hopping strategy of f_h hops/s which limits our time slot durations to the hop duration which is f_h^{-1} seconds. Furthermore, we divide discovery messages of every user into M parts, and modulate each one of them into the time-frequency slots in different discovery blocks in a frame. The proposed discovery structure can be seen in Figure 3.5.

In Figure 3.5, the time and frequency hopping method for resource locations which is formulated in (3.1) is implemented. In this figure, time-frequency slots which are filled with the same pattern belong to a single user. Each time-frequency slot has a duration of 1 hop (e.g., f_h^{-1} seconds), and consists of P parallel channels. For every

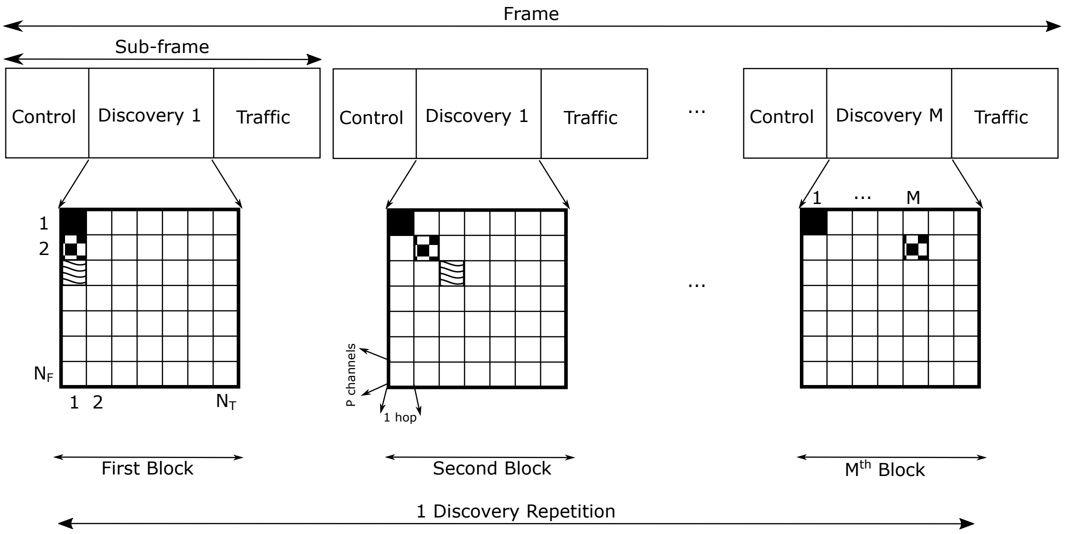


Figure 3.5: Proposed discovery structure

frame, the same peer discovery structure is utilized. In the following subsections, the benefits acquired from this modification are expressed.

3.5.1 PHY Layer Aspects

In Section 3.4, we explained the OFDMA based design for the peer discovery signalling in FlashLinQ. In FlashLinQ, when considering the allocation strategy of the time-frequency slots to users, only half-duplex transmission and ACI problems were taken into account. In this subsection, we will investigate the discovery phase OFDMA time-frequency grid design problem from the diversity perspective. Furthermore, FlashLinQ was designed as a protocol where new users may join and existing users may completely leave the network dynamically. In this thesis work, the protocol was designed and simulated for a deterministic set of users and the time-frequency resources are assigned to them statically. We will also explain the FDM based design strategy for such a network which modifies the method of FlashLinQ.

First of all, since the set of users which are connected to the network is deterministic, we do not need to implement a PDRID allocation strategy. The PDRIDs and the resource locations where users transmit their discovery messages are pre-assigned to the users as in Figure 3.5. For instance, the slot which is filled in black will be used by the same device for discovery message transmission purpose perpetually in that figure. Furthermore, this PDRID assignment is assumed to be tabulated an shared among users by an upper layer service. Using this mapping table of users to resources, devices can cross-check the reliability of discovery messages by comparing their content with their location in the OFDMA grid.

Secondly, there is no diversity in FlashLinQ's discovery structure. By modulating all of the discovery information of a user to a single time slot in every 8 seconds, they have no time or frequency diversity even in the existence of a frequency hopping strategy. For a device in such a system, there is no chance of recovering the discovery information of another user if the channel between them is in deep fade during that particular time slot (i.e., for 2.5 ms). Moreover, if the period of the peer discovery repetition is relatively large (e.g., 8 seconds in FlashLinQ case), this problem will lead to very high delays of discovery, increase the time that it takes for a connection to be established, and result in low QoS. Furthermore, even if the channel conditions are perfect, every user is not able to discover $N_F - 1$ users in a discovery repetition due to half-duplex transmission. In 3.4, we explained that FlashLinQ solves this problem by shifting resources locations for every discovery repetition. On the other hand by splitting the discovery message into M parts and implementing aforementioned shifting method in a discovery repetition as in Figure 3.5, we provide a possibility of discovering all users in a single repetition by using the help of diversity.

In Figure 3.5, we divide discovery messages into M parts and a discovery repetition into M blocks. The discovery blocks are distributed over a frame, and users transmit portions of their messages in different blocks in slots assigned to them. This approach is different than FlashLinQ's fundamentally. They also divide the discovery repetition into 8 blocks as in Figure 3.3, but every user still transmits its message in a single time slot. This kind of separation only scatters users over different discovery blocks, and is proposed to decrease the amount of overhead per time [31]. In Figure 3.5, one of the main purposes of the modification to the OFDMA grid is to mitigate the wireless channel fading problem. We obtain time diversity by modulating the discovery information of users into different time slots, and frequency diversity by implementing a frequency hopping pattern. As an example, a user is assigned to the PDRID slots which is filled in black in Figure 3.3. By using error correction coding, users will be able to extract information transmitted by this user from partially received signals even if some of the symbols is lost due to fading. In [32], Knopp and Humblet proved that the diversity order in block fading channels can be upper bounded as:

$$d^F \leqslant 1 + \left\lfloor F \left(1 - \frac{R}{log_2(|S|)} \right) \right\rfloor , \tag{3.2}$$

and the upper bound can be attained with specifically designed codes. In equation (3.2), F represents the number of independently fading blocks, d^F represents the maximum code diversity of a system where there are F independently fading blocks, R is the code rate, and $|S|$ is the size of the modulation constellation assuming the blocks are constructed using symbols from S. A designer can use the bound in equation (3.2) along with the bandwidth that is utilized and the efficiency constraint that must be satisfied to design an OFDMA grid and a PDRID allocating strategy. We considered this bound and designed the proposed discovery grid according to the expected channel conditions.

3.5.2 Received Power Estimation

Thus far in this chapter, we have discussed the main feature of the peer discovery phase in D2D protocols which is finding potential peers in the environment unaided by any superior control node like a BS. Exemplary systems are using discovery only for this purpose. On the other hand, in addition to its communication controlling functions, the peer discovery phase is fundamentally important in our protocol design for the proper functioning of the connection scheduling algorithm and interference management strategy unlike those proposals.

As it is summarized in [33], FlashLinQ obligates potential D2D transmitters and receivers to send single tone analog signals for interference management, in the time-frequency slots assigned to potential D2D pairs uniquely. The objective of those energy level based single tone signals is to let D2D receivers probe the received power from every D2D transmitter and determine the probable interference that will be caused on them during the data transmission phase. Afterwards, D2D receivers respond with another energy level based signal which broadcasts the received power from their peers to let D2D transmitters determine the amount of interference that they will impinge on receivers [16]. The energy level based signalling strategy of FlashLinQ which is implemented to manage uncontrolled interference in the operational bandwidth (e.g., connection scheduling) ahead of data communication part will be examined in depth in Chapter 4.

Our suggestion to solve the interference management problem and make the connection scheduling algorithm interference-aware is to predict the received powers from every active D2D user in the peer discovery phases. This strategy of long term averaging and keeping the track of received power levels of every user, and using this information in the connection scheduling phase of the signalling without requiring any dedicated signalling to estimate interference as in [16] and [34], is one of the main contributions of this thesis work, and has certain advantages.

One of the main advantages of our average received power tracking approach is the noise and fading-free nature of the received power information. Since the peer discovery messages of D2D users are transmitted over multiple frequency hops, and since D2D users are keeping track of the power levels for multiple peer discovery stages, it is essential to see that nodes will observe the quality of the links in many different channel conditions pertaining to their time varying nature. This method helps D2D users to estimate the received powers from other D2D nodes by averaging the effects of noise, multipath fading and shadowing out. This characteristic is a superiority of our protocol design over the systems offered in [16] and [34] where the interference estimation is done using energy level based signals transmitted in a single time-frequency slot. Especially in networks which employ frequency hopping strategies or in networks having a broad transmission bandwidth, this type of estimation is insufficient since it does not represent the channel conditions that will be experienced during the communication phase where different carrier frequencies and a wide bandwidth will presumably be utilized.

Another advantage of our interference estimation procedure is the assistance to routing protocol forming the accurate route. In multi-hop networks where the peer selection process solely depends on the lowest end-to-end delay and minimum hop-count constraints, transmission delay can be minimized and QoS requirement can be satisfied from the latency perspective. But these can happen only if the channel conditions are perfect and the multiple links which are connecting the peers are free of error which is not the case in a practical network. As it is explained in [35], proper routing should minimize the number of transmissions (including retransmissions) while finding the high-throughput paths. To achieve this, qualities of the paths and the interference among the links should be taken into account. The proposed algorithm allows D2D users to select their peers with the knowledge of the average channel condition of the link between them. By this method, in contrast to the schemes where the link quality and expected interference are estimated after the pairing is established and cannot be used by the routing protocol, users select peers using their routing algorithm and the average received power information. This strategy not only minimizes the probability of unsuccessful transmissions and the latency for a single data link, but also avoids the congestion problem caused by multiple retransmissions in the network.

We estimate the received powers from other devices by averaging the powers received in different time-frequency slots which is the maximum-likelihood (ML) estimation and can be formulated as:

$$\hat{P}_{rec,i} = \frac{1}{n}\sum_{j=1}^{n} P^{j}_{rec,i} - N_0, \tag{3.3}$$

where $\hat{P}_{rec,i}$ is the average received power from user i, n is the number of observations, j is the index of observations, $P^{j}_{rec,i}$ is the j^{th} received power observation from user i, and N_o is the noise power. The derivation of this ML estimate is provided in Appendix A.

The value of n is selected such that all samples in a discovery message frame over M frequency hops which is sent by a user are used in the estimate. In addition, we keep averaging the received power levels continuously. By this way, in case of a mobility, the received power estimates will change smoothly representing the effect of change of the physical channel between users. Finally, it is important to note that since we observe the received power in different frequency bands continuously due to hopping, received power estimates represent the average channel quality for all available operational bandwidth.

All in all, proposed method does not use instantaneous channel state information unlike FlashLinQ. Even though this may be considered as a disadvantage, average received power levels represent the quality of channel fairly for networks where frequency hopping is implemented. It will be seen in Chapter 4 that these levels are enough to provide interference-awareness to the system under consideration.

3.6 Simulations

In this section, we evaluate the performance of the proposed peer discovery protocol via simulations. The performance of the proposed algorithm is investigated through the number of discovered users, the average received power estimation errors, the distance to the discovered devices, and the SNR from the discovered users.

All of the simulations in this thesis work are implemented to their full extent on the physical layer. Firstly, data packets are formed. Then modulation, coding, interleaving, and scattering of symbols over different hops are completed. For every pair of users in the network, independent channels are realized for every frequency hop, and signals are transmitted through them with the inclusion of the noise. All of the signalling between every pair of devices in the network is considered. In addition, the effect of the different path delays on different sets of subcarriers are taken into account. After signal transmission, information over different hops are combined, deinterleaved, decoded, and demodulated. During this procedure, equalization is performed by using pilot symbols. Resulting data are extracted and fed to the MAC layer scheduling protocol directly. As a result, simulations include all possible errors in a typical telecommunications system. Basically, a multi-user network is simulated completely in both PHY and MAC layers.

Finally, we would like to state that in another set of simulations, we included the effect of carrier frequency offset (CFO) by adding a random amount of CFO to the clocks of every device independently. This random CFO had a uniform distribution between -240 and 240 Hz. This set of simulations are implemented to see the benefit of leaving some subcarriers empty to avoid the effect of CFO, and are not included in this thesis.

3.6.1 Simulation Setup

3.6.1.1 Network Environment

Simulations are run on a square shaped area of width $d = 1$ km. D2D users are distributed homogeneously over the two-dimensional area. A snapshot of a typical user distribution can be seen in Figure 3.6. N_T and N_F are set at 17 and 15 respectively. The number of total users is assigned to be $N_T \times N_F = 255$, which is the maximum value that the system can support.

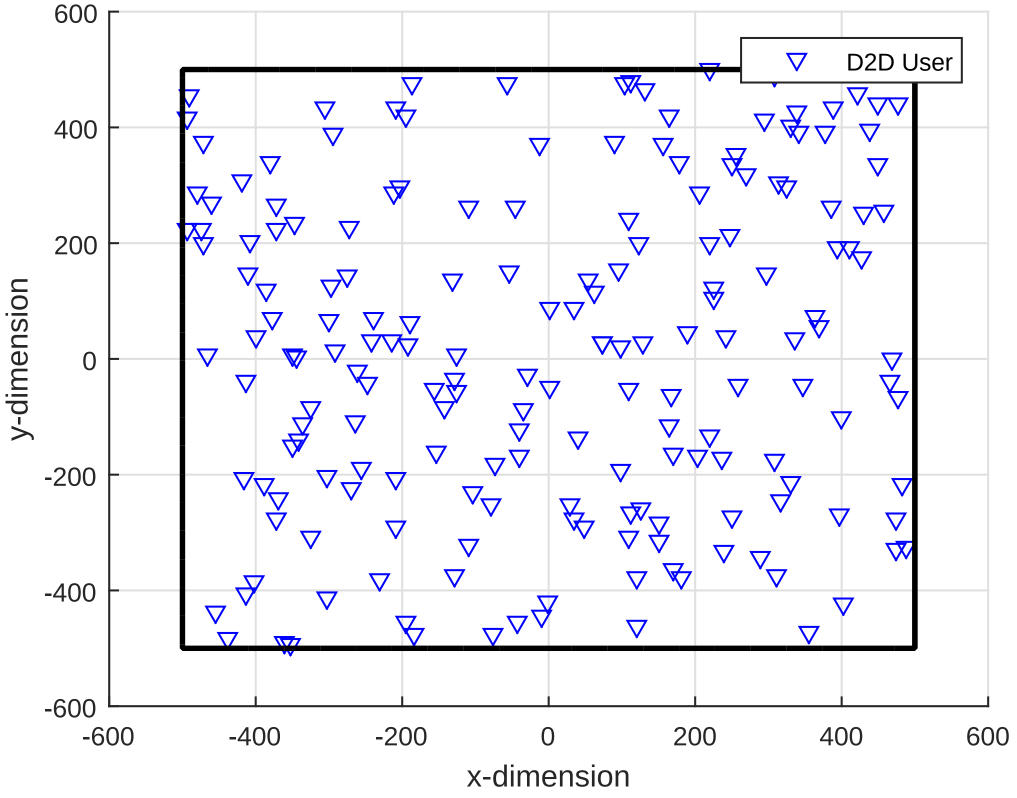


Figure 3.6: A snapshot of the simulated D2D network

Transmit power, P_T, is set to -40 dBW as in [36] and path loss is modelled by:

$$P^{k,l}_{rec,dB} = P^{k}_{tra,dB} + 10 \times log(d^{-\alpha}_{kl}), \tag{3.4}$$

where $P^{k,l}_{rec,dB}$ is the received power of receiver l from transmitter k in dB, $P^{k}_{tra,dB}$ is the transmit power of transmitter k in dB, d_{kl} is the distance between D2D users k and l in meters, and α is the path loss exponent. In simulations, α is set to 3 and 4. For different carrier frequencies, no modifications are applied either on the path loss expression or on the α values considering that the difference will be of insignificant importance.

3.6.1.2 PHY Layer Parameters

In Figure 3.5, the number of time-frequency slots assigned to each user is indicated as M and is an important design parameter. Selection of M depends on the amount of identification and control information bits to be sent in the discovery phase, and the time diversity that is needed to improve the reliability of the discovery messages. We also ensure this reliability by employing a frequency hopping scheme of f_h hops/s imitating the Bluetooth protocol [37]. This frequency hopping strategy puts an upper limit on the duration of a single OFDM symbol since we cannot transmit a single symbol in two or more hops. This limit (f_h^{-1} seconds) is selected as the OFDM symbol duration to benefit from the diversity derived from the hopping strategy by sending the discovery messages over maximum number of hops. Then the number of hops that is used by any user to complete its discovery transmission is also denoted as M.

As the channel realization procedure, statistical propagation model of cooperation in the field of scientific and technical research (COST) 207 is used along with Rayleigh fading and additive white Gaussian noise (AWGN). We implemented the suburban mode of the COST 207 channel which may be a typical scenario [38]. Although this model was developed for GSM links where one of the nodes is located several dozens of meters above the ground and the other is on the ground level typically, we have used it to have a figure of merit nevertheless, and compared the results in between. The power delay profile of the suburban COST 207 channel is given in Figure 3.7. Path loss equally affects all the multipath taps and independent fading is added to taps such that the overall path loss is equal to one calculated from the equation (3.4) in the long term average. In addition, the delays between devices are taken into account and used when modelling the time of arrival of signals to receiving nodes as in a real world practical implementation. The method of compensating different delays on different subcarriers due to multi user characteristic of OFDMA is explained in in Appendix B.

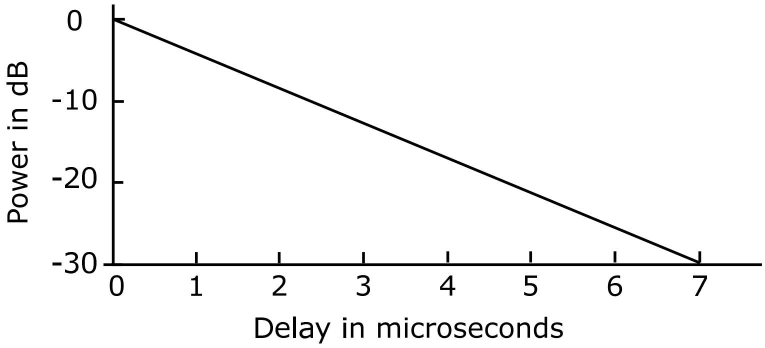


Figure 3.7: Power delay profile of the COST 207 suburban channel

Peer discovery signalling is implemented using the OFDMA structure over a 400-kHz of bandwidth. A frequency hopping strategy is realized as in Bluetooth technology so that every OFDM symbol passes through an independent channel. The hopping rate is adjusted to be $f_h = 2500$ hops/s and the duration of a single hop is $T_{hop} = 400$ μs. Every OFDM symbol has a duration equal to that of the frequency hop and has 150 samples. Cyclic prefix is set to be 25 μs or 10 samples equivalently. Although the maximum transmission delay in the network is 4.7 μs, we extend the guard accounting for residual asynchrony among nodes. In addition, we did not modelled clock timing offsets which can lead to processing a wrong portion of received signal. We considered that the aforementioned extended guard (e.g., CP) also gives enough time to correct these offsets. Frequency spectrum is separated into $N_F = 15$ distinct contiguous group of subcarriers. Subcarrier groups are assigned to users and consist of $P = 10$ subcarriers. Five of the assigned subcarriers remain empty to avoid ACI, and the other five is modulated with the discovery information. Third and the fourth subcarriers are loaded with pilots and channel equalization is done by using the channel gain from one of these middle subcarriers and the phase difference between them as explained in Appendix B. QPSK is utilized as the modulation type. A rate 1/2 convolutional code is implemented. The number of peer discovery blocks is $M = 8$.

The operating frequency is $f_c = 2.4$ GHz to be compatible with the aforementioned systems like Wi-Fi Direct in Section 3.2, Bluetooth in Section 3.3, and FlashLinQ in Section 3.4. Simulations are performed for $MC = 200$ different node distribution scenarios. Simulation parameters are chosen without loss of generality and can be scaled easily. A summary of the simulation parameters is given in Table 3.1.

Table3.1: Simulation Parameters for Peer Discovery Stage

N_T and N_F	17 and 15
Number of discovery blocks, M	8
Frequency hopping rate, f_h	2500 hops/sec
Number of subcarriers per user, P	10 (5 data + 5 empty)
Square shaped area width, d	1 km
Total number of users, N	255
Transmit power, P_T	-40 dBW [36]
Path loss exponent, α	3 and 4
Modulation	QPSK
Coding	Rate 1/2 Convolutional
Noise figure	7 dB
Noise power, N_0	-141 dBm
Bandwidth, BW	400 kHz
Fast/Slow Fading	Rayleigh/None
Carrier frequency, f_c	2.4 GHz
Length of cyclic prefix	25 μsec
Number of subcarriers	150
Multipath channel model	COST 207 Suburban
Monte Carlo Repeats, MC	200

3.6.2 Results

In the following figures, the red bars on black curves indicate standard deviations (i.e., σ). First, multiple (i.e., 200) cumulative distribution functions (CDF) are plotted. Then for a set of probability points, mean values and standard deviations of 200 samples of data are calculated. The black curves are the mean values and the red bars are 2σ long variation bars which are centred at mean values.

The discovery performance of the proposed protocol is illustrated in Figures 3.8 and 3.9. In Figure 3.8, the CDF of total number of discovered users for every device in every node distribution scenario for $MC = 200$ (i.e., 200×255 independent users) are plotted for different α values. It is easy to distinguish that as the path loss exponent α increases, discovery performance worsens. The mean values of the number of discovered devices of a user are 235, and 24 out of possible 254 for α values of 3, and 4 respectively. This significant decrease originates from the fact that as α increases, the path loss value decreases with distance more rapidly, and the links having the same SNR values become closer to each other. Since peer discovery is basically the correct reception of discovery message packets, and the correct reception probability is strictly dependent on the link SNR, it is fundamental to see that as α increases, the discoverable range for devices decreases. As an analogy, we can give the cellular network example. As it is thoroughly explained in [25], cellular systems are built upon the principle that the inter-cell interference caused by BSs is negligible, and frequency reuse is possible for sufficiently distant cells. Assuming that the users can only discover the devices in their cells in such a network, increasing α value corresponds to decreasing the cell sizes, and consequently decreasing the number of discovered users in a system where users are distributed homogeneously.

In Figure 3.9, performance of the discovery protocol is examined from a different perspective. Here we plotted the CDFs of total number of discovered users for each node topology for $MC = 200$ (i.e., 200 different CDFs of 255 different observations for different α values), and the aforementioned standard deviation bars are plotted around the means. This is done to illustrate that $MC = 200$ is enough to have results which are confined in a narrow deviation. As we mentioned, the red bars are of length 2σ and are centred at the mean value for that specific probability point. As it is clear, the characteristics are quite similar to the ones in Figure 3.8 which indicates probabilistic confidence.

In Figure 3.10, we plotted the CDF of distances between all pair of devices (i.e., 255×254 distance values) in a square area of width $d = 1$ km where 255 devices distributed homogeneously. We have also indicated the the maximum number of users that can be discovered by a device in the $\alpha = 4$ case on Figure 3.8. We see that, at most 40 users can be discovered by 99% of devices, which corresponds to 16% of possible discoverable users. In Figure 3.10 we see that in such a network 16% of pairs have a separation distance smaller than 260 m. For $\alpha = 4$, this distance corresponds

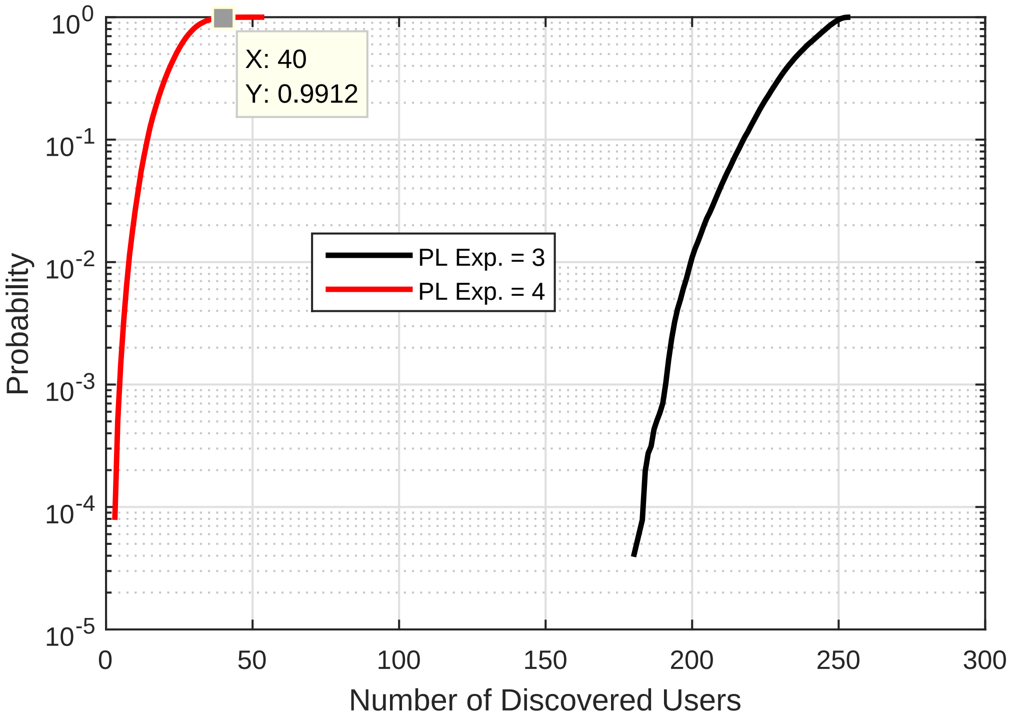


Figure 3.8: CDFs of total number of discovered devices for every device over 200 node distributions where $P_T = -40$ dBW, $\alpha = 3$ and $\alpha = 4$

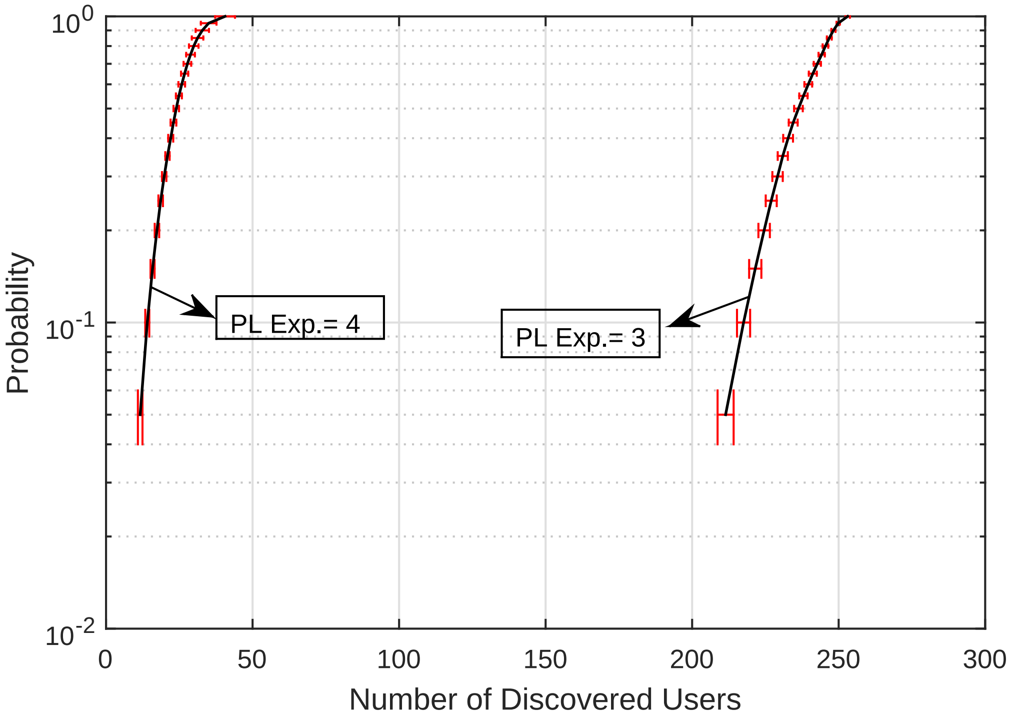


Figure 3.9: CDFs of total number of discovered devices for every device for 200 different node distributions where $P_T = -40$ dBW, $\alpha = 3$ and $\alpha = 4$, with standard deviation bars

to an approximately 5 dB of SNR which we use as a rough SNR requirement for discovery. When we plot the distances to the discovered users, this inference will be further justified.

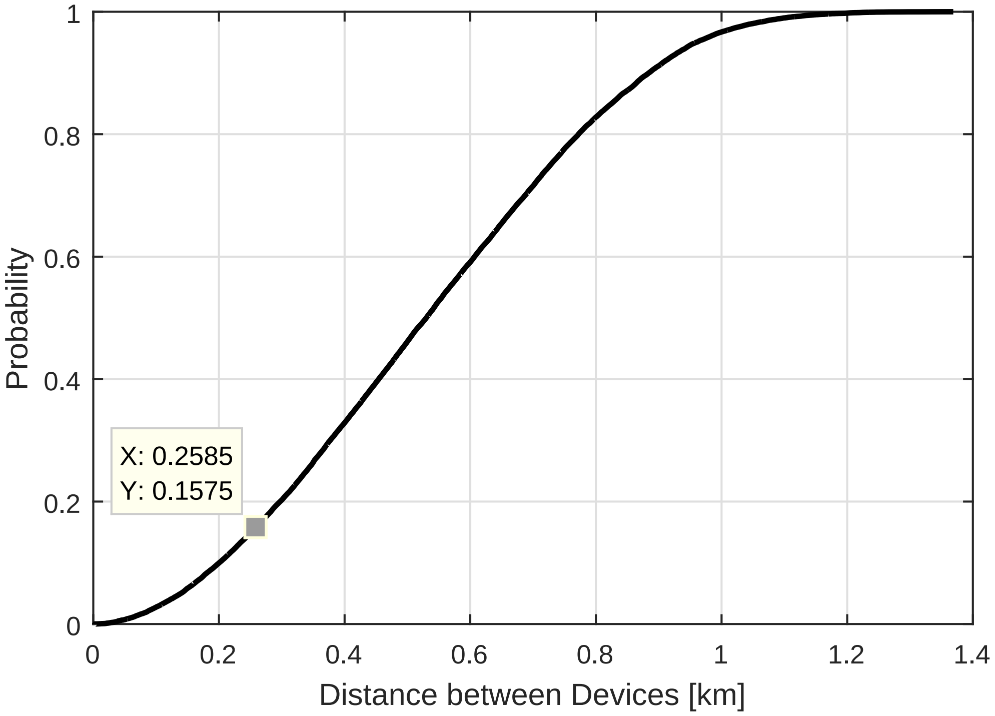


Figure 3.10: CDFs of distances between all possible device pairs for a node distribution for $N = 255$ and $d = 1$ km

In Figure 3.11, we plotted the CDFs of the distances to the discovered users for different α values (i.e., 200 different CDFs of 255×254 different observations for different α values) in the same manner as in Figure 3.9 including the statistical indicators. This performance characteristic is also important to compare our protocol with FlashLinQ which is claimed to have a performance of enabling devices to discover presence of other nodes within a 1-km range [16]. Figure 3.11 shows that the discoverable range for devices in our network increases with decreasing α which matches with our claims about Figure 3.8. It is more explicit now to assert that there is a minimum SNR and consequently a maximum distance requirement to be discoverable, and for low α values the distance requirement becomes larger to match the SNR minimum. We can easily see that for $\alpha = 3$, our system also has the capability of discovering users within up to 1-km range. This range decreases to 260 m for $\alpha = 4$. In addition, we see that the maximum discoverable distance matches our earlier deduction which confirms our inference that since peer discovery is basically a correct packet reception procedure, there is a strict relation with SNR, and consequently distance values. In fact, FlashlinQ uses rateless codes [16] to enable discovery at low SNR values. On the other hand we are using a rate 0.5 convolutional code with QPSK modulation. We claim that by using different types of coding and modulation, the minimum SNR to be discovered can be lowered, and hence the discoverable range can be increased.

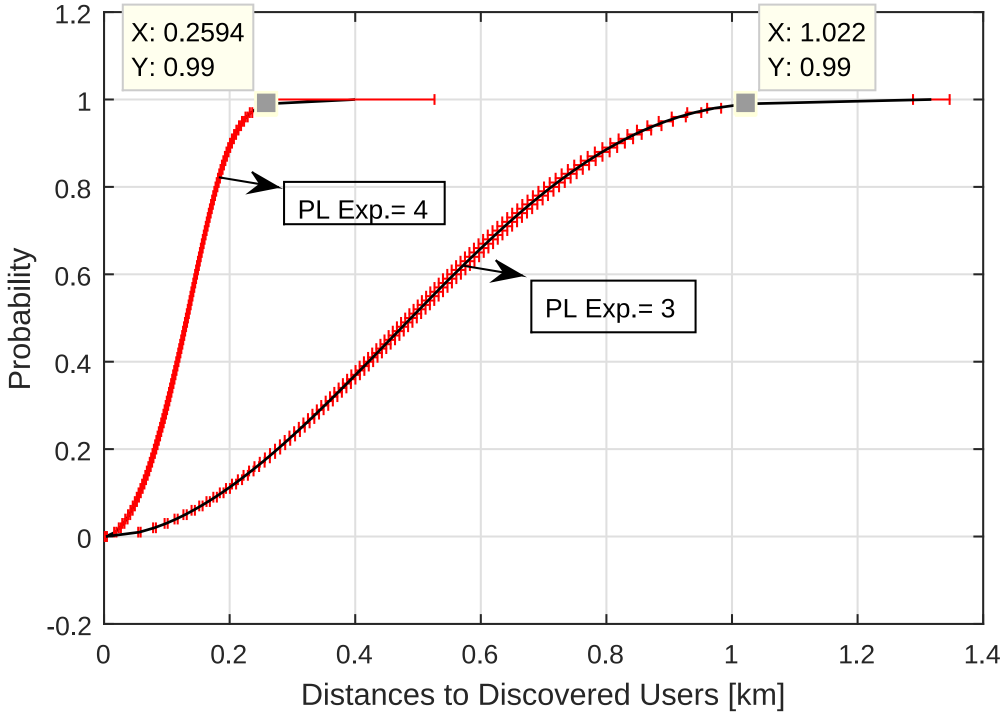


Figure 3.11: CDFs of distances to discovered devices for every device for 200 different node distributions where $P_T = -40$ dBW, $\alpha = 3$ and $\alpha = 4$, with standard deviation bars

In Figure 3.12, we plotted the CDFs of the SNRs from discovered users for $\alpha = 4$ values for all node topologies and all nodes. This figure is plotted to further justify our claim about the SNR requirement to be discovered which we roughly stated as to be 5 dB. We see in Figure 3.12 that for 99% of cases, the link between a device and another device which is discovered has an SNR higher than 4.75 dB. This finally confirms that the parallel channel access supplied by OFDMA allows devices to discover another devices if they are close to each other for a correct packet transmission. This range is determined from the modulation, and coding methods that we used, the block length of the discovery packet, the time and frequency diversities that we have, and channel conditions. As a result, a designer can increase the discovery range of the network under investigation by manipulating the parameters that can alter those.

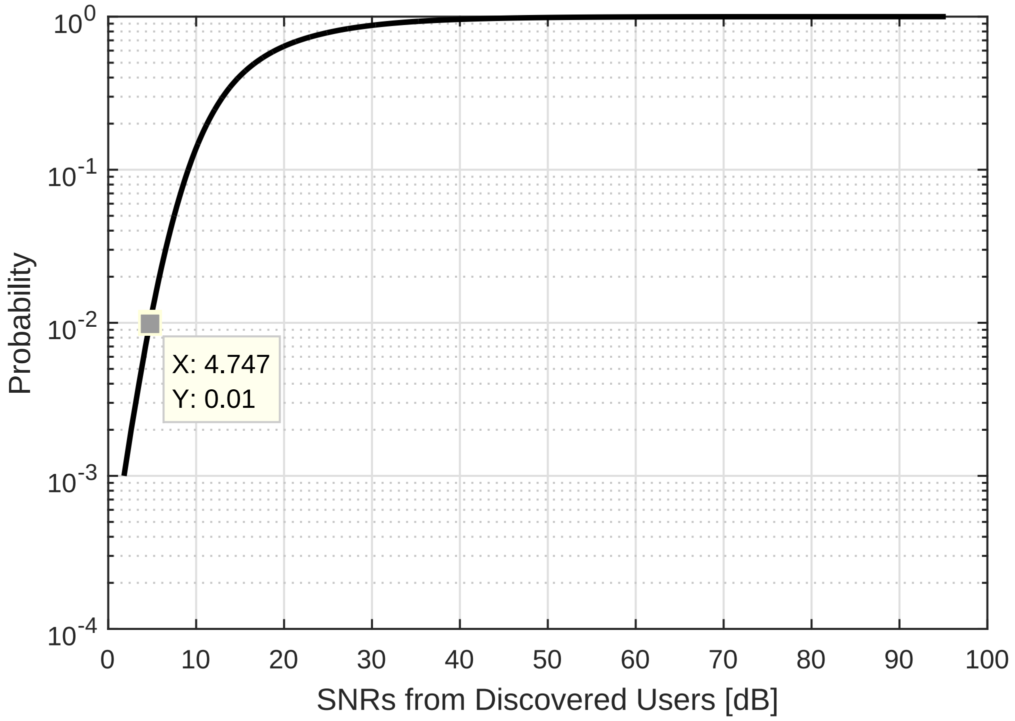


Figure 3.12: CDFs of SNRs from discovered devices for every device for 200 different node distributions where $P_T = -40$ dBW, $\alpha = 4$

Until now, we have shown that in an OFDMA system, peer discovery can be easily implemented by using the parallel channel access feature of OFDM, and the discovery performance is only related with the quality of the channel between devices which are trying to discover each other. Consequently, one can design a system and adjust the required approximate SNR to be discoverable by manipulating our coding, modulation, and diversity adjusting approaches with respect to the characteristics of the network. We also simulated different environments to illustrate the effect of electromagnetic attenuation on the discovery performance.

On the other hand, one of the main contributions of this thesis work is to make FlashLinQ system applicable to networks where there is frequency hopping. For this modification, we need reliable received power estimates which are free of errors due to noise, fading, and frequency dependency. We estimate the received power levels by using maximum-likelihood (ML) estimation which is explained in Appendix A. The received power information is obtained by averaging the power in the 5 data subcarriers over $M = 8$ discovery block as in(3.3).

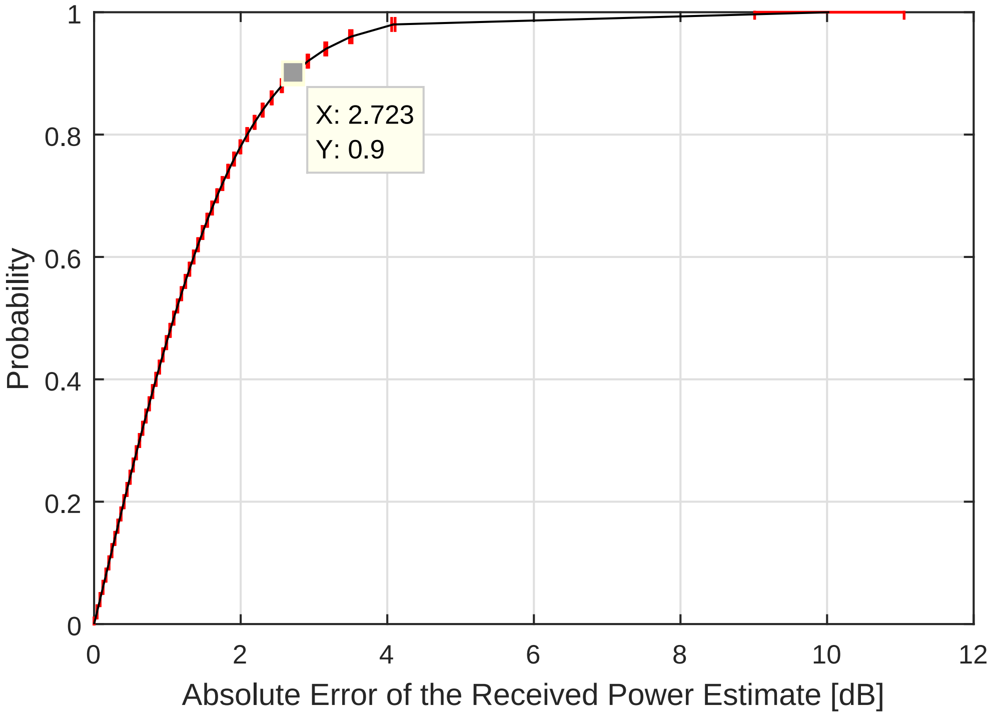


Figure 3.13: CDFs of received power estimation errors for every device for different node distributions where $P_T = -40$ dBW, $\alpha = 4$

The performance of the discovery protocol from the received power estimation perspective can be seen in Figure 3.13. In this figure, we plotted the absolute received power estimation errors which are formed by dividing the estimates by the theoretical values which are free of error. To be fair, we only included received power levels which should be above the noise level. It can be observed that the approach that we offered is quite successful to obtain received power information since 90% of all estimates are at most 2.7 dB away from the ideal value, and half of the errors are smaller than 1.5 dB. Furthermore, this gap will close as the peer discovery phases are repeated and the estimate converges to the ideal value. We would like to emphasize that this result is quite important for networks under consideration where a frequency hopping strategy is employed. In such networks, FlashLinQ's approach for interference management [16] works poorly since they implement an energy-level based signalling and power level probing which last for a single time slot prior to the connection scheduling. In networks with frequency hopping, this estimate includes channel state information about one channel and is taken from a single sample. On the other hand, our approach averages the received power value for all of the peer discovery phases to eliminate the effects of multipath induced and shadow fading, noise, and frequency dependency. It is easy to see that by implementing a single peer discovery phase where $M = 8$, we are able to get nearly error-free estimates.

CHAPTER 4

PAGING AND CONNECTION SCHEDULING

In this chapter, the paging and connection scheduling mechanisms of our protocol will be explained. These parts include the two of the main contribution of this work to FlashLinQ:

- Merging the page and connection (e.g., link) scheduling processes into a single phase,
- Dividing the data transmission phase into slots and scheduling links over multiple slots.

In the following sections, existing solution approaches for this problem will be examined. Then our proposal for both the algorithmic methodology and the physical layer design will be explained in detail.

4.1 Motivation

This thesis work focuses on and makes amendments to the D2D protocol which is offered by FlashLinQ in [16]. The need for a modification on this protocol arises from the dissimilarities between the characteristics of the network that it was offered to and the network that we have considered.

One of the main variations is the sizes of the packets in the network. FlashLinQ is offered without taking the packet sizes in the network into account, and completes the connection scheduling phase by assigning all of the time interval of the data transmission stage to users for all bandwidth. This approach is suitable for networks where the data exchange during the communication phase continues for a relatively long time once a connection is established. However in some networks such as sensor, public safety, intelligent transport etc. networks, communication is bursty in nature. Devices either exchange information or relay the data gathered from other users to a data center, and this information has a small size in practical networks.

This study introduces a new peer-to-peer network architecture in which by providing devices the flexibility of selecting data transmission time slot, spatial reuse is exploited for every time slot separately. In both [16] and [36], resource allocations are performed for the overall interval between two control periods. This strategy, decreases the transmission efficiency which we define as the fraction of time that is reserved for data transmission when users have relatively small amount of information to send. On the other hand, increasing control signalling repetition frequency by shortening the data transmission stage also reduces the transmission efficiency. Since nodes have a few number of packets in most of the networks under consideration, we propose a time slotted scheduling method to improve efficiency while keeping spatial reuse as our primary concern. The superiority of this approach to FlashLinQ in aforementioned networks will be demonstrated by taking efficiency into account when considering the network throughput.

4.2 Problem Definition

In a D2D system, devices should complete a device discovery procedure to connect to the network. After discovery, devices are eligible to establish a connection for communication. We call these devices as potential D2D peers.

Any potential D2D transmitter can determine a suitable receiver to form a D2D pair. This peer selection depends many parameters in the network and is fulfilled by an upper layer routing algorithm. In general, the routing algorithm makes an effort to reduce delays, save energy, and satisfy QoS while managing the interference. This interference management can be considered as a primary and a rough management to avoid the uncontrolled interference in a distributed network, and affects the peer selection. The secondary and extensive interference management is performed in the connection scheduling part.

In D2D communications underlying a cellular network, the routing protocol is also responsible for the mode selection feature [12]. Mode selection is basically the decision of communicating whether directly or via the cellular network. This process is out of the scope of this thesis work and will not be explained any further.

Having accomplished the peer selection, a potential D2D transmitter should page its corresponding receiver. This paging is to inform the receiver about the communication request. After the paging process, the peers which are eligible to have connections must be scheduled in time and/or frequency in order to satisfy QoS requirements while restricting the interference to the harmless range. The design of these paging and connection scheduling processes is an important issue and will be examined thoroughly in this chapter.

4.3 Wi-Fi Direct's Connection Establishment

Following the P2P discovery phase of the Wi-Fi Direct, devices run a group formation procedure to establish a communication. The group formation phase of Wi-Fi Direct corresponds to paging and connection scheduling (P/CS) phase in our proposed D2D protocol. This procedure consists of a number of phases such as group owner negotiation, provisioning, and connection setup [39].

When two devices discover each other, a P2P group must be constituted to start the conversation. This group formation starts with a group owner negotiation phase. A Wi-Fi Direct P2P group is another form of a piconet in the Bluetooth protocol. The master-slave relationship in the piconets resembles the owner-client relation in the Wi-Fi Direct P2P groups. To designate one of the P2P users as the owner of the P2P group, a three-way handshake mechanism is implemented. The three signalling types in the group owner negotiation phase are called as *request*, *response*, and *confirmation*.

Devices start the group owner negotiation phase as a part of the group formation procedure right after they found each other in the find phase of the device discovery which is explained in 3.2. For this negotiation signalling, devices use the channel that they exchanged probe request and response frames in the find phase. In this signalling, one of the devices sends the negotiation request message including the device information, channel list, and a number which represents the desire of the device to became the group owner. The device which hears the request frame, examines the content and replies with a negotiation response frame. This frame also includes a number for the same purpose. The device sending the number with higher value is chosen as the group owner. The details of this owner selection procedure can be found in [39]. Finally the device which gets the response frame, replies it with a confirmation frame which completes the owner-client assignment and the first phase of the group formation. An example group formation procedure is shown in Figure 4.1.

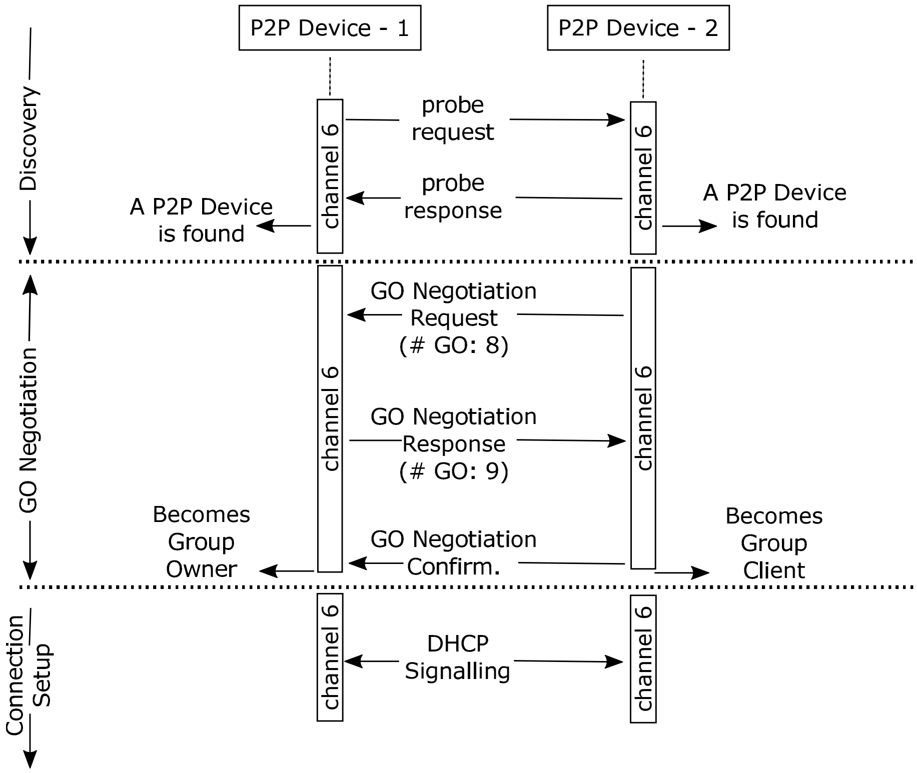


Figure 4.1: Example Wi-Fi Direct in-band connection establishment procedure

In Figure 4.1, two P2P devices which discovered each other in Figure 3.1 start negotiating that which one will be the group owner. They maintain this signalling in the channel that they met and discovered each other. One of the devices uses eight as the number for group owning, and the other uses nine. The latter device becomes the group owner after the confirmation packet. This three-way handshake establishes the link between peers and determines the roles.

After the group owner negotiation is completed, provisioning and address configuration phases of the connection establishment procedure start. These parts of the connection setup are more related to the higher layer services such as dynamic host configuration protocol (DHCP) which are not among the interests of this thesis work.

4.4 Bluetooth's Connection Establishment

In Bluetooth, page process is implemented for connection establishment. We should note that the maximum frequency hopping rate of the devices during this process is 3200 hops/s as in the inquiry process. In the page process, there are four substates for devices to be in, which are [27]:

- Page Substate which master devices switch to when they plan to form a new connection with a device,
- Page Scan Substate which slave devices go into when they want to listen to the page messages sent by paging masters to join a piconet,
- Slave Response Substate which slaves switch to respond to the page messages,
- Master Response Substate which paging masters get into to reply slave page responses with a Frequency Hopping Synchronization (FHS) packet.

Following the completion of the inquiry process which is explained in Section 3.3, a master device can decide to form a new piconet by establishing a connection with a device or can decide to add another slave to its piconet. For this purpose, master device starts the page procedure as a response to the inquiry response message that it received at the end of the inquiry process which is illustrated in Figure 3.2.

A slave device which wants to hear page messages to join a piconet, switches to the page scan substate. Beforehand, this device can be in the standby state to save power by being idle or can be in the connection state and be a part of another piconet. The page scan substate is as the same as the inquiry scan substate explained in Section 3.3. Devices (page scanners) listen to the 32 page scan channels and change the page scan channel at a very slow rate. They change the frequency that they listen in every 1.28 seconds and scan for that channel for 11.25 ms only. The frequency hopping pattern is extracted from the addresses of the slave devices to minimize the probability for two slaves to reply the page messages simultaneously. This slow hopping is designed to ensure maximize probability of correct reception of page messages since there is no exact synchronization between devices. During the page scan, slaves wait for page messages sent by potential masters. When they get a page, they get into the slave response substate, wait for 625 μs, and they send a page response on the same channel. Page responses include information for the communication setup. Before page scanners return to the page scan substate, they wait for an FHS packet from the master to align the hopping pattern and packet timing. In case of a FHS reception, page scanners acknowledge the master and enter the connection state to wait for traffic packets. Otherwise they return to the page scan substate.

Master devices (e.g., paging devices) switch to the page substate to connect to the potential slaves in the page scan. Beforehand, these devices can be in the standby state to save power by being idle or can be in the connection state and be masters of another piconets. In the latter case, masters can pursue their communication by periodically entering and leaving the page state. A paging device tries to encounter with a slave device's scan channel by repeatedly broadcasting a page message and listening in between transmit intervals until it receives a response from the slave [40]. These page messages include master device's access code. As in the inquiry substate, the master transmits this information in different hop channels where hopping rate is 3200 hops/s. The paging device roughly aligns its slot timing with the slave's using the inquiry response message. In the page substate, masters hop through all possible 32 page scan channel frequencies in a randomized fashion at a hopping rate of 3200 hops/s. As in the inquiry substate, they send two page messages at two different frequencies in a single slot and in the following time slot, they listen for the responses in those channels. When a master device gets a page response, it goes into master response substate and sends a FHS packet. This packet provides all information to construct the connection. After sending the FHS packet, paging device listens for acknowledgement and when it receives, it enters connection state to send traffic packets. An example page process timing diagram is given in Figure 4.2.

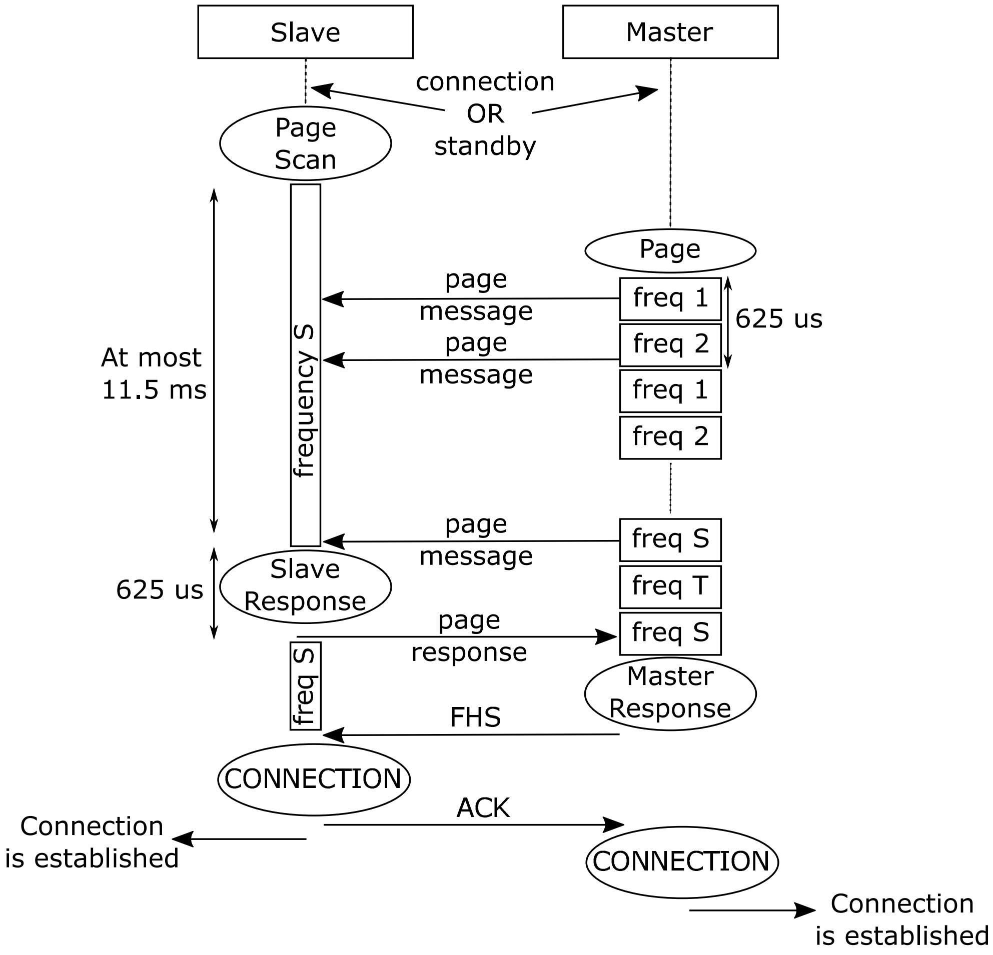


Figure 4.2: Example Bluetooth in-band paging procedure

In Figure 4.2, slave device decides to form a connection and turn into the page scan substate. Then a master device which decides to add a new device to its piconet and switches to the page substate. The slave device listens a page scan channel for at most 11.25 ms. On the other hand the master device sends two page messages on different channels in a time slot and listens to the answers consecutively which doubles its hopping rate. When the master device sends a page message in the potential slave's page scan channel (e.g., frequency S), the slave device sends a page response in the same channel 625 μs later. Master replies this page response with an FHS packet, and when the slave acknowledges it; both of the devices enter the connection state and start communication by exchanging traffic packets.

4.5 FlashLinQ's Connection Establishment

The general frame structure of the FlashLinQ can be seen in Figure 4.3 where the peer discovery parts are neglected for the sake of simplicity. Fundamentally, FlashLinQ protocol consists of 4 parts: Peer discovery, paging, scheduling, and data transmission. Peer discovery phase is described in detail in Section 3.4. The paging part is implemented for devices to inform each other about their desire to establish a connection and obtain connection ID (CID). The scheduling part includes connection and rate scheduling phases. The purpose of the connection scheduling is to determine the set of links which will communicate in the data transmission phase, and the purpose of the rate scheduling is to specify the physical layer aspects of their communication (e.g., adaptive coding and modulation) by sending wideband pilots and receiving channel quality indicators (CQI) [16].

The paging procedure is implemented after the peer discovery. In this procedure, devices exchange user identity information and it is repeated every second. In this phase, devices have to page each other to become peers, and acquire a CID number which indicates that they are potential candidates to participate in the connection scheduling signalling. For this CID acquisition process, an energy-level based approach is implemented resembling the PDRID assignment (e.g., greedy selection) which is explained in Section 3.4. This assignment generates a local uniqueness of CIDs. Local uniqueness is the situation in which the links which are closer to each other are unlikely to get the same CID number. This uniqueness will be significant when we explain the connection scheduling procedure of the FlashLinQ.

A CID is an index between 1 and 112 and corresponds to a pair of single tones in the OFDMA time-frequency grid which can be seen in Figure 4.3. One of these single tones resides in the Tx-block and the other in the Rx-block. The mapping between

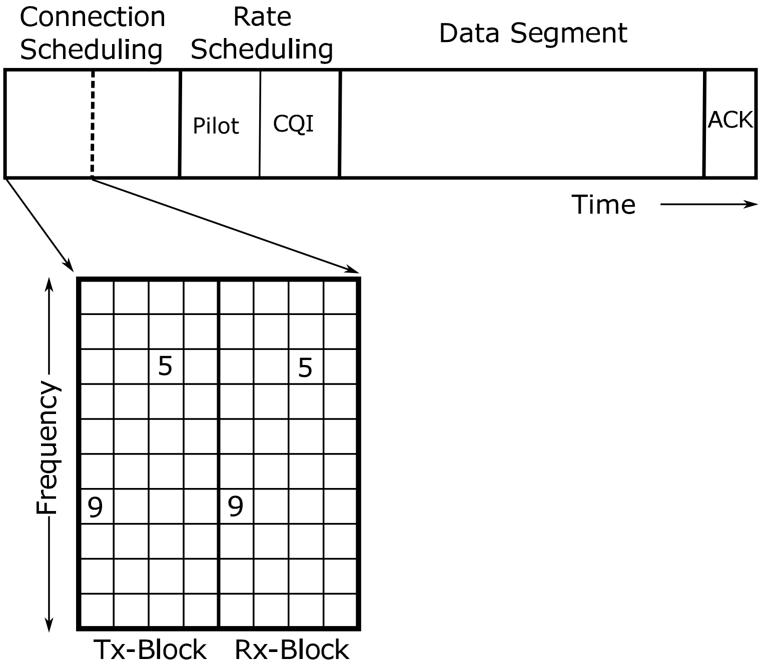


Figure 4.3: FlashLinQ frame structure

CIDs and the pairs of single tones can be deterministic or random according to the needs of the application. Both CID and single tone will be used interchangeably in this context.

The locations of the single tones in the time-frequency grid represent a priority assigned to the links. FlashLinQ implements a priority list such that as the time and frequency indexes of the location of a tone increase, the priority decreases while the frequency index having the higher priority. As an example, in Figure 4.3, locations of both of the tones for two links are marked. In this figure, the link which has the CID number of 9 has the higher priority with respect to the link with CID number of 5. In cases where the mapping of CIDs to the tone locations are random, a fairness is established between the links since the priority assignment is also randomized by its nature. On the other hand in cases where this mapping is deterministic, the CID acquisition procedure ensures this fairness. Since users probe the frequency channels to find the least congested one to select as the CID of their connection, time varying nature of the connectivity in the network provides fairness among the users.

After the completion of the paging procedure, a pair of devices (e.g., peers) are called as a link with an assigned CID. For a link to be scheduled and to establish a connection, a connection scheduling procedure is implemented. In this procedure, transmitters and receivers will broadcast energy-level based single tone signals in the locations assigned to their CIDs in the OFDMA grid. These signals are called the *direct power signal* and the *inverse power echo* in [16].

In the Tx-block which is indicated in Figure 4.3, transmitters of the links broadcast a single tone signal at a power which is the transmit power that they plan to use in the data communication phase. These signals are called direct power signals. Every potential receiver listen to the all of the channels to estimate the transmit powers of the potential transmitters. By using these power estimates, receivers calculate their expected received SINR in the data transmission phase. Receivers do this calculation by using only the power values of higher priority transmitters since lower priority ones should themselves waive the communication for the sake of higher priority links in cases where they cause a destructive amount of interference on their receivers.

In the Rx-block, receivers of the links broadcast a single tone signal at a power value which indicates the power that they received form their potential transmitters. These signals are called the inverse power echo signals and their power can be written as

$$P_{inverse,i} = \frac{K}{P_{T,i}|h_{c,i}|^2}, \tag{4.1}$$

where $P_{inverse,i}$ is the power of the inverse echo signal which is broadcast by the receiver of the link i, $P_{T,i}$ is the transmit power of the transmitter of the link i, $h_{c,i}$ is the channel gain of the channel of the link i, and K is a scaling constant which is known to all of the users. Every potential transmitter listens to all of the channels to estimate the received powers of the potential receivers from their corresponding transmitters. By using these power estimates, transmitters calculate the expected SINRs of only the higher priority receivers since lower priority receivers should themselves yield for the sake of their communication. Transmitters do this calculation by including the interference that they will cause on higher priority receivers to decide whether or not to yield.

To explain this two-way signalling mechanism we refer to the Figure 4.4.

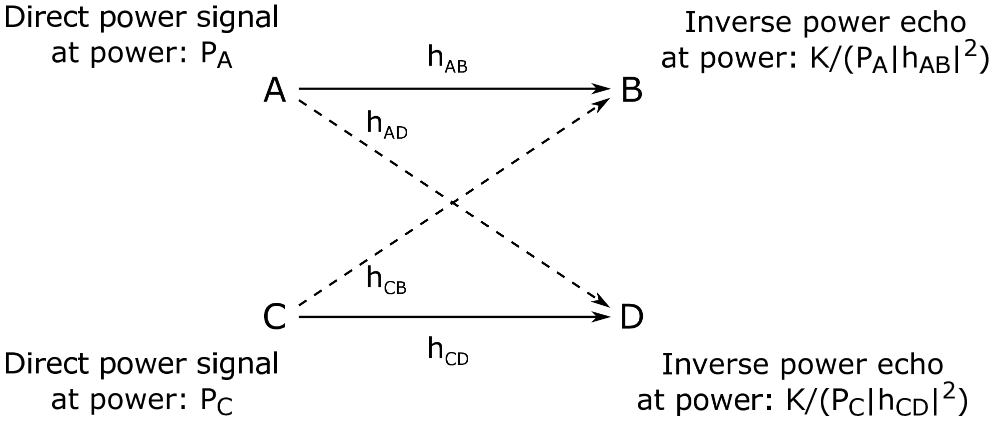


Figure 4.4: Connection scheduling signals in FlashLinQ

In Figure 4.4, there are two links which are denoted as AB and CD, and they have their CID numbers which can be translated into a pair of locations in the OFDMA grid for connection scheduling. In this configuration, link AB has a higher priority than link CD. In the Tx-block of the connection scheduling phase, both transmitters A and C broadcast their transmit powers P_A and P_C in their respective single tone locations, and both receivers B and D listen to estimate those power values which are including the channel gains h_{AB}, h_{AD}, h_{CD}, and h_{CD}. Since B is the highest priority receiver in the network, it presumes that there will be no detrimental interference in data transmission and lower priority transmitters will ensure this. However , D calculates the expected signal-to-interference ratio (SIR) value in the data transmission phase including the interference that will be caused by higher priority transmitters (A in this case). This calculation can be written as:

$$\frac{P_C|h_{CD}|^2}{P_A|h_{AD}|^2} > \gamma_{RX}, \tag{4.2}$$

where γ_{RX} is the receiver side yielding threshold [33]. If the estimate of the expected SIR doesn't satisfy the criterion in (4.2), D yields and doesn't send an inverse power echo signal. This process is referred to as Rx-yield procedure and is to impose the requirement that receiver D should predict acceptable SIR if scheduled [16]. This condition provides SIR-protection for the lower priority link.

After the Tx-block, receivers which didn't yield transmit the inverse power echo signals in the Rx-block, and both transmitters A and C listen to measure those power values which are including the channel gains h_{AB}, h_{AD}, h_{CD}, and h_{CB}. Since A is the highest priority transmitter in the network, it assumes that the receivers which will be damaged in the data transmission phase due to its transmission yielded. However, C estimates the expected SIR value of the higher priority receiver B. In this estimation, C includes its transmission power to see the effect of its transmission on B. This value can be written as:

$$\frac{P_A|h_{AB}|^2}{P_C|h_{BC}|^2} > \gamma_{TX}, \tag{4.3}$$

where γ_{TX} is the transmitter side yielding threshold [33]. The SIR value in (4.3) can be calculated by the inverse power echo signal which is sent by A at the power level which is indicated in (4.1). This value will be received including the channel gain h_{BC}. C can estimate the SIR value by taking the inverse of this value, dividing it by P_C and multiplying it by K. If this SIR value is below the threshold, transmitter yields, and if neither the transmitter nor the receiver yields, the link is scheduled. This condition provides SIR-protection for the higher priority link.

The connection scheduling phase is repeated over multiple rounds [16] as it can be seen in Figure 4.3, where the Tx-block and the Rx-blocks of the scheduling part are repeated multiple times. This is implemented to ensure the spatial packing of the links such that in consecutive rounds, new links can be added while the previously scheduled connections will continue. A multi-round scheduling example can be examined as in Figure 4.5.

In Figure 4.5, there are three links which are AB, CD, and EF. The priorities of these links are alphabetically decreasing. In the first round of connection scheduling, all of the transmitters will transmit their direct power signal in the Tx-block. In the Rx-block, all of the receivers will send inverse power echo signals, assuming that Figure 4.5 is representing relative locations of the links correctly, and no higher priority transmitter is closely located to a lower priority receiver, and the SIR conditions are satisfied at the receivers. In short, none of the receivers will yield.

At the end of the Rx-block of the first round, transmitters C and E will yield since

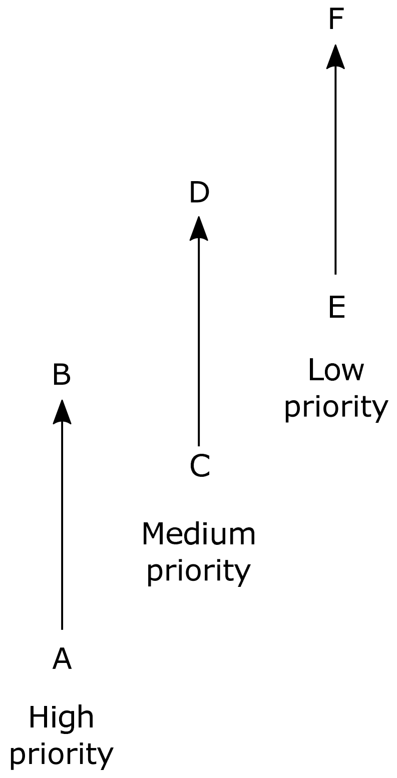


Figure 4.5: Multi-round scheduling example

there are higher priority receivers which are located close to them (B and D, respectively), and the SIR conditions at these transmitters are not satisfied. In the first round of the scheduling, only the link AB will be scheduled.

In the second round, only nodes A and B will transmit, and the other links will remain silent. After the second round is complete, transmitter E will realize that there is no higher priority link that it can cause too much interference on since link CD is yielded. In the third round of scheduling, node E will again participate in the scheduling procedure, and since link CD is not active, link EF will be scheduled. By this way, instead of implementing a single round and scheduling one link, we can schedule two links by using three rounds. This approach will pack the links and schedule them by using spatial diversity.

4.6 Proposed Method

One of the main contributions of this thesis work is to combine the paging and the connection scheduling phases of FlashLinQ [16] into a single section which we denote as P/CS. In this integrated phase, users both transmit their page messages to indicate their communication intentions as in the paging stage of the FlashLinQ, and check whether the SINR conditions to be scheduled are satisfied or not as in the connection scheduling stage of the FlashLinQ. Moreover, in the protocol that we are offering, devices are not computing those SINR values by using the energy-level based single tone signals which are sent during the connection scheduling phase in a single time slot, rather they use the long term averaged received power levels which are gathered during the peer discovery phase. This property of our algorithm is significantly important for networks where frequency hopping is employed since in such networks, these instantaneous energy-level based signals would not resemble the characteristics of the link between peers correctly.

Furthermore we are implementing a data transmission time slot selection feature to schedule users over multiple slots rather than a single one. This feature of our algorithm is substantial for networks where the packet sizes are relatively small, and scheduling users for all of the data transmission phase is inefficient.

The proposed frame structure, and both the receiver and the transmitter side scheduling algorithms will be explained in the following subsections.

4.6.1 Frame Structure

As we denoted in Section 3.5, we use the OFDMA grid which is illustrated in Figure 3.5 for peer discovery. In addition, we employ a frequency hopping strategy to create frequency diversity.

In the discovery OFDMA grid, the M peer discovery blocks can be located consecutively at the beginning of a frame. However, such a placement increases the time that it takes for a device to connect to the network since devices can only join the network by sending discovery messages and the discovery part is repeated once in a frame. To avoid this problem, we should let devices which become active connect to the network as soon as possible. This can be performed by decreasing the length of a frame and increasing the frequency of repetition of the peer discovery phase which will decrease the data transmission efficiency in turn. Alternatively, we can scatter M discovery blocks over a frame and decrease the maximum time that can take to connect to the network. We put this approach to use and name every division of a frame which includes only one discovery block as a *sub-frame*. The structure of a sub-frame can be seen in Figure 4.6.

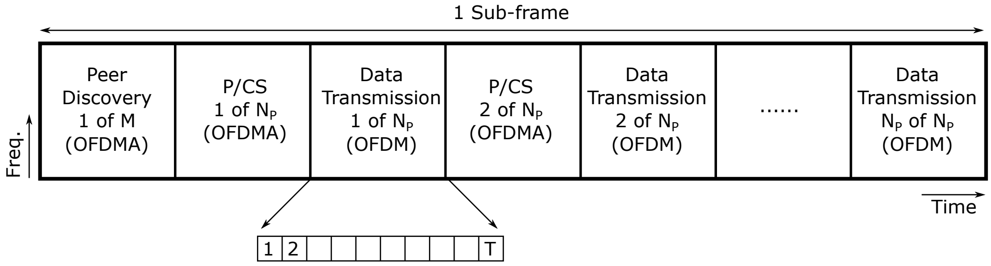

1 Sub-frame
Peer Discovery 1 of M (OFDMA)
P/CS 1 of N_P (OFDMA)
Data Transmission 1 of N_P (OFDM)
P/CS 2 of N_P (OFDMA)
Data Transmission 2 of N_P (OFDM)
......
Data Transmission N_P of N_P (OFDM)
Freq.
Time
1
2
T

Figure 4.6: Proposed sub-frame structure

Every frame consists of M consecutive sub-frames. At the begining of each sub-frame, there is a peer discovery block. In this block, devices transmit a piece of their discovery information in the time-frequency slots assigned to them, and monitor the environment to discover possible peers in their neighborhood. After the peer discovery phase, we propose a hybrid P/CS phase in which there are N_P P/CS blocks followed by N_P data transmission sessions alternately. Every data transmission session consists of N_S time slots. The detailed structure of a P/CS block is presented in Figure 4.7.

In the P/CS grid, every user has a paging slot which is located in one of the Tx-blocks. Two such paging slots are illustrated in Figure 4.7 by filling in with a wavy and a checker-board pattern. These paging slots are not uniquely assigned to users. Instead, there are N_{CS} users which are assigned to a paging slot and they will send their page messages in the same slot. This assignment is done assuming that the number of active users will probably be a small fraction of the number of all users for a given instant. In addition, we assume that for the cases where there are more than

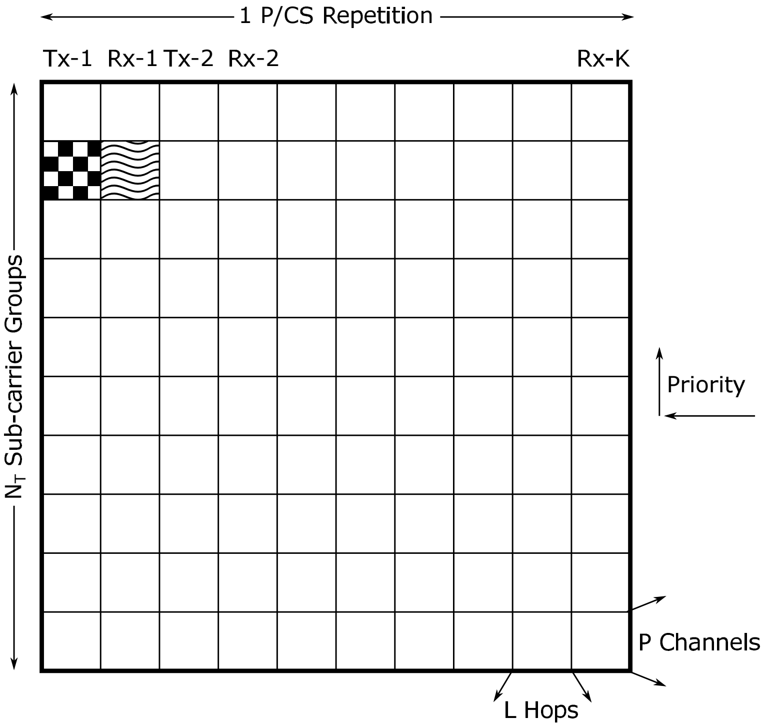


Figure 4.7: Proposed P/CS OFDMA grid

one user which broadcast their page messages in the same paging slot, the probability of one transmitter causing intolerable interference on another is also small due to spatial diversity. As an example, the system which has the P/CS block structure in Figure 4.7 can support a maximum of $N = N_F \times K \times N_{CS}$ users simultaneously. As in the PDRID case, this paging slot assignment to devices is deterministic and known by all users in the network. This lets devices which get page messages to cross-check the identity of the paging device by using the paging slot location.

4.6.2 Receiver Side Scheduling Algorithm

In every P/CS block, each idle node listens to the environment and forms a scheduling table. This table contains information about the links which are scheduled and about the links which yield to communicate in advance. Basically, nodes only update this scheduling table until they get a page or they decide to establish a communication with a peer. Devices which decide to establish a connection with their peers send a page message in the Tx-block in the slot assigned to them. For example, node A broadcasts its paging message in the slot filled with checker board pattern as illustrated in Figure 4.7. The page messages consist of IDs of both peers, data transmission time slot proposed by transmitter, and the average SNR value of the potential D2D link acquired from the prior peer discovery blocks.

After a Tx-block is completed, receivers which heard a page message from their potential D2D transmitters, run their side of the proposed scheduling algorithm using their scheduling table and the information that they collected during peer discovery phases. This algorithm is constrained by a minimum received SINR threshold for receivers, γ, and a priority list assigned to links. Priority levels are assigned to links according to the location of the paging slot of the transmitter side as in FlashLinQ. By changing the mapping of paging slots to the users continuously, we can provide fairness among the users. Furthermore, by mapping the paging slots to users with respect to their rank in the network, we can provide a variety of QoS levels to users.

The objective of the proposed scheduling algorithm on the receiver side is to find a suitable data transmission time slot. For this purpose, receivers first check the suitability of the time slot that is requested by the transmitter. If the requested time slot is appropriate for communication from the interference perspective, receivers send their page answers indicating that the connection is available to be scheduled in the requested data transmission time slot.

In cases where the requested time slot is not suitable for communication, receivers check whether there are other suitable time slots or not. If any of the other time slots is feasible to be scheduled in, receivers select one of them randomly, and send their page answers indicating that particular time slot. If there is no possible time slot to be scheduled, receivers remain silent.

Receivers determine the suitability of time slots by estimating the expected received SINR levels in case of communicating in these slots. SINR estimations are performed by using the average received power levels that are gathered during peer discovery phases, and by using the scheduling table. A time slot is eligible for a receiver to be scheduled, if the expected received SINR is above γ. In this estimation, receivers only consider the interference caused by transmitters with higher priority in the time slot under investigation.

In case of a positive decision, receivers send a page answer in the immediate Rx-block. Page answers consist of IDs of both peers, data transmission time slot selected by receiver, and the expected received SINR in that time slot. Receivers use the same frequency slot that they heard the paging message. For example, if node A sent its page message to node B in the slot filled with checker board pattern in Figure 4.7, node B responds in the slot filled with the wavy pattern.

4.6.3 Transmitter Side Scheduling Algorithm

After an Rx-block is completed, transmitters which heard a page answer form their potential peers, run their side of the proposed scheduling algorithm using their scheduling table, and using the information that they collected during peer discovery phases. This algorithm is also constrained by a minimum received SINR threshold for receivers, γ, and a priority list assigned to links.

The objective of the proposed scheduling algorithm on the transmitter side is to check whether it is possible to communicate in the time slot that is indicated in the page response or not. This check is implemented to avoid causing unfavourable interference on the higher priority receivers as in FlashLinQ. If the time slot which is indicated in the page answer is suitable for communication, transmitters decide to be scheduled and start transmitting traffic packets in the determined time slot. Otherwise, transmitters remain silent during the data transmission phase.

As explained in Section 4.5, if the interference that a transmitter will cause during the planned data transmission time slot is expected to reduce the received SINR value of any receiver with higher priority in the same time slot below γ, transmitter yields. To calculate the effect of its interference on other receivers, transmitters use the SINR information broadcast by the receivers in their page answers. This SINR value can be written basically as

$$SINR = \frac{P_r}{I + N}, \tag{4.4}$$

where P_r is the received power from the D2D peer, I is the amount of received power from other D2D transmitters scheduled in the same time slot, and N is the thermal noise power.

Transmitters use this SINR value and the average received power information gathered during the peer discovery phase to check if the SINR threshold is satisfied for all of the receivers with higher priority. Here we assume that the links are reciprocal which means that the channel gains between A and B in both directions are the same. A transmitter have to know either the I or the P_r term in (4.4) to include the interference that it will cause on a higher priority receiver in the SINR expression. Transmitters use the minimum SNR threshold for peer selection, SNR_{min}, which will be explained in Subsection 4.8.1.1 as the P_r term in (4.4) since it is the worst case scenario.

4.7 Performance Criteria

To investigate the performance of the algorithm that we suggested, we should use some metrics to translate its outputs to meaningful numbers with proper units such as network throughput in bits per seconds (bps), area efficiency in links/m^2, average SINRs and link distances for scheduled links, and the frequency reuse gain.

We consider an arbitrary D2D pair with index $j \in$ {1,2,...,N}. The SINR at the j^{th} receiver is defined as:

$$SINR_j = \frac{P_t^j d_{jj}^{-\alpha} |h_{jj}|^2}{\sum\limits_{k,k\neq j} P_t^k d_{kj}^{-\alpha} |h_{kj}|^2 + N_0}, \tag{4.5}$$

where h_{ab} describes the channel gain from node a to node b, P_t^i is the transmit power of the i^{th} transmitter, and N_0 is the AWGN power. In (4.5), k represents the indices of transmitters which are scheduled in the same time slot with j.

Our fundamental performance metric is the network throughput or the sum rate of the network normalized over the bandwidth. Assuming Gaussian codebooks, capacity achieving adaptive coding and modulation, and treating interference as noise as in [41] and [36], we can write the sum rate as:

$$\Lambda_c = \frac{T_{slot}}{T_{data} + T_{sch}} \sum_{i=1}^{n} log_2(1 + SINR_i), \tag{4.6}$$

where Λ_c is the sum of capacities of all scheduled links, and n is the number of scheduled connections in a single data transmission stage which consists of N_S time slots, T_{slot} is the duration of a single data transmission time slot, T_{data} is the duration of data transmission stage, and T_{sch} is the duration of the P/CS stage. Since connections are scheduled for single time slots, it is critical to normalize the sum rate over the time that is utilized by users.

To compare our algorithm for different N_S values and with FlashLinQ, we should somehow take the transmission efficiency into account when calculating the sum rate. In (4.6), the summation represents the sum of achievable rates of scheduled users in a data transmission stage which consists of N_S time slots, and its unit is bits per second per hertz (bps/Hz). The remaining multiplicative factor is the percentage of time that is utilized by scheduled users, and it is unitless. Consequently, the unit of the Λ_c is bps/Hz, and it gives us a metric for fair comparison.

In (4.6), we put no constraint on the rate since nodes have information about the link quality, and it is practically possible to achieve near the channel capacity. But we can take the finite constellation size constraint into account if need be, and write the sum rate as:

$$\Lambda_a = \frac{T_{slot}}{T_{data} + T_{P/CS}} \sum_{i=1}^{n} min\Big\{R, log_2(1 + SINR_i)\Big\}, \tag{4.7}$$

where Λ_a is the sum of achievable capacities of all scheduled links, and R is the maximum achievable spectral efficiency by a transmitter.

4.8 Simulations

In this section, we evaluate the performance of the proposed paging and connection scheduling protocol via simulations. The performance of the proposed algorithm is compared for different N_S values to investigate the effect of time slotted scheduling. Along with the sum rate, distances of scheduled links, actual SINRs of scheduled links, and reuse gain which we defined as the number of simultaneous scheduled connections in the area are used as performance indicators for different N_S values.

4.8.1 Simulation Setup

4.8.1.1 Peer Selection

The length of each link, which is the distance between its transmitter and receiver, is set to 20 m in [16], and 50 m in [36]. This type of setting does not really represent the real life situations where randomly deployed sensors or arbitrarily moving cellular users do not have the opportunity of finding a peer at a predetermined distance. To conceive the effect of link length variations on the accessible D2D connection range, transmitter-receiver separation distance is taken to be a uniform random variable in the interval [2,65] m in [34], and [0,100] m in [42] which offers an enhancement to [34]. We mimic this approach and set link lengths as a random variable which is uniformly distributed in [d_{min}, d_{max}].

The maximum distance that a link can be established is indicated as d_{max}. In cases where there is no interference in the peer discovery case, this maximum link distance circle also corresponds to a minimum SNR circle since we use (3.4) as our path loss model. As a result, the minimum link SNR, SNR_{min}, which is equivalent to the maximum link distance, d_{max}, can be written as:

$$SNR_{min} = P_{T,dB} + 10 \times log_{10}(d_{max}^{-\alpha}) - N_{dB}, \tag{4.8}$$

where $P_{T,dB}$ is the transmit power in dB, α is the path loss exponent, and N_{dB} is the noise power in dB.

4.8.1.2 Network Environment

We consider a D2D network in which there is no BS support and multiple D2D links are communicating using the same spectrum. The area where D2D users are deployed is denoted by a circle C of radius r. A snapshot of a typical user distribution with $r = 500$ m can be seen in Figure 4.8.

D2D transmitters are distributed over the two-dimensional area, C, homogeneously and their corresponding receivers are placed around them as explained in 4.8.1.1 by taking d_{min} and d_{max} as 20 m and 100 m respectively. To see the performance of the network for different utilization levels (e.g., activity factor), we deployed N_L links over the area and varied N_L between 16 and 256. Consequently, the maximum total number of users is $N = 512$.

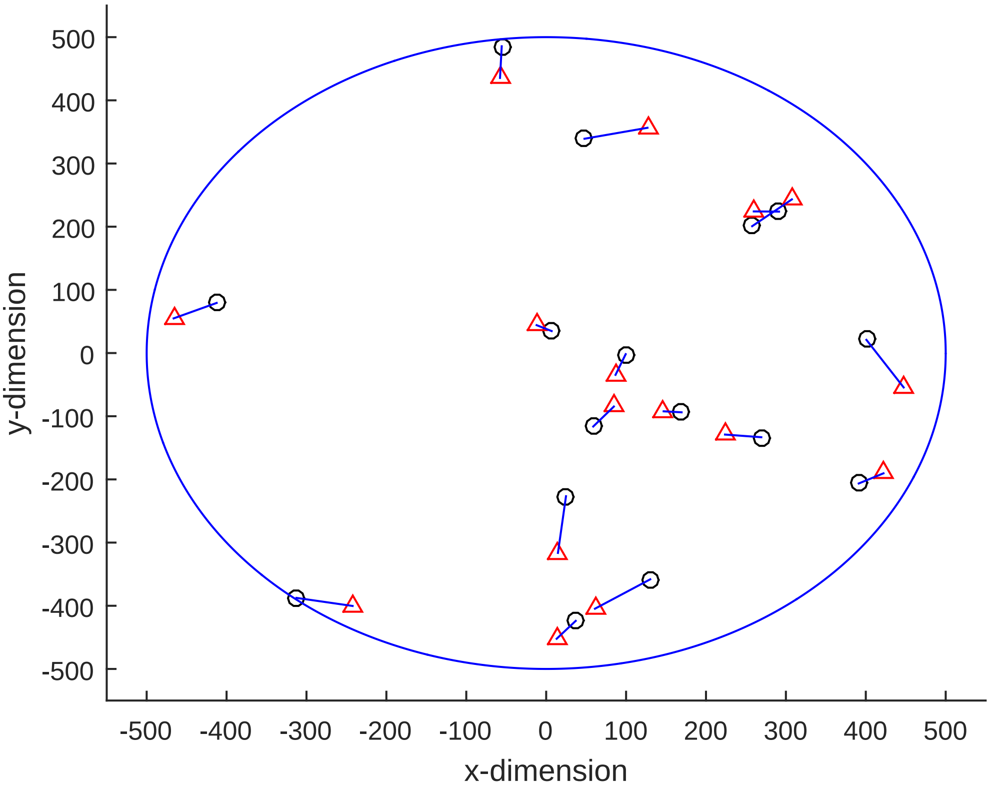


Figure 4.8: A snapshot of the simulated D2D network

Received powers estimations are assumed to be done perfectly over multiple peer discovery phases since we show in 3.6.2 that our estimation procedure is quite successful even in one peer discovery repetition. Path loss is modelled as in (3.4). Both 3 and 4 is used as α value. Transmit power, P_T, is taken to be -30 dBW as in Bluetooth Low Energy (BLE) [43]. In contrast to the case where we implemented the device discovery protocol by using -40 dBW of transmit power, -30 dBW is selected in this case to show the operability of the proposed system under different power constraints. In the literature, values between 0.1 mW and 100 mW are used typically as the transmit power of the devices in D2D systems as in [16], [36], and [43]. Finally, yielding SINR thresholds are set to $\gamma = 5$ dB. All of the results are obtained by averaging over 100 realizations.

4.8.1.3 PHY Layer Parameters

During the simulations, P/CS signalling is completely implemented over a bandwidth of $BW = 400$ kHz. For this signalling, OFDMA is utilized. Frequency spectrum is separated into $N_F = 15$ distinct group of sub-carriers. Sub-carrier groups are assigned to users and consist of $P = 10$ sub-carriers which leads to a total 150 subcarriers. Four of the assigned subcarriers remain empty to avoid ACI, and the remaining six is modulated with the discovery information. Third and the fourth subcarriers are loaded with pilots and channel equalization is done by using the channel gain from one of these middle subcarriers and the phase difference between them as explained in Appendix B. As the modulation scheme, 16-quadrature amplitude modulation (16-QAM) is used unlike the peer discovery where 4-QAM is utilized. This is because the links are guaranteed to have a high SNR. As the coding strategy, a convolutional code with rate 3/4 which is acquired from [44] is utilized. In addition, the delays between devices are taken into account and used when modelling the time of arrival of signals to receiving nodes. The method of compensating different delays on different subcarriers due to multi user characteristic of OFDMA is also explained in in Appendix B.

Since there are at most 512 users in the area, 9 bits are required to represent a receiver's ID in a paging message. In addition, we used 3 bits to quantize SNR/SINR information which is transmitted in paging messages. Moreover, the maximum number of bits required to indicate the time slot index is 3 since there are 8 time slots at most. Finally, since $N_F = 15$ and we set the number of page block repetitions, K, to 2, we should assign 30 paging slots to 512 users. This leads to the case where 18 users can share the same paging slot to transmit their paging messages. Consequently we require 5 bits to represent a transmitter's ID.

Eventually, paging messages consist of 20 bits. Taking the constraint length of the convolutional code into account, the coded and modulated page messages consist of 8 symbols which requires page blocks to continue for 2 hops where in each hop, every user is able to utilize 4 data subcarriers. As a result, L is set at 2 and the duration of the scheduling stage, $T_{sch} = 2KL$, becomes 8 hops.

The duration of data transmission time slots, T_{slot}, are taken to be 1,2, and 100 hops long to demonstrate the effect of transmission efficiency. Then T_{data} is $T_{slot} \times N_S$ hops, and T_{sch} is $2KL$ hops. The transmission efficiency for different N_S values is:

$$\zeta = \frac{T_{slot} \times N_S}{T_{slot} \times N_S + 2KL}. \tag{4.9}$$

User packet sizes for different T_{slot} values are 450, 900, and 45000 bits. The ultimate goal of this investigation of network throughput performance of the proposed algorithm for different number of data transmission time slots, N_S, is to justify our claim that as the packet sizes in the network get smaller, it is more convenient to accomplish scheduling over multiple time slots.

Lastly, a frequency hopping strategy is realized as in Bluetooth technology where $f_{hop} = 2500$ hops/s so that every OFDM symbol passes through an independent channel. Scheduling is performed for data transmission sessions which consist of $N_S \in \{2, 4, 8\}$ time slots. Simulations are repeated $MC = 500$ times. A summary of the parameters used in P/CS signalling is given in Table 4.1. Note that parameters are selected without loss of generality and can be scaled to satisfy system requirements.

Table4.1: Simulation Parameters for P/CS Stage

Circularly shaped area radius, r	500 m
Number of deployed links, N_L	16, 32, 64, 128, and 256
Link distance range, $[d_{min}, d_{max}]$	[20, 100 m]
Maximum total number of users, N	512
Frequency hopping rate, f_{hop}	2500 hops/sec
Multipath channel model	COST 207 Suburban
Fast/Slow fading	Rayleigh/None
Carrier frequency, f_c	2.4 GHz
Bandwidth, BW	400 kHz
Transmit power, P_T	-30 dBW [43]
Path loss exponent, α	3 and 4
Modulation	16-QAM
Coding	Rate 3/4 Convolutional
Noise figure	7 dB
Noise power, N_0	-141 dBm
Number of sub-carriers	150
N_F and N_{CS}	15 and 18
Number of subcarriers per user, P	10 (6 data + 4 empty)
Length of cyclic prefix	25 μsec
Number of data transmission time slots, N_S	2,4, and 8
Duration of a time slot, T_{slot}	1,2, and 100 hops
Data transmission packet sizes	450, 900, and 45000 bits
Duration of the data transmission stage, T_{data}	$N_S \times T_{slot}$
Number of scheduling block pairs, K	2
Number of hops in a scheduling block, L	2
Duration of the scheduling stage, T_{sch}	8 hops
SINR yielding threshold, γ	5 dB
Monte Carlo Repeats, MC	500

4.8.2 Results

In Figures 4.9, 4.10, and 4.11, we plotted the overall network throughput normalized over bandwidth for various cases. These sum rate values include the effect of transmission efficiency as defined in (4.9). Furthermore, we plotted the standard deviations, σ, for every data point which are averaged over 500 realizations. The length of the standard deviation bars are 2σ which is symmetrical around the mean. In these figures, $\alpha = 4$ and the $P_T = -30$ dBW. The number of links which are deployed over the area is in the range between 16 and 256.

At first glance, we see that the performance from the network throughput perspective improves as the number of data transmission time slots, N_S, increases for the cases

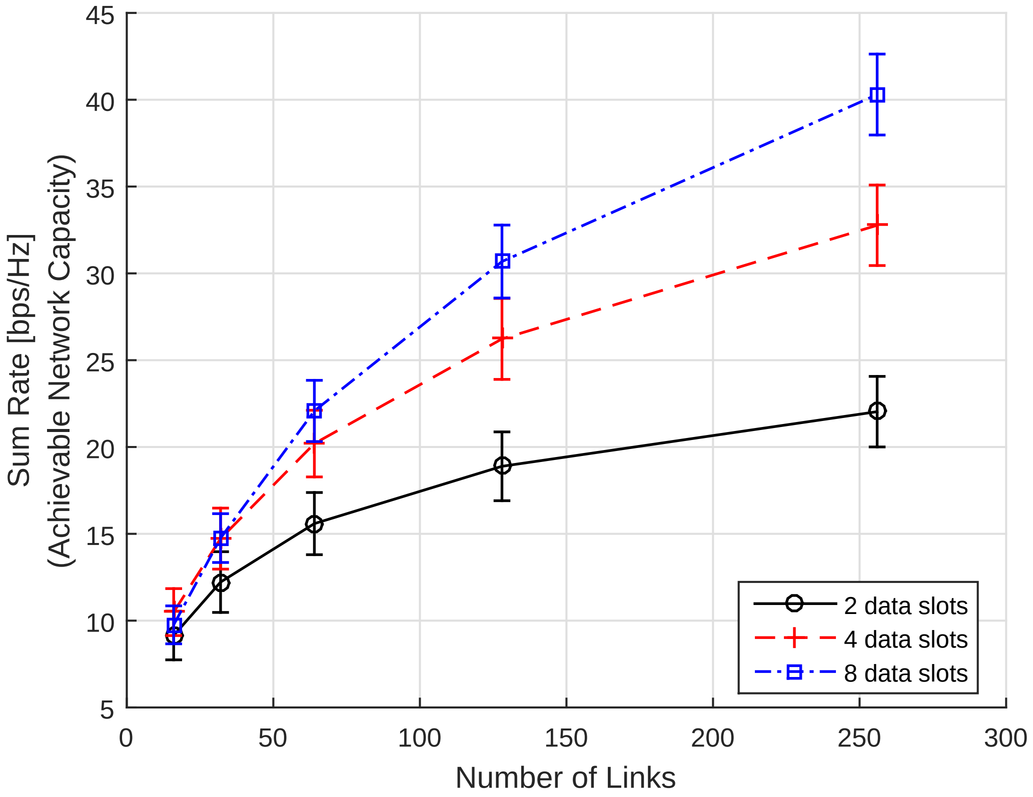


Figure 4.9: Network sum rate for $N_S = 2, 4$, and 8 and $\alpha = 4$ when packet sizes are 450 bits (i.e., 1 packet)

where the size of the data packets in the network is relatively small. In Figure 4.9, time slots are of length $T_{slot} = 1$ hop or equivalently 1 packet of 450 bits. In this case, the transmission efficiencies are extremely low since T_{sch} is fixed at 8 hops. In such a case, implementing the scheduling algorithm for a single slot and repeating it before every transmission slot is needlessly inefficient. Moreover, scheduling users again for a single but this time longer time slots can increase the efficiency. But this approach is also unprofitable since in the network, users don't have data to transmit for long time durations.

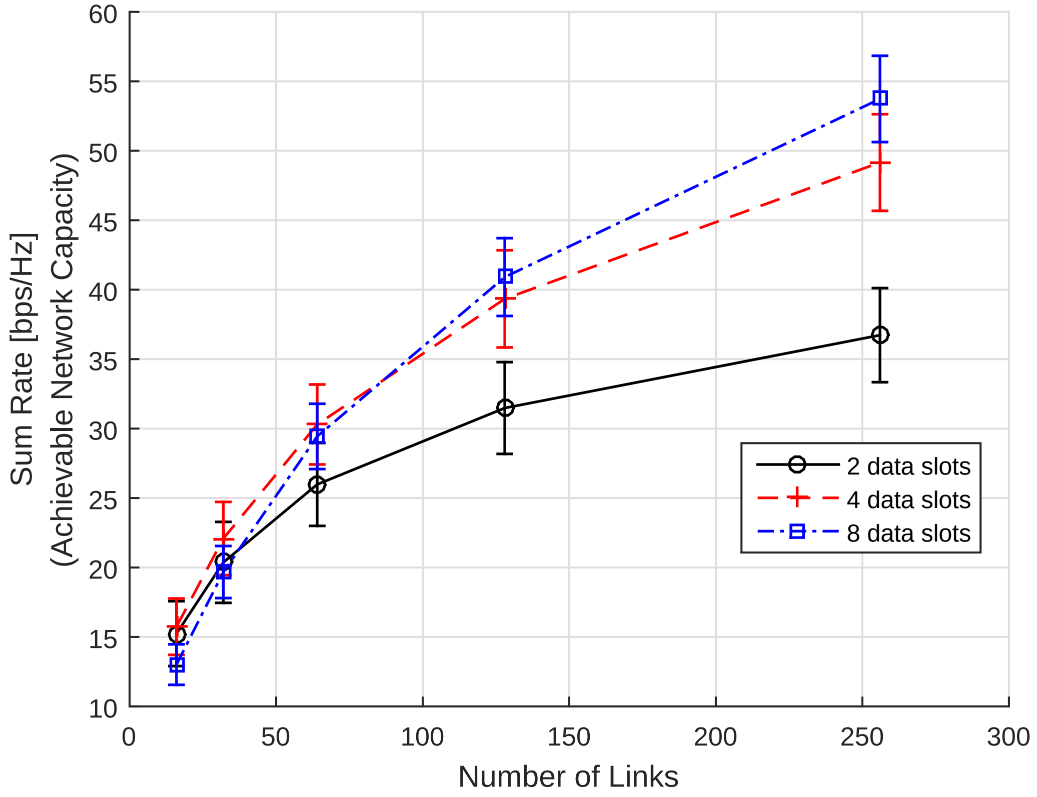


Figure 4.10: Network sum rate for $N_S = 2, 4,$ and 8 and $\alpha = 4$ when packet sizes are 900 bits (i.e., 2 packets)

Constrained by the low transmission efficiency and the small packet sizes, our claim for a performance improvement was to schedule connections in the network for multiple time slots (i.e., $N_S = 2$, 4, and 8 in our simulations). By this way, while keeping the time which is spent for scheduling, T_{sch}, fixed, we are increasing the duration of the data transmission phase, and the transmission efficiency equivalently. Moreover, since we are increasing the number of data transmission time slots, N_S, while keeping the time slot durations, T_{slot}, the same, we are basically allocating the time resources to users as they need. These lead to the case where all of the data transmission time is efficiently utilized.

In Figure 4.10, time slots are of length $T_{slot} = 2$ hops or equivalently 900 bits. In this case, since the T_{sch} is still fixed, transmission efficiencies are a little bit increased. This improvement obviously closes the gap between different number of time slot cases, and the amount of throughput surplus of $N_S = 8$ case over $N_S = 2$ case drops to 44% from 77%.

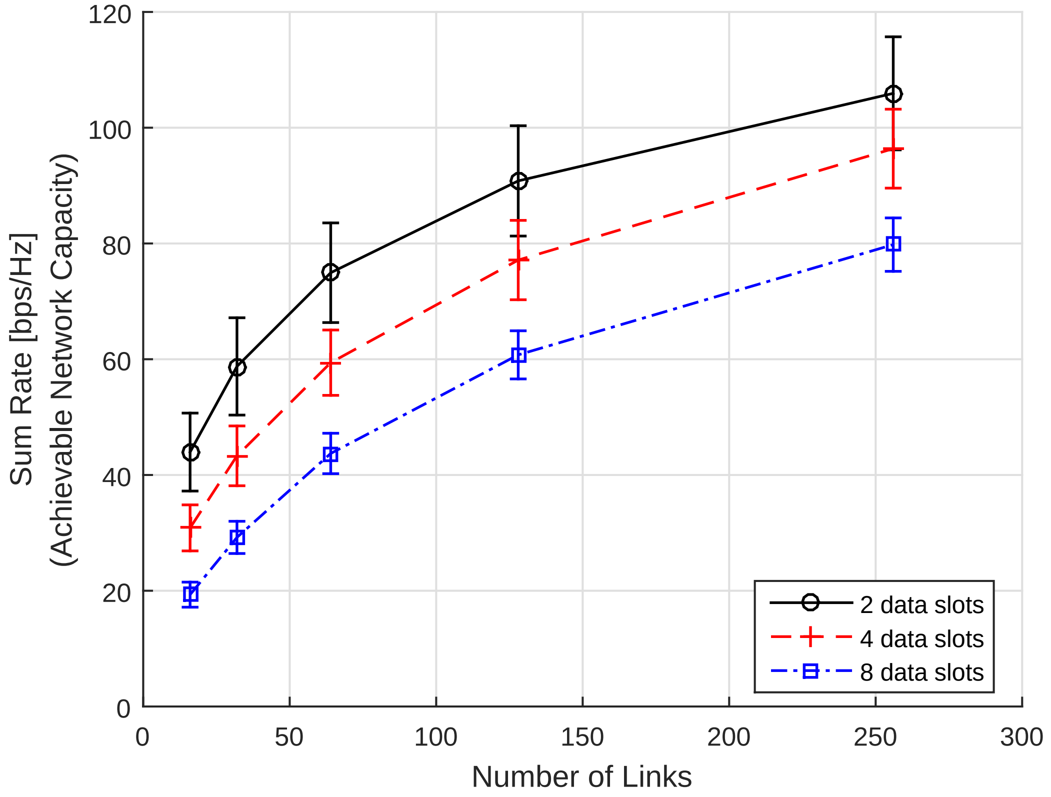


Figure 4.11: Network sum rate for $N_S = 2, 4,$ and 8 and $\alpha = 4$ when packet sizes are 5625 bytes (i.e., 100 packets)

Finally, in Figure 4.11, time slots are of length $T_{slot} = 100$ hops or equivalently 45000 bits. This case is implemented to illustrate the extreme situation where the fixed scheduling time, $T_{sch} = 8$ hops, is negligible with respect to the data transmission durations. In such a situation, the efficiencies for all of the different number of N_S cases converge to 1 and our approach becomes unnecessary. As seen in Figure 4.11, scheduling connections over more time slots degrades the performance. One can ask the question, why we do not observe the same performances approximately if the efficiencies are the same. To answer this question we refer to Figure 4.12.

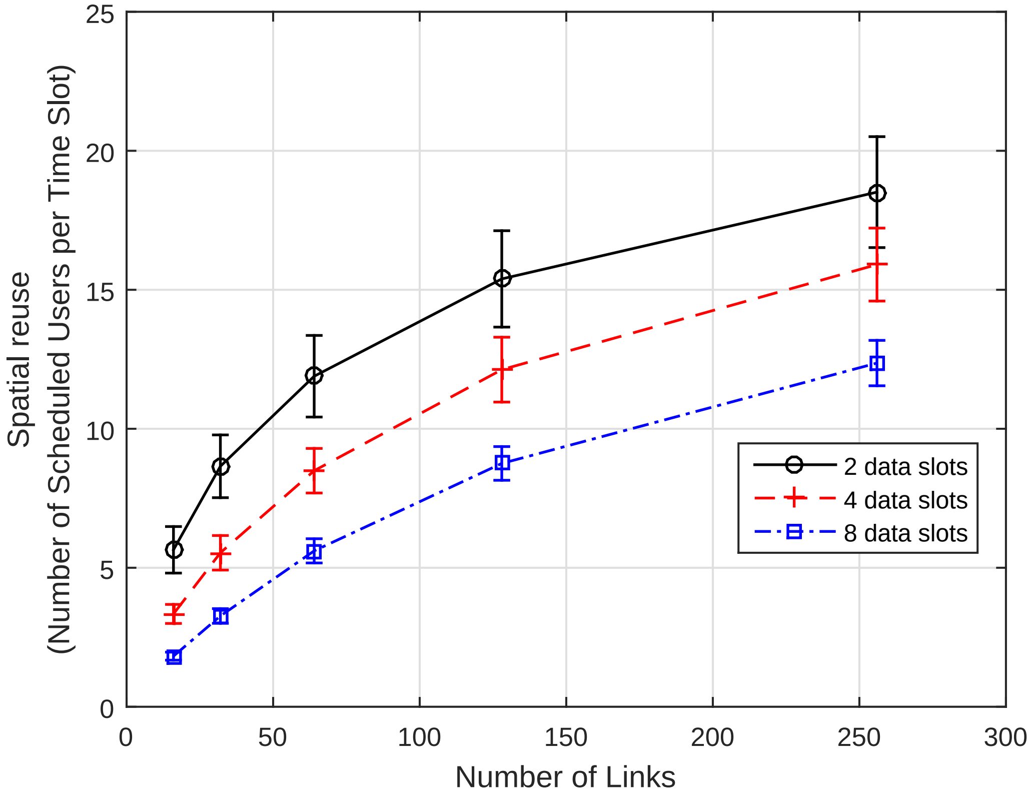


Figure 4.12: Spatial reuse values (i.e., number of scheduled links per time slot) for $N_S = 2, 4$, and 8 and $\alpha = 4$

In Figure 4.12, we plotted the spatial reuse performance of the algorithm along with the standard deviations for various N_S values when $\alpha = 4$ and $P_T = -30$ dBW. The number of links, N_L, is again in the range between 16 and 256. By spatial reuse, we mean that the number of links scheduled per time slot. For example the rightmost square data point in Figure 4.12 indicates that there are approximately 12 scheduled links per time slot for the $N_S = 8$ slots case where 256 links are deployed over the area. This indicates that the algorithm has scheduled 96 of the 256 users in the area over 8 time slots. Equivalently the rightmost circle data point illustrates that there are approximately 36 scheduled users in the $N_S = 2$ slots and $N_L = 256$ links case.

As we can see in Figure 4.12, scheduling performance from the total number of scheduled users perspective is better for higher N_S values. On the other hand, a fair comparison should be made from the number of scheduled users per time slot perspective. It is fundamental to see that our algorithm is more successful to compress links in time slots when the number of time slots is smaller. This because the decentralized (i.e., distributed) nature of the algorithm. Since there is no global information on a central superior node about the links which are competing to get resources allocated and their qualities, it is impossible to schedule users optimally. In a hypothetical situation where a base station is exists and has all of the information that it needs, one can write a cost function to maximize the network throughput and solve this equation by using optimization tools. In such a case, the spatial reuse performances of different N_S would be close. The performance gain of the $N_S = 2$ case in Figure 4.11 arises from this fact. Since our system works in a distributed fashion and nodes run the algorithm by utilizing the partial knowledge that they have, the reuse performance worsens as N_S increases. This is because the fact that for lower N_S values, same number of links are fighting for resource allocation for a few time slots. For instance in $N_S = 2$ case, this forces the algorithm to pack more and more users in 2 time slots. Since the only option for a link in case of an unsuccessful SINR threshold (i.e., γ) check is to ask for the only other remaining time slot, we can think this as links are taking their chances for all of the possible time slots. On the other hand, as the number of time slots, N_S, increases, links are still trying to be scheduled in at most 2 different time slots. This means since there is no global information at the nodes, they do not know the optimum time slot for them to be scheduled. And as the N_S increases, the probability that they ask for the optimum time slot to be scheduled in the scheduling phase decreases. To sum up, as N_S increases, the algorithm performs poorly from spatial packing perspective due to its distributed nature and lack of global knowledge. On the other hand, this is dominated by the low efficiencies in the cases where packet sizes are small, and the throughput performance is still better for higher N_S values as in Figure 4.9.

Another deduction which can be made from Figures 4.9 and 4.10 is about the performance change as the number of links deployed over the area, N_L increases. For both of the figures, one can see that the throughput performance of the $N_S = 8$ case over $N_S = 4$ and 2 cases improves as the number of links in the area increases. In Figure 4.9, the $N_S = 8$ case increases the performance gap over $N_S = 4$ case, and in Figure 4.10, $N_S = 8$ case closes the gap under $N_S = 4$ case and outdoes it. We conclude from this observation that, even though the performance gain for higher N_S values seem to be lost as the packet sizes grow gradually, for higher number of links in the network (i.e., higher congestion) the scheduling over multiple time slots approach is still feasible. This is due to the fact that, although the efficiencies become closer to each other for higher T_{slot} durations, since the number of links, N_L, competing for the resources are immensely large, the performance degradation caused from poor spatial reuse performance gets smaller. To see this effect, we can have another look at Figure 4.12. As it can be seen, as the number of links in the network increase from 64 to 256, the spatial reuse performance gap between $N_S = 2$ and $N_S = 8$ reduces to 31% from 88%. This indicates that the poor spatial reuse performance of higher N_S values can be compensated by the link density in cases where N_L is large.

In addition, we could vary K value to see the effect multiple page messages which are sent in the same channel. Increasing K would decrease N_{CS} for fixed N. Consequently, the expected number of page messages in a channel would decrease. This may improve the performance due to the decreasing amount of interference. However, we observed that for $\alpha = 4$, this interference is not affecting the performance of scheduling significantly. As a result, we did not include the results for different K values.

As an intermediate conclusion, we can claim that there is a multidimensional tradeoff between number of links, N_L, packet sizes, T_{slot}, and the number of time slots to be scheduled, N_S. Although we did not come up with an analytical explanation of this relation, we can intuitively make some points:

- For smaller packet sizes (i.e., smaller T_{slot} values), scheduling over multiple time slots (i.e., higher N_S) approach brings out a performance gain from the network throughput perspective due to the limitation caused by low transmission efficiencies.

- As the packet sizes increase (i.e, higher T_{slot} values), the transmission efficiencies converge to 1, and the performance gain for higher N_S values becomes lost. This is due to the poor spatial packing performance in high N_S situations caused by the distributed nature of the scheduling algorithm.
- The performance loss of our algorithm for relatively large packet sizes as N_S increases is small for high number of links in the network, N_L, due to the fact that the spatial reuse performances of different N_S values are closer to each other in networks with high link density.

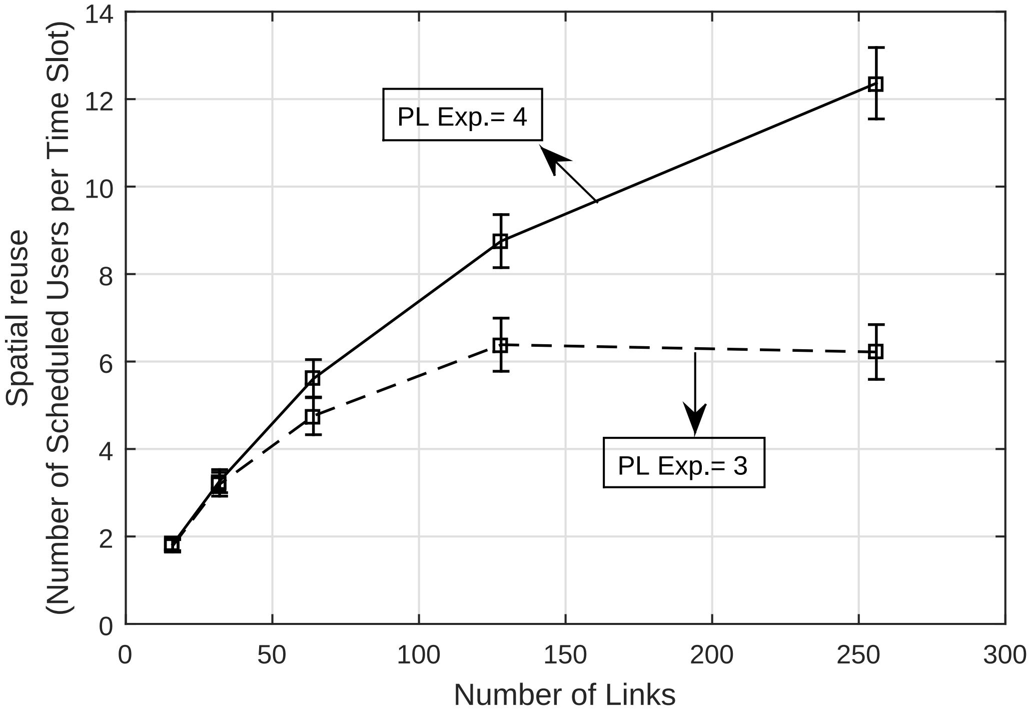


Figure 4.13: Spatial reuse values (i.e., number of scheduled links per time slot) for $N_S = 8$, $P_T = -30$ dBW and $\alpha = 3$ and $\alpha = 4$

Our claim about the compensation of poorer spatial reuse performances of the cases with higher N_S values by increasing link density of the area, (i.e., increasing N_L), can be investigated from another point of view. Basically, spatial reuse is directly proportional to the success of compressing connections into the time resource as much as possible. The primary obstacle to this procedure is the detrimental interference that the users cause on each other. Actually, the major objective of this thesis work is to manage this uncontrolled interference in a distributed manner, and plan connections over time with a guaranteed QoS (i.e., SINR in our case). On the other hand, besides the performance of our distributed algorithm and our time slotted approach, the amount of spatial reuse in a network is initially related with the propagation characteristics of the wave and the physical medium. In environments where the path loss increases rapidly with distance (i.e., high α values) the harmful range of the interference caused by a device is limited to a narrow area. This fact generates virtual cells in the area with scheduled transmitters located at their centres where the inter-cell interference is negligible as in a cellular network. Defining the environment in such a simple manner, connection scheduling procedure with interference management becomes a task where the distributed algorithm tries to find a set of links consisting of a maximum number of connections. Since not all of the pairs are separated with the same distance, those cells are not of the same size, and connection scheduling is then equivalent to partitioning the area into sections with different sizes.

Figure 4.13 shows the spatial reuse performance of our algorithm when $N_S = 8$ and the $P_T = -30$ dBW for different α values. As it is seen, when α is higher meaning that the harmful interference caused by a device is confined in a small range, the spatial reuse performance is significantly better. Furthermore, the performance gap between $\alpha = 3$ and $\alpha = 4$ cases is increasing with the number of links deployed in the area, N_L. In fact, the $\alpha = 3$ case reaches to a maximum when N_L surpasses 128, where $\alpha = 4$ case continues to improve its spatial reuse performance even when $N_L = 256$. This performance gain can be used to strengthen our third claim which states that our algorithm's performance loss in networks with relatively large packet sizes, T_{slot}, can be compensated with increasing number of links, N_L, due to the fact that the spatial reuse performances of different N_S values are closer to each other in networks with high link density. We see from Figure 4.13 that especially in networks with high electromagnetic attenuation, as the link density becomes larger and larger, aforementioned spatial reuse performance convergence occurs more rapidly. In short, to eliminate the effect of poor spatial reuse performance in high N_S cases in networks with relatively large packet sizes, the required congestion increment is smaller in environments with high attenuation.

This conclusion seems complex at first appearance. But it just states that our multidimensional trade-off includes one more parameter, α. In typical examples of networks where a D2D communication system can be implemented such as a sensor network or proximity aware advertising in a crowded shopping mall, α is modelled to be relatively higher [25] due to grounded sensors and reflections from earth, and attenuation by floors or objects, respectively. Considering that the research towards 5G will enable communication designers to use higher carrier frequencies to have higher data rates, and new use cases of 5G will need Internet of Things to exchange information continuously, it is reasonable to expect that there will be networks with high attenuation characteristics, with small packets, and with an immense number of short distance connections. We claim that our time slotted connection planning idea will find an area of utilization in such networks.

Distances of scheduled links are also important when the multi-hop structure of the network, QoS, and latency issues are considered. Figure 4.14 shows the CDFs of the distances of links which are scheduled by the proposed method for varying number of data transmission time slots, N_S = 2, 4, and 8. Here we see a uniform-like distribution for the link distances in all cases. We observe from this figure that the link distances are not necessarily very short to have a high value of overall throughput. Moreover, the algorithm does not have a bias against relatively longer links which tend to have smaller SINR values. In the simulated network, the homogeneously distributed characteristic of the link lengths between d_{max} and d_{min} are preserved after scheduling. Considering that 20m and 50m are the fixed distances assigned to links in [16], and in [36] respectively, we see that the proposed time slot allocation based resource management algorithm schedules links which have a large range of transmitter-receiver separation distances. We observe from Figure 4.14 that the performance of the proposed algorithm enables upper layers to find variety of multi-hop paths with different lengths and qualities, and may satisfy latency requirements.

SINR values of the scheduled links are also important when the QoS requirements

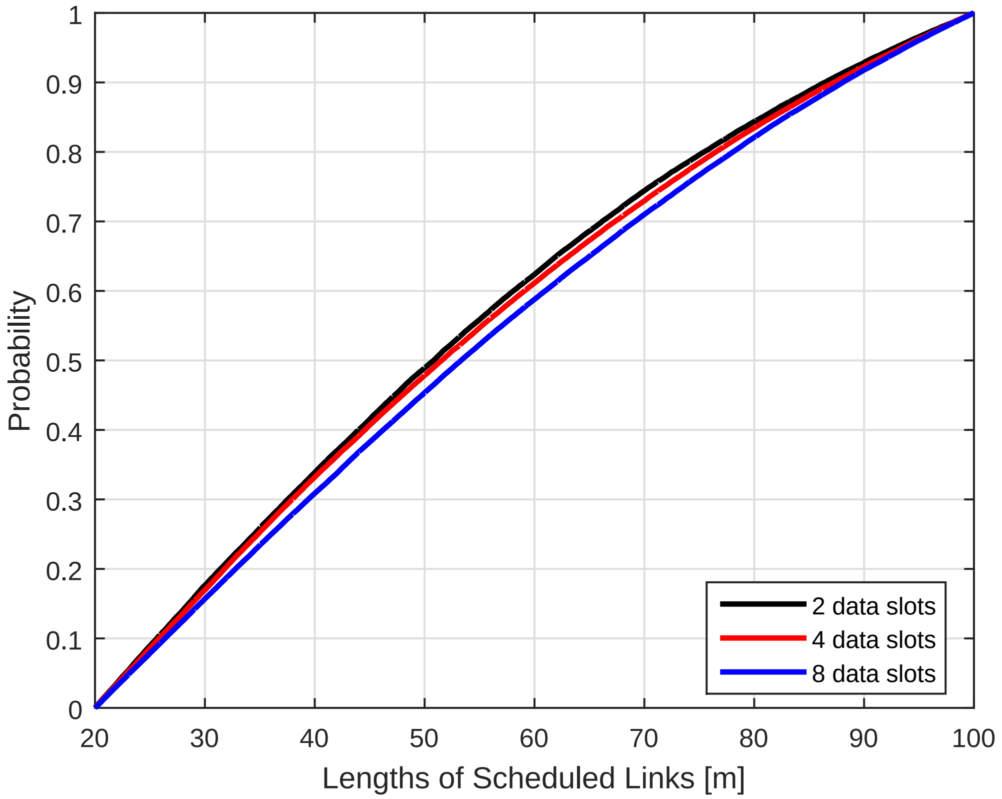


Figure 4.14: CDFs of lengths of the scheduled links for $N_S = 2, 4$, and 8, $\alpha = 4$, and $P_T = -30$ dBW

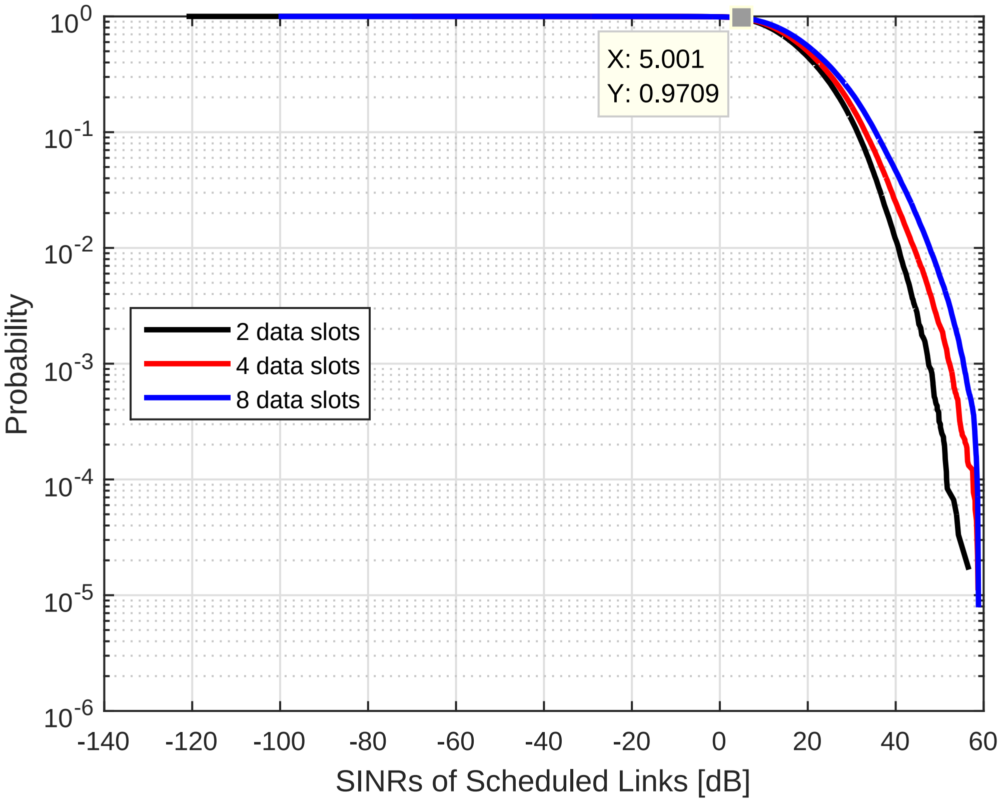


Figure 4.15: CDFs of SINRs of the scheduled links for $N_S = 2, 4$, and 8, $\alpha = 4$, and $P_T = -30$ dBW

and the success of the proposed algorithm are considered. Figure 4.15 shows the CDFs of the SINR values of links which are scheduled by the proposed method for varying number of data transmission time slots, N_S = 2, 4, and 8. Here we see that our algorithm is quite successful satisfying the minimum SINR constraint to be scheduled, γ, which is set at 5 dB in our simulations. For all cases, approximately 97% of the scheduled links have an SINR value above the threshold. The remaining erroneously scheduled links are scheduled due to the fact that the algorithm runs in a distributed manner. At the transmitter decision phase, more than one transmitters can find out that they will not decrease SINR value of any receiver with higher priority below γ. Scheduled and combined, those transmitters can decrease the SINR value of a higher priority receiver below threshold. On the other hand, Figure 4.15 shows that this situation occurs quite rare and it is reasonable to say that the algorithm works as intended.

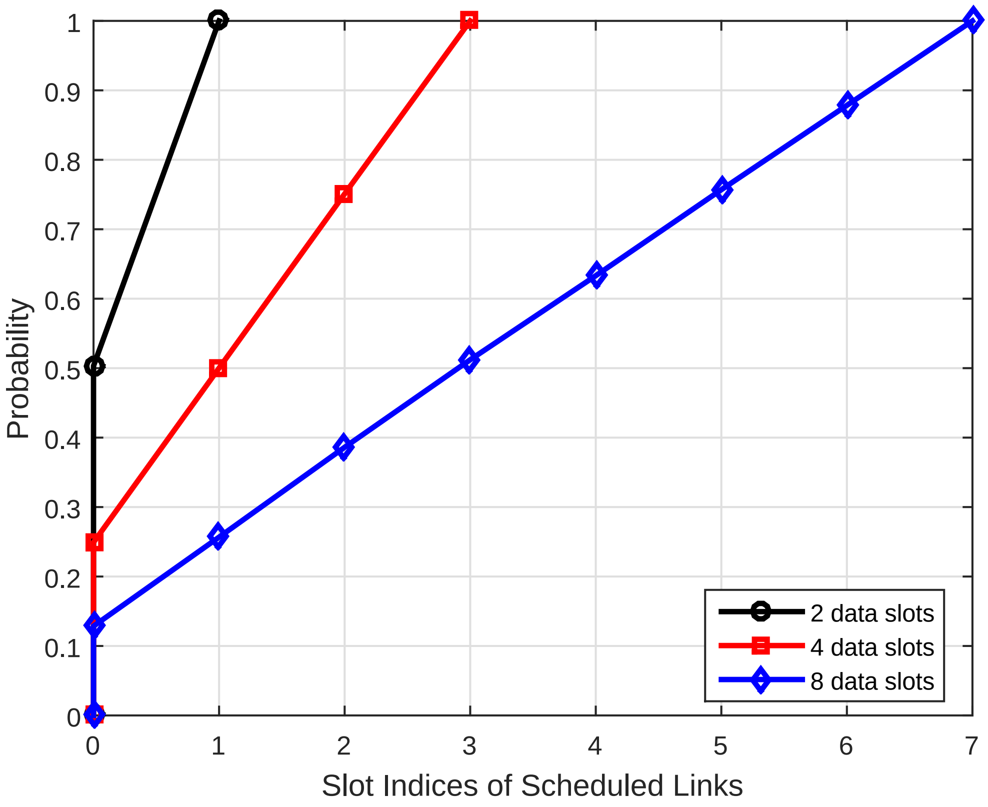


Figure 4.16: CDFs of indices of time slots that the links are scheduled in for $N_S = 2, 4$, and 8, $\alpha = 4$, and $P_T = -30$ dBW

To investigate the performance of the proposed algorithm from many perspective, we plotted the indices of the time slots that the links are scheduled in for varying number of data transmission time slots, N_S = 2, 4, and 8 in Figure 4.16. Although it is unusual prima facie, this figure indicates that half of the links are scheduled in a time slot in $N_S = 2$ case, quarter of the links are scheduled in a time slot in $N_S = 4$ case, and eighth of the links are scheduled in a time slot in $N_S = 8$ case. These results indicate that the random time slot selection procedure works decently, and there is no bias of the algorithm against any particular time slot.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

In this study, we have focused on the design of D2D communication protocols. Based on the previously proposed peer-to-peer solution FlashLinQ, we proposed an OFDMA based distributed solution for a specific type of networks. First, we implemented a received power estimation feature to the peer discovery phase of FlashLinQ. In other words, we keep track of received power levels by constant averaging to acquire estimates free of errors induced by noise, fading, and frequency selectivity. These estimates should be reliable to provide interference-awareness even in networks where there is a frequency hopping strategy is implemented. In such networks, instantaneous energy probing approach of FlashLinQ remains incapable of representing average channel conditions. In addition, we proposed an altered OFDMA resource allocation scheme for peer discovery signalling to create time and frequency diversities. These diversities increase the discovery probability of users in frequency hopping networks and provide more reliable discovery signalling. Moreover, receiving discovery messages in more than one time-frequency slots diversifies the observed channel conditions and increases the consistency of the received power estimates more quickly.

Simulation results showed that depending on the path loss characteristics of the network, the peer discovery signalling scheme works as expected. In cases where path loss is decreasing with distance slowly, users are able to discover nearly all of the other devices in a 1 km x 1 km area by taking the advantage of parallel channel access feature of OFDMA. On the other hand, as the increasing rate of the path loss values with distance increases, the discovery performance worsens naturally. The reasonable deduction is that depending on the modulation, coding and diversity, there is a correct packet reception probability of the proposed system for a given SNR. Consequently, devices for which the link between them ensures an SNR requirement are likely to discover each other.

Moreover, the performance of the received power estimation procedure is also satisfying. We observed that even in one discovery block repetition, we are able to get good estimates which are at most 4 dB away from the real value nearly all the time. In addition, the success of the estimate will improve as the discovery block repetitions increase. Since the received power information is very important in the distributed scheduling phase, this success is one of the main boosters of the performance of our connection establishment procedure.

The most important result of our work is the performance of our scheduler from the throughput perspective. We proposed a hybrid paging and connection scheduling signalling phase for connection establishment purpose, to increase the effective network sum rate of the networks where the packet sizes are small. In such networks either repeating the scheduling phase frequently or scheduling users for long data transmission stages are inefficient, since the first strategy decreases transmission efficiency and second one assigns time resource to users more than needed (which is equivalent to decrease transmission efficiency).

Firstly, our method combines the paging and link scheduling sections of FlashlinQ, and completes these procedures in a single phase which decreases overall overhead. This is because FlashLinQ first implements a paging stage and then transmits energy-level based signals for scheduling and interference management while we do both of them simultaneously and know the interference amounts from our received power estimation strategy. Secondly, for networks where the communication is bursty, the approach of FlashLinQ remains insufficient to increase network throughput when we consider the transmission efficiency. So rather than scheduling connections for long periods of data transmission, we divide the data transmission phase into multiple time slots and accomplish the scheduling over time slots. By exploiting the spatial reuse over multiple time slots, we confirmed by simulations that it is possible to attain larger network throughputs while keeping the overheads as small as possible.

The performance gain of the time slotted approach from the network throughput perspective is significant especially when packet sizes in the network decrease. When the packet sizes increase, transmission efficiencies become closer to each other since time spent for scheduling is negligible, and the throughput performance depends only on the spatial reuse performance. We observed that due to the distributed nature of the scheduling strategy, as the number of time slots to be scheduled increase, the spatial performance worsens since scheduling optimum (from throughput perspective) sets of users to every time slot becomes harder in the lack of global knowledge. This leads to the situation that if efficiency loss of a low number of time slots case is small due to size of payload, time slotted approach does not perform better. In such cases, if the number of links competing for the resource allocation is large, spatial performances also became closer to each other since there are more than enough connections to be spatially packed over time slots. Especially in interference dominant networks, spatial performance of the multiple time slot approach increases more rapidly with increasing number of links which leads to the conclusion that for relatively large packet sizes, time slotted approach is still feasible.

To sum up, our conclusions for the peer discovery part are:

- Using the parallel channel access speciality of OFDMA, if a link satisfies the conditions for high probability of correct packet reception in that specific configuration of PHY layer design, devices at both ends of that link are likely to discover each other,
- Even in one discovery repetition, it is achievable for a device to estimate the average channel conditions between it with other devices with small error, and this information will both decrease the overhead and be of good quality than a single slot energy-level probing.

Finally, our conclusions for the distributed time slotted scheduler indicates that there is a multidimensional trade-off between number of time slots to be scheduled, number of connections in the network, packet sizes, and interference characteristics. In short:

- For networks where size of data packets is small, it is more efficient to implement scheduling over multiple time slots from throughput perspective. This is because of the low transmission efficiency due to considerable amount of time spent for scheduling,

- As the packet sizes increase, the time spent for scheduling becomes negligible, transmission efficiencies become closer to each other, and the throughput performance depend only on spatial reuse. Since spatial reuse performance is worse for higher number of time slots due to distributed nature of algorithm, performance gain is lost,
- On the other hand for networks where there is a large number of connections to be established, the difference between spatial reuse performances become small since there is a congestion. So the multiple time slot scheduling approach again becomes feasible,
- If interference is dominant in the network along with the high number of connections, spatial reuse performance and consequently the the network sum rate performance of the multiple slot approach is better even for relatively larger packet sizes.

We think that deriving an analytical explanation for this multi parameter relation would be a worthwhile future direction for this thesis work. Furthermore, considering this interference limited distributed scheduling problem from multi user information theory perspective would strengthen our claims and broaden our endeavors. Last but not least, considering a dynamic network where new devices can join, and designing the pre-allocation of discovery and paging resources in a time varying (from number of users perspective) fashion can be of practical importance in future.

Finally, to broaden our viewpoint from the network layer perspective, investigating proposed approaches in the multi-hop network setting by considering the delays, throughputs, and QoS satisfaction in the network can be a reasonable and productive future work.

REFERENCES

[1] A. Osseiran, F. Boccardi, V. Braun, K. Kusume, P. Marsch, M. Maternia, O. Queseth, M. Schellmann, H. Schotten, H. Taoka, H. Tullberg, M. Uusitalo, B. Timus, and M. Fallgren, "Scenarios for 5G mobile and wireless communications: the vision of the METIS project," *IEEE Communications Magazine*, vol. 52, no. 5, pp. 26–35, May 2014.

[2] W. H. Chin, Z. Fan, and R. Haines, "Emerging technologies and research challenges for 5G wireless networks," *IEEE Wireless Communications*, vol. 21, no. 2, pp. 106–112, April 2014.

[3] C.-X. Wang, F. Haider, X. Gao, X.-H. You, Y. Yang, D. Yuan, H. Aggoune, H. Haas, S. Fletcher, and E. Hepsaydir, "Cellular architecture and key technologies for 5G wireless communication networks," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 122–130, February 2014.

[4] T. Rappaport, S. Sun, R. Mayzus, H. Zhao, Y. Azar, K. Wang, G. Wong, J. Schulz, M. Samimi, and F. Gutierrez, "Millimeter Wave Mobile Communications for 5G Cellular: It Will Work!" *IEEE Access*, vol. 1, pp. 335–349, 2013.

[5] L. Lu, G. Li, A. Swindlehurst, A. Ashikhmin, and R. Zhang, "An Overview of Massive MIMO: Benefits and Challenges," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 5, pp. 742–758, Oct 2014.

[6] L. Song, D. Niyato, Z. Han, and E. Hossain, *Wireless Device-to-Device Communications and Networks*. Cambridge University Press, 2015, cambridge Books Online. [Online]. Available: http://dx.doi.org/10.1017/CBO9781107478732

[7] A. Ghosh, R. Ratasuk, B. Mondal, N. Mangalvedhe, and T. Thomas, "LTE-advanced: next-generation wireless broadband technology [Invited Paper]," *IEEE Wireless Communications*, vol. 17, no. 3, pp. 10–22, June 2010.

[8] M. Tehrani, M. Uysal, and H. Yanikomeroglu, "Device-to-device communication in 5G cellular networks: challenges, solutions, and future directions," *IEEE Communications Magazine*, vol. 52, no. 5, pp. 86–92, May 2014.

[9] Y. Cao, T. Jiang, and C. Wang, "Cooperative device-to-device communications in cellular networks," *IEEE Wireless Communications*, vol. 22, no. 3, pp. 124–129, June 2015.

[10] T. Peng, Q. Lu, H. Wang, S. Xu, and W. Wang, "Interference avoidance mechanisms in the hybrid cellular and device-to-device systems," in *2009 IEEE 20th International Symposium on Personal, Indoor and Mobile Radio Communications*, Sept 2009, pp. 617–621.

[11] P. Phunchongharn, E. Hossain, and D. Kim, "Resource allocation for device-to-device communications underlaying LTE-advanced networks," *IEEE Wireless Communications*, vol. 20, no. 4, pp. 91–100, August 2013.

[12] S. Mumtaz and J. Rodriguez, *Smart Device to Smart Device Communication*. Springer Publishing Company, Incorporated, 2014.

[13] "Wi-Fi Peer-to-Peer (P2P) Technical Specification v1.5," Wi-Fi Alliance, Tech. Rep., March 2014.

[14] "ZigBee Specification," ZigBee Alliance, Tech. Rep., January 2008.

[15] "Specification of the Bluetooth System v4.2," Bluetooth Special Interest Group, Tech. Rep., December 2014.

[16] X. Wu, S. Tavildar, S. Shakkottai, T. Richardson, J. Li, R. Laroia, and A. Jovicic, "FlashLinQ: A Synchronous Distributed Scheduler for Peer-to-Peer Ad Hoc Networks," *IEEE/ACM Transactions on Networking*, vol. 21, no. 4, pp. 1215–1228, Aug 2013.

[17] A. Asadi, Q. Wang, and V. Mancuso, "A Survey on Device-to-Device Communication in Cellular Networks," *IEEE Communications Surveys Tutorials*, vol. 16, no. 4, pp. 1801–1819, Fourthquarter 2014.

[18] B. Zhou, H. Hu, S.-Q. Huang, and H.-H. Chen, "Intracluster Device-to-Device Relay Algorithm With Optimal Resource Utilization," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 5, pp. 2315–2326, Jun 2013.

[19] L. Lei, Z. Zhong, C. Lin, and X. Shen, "Operator controlled device-to-device communications in LTE-advanced networks," *IEEE Wireless Communications*, vol. 19, no. 3, pp. 96–104, June 2012.

[20] N. Golrezaei, A. Molisch, and A. Dimakis, "Base-station assisted device-to-device communications for high-throughput wireless video networks," in *IEEE International Conference on Communications (ICC), 2012*, June 2012, pp. 7077–7081.

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

[22] X. Bao, U. Lee, I. Rimac, and R. R. Choudhury, "DataSpotting: Offloading Cellular Traffic via Managed Device-to-device Data Transfer at Data Spots," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 14, no. 3, pp. 37–39, Dec. 2010. [Online]. Available: http://doi.acm.org/10.1145/1923641.1923655

[23] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, pp. 393–422, 2002.

[24] Y. S. Tieu J., "Wi-Fi Direct Services," Master's thesis, Lund University, Sweden, June 2014.

[25] A. Goldsmith, *Wireless communications*, 1st ed. Cambridge University Press, 2005.

[26] "IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007)*, pp. 1–2793, March 2012.

[27] B. Peterson, R. Baldwin, and J. Kharoufeh, "Bluetooth Inquiry Time Characterization and Selection," *IEEE Transactions on Mobile Computing*, vol. 5, no. 9, pp. 1173–1187, Sept 2006.

[28] M. Duflot, M. Kwiatkowska, G. Norman, and D. Parker, “A formal analysis of Bluetooth device discovery,” *International journal on software tools for technology transfer*, vol. 8, no. 6, pp. 621–632, 2006.

[29] A. Franssens, “Impact of multiple inquirers on the Bluetooth discovery process and its application to localization,” Master’s thesis, University of Twente, Netherlands, July 2010.

[30] Z. Yang, J. Huang, C. Chou, H. Hsieh, P. Yeh, and C. Hsu, “Fast Device Discovery for Device to Device Communication,” Jul. 24 2014, uS Patent App. 14/161,924. [Online]. Available: http://www.google.com/patents/US20140204898

[31] F. Baccelli, N. Khude, R. Laroia, J. Li, T. Richardson, S. Shakkottai, S. Tavildar, and X. Wu, “On the design of device-to-device autonomous discovery,” in *Fourth International Conference on Communication Systems and Networks (COMSNETS), 2012*, Jan 2012, pp. 1–9.

[32] R. Knopp and P. Humblet, “On coding for block fading channels,” *IEEE Transactions on Information Theory*, vol. 46, no. 1, pp. 189–205, Jan 2000.

[33] H.-W. Yoon, J. S. Kim, S. J. Bae, B.-G. Choi, and M. Y. Chung, “Performance analysis of FlashLinQ with various yielding threshold values,” in *International Conference on ICT Convergence (ICTC), 2012*, Oct 2012, pp. 477–478.

[34] N. Naderializadeh and A. Avestimehr, “ITLinQ: A New Approach for Spectrum Sharing in Device-to-Device Communication Systems,” *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1139–1151, June 2014.

[35] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris, “A High-throughput Path Metric for Multi-hop Wireless Routing,” *Wirel. Netw.*, vol. 11, no. 4, pp. 419–434, jul 2005. [Online]. Available: http://dx.doi.org/10.1007/s11276-005-1766-z

[36] Z. Chen and M. Kountouris, “Distributed SIR-aware opportunistic access control for D2D underlaid cellular networks,” in *IEEE Global Communications Conference (GLOBECOM), 2014*, Dec 2014, pp. 1540–1545.

[37] N. Golmie, O. Rebala, and N. Chevrollier, “Bluetooth adaptive frequency hopping and scheduling,” in *MILCOM ’03 IEEE Military Communications Conference, 2003*, vol. 2, Oct 2003, pp. 1138–1142 Vol.2.

[38] J. Proakis and M. Salehi, *Digital communications*, 5th ed. McGraw-Hill Education, 2007.

[39] D. Camps-Mur, A. Garcia-Saavedra, and P. Serrano, “Device-to-device communications with Wi-Fi direct: overview and experimentation,” *IEEE Wireless Communications*, vol. 20, no. 3, pp. 96–104, June 2013.

[40] T. Thamrin and S. Sahib, “The Inquiry and Page Procedure in Bluetooth Connection,” in *SOCPAR ’09. International Conference of Soft Computing and Pattern Recognition, 2009*, Dec 2009, pp. 218–222.

[41] C. Geng, N. Naderializadeh, A. Avestimehr, and S. Jafar, “On the Optimality of Treating Interference as Noise,” *IEEE Transactions on Information Theory*, vol. 61, no. 4, pp. 1753–1767, April 2015.

[42] J. W. Kang, C. Lim, and S.-H. Kim, “A distributed link scheduling with on-off interference map for device-to-device communications,” in *International Conference on ICT Convergence (ICTC), 2013*, Oct 2013, pp. 1065–1066.

[43] A. Dementyev, S. Hodges, S. Taylor, and J. Smith, “Power Consumption Analysis of Bluetooth Low Energy, ZigBee, and ANT Sensor Nodes in a Cyclic Sleep Scenario,” in *Proceedings of IEEE International Wireless Symposium (IWS)*. IEEE, April 2013. [Online]. Available: http://research.microsoft.com/apps/pubs/default.aspx?id=192688

[44] S. Lin and D. J. Costello, *Error Control Coding, Second Edition*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2004.

APPENDIX A

MAXIMUM-LIKELIHOOD ESTIMATION OF RECEIVED POWER

The received signal from a single user in peer discovery section which is explained in Chapter 3 is:

$$y_{i,j} = h_{i,j}x_{i,j} + n_{i,j}, \qquad i = 1, 2, \ldots, N_i \quad \text{and} \quad j = 1, 2, \ldots, M \tag{A.1}$$

where $h_{i,j}$ is the channel gain and $x_{i,j}$ is the 4-QAM symbol in the i^{th} subcarrier in the j^{th} discovery block, M is the number of peer discovery blocks, and N_i is the number information subcarriers. The distributions of $h_{i,j}$'s and $n_{i,j}$'s are:

$$h_{i,j} \sim \mathcal{CN}(0, P_r), \quad \text{and} \quad n_{i,j} \sim \mathcal{CN}(0, N_0), \tag{A.2}$$

where N_0 is the noise power, and P_r is the quantity we are trying to estimate which the ideal received power which is free of errors caused by noise and fading. Since $x_{i,j}$'s are unit-norm complex numbers, the probability distribution of the received signal becomes:

$$y_{i,j} \sim \mathcal{CN}(0, P_r + N_0). \tag{A.3}$$

Since $N_i = 5$ and $M = 8$ in our system, we have 40 independent received signal observations which leads to a probability distribution for an observation vector $\mathbf{y}_{1\times 40}$ as:

$$f_{\mathbf{Y}}(\mathbf{y}) = \prod_{k=1}^{40} \frac{1}{\pi(P_r + N_0)} e^{-\frac{|y_k|^2}{(P_r+N_0)}}. \tag{A.4}$$

Then ML estimate for P_r can be written as:

$$\hat{P}_{r,ML}(\mathbf{y}) = \arg\max_{P_r} \quad ln\Big(f_{\mathbf{Y}}(\mathbf{y}|P_r)\Big). \tag{A.5}$$

To achieve maximization, we can take derivative with respect to P_r and then equate it to zero as:

$$\frac{d}{dP_r} ln\Big(f_{\mathbf{Y}}(\mathbf{y}|P_r)\Big) = \frac{d}{dp}\Bigg\{ - 40ln(\pi) - 40ln(P_r + N_0) - \sum_{j=1}^{40} \frac{|y_k|^2}{(P_r + N_0)} \Bigg\}, \quad \text{(A.6)}$$

$$0 = -40\frac{1}{P_r + N_0} + \sum_{j=1}^{40} \frac{|y_k|^2}{(P_r + N_0)^2}, \quad \text{(A.7)}$$

and the estimate becomes:

$$\hat{P}_{r,ML}(\mathbf{y}) = \Bigg(\frac{1}{40}\sum_{j=1}^{40} |y_k|^2\Bigg) - N_0. \quad \text{(A.8)}$$

Note that P_r cannot be smaller than zero since it represents power. Due to this constraint, we modify equation (A.8) such that the estimate becomes:

$$\hat{P}_r(\mathbf{y}) = max\Bigg\{ 0, \hat{P}_{r,ML}(\mathbf{y}) \Bigg\}, \quad \text{(A.9)}$$

and it can be shown that this is the optimum maximum a posteriori (MAP) solution of this problem by using the unimodality of chi-squared distribution.

APPENDIX B

MITIGATION OF EFFECTS OF FADING AND PATH DELAY

In [25], the purpose of adding a cyclic prefix (CP) in front of every time domain OFDMA signal is thoroughly explained. By selecting the last μ values of the N-tap $x[n]$ sequence which is the output of the inverse discrete Fourier transform (IDFT) operation as the CP, we form a new sequence $\widetilde{x}[n]$ of length $N+\mu$ which is periodic with period N in the region of interest which can be written as:

$$\widetilde{x}[n] = x[n]_N \quad \text{for} \quad -\mu \leqslant n \leqslant N-1. \tag{B.1}$$

Suppose that $\widetilde{x}[n]$ is the input sequence to the channel with an impulse response $h[n]$ of length $\mu+1$ where we can designate $\mu+1$ as the maximum delay spread of that particular channel . Then we can write:

$$y[n] = \widetilde{x}[n] * h[n], \tag{B.2}$$

$$y[n] = \sum_{k=0}^{\mu-1} h[k]\widetilde{x}[n-k], \tag{B.3}$$

$$y[n] = \sum_{k=0}^{\mu-1} h[k]x[n-k]_N, \tag{B.4}$$

$$y[n] = x[n] \otimes h[n] \quad \text{for} \quad 0 \leqslant n \leqslant N-1, \tag{B.5}$$

where $y[n]$ is the output of the channel. The resulting equation (B.5) indicates that by appending the CP, the linear convolution which is modelling the effect of channel becomes a circular convolution for the region of interest. Then taking the discrete Fourier transform (DFT) of the channel output, we get:

$$Y[i] = DFT\Big\{y[n] = x[n] \otimes h[n]\Big\} = X[i]H[i], \tag{B.6}$$

where the input sequence can be recovered in the region of interest as

$$x[n] = IDFT\Big\{Y[i]/H[i]\Big\}, \tag{B.7}$$

which lets us recover the input signal easily (by just a division) if we know the channel impulse response. To estimate the channel impulse response one can use pilots.

However, we implemented the multiple access version of the OFDM where different users utilize different sets of subcarriers. Since distances between device pairs in the network are not identical, one can easily see that for a particular receiving node, signals in different subcarriers arrive at different time instants due to the effect of different amounts of delays as illustrated in Figure B.1. Note that in Figure B.1, channels consist of multiple subcarriers (5 or 6 in our case).

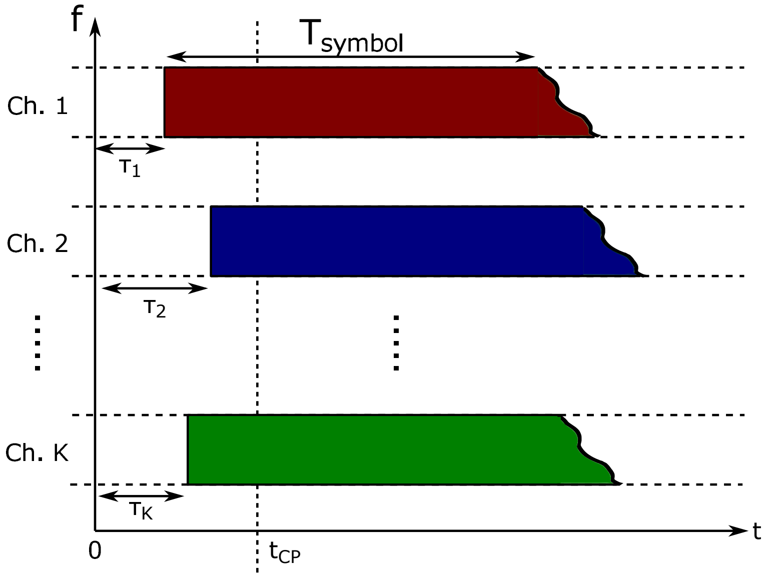


Figure B.1: Effect of different delays on channels due to device locations

Theoretically, we should receive the transmitted OFDMA signal starting from the $t = 0$ instant, and use the portion starting from t_{CP} to avoid intersymbol interference (ISI). On the other hand, there are path delays due to the finite speed of electromagnetic waves and the distance between nodes. Consequently, signals arrive with a delay to the receiver. Furthermore, signals in different subcarriers will arrive with different delays in our case since they are utilized by different users. For instance, the path delay between the receiver and the device which is utilizing channel 1 is τ_1 in Figure B.1 and so forth. Mathematically, instead of receiving $y(t)$, receiver observes:

$$y'(t) = y(t - \tau_1), \tag{B.8}$$

which leads to the received signal:

$$y[n] = y'(t)|_{t=nT_s}, \tag{B.9}$$

on the first channel. We know that the time delay in the continuous time signal will be observed as an additive linear phase on the discrete time samples when we apply the DFT operation due to the following property of Fourier transform:

$$x[n-k] \xrightarrow{\mathfrak{F}} X(e^{j\omega})e^{-j\omega k}. \tag{B.10}$$

To mitigate this effect of the path delays on the received samples, we implemented a simple method. As it is explained in Chapters 3 and 4, we either assign 5 or 6 subcarriers to a single user. We inserted pilots to the third and the fourth subcarriers on the transmitter side. On the receiver side, we extracted two complex coefficients from the pilots which are due to the multipath induced fading in the channel. To mitigate the effect of fading, we used one of these two complex coefficients and divide samples by it assuming that the channel is flat over the utilized subcarriers. On the other hand, to mitigate the effect of path delay which we showed to be equivalent to an additive linear phase on samples, we determine the phase difference between received pilot symbols and remove this phase from other samples accordingly. This operation can be written as:

$$y[n] = \Big[y[1] \quad y[2] \quad \widetilde{p_1} \quad \widetilde{p_2} \quad y[5] \quad y[6]\Big], \tag{B.11}$$

where $\widetilde{p_1}$ and $\widetilde{p_2}$ are the received pilots. Then we extract the channel coefficient as:

$$h = \frac{\widetilde{p_1}}{p_1}. \tag{B.12}$$

In addition, we extract the phase difference between two pilots as:

$$\theta = \angle\Big\{\frac{\widetilde{p_2}}{\widetilde{p_1}}\Big\}. \tag{B.13}$$

Finally, the operation that we have used to mitigate the effects of fading and path delay can be written as:

$$\hat{y}[n] = \frac{1}{h}\Big[y[1]e^{j2\theta} \quad y[2]e^{j\theta} \quad y[5]e^{-j2\theta} \quad y[6]e^{-j3\theta}\Big]. \tag{B.14}$$

It is worth noting that $H[i]$ includes effects of both multipath induced fading and path delay for each sample in practice since the channel impulse response is extracted by interpolating multiple complex channel coefficients from pilots. Consequently, there is no need for such a phase correction in addition to the single-tap equalization. However, we are using a single complex channel coefficient for a set of subcarriers which can only represent the fading characteristics. This leads to the requirement that the path delay induced additive linear phase on samples should be corrected separately.