Shourov Kumar Roy

Quantifying Device-to-Device Data Transfer in 802.11 Wireless Networks

School of Electrical Engineering

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 10.06.2015

Thesis supervisor:

Professor Jörg Ott

Thesis advisor:

M.Sc. Teemu Kärkkäinen

Aalto University
School of Electrical
Engineering

AALTO UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING

ABSTRACT OF THE
MASTER'S THESIS

Author: Shourov Kumar Roy

Title: Quantifying Device-to-Device Data Transfer in 802.11 Wireless Networks

Date: 10.06.2015 Language: English Number of pages: 10+92

Department of Communications and Networking

Professorship: Networking Technology Code: S-38

Supervisor: Professor Jörg Ott

Advisor: M.Sc. Teemu Kärkkäinen

Opportunistic contacts can be very short-lived, i.e., they usually last for a few seconds to several minutes. During the short connection window of an opportunistic contact, it is important to maximize the data transfer between the nodes. The amount of data that can be transferred during an opportunistic contact (i.e., contact capability) will depend on the link layer connection establishment delay, IP address acquisition delay as well as on the data throughput. If the opportunistic network is aimed for service provisioning, the delay of the service discovery phase is also needed to be taken into account while defining the capability of the contact. Moreover, the wireless environment where the opportunistic contacts take place, can influence the overall capability of the opportunistic contacts. In the thesis, we get the opportunity to put emphasize on all these factors while quantifying the opportunistic contacts (i.e., device-to-device data transfer) in indoor 802.11b Wi-Fi networks (Infrastructure mode).

This thesis makes several original contributions. First, we carefully design the indoor wireless testbed and to facilitate the experiments we build a Service Browser and Service Publisher application. Second, we conduct a site survey in the testbed area to gain an understanding on the indoor RF wave propagation characteristics. Third, we perform experiments, where we collect traces during the link-layer connection establishment, IP address acquisition and service discovery phases of opportunistic contacts. Using the collected data, we measure the delays of the different events/steps that take place during the phases mentioned above. Furthermore, we run experiments to investigate the throughput performance of data transmission between the nodes when TCP is used as the transport layer protocol and thus to check the suitability of TCP in opportunistic networks.

Keywords: Opportunistic contact, IEEE 802.11b, Infrastructure mode, Indoor RF wave propagation, Service discovery, Avahi, mDNS/DNS-SD, TCP

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor, Prof. Jörg Ott for providing me the opportunity to pursue this Master's thesis topic. I am indebted for his continuous support, guidance and the resources he provided. His consideration, patience and words of encouragement helped me a lot during the difficult times. Without his cooperation it would be not possible to complete the thesis.

I am grateful to have Teemu Kärkkäinen as my instructor. It's been a pleasure working with an outstanding researcher like him. The technical guidance he provided always kept me on the right track and helped me achieving the goals of the thesis. I would like to thank him for his guidance, patience as well as for keeping an open door always.

I would like to thank Viktor Nässi for providing the resources that were needed to carry out the experiments in the thesis work. His experience also proved very effective in planning the experiments.

Deepest thanks also goes to Jenni Tulensalo for being always so cooperative with the study related administrative issues. Her cooperation always helped me to stay focused on my work without worrying much about the administrative issues.

Special thanks to Mohammad Ashraful Hoque for his valuable suggestions on different technical issues as well as for his cooperation. I would also like to thank Md. Tarikul Islam for all the advices he provided.

I would like to thank my friends: Stefano, Tarja, Liisa, Prajwal, Elisabet, Anja, Sumon, Jewel, Ripu, Judith and Alexandra who have always inspired and supported me in so many ways. Many thanks also goes to the family of Meri, Mikko and little Eero. A family that has made my life easier and enjoyable in Finland and thus helped me a lot in concentrating on the thesis.

Finally, my heartfelt thanks to my family for all the good advices, opportunities, encouragement and support they have given me all these years. I am extremely grateful for the sacrifices that my parents have made for my education. A very special thanks to my little sister, Sugandha for being an unending source of encouragement as well as for being the best little sister ever.

Espoo, 10.06.2015

Shourov Kumar Roy

Contents

List of Figures

List of Tables

Abbreviations

ARP	Address Resolution Protocol
BSS	Basic Service Set
BSSID	Basic Service Set Identifier
DTN	Delay-Tolerant Networking
DNS	Domain Name System
DNS-SD	DNS Service Discovery
DHCP	Dynamic Host Configuration Protocol
IBSS	Independent Basic Service Set
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
LOS	Line-of-sight
mDNS	Multicast DNS
RF	Radio Frequency
SSID	Service Set Identifier
TCP	Transmission Control Protocol
WEP	Wired Equivalent Privacy

1 Introduction

In opportunistic networks content is forwarded between devices in the absence of global connectivity by taking advantage of communication opportunities that arise when devices come close to each other. Opportunistic networks can be characterized by sparse connectivity, forwarding through mobility and fault tolerance. Opportunistic networking is considered as a subclass of Delay-Tolerant Networking (DTN). This kind of network seems to be a promising future network technology and has already been successfully implemented in many projects like wildlife tracking [44, 78], developing remote area communication [58], interplanetary communication [53, 18], disaster management [86] and VANETs (Vehicular ad-hoc networks) [39, 62].

Opportunistic networking shows great promise for enabling entirely new types of applications, services as well as new ways of using old applications. These applications and services can rely purely on direct communication between devices while also exploiting static infrastructure components, such as network access points, when available. We can identify two device-to-device opportunistic network paradigms: Direct (e.g., IEEE 802.11 Ad-hoc mode) and Infrastructure-assisted (e.g., IEEE 802.11 Infrastructure mode) [60].

Opportunistic communications have gained interest recently due to the increasing number of mobile devices people carry today and their power in terms of CPU, storage space and available networking interfaces. Alcatel-Lucent estimated that the number of smartphone connections would grow from 500 million in 2010 to 2.5 billion in 2015 [12]. The penetration of 802.11 (Wi-Fi) networks around the world has been phenomenal. As mentioned earlier, beside the device-to-device direct communication, opportunistic contact between two nodes is possible via access point as well. There was an estimation by Informa Telecoms & Media that the number of private and public hotspots was 233 million worldwide in 2009 and it would grow at a CAGR (Compound Annual Growth Rate) of more than 18% over the next five years, reaching more than 652 million worldwide in 2015 [56]. Also, making use of Wi-Fi in unlicensed spectrum as a network offloading solution is an increasingly attractive solution for mobile operators. More than 7 million carrier-grade Wi-Fi access points have been deployed by telecom operators worldwide as of year-end 2012 and growing at a CAGR of 13.5%, that is going to reach 15 million by 2018 [13].

All these statistics above show that there is a huge scope for the opportunistic network based technologies and applications to exploit the Wi-Fi networks that are available all around us. The thesis is concerned with an important technical question surrounding such a system (Wi-Fi): can the IEEE 802.11 Wi-Fi (Infrastructure mode) provide reasonable performance for the opportunistic network based applications and solutions? In particular, this thesis tries to quantify the different phases of device-to-device data transfer in 802.11 Infrastructure-assisted wireless networks.

1.1 Problem Statement

Opportunistic networks are networks in which the existence of simultaneous end-to-end paths between a sender and a receiver is not assumed. An opportunistic contact occurs when two devices come within the communication range of the same access point (in Infrastructure-assisted Wi-Fi network) and establish connection. The connection between the nodes in an opportunistic contact may last from seconds to several minutes. During this short connection window of opportunistic contact, it is important to maximize the data transfer between the nodes. But before the actual data transfer can take place, the nodes have to go through the connection establishment process with the access point. The connection set-up process consists of different phases, e.g., scanning (network discovery), joining, authentication, association and IP address acquisition. If these phases of the connection establishment process take the majority of the short contact time of an opportunistic contact, there would be very little time left for the actual data transfer. That is why in the thesis work, one of our goals is to quantify the 802.11 link-layer connection (Infrastructure mode) set-up time as well as the IP address acquisition time that have significant effect on the data transfer capability of short lived opportunistic contacts.

The central idea in many of the proposed applications for opportunistic networking is to substitute a fixed routing infrastructure by an opportunistic message routing system built on pervasive wireless nodes. Such a routing medium focuses on moving data from sources to destinations (e.g., from sensor nodes to database servers) or on providing access to certain data (e.g., in Content Centric Networking). But we can widen our view and consider opportunistic networking as a potential provider of services rather than as a simple message routing system [60]. But this service providing scenario will require additional steps to be added between the connection establishment process and the data transfer phase of an opportunistic contact. In our work the extra steps added for service provisioning are name reservation and service discovery. In this scenario, once a station (i.e., client/browser) joins the opportunistic network and has the IP address, next steps will be to reserve a domain name for its own and discover an appropriate service offered by other node (i.e., the server/publisher) in the Infrastructure-assisted Wi-Fi network. In the thesis, we characterize these steps of service provisioning as well cause these steps also take place during the short window of an opportunistic contact and may influence the capability of the contact significantly.

During a short connection period of an opportunistic contact the amount of data that can be transferred (i.e., capability of a contact) also depends on the data throughput that can be achieved after the connection is established [59]. Transmission Control Protocol (TCP) [63] is the most popular transport layer protocol for point-to-point, connection-oriented, in-order, reliable data transfer in the internet. TCP was primarily designed for wired network. In TCP, reliability is achieved by retransmitting lost packets. Packets will be retransmitted if the sender receives no acknowledgement within a certain timeout interval or receives duplicate acknowledgements. Due to the inherent reliability of wired networks, there is an implicit assumption always made by TCP that any loss is due to congestion. To reduce con-

gestion, TCP invokes its congestion control mechanism whenever any packet loss is detected [21]. But in wireless networks, packets can be lost due to the characteristics of the wireless environment (e.g., channel contention, fading, mobility, limited power etc.). TCP will yield poor throughput performance if it interprets all the losses in wireless networks as congestion and consequently invokes congestion control and avoidance procedures [51]. So in our work, we have investigated how the characteristics specific to wireless networks affect the throughput of TCP connection.

IEEE 802.11 Wi-Fi is used as the underlying technology throughout our work and we carry out our experiments in an indoor environment. Indoor wireless signal propagation can be very dynamic and complex in nature [50]. The indoor wireless environment can have great impact on the performance of the networks and thus it can influence the overall capability of the opportunistic contacts. So to study the characteristics of the wireless testbed area, we also have carried out a site survey in our work.

1.2 Contribution of the Thesis

In the thesis, we characterize the device-to-device data transfer during opportunistic contacts. First of all, we carefully design the indoor wireless testbed. A wireless testbed, when designed and administered properly, can provide profound insights on the deployed wireless networks as well as on the experiments that have been carried out on it. So the testbed we design to carry out the experiments is one of the major contributions of the thesis.

We are interested in characterizing the opportunistic contacts between two stations when they communicate via an access point in an indoor wireless environment. Due to the impact of indoor wireless environment on the overall performance of the Wi-Fi networks, understanding the indoor wireless characteristics is also a key concern in our work. That is why the thesis work includes the site survey of the indoor testbed area.

In our work, IEEE 802.11 Wi-Fi (Infrastructure mode) is used as the underlying communication technology of the opportunistic contacts between the devices. We measure and analyze each step of the link layer connection establishment procedure (i.e., scanning, joining, authentication, association) as well as the IP address acquisition procedure that take place during a contact. Measurement and analysis of these steps give us a clear idea on the duration of the overall procedure of connection establishment and thus it helps us to define the potential capability of a short lived opportunistic contact.

Next we deal with the service discovery phase. The client node that needs a service, performs a discovery phase which typically initiates queries for the appropriate service offered by the other node (i.e., the server) in the network. In our work, "DtnUpload" is the service that the client is looking for and is offered by the server. To accomplish the tasks of service browsing (i.e., querying) and service publishing we have built an Application (Service Publisher, Service Browser) using Avahi-client API [82]. The API is based on Multicast DNS (mDNS) and DNS Service Discovery

(DNS-SD) mechanisms [48, 49]. This phase of the thesis contributes to the understanding of Avahi Service Discovery mechanism and from the experimental data, we draw the timeline of the different steps that the client node go through during the service discovery phase.

Then we investigate the data transfer over TCP in opportunistic contacts. The application (i.e., Service Browser) running at the client starts uploading data to the server using the "DtnUpload" service. In our work the data transmission between the nodes is done over TCP. Through the experiments of this phase we can study the impact of indoor wireless characteristics on the data throughput of TCP connection and thus gain an understanding on the performance of TCP as a transport layer protocol in opportunistic networks.

1.3 Scope and Goals

In the thesis, we characterize the opportunistic contacts that take place in indoor single-hop Wi-Fi networks. As we use 802.11b Wi-Fi as the underlying technology for the opportunistic contacts between the devices, we measure the delays corresponding to the 802.11 link layer connection establishment phase in our work. Though Wi-Fi Ad-hoc mode can be used for opportunistic networks, in our work we emphasize only on the Wi-Fi Infrastructure mode. We use DHCP (Dynamic Host Configuration Protocol) as the IP address acquisition mechanism in the experiments [29]. To enable service provisioning in the opportunistic networks we use Avahi in our work which is an implementation of mDNS/DNS-SD mechanism [82]. During the service discovery phase, while measuring the delays we only focus on the steps/ events that the client node goes through (i.e., name reservation, service browsing and service resolution steps). The delays, the server node experiences (i.e., the delay for service registering, service publishing etc.), are not included in our work. Avahi also has an self-assigned link-local IP addressing mechanism. In our work we have not used the link-local IP addressing mechanism of Avahi, instead we have used DHCP. Though ARP (Address Resolution Protocol) is not part of the service discovery phase, at this stage we also measure the ARP delay to have a complete timeline view of all the steps that a station (i.e., a client node) goes through during an opportunistic contact [61]. For the data transmission phase, we use TCP as the transport layer protocol in order to check its performance in opportunistic networks. Some issues that are linked with opportunistic networking, e.g., inter-contact time, routing in multi-hop networks, storage management, energy efficiency of wireless devices etc., have been kept beyond the scope of the thesis.

In our work, the goal is to quantify the opportunistic contacts through the measurement of delays corresponding to the link layer connection establishment, layer-3 IP address acquisition phase and service discovery phase as well as through the analysis of throughput performance of data transmission over TCP. We also investigate the impact of indoor wireless environment on the capability of opportunistic contacts. The 802.11 link layer connection establishment phase involves several steps: scanning, joining, authentication and association. Our goal is to measure the delays that a station goes through during these steps. Then we measure the delays that

a station experiences to acquire an IP address through DHCP. During the service discovery phase the station (i.e., the client node) goes through several steps, namely, name reservation, service browsing and service resolution. So we aim to measure the delays that are corresponding to these steps. Then in order to check the suitability of TCP as a transport layer protocol in opportunistic networks, we investigate the throughput performance of data transmission over TCP. While carrying out the experiments of the different phases, the station is positioned on different locations as one of our goals in the thesis is to investigate the impact of indoor RF wave propagation characteristics on opportunistic contacts.

Opportunistic networking is a new network paradigm. Running stock implementations of standards, protocols and service provisioning mechanisms (e.g., IEEE 802.11b standard, DHCP, Avahi, TCP etc.) may not be optimal for the opportunistic networks where the packet loss rate is high and the connections are short-lived and intermittent. The unique differences that draw the line between opportunistic networks and other legacy networking environments call for new approaches for system development. In the thesis, we work on the characterization of device-to device data transfer during opportunistic contacts using some of the existing and widely used standards, protocols and services with an intention that the study, we carry out in this thesis, will contribute in the design, development and implementation of the systems (i.e., protocols, applications, service discovery mechanism etc.) that will perform optimally in opportunistic networks.

1.4 Structure

The thesis is logically structured to provide the reader with suitable background knowledge before diving deep into the details of experimentation and subsequent analysis. After introducing the work in Chapter 1, an overview on topics such as, opportunistic contacts, indoor radio wave propagation characteristics, connection establishment in IEEE 802.11 networks, IP address acquisition procedure through DHCP, service discovery mechanism and wireless data tramission over TCP are presented in Chapter 2. The original work is presented in Chapters 3, 4 and 5. In Chapter 3, we discuss the testbed design where we provide overviews on different hardware and software components as well as the details of the testbed environment (i.e., testbed set-up). Chapter 4 includes the site survey results and analysis. In Chapter 5, our experimental methods and set-ups of characterizing the opportunistic contacts (phases: connection establishment and IP acquisition, service discovery, data transmission over TCP) are described as well as the findings from the experiments are discussed in detail. The thesis concludes with a summary (Chapter 6) of the major findings of our research work and presents observations on various future work directions.

2 Different Phases of Opportunistic Communication

Opportunistic networking is a communication paradigm based on proximity of mobile users and their capability to store data on their devices (e.g., smartphones, tablets, laptops etc.), carry it through their mobility and forward it to other users they meet. An event when the devices of two users are found within transmission range is called a contact opportunity. When such an opportunity arises, two users are able to establish wireless connection between their devices to exchange data based on their interests.

Connectivity among opportunistic devices is possible through Bluetooth but this protocol has severe limitations due to its short range, limited bandwidth and pairing difficulties. While Wi-Fi Ad-hoc might appear to be better suited for the task, device support is sparse at best with no signs of improvement in sight. Manufacturers are instead focusing on Wi-Fi Direct which, however, due to its complex pairing procedure, cannot effectively enable opportunistic communication that happens in very dynamic environments [85]. In our work, Infrastructure-assisted IEEE 802.11 Wi-Fi is used as the underlying communication technology for the opportunistic contacts between the devices. This scenario in opportunistic networking is desirable for several reasons. Nowadays there are numerous hotspots (deployed Wi-Fi access points) - belonging to network operators, businesses or private households that can be used for opportunistic networking. An access point used in Infrastructure mode might offer a greater coverage area for opportunistic contacts than the technologies of Bluetooth, Wi-Fi Ad-hoc and Wi-Fi Direct. Moreover, they tend to have some other capabilities as well, such as support for address allocation, unlimited power supply etc.

Opportunistic networking can leverage the powerful capabilities and pervasive nature of modern mobile devices to provision services to nearby nodes. In case of Infrastructure-assisted opportunistic networks, the service discovery usually starts after the nodes have come into the physical communication range of the same access point and negotiated a radio contact through which communication can be established. The purpose of the service discovery is to find knowledge about existing service(s) and their contact information (e.g., protocols and ports) as well as access to the service [60]. In our work to enable the service provisioning in the opportunistic networks we use Avahi which is an implementation of multicast DNS (mDNS)/ DNS Service Discovery (DNS-SD) mechanism [82].

Transmission Control Protocol (TCP) [63] is the most popular transport layer protocol for point-to-point, connection-oriented, in-order, reliable data transfer in the internet. TCP was primarily designed for wired network. In our work, we have investigated how the characteristics specific to wireless networks affect the throughput of TCP connection and thus to have an understanding on its suitability as a transport layer protocol in opportunistic networks.

We carry out our experiments in an indoor wireless environment. That is why understanding the characteristics of indoor radio wave propagation is also a part of

our work as those characteristics may have some impacts on the the overall capability of the opportunistic contacts.

In this chapter, after providing a brief overview on opportunistic contacts (section 2.1), we continue our discussion with the lower layer physical medium characteristics i.e., indoor radio wave propagation characteristics (section 2.2) and subsequently go higher in the stack, i.e., we discuss IEEE 802.11 link layer connection establishment in section 2.3, layer-3 IP address acquisition in section 2.4, service discovery mechanism in section 2.5 and at last data transmission over TCP in wireless medium in section 2.6.

2.1 Opportunistic Contacts

Opportunistic networks are commonly defined as a type of networks where communication is challenged by sporadic and intermittent contacts as well as frequent disconnections and reconnections and where the assumption of the existence of an end-to-end path between the source and the destination is relinquished [88]. In opportunistic networking data exchanges between mobile devices take place based on the connection opportunities that arise whenever the devices happen to come into wireless range of each other or to the wireless range of the same access point due to the mobility of their users. Thus opportunistic networking focuses on the space where the devices take advantage of ephemeral contact opportunities [20]. As we are interested in characterizing the opportunistic contacts, the duration of these connection opportunities (i.e., contact duration) play an important role in our work.

Contact duration: The contact duration is the time interval for which two devices (i.e., stations) can communicate when they come into the range and it is an important factor in determining the capacity of opportunistic networks. It gives insight on how much data can be transferred at each opportunity. But as mentioned in section 1.1 before the actual data transfer can take place during a contact, the devices have to go through the different phases, e.g., connection establishment, IP address acquisition and service discovery. If these phases take the majority of the short contact time of an opportunistic contact, there would be very little time left for the actual data transfer. That is why in the Thesis work, we characterize the steps that a station goes through during the short duration of an opportunistic contact.

2.2 Indoor Radio Wave Propagation Characteristics

Radio Frequency (RF) communications are based on laws of physics that describe the behaviour of electromagnetic waves. Although the laws of physics describe very accurately the different aspects of electromagnetic wave propagation, it is vital to understand that the complexity of practial life makes the actual propagation loss very difficult to predict. In an ideal free space scenario the traveling wave can be described as propagating in a direct ray from the transmitter to the receiver. However, most applications in today's short range RF scene operate in environments that are far different from the ideal free-space scenario. Radio propagation in indoor systems is very complex to calculate and the main reason for this difficulty is that the

signal propagated from the transmitter antenna will experience many different signal transformations and paths with a small portion reaching the receiver antenna. The mechanisms behind indoor electromagnetic wave propagation are diverse, but can generally be attributed to reflection, diffraction and scattering [66]. The reflection, diffraction and scattering phenomena cause additional radio propagation paths to the direct LOS path between the transmitter and receiver. The term path loss or attenuation, fading are used in order to describe all the phenomena.

Reflection

Reflection occurs when a propagating electromagnetic wave impinges upon an object which has very large dimensions when compared to the wavelength of the propagating wave (Figure 1). The degree of reflection strongly depends on the physical attributes of the object and the signal properties. In general, when a radio wave impinges on another object or medium which has different electrical properties from previous medium, the wave is reflected and may be partially refracted unless the second medium has perfect conducting properties. Major contributors of reflection and refraction in indoor applications are walls, floors, ceilings, furniture etc. [65, p. 40]

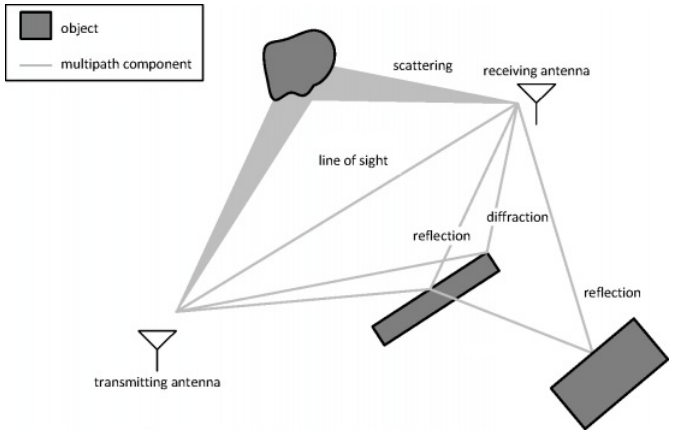


Figure 1: The reflection, diffraction and scattering phenomena. Picture adopted from [4].

Diffraction

Diffraction occurs at the edge of an impenetrable body that is large compared to the wavelength of the radio wave. When a radio wave encounters such an edge, waves propagate in different directions with the edge as the source [79, p. 116]. The secondary waves resulting from the obstructing surface are present throughout the space and even behind obstacle, giving rise to a bending of waves around the obstacle, even when a line-of-sight path does not exist between transmitter and receiver (Figure 1) [66].

Scattering

Scattering occurs when the wave length of transmitted signal is large compared to the object dimension. Small objects, rough surfaces usually are the major contributors of scattering the signals (Figure 1) [28].

2.2.1 Multipath Propagation

The phenomenon of reflection, diffraction and scattering all give rise to additional radio propagation paths beyond the direct line-of-sight (LOS) path between the radio transmitter and receiver. So the signal arriving at the receiver is in general a summation of both direct LOS and several multipath components. Reflection, diffraction, scattering cause radio signal distortions as well as additional signal propagation losses in a wireless communication system. The relative importance of these propagation mechanisms depends on the particular environment, e.g., if there is a direct LOS between terminals, then reflection dominates the propagation, whilst if the mobile is in a heavily cluttered area with no LOS path, diffraction and scattering usually play a major role. Combination of all these mechanisms is seen to account for all the observed effects in the radio wave propagation [16, p. 184].

When multiple signal propagation paths exist, caused by whatever phenomenon, the actual received signal level is vector sum of all the signals incident from any direction or angle of arrival. Some signals will aid the direct path, while other signals will subtract (or tend to vector cancel) from the direct signal path. The total composite phenomenon is thus called multipath. Two kinds of multipath exist: specular multipath that arises from discrete, coherent reflections from smooth metal surfaces and diffuse multipath that arises from diffuse scatterers and sources of diffraction. Both forms of multipath are bad for radio communications. Diffuse multipath provides a sort of background "noise" level of interference, while specular multipath can actually cause complete signal outages and radio "dead spots" within a building. This problem is especially difficult in underground passageways, tunnels, stairwells and small enclosed rooms [7].

2.2.2 Path Loss

Path loss or attenuation is the reduction in power density of an electromagnetic waves it propagates through space. The path loss elements include free-space loss, fading loss due to multipath and other miscellaneous effects based on frequency and the environment. Indoor path loss can change dramatically with either time or position because of the amount of multipath present and the movement of people, equipment, doors etc.

2.2.3 Fading

A unique characteristic in a wireless channel is a phenomenon called fading, the variation of the signal amplitude over time and frequency. Fading may either be due to multipath propagation, referred to as multi-path (induced) fading or to shadowing

from obstacles that affect the propagation of a radio wave, referred to as shadow fading.

The fading phenomenon can be broadly classified into two different types: large-scale fading and small-scale fading.

Large-scale fading: Large-scale fading occurs as the mobile moves through a large distance, for example, a distance of the order of cell size. It is caused by path loss of signal as a function of distance and shadowing by large objects. In other words, large-scale fading is characterized by average path loss and shadowing.

Small-scale fading: Small-scale fading refers to rapid variation of signal levels due to the constructive and destructive interference of multiple signal paths when the mobile station moves short distances. Depending on the relative extent of a multipath, frequency selectivity of a channel is characterized by frequency-selective or frequency flat for small-scaling fading. Meanwhile, depending on the time variation in a channel due to mobile speed (characterized by the Doppler spread) short-term fading can be classified as either fast fading or slow fading [8].

2.2.4 Interference

While an understanding of indoor propagation is essential, another important element of indoor wireless operation that should be considered is interference. Unlike outdoor environments, where the operating distances are greater, in an indoor environment, it is common to have an interfering system operating within a few feet or less of a given system. It is important to account for this interference and to understand that communication link problems in indoor environments may not be propagation issues, but rather interference issues [76, p. 208].

2.2.5 Communication Link and Range

The principal characteristics of an indoor RF propagation environment that distinguish it from an outdoor environment are that the multipath is usually severe, a line-of-sight path may not exist and the characteristics of the environment can change drastically over a very short time or distance. Walls, doors, furniture and people can cause significant signal loss. Quantifying the range performance of RF communication systems can be difficult. In order to estimate the transmission range four factors must be considered:

Transmit power: The power that is broadcast by the transmitter. This is usually measured in watts (W), milliwatts (mW) or in dBm. In US and Canada maximum transmit power for 2.4 GHz band is 1 W (specified by FCC), whereas in Europe, ETSI specifies 100 mW maximum power for the 2.4 GHz band [35].

Receiver sensitivity: It is a measure of the ability of a receiver to demodulate and get information from a weak signal. Receiver sensitivity is quantified as the

lowest signal power level from which a receiver can get useful information.

Antenna gain: The gain of an antenna (in any given direction) is defined as the ratio of the power gain in a given direction to the power gain of a reference antenna in the same direction. It is standard practice to use an isotropic radiator as the reference antenna in this definition. An isotropic radiator is a hypothetical lossless antenna and it would radiate its energy equally in all directions. That means that the gain of an isotropic radiator is G = 1 (or 0 dB). It is customary to use the unit dBi (decibels relative to an isotropic radiator) for gain with respect to an isotropic radiator [25].

Path loss: The signal decrease that occurs as the radio waves travel through the air or through obstacles. Path loss or attenuation of RF signals occurs naturally with distance. Obstacles between the transmitter and receiver also attenuate signals. The amount of attenuation varies with the frequency of the RF signal and the obstructing materials‘ type and density. It also depends on antenna height, receive terminal location relative to obstacles and reflectors.

Knowing how strong the communication link is or just how close a system is to failure can be important in some situations. Link margin is a parameter that is used to measure how close the link is to failing. It is the difference between the system gains and the system losses. Successful communication takes place when the link margin is greater than zero.

$$\begin{aligned}\text{Link Margin} =& \text{Transmit Power} - \text{Receiver Sensitivity} \\ &+ \text{Antenna Gain} - \text{Path Loss}\end{aligned}$$

In line-of-sight conditions, every 6 dB of link margin will double the transmission range. Figuring out the range for non line-of-sight and indoor communication systems is a lot more difficult and can involve a lot of obstructions and variables. The different obstacles and materials that are found in typical indoor environments make it difficult to determine the actual path loss in a given situation [3].

2.3 Connection Establishment in IEEE 802.11 Networks

In an opportunistic contact, at first wireless nodes have to go through the steps of link layer connection establishment procedure. IEEE 802.11 Wi-Fi facilitates the link layer connection set-up phase. This section focuses on IEEE 802.11 Wi-Fi specification and later it describes the overall procedure of the connection establishment in an Infrastructure-assisted Wi-Fi network.

2.3.1 Wi-Fi Network (IEEE 802.11 Standard)

The IEEE 802.11 refers to a family of specifications developed by the IEEE for over-the-air interface between a wireless client and an access point (AP) or between two wireless clients [36]. The IEEE 802.11 standard covers the physical (layer 1) and

data link (layer 2) layers of the OSI Model. To be called 802.11 device, a device must conform the medium access control (MAC) and physical layer specifications of IEEE 802.11 standard [17]. A Wi-Fi network, in reality, is a network that complies with the IEEE 802.11 standard.

Standard	Year Introduced	Frequency Band (GHz)	Maximum Data Rate (Mbps)	Modulation
802.11	1997	2.4	2	DSSS, FHSS
802.11a	1999	5	54	OFDM
802.11b	1999	2.4	11	DSSS (CCK)*
802.11g	2003	2.4	54	DSSS, OFDM
802.11n	2009	2.4/ 5	600	OFDM

* CCK is the modulation scheme used in 802.11b to achieve higher data rates (i.e., 5.5 Mpbs, 11 Mbps)

Figure 2: IEEE 802.11 Standards

The IEEE 802.11 standard is the earliest standard, allowing 1-2 Mbps of bandwidth. Amendments have be made to the original standard in order to optimize bandwidth or to better specify components in order to ensure improved security or compatibility. These newer standards include 802.11a, 802.11b, 802.11g, 802.11n etc. (Figure 2). All the 802.11 Wi-Fi standards operate within the ISM (Industrial, Scientific and Medical) frequency bands (Information on ISM frequency bands in [30]). No license is required for operation within the ISM frequency bands. This makes them ideal for a general system for widespread use [10].

IEEE 802.11 Frames

The 802.11 link layer is much more complicated than the Ethernet one. The main reason is that wireless links have lower reliability compared to the reliability of wired links and therefore the 802.11 link layer has features to reduce the effects of frame loss. For example, every data frame is acknowledged with an acknowledgement (ACK) frame. Moreover, the protocol needs to support access point discovery, association and disassociation, authentication, wired/ wireless bridging and many other features that are not necessarily needed in a wired link layer [71]. The 802.11 standard defines three major categories of frames. The major types and their subtypes are as follows [42]:

Data frames: The main purpose of having a wireless LAN is to transport data. Data frames are used to carry packets from higher layers (e.g., web pages, printer control data, etc.) within the body of the frame.

Control frames: 802.11 control frames assist in the delivery of data frames between stations. They are used to improve the reliability characteristics of the link. The following are common 802.11 control frame subtypes:

- Acknowledgement

- Request to Send
- Clear to Send

Management frames: : 802.11 management frames enable stations to establish and maintain communications. The following are common 802.11 management frame subtypes:

- Beacons
- Probe Request/ Probe Response
- Authentication/ Deauthentication
- Association Request/ Association Response
- Reassociation Request/ Reassociation Response
- Disassociation

IEEE 802.11 Operational Modes

The 802.11 specification defines two types of operational modes: Ad-hoc mode and Infrastructure mode.

Figure 3: Ad-hoc Mode. Picture adopted from [9].

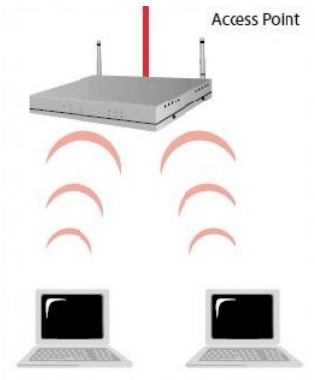


Figure 4: Infrastructure Mode. Picture adopted from [9].

Ad-hoc mode: In the Ad-hoc mode, each station is a peer to the other stations and communicates directly with other station(s) within the network (Figure 3). No

access point is involved. All stations can send beacon and probe frames. The Ad-hoc mode stations form an Independent Basic Service Set (IBSS) [17].

Infrastructure mode: In the Infrastructure mode, an access point is used for all communications between the stations in the same service area (Figure 4). A Basic Service Set (BSS) is a set of stations that are logically associated with each other and controlled by a single AP. If one mobile station in an Infrastructure BSS needs to communicate with a second mobile station, the communication must take two hops. First the originating mobile station transfers the frame to the access point. Second, the access point transfers the frame to the destination station [32]. Every BSS has an ID called the BSSID (Basic Service Set Identifier), which is the MAC address of the access point servicing the BSS and a text identifier called the SSID (Service Set Identifier). Most products refer to the SSID as the network name because the the string of bits is commonly set to a human-readable string. Each BSS operates on a particular channel, i.e., the access point and all of the wireless clients within a BSS communicate over a common channel. The radio card of the mobile station automatically tunes its transceiver to the frequency of the access point [40].

IEEE 802.11b Specification

Throughout the Thesis work we have used the IEEE 802.11b standard. So here 802.11b has been highlighted with some of its features. IEEE 802.11b is an amendment to the IEEE 802.11 wireless networking specification that extends throughput from 2 Mbps to 11 Mbps using the same 2.4 GHz band. The basic architecture, features and services of 802.11b are defined by the original 802.11 standard. The 802.11b specification affects only the physical layer adding higher data rates and more robust connectivity. The dramatic increase in throughput of 802.11b (compared to the original standard) along with simultaneous substantial price reductions led to the rapid acceptance of 802.11b as the definitive wireless LAN technology. Although 802.11b cards are specified to operate at a basic rate of 11 Mbps, the system monitors the signal quality. If the signal falls or interference levels rise, then it is possible for the system to adopt a slower data rate in order to decrease the rate of re-broadcasts that result from errors. Under these conditions the system will first fall back to a rate of 5.5 Mbps, then 2 and finally 1 Mbps. This scheme is known as Adaptive Rate Selection (ARS) [74].

IEEE 802.11b Frequency Usage

There is a total of fourteen channels defined for use by Wi-Fi 802.11b. Depending on the country a user lives in and where he or she will be installing a WLAN, there are certain governmental restrictions. In North America, the FCC (Federal Communications Commission) and IC (Industry Canada) allow manufacturers and users to use channels 1 through 11, per ETSI (European Telecommunications Standards Institute) approval most of Europe can use channels 1 through 13, while in Japan, users have all 14 channels available [9].

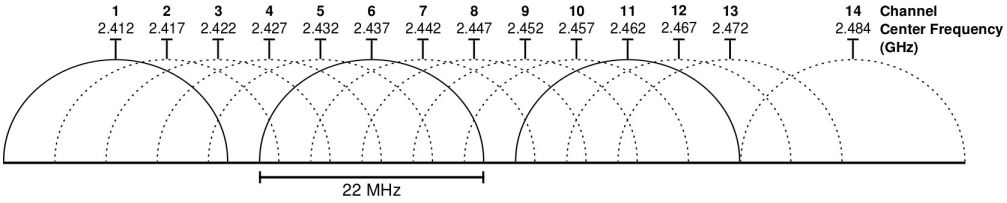


Figure 5: 802.11b channels in 2.4 GHz band. Picture adopted from [2].

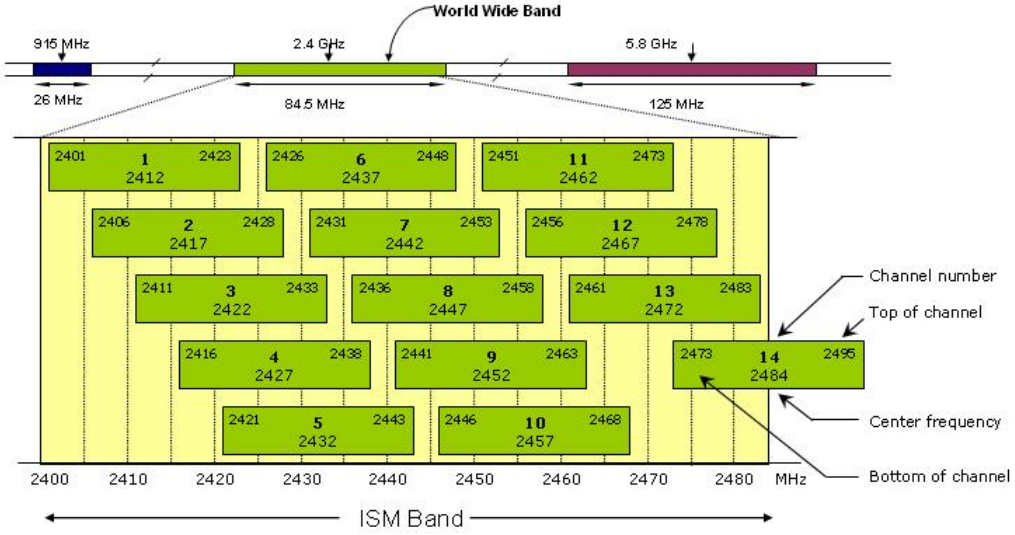


Figure 6: Non-overlapping channels in 802.11b standard. Picture adopted from [27].

In Europe, the frequency ranges from 2.401 to 2.483 GHz is divided into 13 channels of 22 MHz wide, and spaced 5 MHz between them, where channel 1 is centered on 2.412 GHz and the channel 13 is located at 2.472 GHz (Figure 5) [74].

The channels used for Wi-Fi are separated by 5 MHz but have a bandwidth of 22 MHz. As a result channels overlap and it is possible to find a maximum of three non-overlapping channels (Figure 6). Therefore if there are adjacent pieces of WLAN equipment that need to work on non-interfering channels, there is only a possibility of three. There are five combinations (sets) of non overlapping channels possible. From figure 6 it can be seen that Wi-Fi channels 1, 6, 11 or 2, 7, 12 or 3, 8, 13 or 4, 9, 14 (if allowed) or 5, 10 (and possibly 14 if allowed) can be used together as sets. Often Wi-Fi access points are set to channel 6 as the default and therefore the set of channels 1, 6 and 11 is possibly the most widely used [11].

2.3.2 Connection Establishment Procedure in Infrastructure-assisted Wi-Fi Network

One of the two modes of operation defined in Wi-Fi is the Infrastructure mode and this is the mode that has been used throughout the Thesis work (reason behind such selection is discussed in chapter 2). On these widely used infrastructure-based networks, each station communicates via access point. If a station wishes to send or receive data to another station, it first needs to associate with an access point. The access point acts as a bridge and forwards data packets to appropriate destination. Similarly, all the data packets targeted to stations are passed through their respective access points. The whole process, a station being associated with an access point, includes several phases: Scanning, Joining, Authentication and Association.

Scanning (Network Discovery)

In wireless network, stations must identify/discover a compatible network before joining it. The process of discovering wireless networks in the area is called scanning. The IEEE 802.11 standard defines two types of scanning procedures: Passive scanning and Active scanning.

Passive Scanning: In wireless networks, access points contend with other wireless devices to gain access to the wireless medium and periodically broadcast Beacon frames to announce its presence and relay information, such as timestamp, SSID, supported rates, capability information etc. In the passive scan mode, the station listens to each channel of the physical medium one by one for the Beacon frames, in an attempt to locate potential access points. Using information obtained from the Beacon frames, the station selects an access point to associate with [70].

Active Scanning: In active scanning, the station actively searches for the network to join. This process involves the exchange of probe frames. Rather than listening for Beacon frames, a station wishing to join a network will broadcast Probe Request frames on each channel and wait for Probe Responses from access points

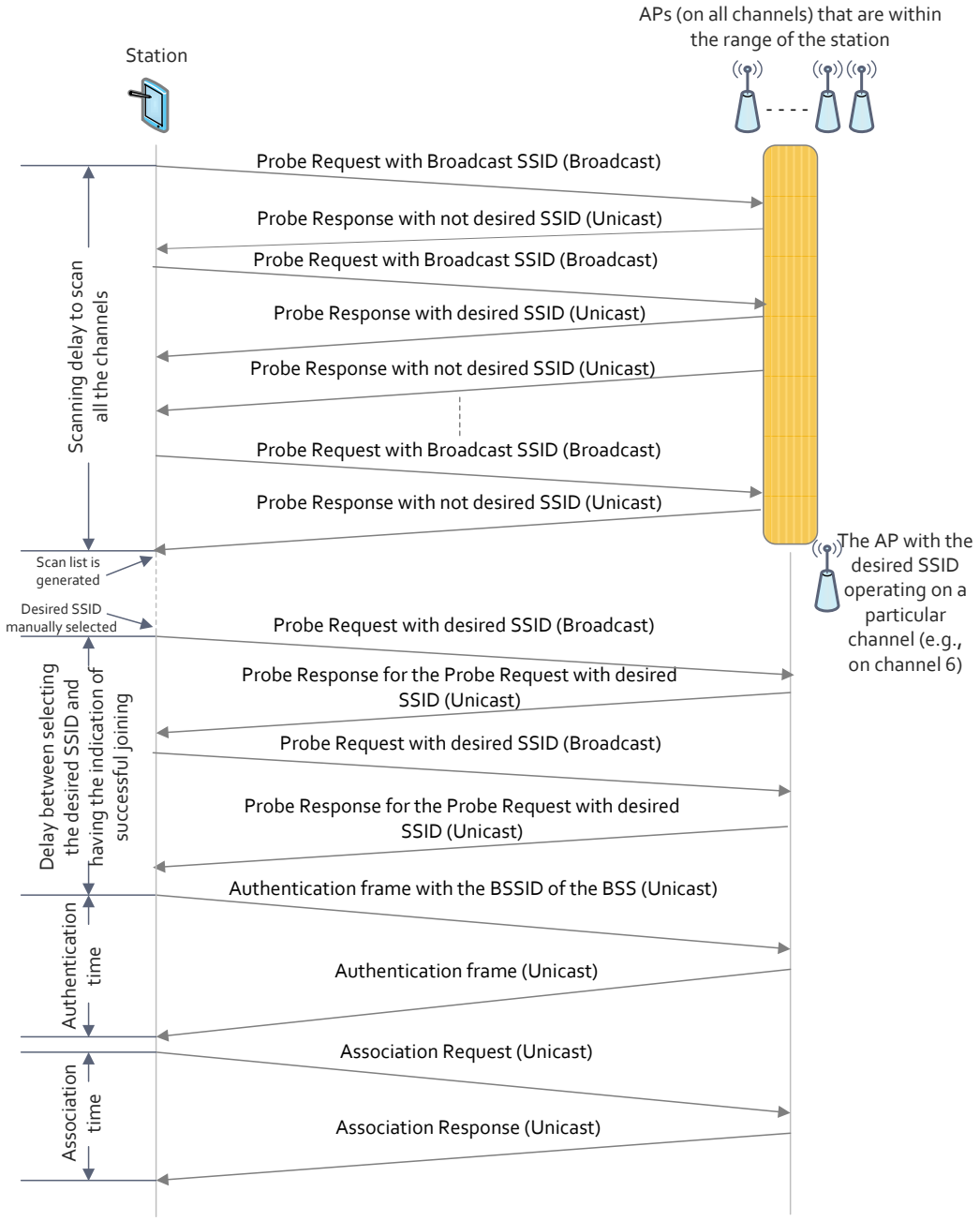


Figure 7: IEEE 802.11 link layer connection establishment procedure (The diagram shows Active Scanning, Joining, Authentication and Association phases).

(Figure 7). The IEEE 802.11 standard [37] defines two timers, namely MinChannelTime and MaxChannelTime, to determine the time a station needs to wait on a channel after having sent a Probe Request. The station sends a Probe Request packet on each probed channel and waits MinChannelTime for a Probe Response packet from each reachable access point. If at least one Probe Response packet is received, the station extends the sensing interval to MaxChannelTime in order to obtain more responses. On the contrary, if during MinChannelTime the station does not receive any Probe Response on the channel, the station considers that channel empty and starts scanning the next channel [37]. Thus being controlled by two timers the waiting time on each channel is irregular (whereas in passive scanning the waiting time is prearranged) and it makes the scanning process faster[70].

When the active scanning starts, the station starts sending Probe Request frames with broadcast SSID (Broadcast Probe). Access points receiving Probe Request frames from the station responds with Probe Response only if the SSID in the Probe Request is the broadcast SSID or matches the specific SSID of its own. Probe Response frames will be sent as directed frames to the address of the station that generated the Probe Request (Figure 7) [37] .

The scanning process requires wireless station to switch between and scan each channel independently. A scan report is generated at the conclusion of scanning process. The report lists all the discovered BSSs and the parameters (BSSID, SSID, BSSType, Beacon interval, Timing parameters, PHY parameters, BSSBasicRateSet etc.) associated with those BSSs. The complete parameter list enables the scanning station to send join request to any of the networks that it discovered.

Active scanning scans access points faster though it consumes more power than passive scanning [70, 77]. In opportunistic communications, network discovery is a vital part and scanning latency plays a key role on the overall capability of the opportunistic contacts. That is why in our approach, we have used the faster scanning mode, i.e., active scanning.

Joining

After compiling the scan results, a station can elect to join one of the BSSs. Joining is a precursor to association. It does not enable network access. Before this can happen, both authentication and association are required.

Choosing which BSS to join is an implementation-specific decision and may involve user intervention. This selection process involves broadcasting of a Probe Request but this time with an specific SSID (Directed Probe). The access point that has the same SSID responds with the Probe Response frame (Figure 7). It cannot be told exactly when the station has joined the network because the joining process is internal to a node and it involves matching local parameters to the parameters required by the selected BSS. At this stage one of the most important tasks is to synchronize timing information between the station and the rest of the network. Some other important parameters are PHY parameters, BSSID etc. PHY parameters guarantees that any transmissions with the BSS are on the right channel. Using the BSSID ensures that transmission are directed to the correct set of

stations and ignored by stations in another BSS [32].

Authentication

Having ended with the Scanning and Joining phases, the station starts the next phase of connection establishment: Authentication. It is a process whereby the selected access point either accepts or rejects the identity of the station. According to IEEE 802.11 specifications [37] there are two authentication methods: Open System authentication and Shared Key authentication.

Open System authentication utilizes a two-message authentication transaction sequence. The first message originated by the station asserts identity and requests authentication. The second message originated by the access point returns the authentication result (Figure 7). If the authentication frame from the access point is received with a status value of "successful", the station and the access point are then declared mutually authenticated [37].

Shared Key authentication method assumes the existence of a secret key shared between the station and the access point represented by a Wired Equivalent Privacy (WEP) key. An extra two packets (challenge - response) are exchanged during the authentication phase, in which the station must decrypt a text provided by the access point. The method of Shared-Key Authentication requires therefore the exchange of four messages [37, 70]. As in opportunistic networking the intention is to utilize the public hotspots, this method of Shared Key authentication is ignored in the Thesis work.

Association

Once authenticated, the station sends an Association Request to the access point to associate itself with the access point. When the access point receives the Association Request frame from the already authenticated station, it transmits an Association Response with a status value of "successful". Also an Association ID (AID) assigned to the station will be added in the response. When the station receives the Association Response from the access point with a status value of "successful", the station is now associated with the access point (Figure 7) [37].

2.4 IP Address Acquisition

The successful association to the access point indicates the completion of link layer connection establishment procedure. But to communicate with the other device(s) in the network the station needs a layer-3 IP address (Internet Protocol address) [47].

2.4.1 Internet Protocol (IP)

Internet Protocol (IP) protocol works at Network layer of OSI model and at Internet layer of TCP/IP model. When the data segments from Transport layer are passed to Network/ Internet layer, Internet Protocol encapsulates those segments into IP

datagrams (or packets) [46]. This protocol has the responsibility of identification of the hosts based on their logical addresses and to route data among them over the underlying network. IP provides a mechanism to uniquely identify a device on an IP network by IP addressing scheme. IP version 4 (IPv4) addresses are 32 bits in length and are typically communicated in a format known as dotted decimal. IP version 6 (IPv6) is the next generation of IP addressing. IPv6 quadruples the number of network address bits from 32 bits (in IPv4) to 128 bits, which provides enough globally unique IP addresses for every networked device on the planet. IPv6 is an important protocol for the future of IP networking [26]. The version used in our work is IPv4 because still it is the most widely deployed Network/Internet layer protocol.

There are different ways how an IP address is allocated to a device, e.g., Static allocation, Dynamic allocation.

Static allocation: Static IP addresses are manually assigned to a device by a user or by an administrator.

Dynamic allocation: Dynamic allocation is the automatic assignment of an IP address to a device. There are different methods of dynamic IP address allocation: DHCP (Dynamic Host Configuration Protocol) and link-local auto-configuration. DHCP provides a means for a computer or other device to ask a central server (i.e., a DHCP server) for an IP address suitable for use on the network [15]. In case of link-local auto-configuration the device assigns itself an IP address from a link-local range.

2.4.2 DHCP Procedure

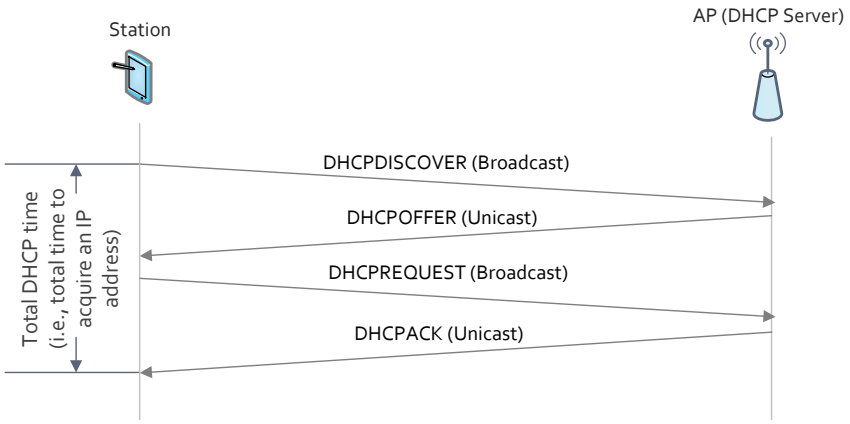


Figure 8: IP acquisition through DHCP.

In our work, we have used DHCP as the IP address allocation mechanism for the station as it is most commonly used mechanism in Wi-Fi networks. We also

configure the access point (AP) to operate as a DHCP server. Here we discuss on the overall procedure of IP acquisition through DHCP.

IP address acquisition through DHCP begins with the station broadcasting a DHCP-DISCOVER packet (Figure 8) when the station is configured to obtain IP address automatically. The DHCPDISCOVER packet includes options that suggest values for the IP address and lease duration. This packet informs the DHCP server (AP) in the network regarding the presence of a new node. In response to the DHCPDISCOVER, the DHCP server (AP) sends DHCPOFFER packet to the station that includes an available IP address. When allocating a new address, the server should check that the offered IP address is not already in use, e.g., the server may probe the offered address with an ICMP Echo Request.

When a DHCPOFFER packet is received, the station broadcasts a DHCPREQUEST packet requesting the offered IP Address. The DHCP server (AP) responds with a DHCPACK packet containing the configuration parameters for the requesting station. If the DHCP server is unable to satisfy the DHCPREQUEST packet (e.g., the requested IP address has already been allocated), it should respond with a DHCPNAK packet.

When the station receives the DHCPACK packet with configuration parameters, it may perform a final check on the parameters (e.g., ARP for allocated IP address) and notes the duration of the lease specified in the DHCPACK packet. At this point, the station is configured with the newly allocated IP address. If the station detects that the address is already in use (e.g., through the use of ARP), it must send a DHCPDECLINE packet to the server and restarts the configuration process. The station should wait a minimum of ten seconds before restarting the configuration process to avoid excessive network traffic [29, 75]. Once an IP address is acquired, communication between the station and other station(s) of the network can commence.

2.5 Service Discovery Mechanism

In our work, we consider opportunistic networking as a potential provider of services. There are different service discovery protocols, including, Avahi, Bonjour, UPnP (Universal Plug and Play), Jini etc. Service discovery protocols are network protocols which allow automatic detection of devices and services offered by these devices on a network. To enable the service provisioning feature, we have adopted the Avahi service discovery mechanism in our work. Avahi is a Zero Configuration Networking (Zeroconf) implementation for Linux and BSDs, including a system for multicast DNS (mDNS)/ DNS Service Discovery (DNS-SD) [82, 83, 48, 49]. It allows programs to publish and discover services and hosts running on a local network with no specific configuration. The Zeroconf working group‘s requirements and proposed solutions for Zero Configuration Networking over IP essentially cover three areas [83]:

- Addressing (allocating IP addresses to hosts)
- Naming (using names to refer to hosts instead of IP addresses)

- Service discovery (finding services on the network automatically)

Avahi has a Zeroconf solution for all three of these areas. This section includes the discussion on these three areas that is followed by the architectural overview of Avahi.

2.5.1 Addressing

Avahi has a mechanism of self-assigned link-local IP addressing. Link-local is intended for two main scenarios: i) for tiny Ad-hoc networks where communication is desired without the overhead of setting up a DHCP server and ii) to provide a minimum safety-net level of service on networks where there is supposed to be a DHCP server but it is failed [23, p. 21]. When the IP address option in a host is set to link-local IP, the host selects an address from the range of 169.254.1.0–169.254.254.255. After the selection of an IPv4 link-Local address, the host must test to see if it is already in use before starting to use it. This test is done by broadcasting the ARP probes. If there is any address conflict, the host will select a new address from the 169.254.1.0–169.254.254.255 range and repeat the testing process [22]. In case of IPv6 address selection, a link-local address is formed by combining the link-local prefix FE80::0 with an interface identifier. To ensure the uniqueness of the IPv6 address the IPv6 node runs a "duplicate address detection" algorithm on the address before assigning it to an interface [55]. In our work, we have not used the link-local addressing feature of Avahi. The stations in our work acquire the IP address through DHCP as it is the widely deployed mechanism in public hotspots. Details of the IP acquisition through DHCP are discussed in section 2.4.2.

2.5.2 Naming

This section is about the name-to-address translation. Identifying endpoints on the network by name rather than by address provides operational stability when an endpoint's address changes because its name remains the same. For name-to-address translation on a local network Avahi uses Multicast DNS (mDNS), in which DNS-format queries are sent over the local network using IP multicast [48]. Because these DNS queries are sent to a multicast address, no single DNS server with global knowledge is required to answer the queries. Each service or device can provide its own DNS capability. When it sees a query for its own name, it provides a DNS response with its own address.

For name-to-address translation to work properly, a unique name on the local network is necessary. Multicast DNS (mDNS) uses .local top level domain for names in the local link. As the .local domain is not managed by a central station conflicts may occur. So when joining the network, the node has to claim a name in the .local domain to check whether the name is already taken on the network or not. To avoid the conflicts mDNS follows two steps: Probing and Announcing [48].

When the host wants to claim its name, it first probes in the network by sending a multicast query message. But before sending any multicast messages the node that has IPv4 address needs to report its IP multicast group membership to neighboring

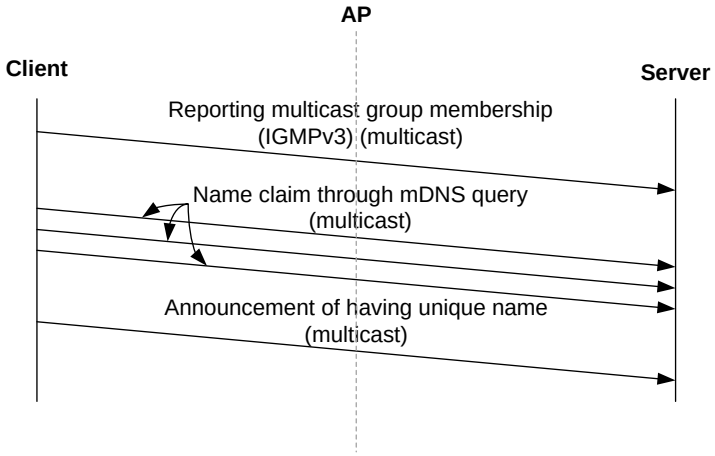


Figure 9: Reporting of multicast group membership and name reservation.

multicast nodes by sending IGMPv3 message (Figure 9) [19]. After sending the IGMPv3 message the node now can do the probing by sending multicast query message. The query message asks for any record types assigned to the name it desires. 250ms after the first query the host sends a second, then 250ms after that a third query message. If, by 250ms after the third probe, no conflicting Multicast DNS responses have been received, the host moves to the next step: announcing. In this step, the host sends an unsolicited Multicast DNS response containing all the Resource Records it now claims ownership of. In the case of records that have been verified to be unique, they are placed into the Answer Section of the DNS Response with the most significant bit (mDNS "cache-flush" bit) of the rrclass set to one. In case of name conflicts, the host has to assume that another host in the network already has taken the desired name and it must choose another name [48].

2.5.3 Service Discovery

The final element of Avahi is service discovery. Service discovery allows applications to find all available instances of a particular type of service and to maintain a list of named services and port numbers. The application can then resolve the service hostname to IP address. Avahi service discovery mechanism is based on DNS Service Discovery (DNS-SD). In our work DNS-SD is used with mDNS.

According to the RFC 6763 [49], in DNS-SD, given a type of service that a client is looking for, and a domain in which the client is looking for that service, this allows clients to discover a list of named instances of that desired service, using standard DNS queries. This service-centric approach is very different from the traditional device-centric idea of network services where the network consists of a number of devices or hosts, each with a set of services. For example, the network might consist of a server machine and several client machines. In a device-centric browsing scheme, a client queries the server for what services it is running, gets back a list (e.g., FTP,

HTTP and so on) and decides which service to use. The interface reflects the way the physical system is organized. But this is not necessarily what the user logically wants or needs. Users typically want to accomplish a certain task, not query a list of devices to find out what services are running. The device-centric approach is not only time-consuming, it generates a tremendous amount of network traffic, most of it useless. The service-centric approach sends a single query, generating only relevant replies.

In addition, DNS-SD has a service-oriented view. In DNS-SD, queries are made according to the type of service needed (not according to the hosts providing them). Applications that use DNS-SD store service names, not address. So if the IP address, port number or even host name gets changed, the application can still connect.

2.5.4 Architectural Overview of Avahi

The architecture of network services in Avahi includes service publication, service browsing and service resolution. Here we discuss on these three fundamental operations of Avahi.

Service Publication

To publish a service an application or a device registers the service with the Avahi-daemon (Multicast DNS responder). After the registration of the service, the Avahi-daemon starts advertising the service in the network (Figure 10). When a service is registered, three related DNS records are created: a service (SRV) record, a pointer (PTR) record and a text (TXT) record [49][14].

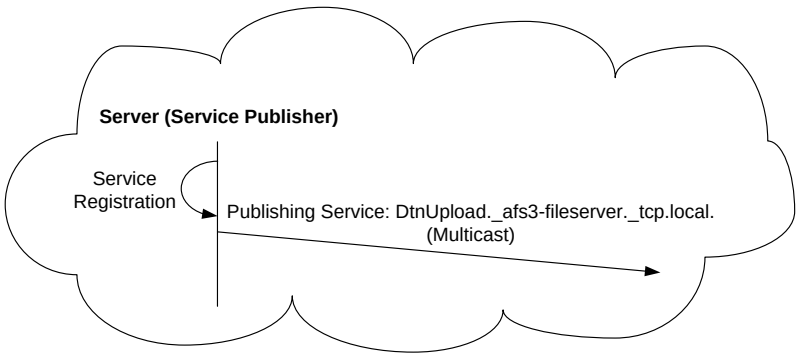


Figure 10: Service publication.

Service Records: The SRV record maps the name of the service instance to the information (i.e., host name and port number) needed by a client to actually use the service. Clients then store the service name as a persistent way to access the service and perform a DNS query for the host name and port number when it's time to connect. This additional level of indirection provides for two important features. First, the service is identified by a human-readable name instead of a domain name

and port number. Second, clients can access the service even if its port number, IP address, or host name changes, as long as the service name remains the same.

The SRV record contains host name and port number to identify a service. The host name is the domain name where the service can currently be found. The reason a host name is given instead of a single IP address is that it could be a multi-homed host with more than one IP address, or it could have IPv6 addresses as well as IPv4 addresses and so on. Identifying the host by name allows all these cases to be handled gracefully. The port number identifies the UDP or TCP port for the service [49][14].

SRV records are named according to the following convention: <Instance Name>. <Service Type>.<Domain>

<Instance Name>, the name of a service instance, can be any UTF-8-encoded Unicode string, and is intended to be human readable. In our work we have used DtnUpload as the Service Instance Name.

<Service Type>is a standard IP protocol name, preceded by an underscore, followed by the host-to-host transport protocol (TCP or UDP), also preceded by an underscore. For example, a Trivial FTP service running over UDP would have a service type of _tftp._udp. In our work Service Type is _afs3-fileserver._tcp.

<Domain>is a standard DNS domain. This may be a specific domain, such as aalto.fi., In our case the domain is local. as the service is accessible only on the local link.

In our work, the following SRV record for the service named DtnUpload (running on TCP port 7000) is created on the skroy-Latitude-D830.local. host (server), DtnUpload._afs3-fileserver._tcp.local. 120 IN SRV 0 0 7000 skroy-Latitude-D830.local. Here, the initial 120 represents the time-to-live (TTL) value which is used for caching. The two zeros are weight and priority values, used in traditional DNS when choosing between multiple records that match a given name. For multicast DNS purposes, these values are ignored.

Pointer Records: PTR records enable service discovery by mapping the type of the service to a list of specific service instance names of that type of service [49][14]. The PTR Record from our work is as follows, _afs3-fileserver._tcp.local. 4500 PTR DtnUpload._afs3-fileserver._tcp.local. Here, the 4500 is the time-to-live (TTL) value, measured in seconds.

Text Records: The TXT record has the same name as the corresponding SRV record and can contain a small amount of additional information about the service instance, typically no more than 100-200 bytes at most. This record may also be empty. Historically, this record has been used for multiple service running on the same port at the same IP address. For example, multiple print queues running on the same print server. In this case, the TXT record can be used to identify the intended print queue [49][14].

Service Browsing

Service browsing makes use of the DNS records registered during service publication to find all named instances of particular type of service. To do this an appli-

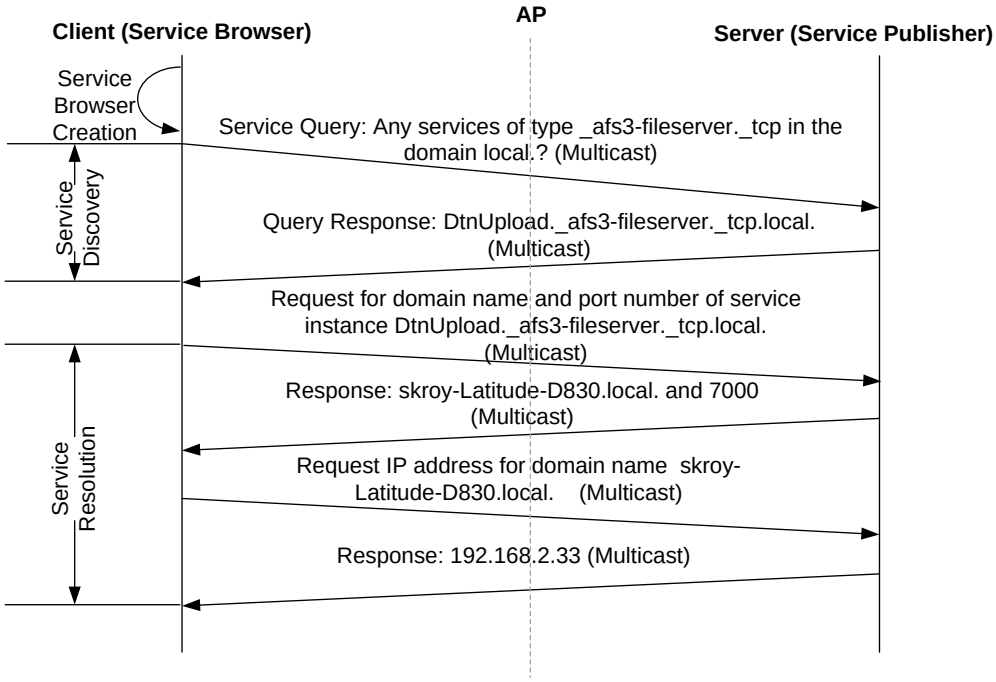


Figure 11: Service browsing and service resolution.

cation performs query for PTR Records matching a service type [49][14]. Figure 11 shows the client node querying for the service instances of service type _afs3-fileserver._tcp. Avahi Daemon running on the server returns PTR records with service instance name DtnUpload._afs3-fileserver._tcp.local.

Service Resolution

After the service browsing phase the client has service instance name(s) of the desired service type. But to use that service the client requires more information (e.g., Port, IP address). To resolve a service the application at the client node performs a DNS lookup for a SRV Record with the name of the service. The Avahi Daemon of the server side responds with the SRV record containing the required information [49][14]. But the whole resolution process requires several steps to go through.

Figure 11 shows that, the service resolution process starts with a Multicast DNS query asking for the DtnUpload._afs3-fileserver._tcp.local. SRV record. This query returns the service's host name (skroy-Latitude-D830.local.) and port number (7000). Then the client sends out a multicast request for the IP address. This request resolves to the IP address of the host (server) 192.168.2.33 . When the address is resolved, the client can use the IP address and port number to connect to the service. This process takes place each time the service is used, thereby always finding the service's most current address and port number.

Traffic Reduction: DNS-SD has an efficient traffic reduction mechanism in it [49]. When Avahi Daemon (Multicast DNS Responder) at the server node answers a query for a PTR record for a specific service, it creates additional records in the Additional Section of the DNS Message. Generally when a client requests a PTR record it is very likely that it will request shortly at least once for SRV and TXT Resource Record for a given instance name and for the appropriate address records. So without waiting for the SRV and TXT queries from the client, the responder adds all SRV and TXT records for the instance name in the PTR resource data and all address records mentioned in any SRV records and thus the traffic reduced in the network [49].

2.6 Data Transmission over TCP in Wireless Medium

Opportunistic contacts may last from seconds to several minutes. During this short connection window, nodes get connected with each other via an access point and transfer data. Maximizing data transfer within the short period is one of the key issues in opportunistic contacts. Currently, the Transmission Control Protocol (TCP) is the most popular transport layer protocol for connection-oriented, in-order, reliable data transfer [63, 51]. In this section, we discuss on the basic mechanism of TCP as well as on its suitability as a transport layer protocol in wireless opportunistic networks.

2.6.1 Basic Operation of TCP

TCP is a connection-oriented, end-to-end reliable protocol. TCP provides reliable inter-process communication between pairs of processes in host computers attached to distinct but interconnected computer communication networks. So the primary purpose of the TCP is to provide a reliable logical circuit or connection service between pairs of processes and to provide this service on top of a less reliable communication system (e.g., Internet) requires facilities in the following areas [63]:

- Basic Data Transfer
- Reliability
- Flow Control
- Multiplexing
- Connections

The basic operation of the TCP in each of these areas is described below.

Basic Data Transfer: The TCP is able to transfer a continuous stream of octets in each direction between its users by packaging some number of octets into segments for transmission through the network system. In general, the TCPs decide when to block and forward data at their own convenience.

Reliability: The TCP must recover from data that is damaged, lost, duplicated or delivered out of order by the internet communication system. This is achieved by assigning a sequence number to each octet transmitted and requiring a positive acknowledgment (ACK) from the receiving TCP. If the ACK is not received within a timeout interval, the data is retransmitted. At the receiver, the sequence numbers are used to correctly order segments that may be received out of order and to eliminate duplicates. Damage is handled by adding a checksum to each segment transmitted, checking it at the receiver and discarding damaged segments. As long as the TCPs continue to function properly and the network system does not become completely partitioned, no transmission errors will affect the correct delivery of data. TCP recovers from communication system errors.

Flow Control: TCP provides a means for the receiver to govern the amount of data sent by the sender. This is achieved by returning a "window" with every ACK indicating a range of acceptable sequence numbers beyond the last segment successfully received. The window indicates an allowed number of octets that the sender may transmit before receiving further permission.

Multiplexing: To allow for many processes within a single Host to use TCP communication facilities simultaneously, TCP provides a set of addresses or ports within each host. Concatenated with the network and host addresses, it forms a

socket. A pair of sockets uniquely identifies each connection, i.e., a socket may be simultaneously used in multiple connections. The binding of ports to processes is handled independently by each host.

Connections: The reliability and flow control mechanisms described above require that TCPs initialize and maintain certain status information for each data stream. The combination of this information, including sockets, sequence numbers, and window sizes, is called a connection. Each connection is uniquely specified by a pair of sockets identifying its two sides. When two processes wish to communicate, their TCP's must establish a connection (initialize the status information on each side) first. When their communication is complete, the connection is terminated or closed to free the resources for other uses. Since connections must be established between unreliable hosts and over the unreliable internet communication system, a handshake mechanism with clock-based sequence numbers is used to avoid erroneous initialization of connections.

2.6.2 TCP Performance in Wireless Environment

TCP was primarily designed for wired networks. In a wired network, random bit error rate (BER), a characteristic usually more pronounced in the wireless network, is negligible and congestion is the main cause of packet loss.

Each TCP packet is associated with a sequence number and only successfully received in-order packets are acknowledged to the sender by the receiver, by sending corresponding packets (acknowledgements) with sequence numbers of the next expected packets. On the other hand, packet loss or reception of out-of-order packets indicates failures. To eradicate such failures, TCP implements flow control and congestion control algorithms based on the sliding window and additive increase multiplicative decrease (AIMD) algorithms [84, 24]. Based on the assumption that packet losses are signals of network congestion, the additive increase multiplicative decrease congestion control of the standard TCP protocol reaches the steady state, which reflects the protocol's efficiency in terms of throughput and link utilization. However, this assumption does not hold when the end-to-end path includes wireless link. Factors such as high BER, unstable channel characteristics (e.g., MAC layer contention and collision, interference etc.) and user mobility may all contribute to the packet losses in wireless networks. So TCP yields poor performance when it interprets such losses in wireless links as congestion and consequently invokes congestion control mechanism [34, 84].

2.7 Summary

In this chapter, we have discussed opportunistic communication and its different phases. While discussing on the physical medium (i.e., lower layer) we see that in an indoor RF propagation environment the presence of multipath is usually severe, a line-of-sight path may not exist, possibility of having interference from other wireless

nodes is higher and the characteristics of the environment can change drastically over a very short time or distance. All of these issues can have some great impact on the link quality and thus can influence the capability of the opportunistic contacts. After the discussion on physical medium characteristics, we discuss on link layer connection establishment phase. This phase includes several steps, namely, scanning, joining, authentication and association. We discuss on different scanning modes and find active scanning faster than the passive scanning. During the joining phase the most important task that takes place is, the synchronization between the station and the rest of the network. After the joining phase, the authentication process starts where the selected access point either accepts or rejects the identity of the station. We discuss that, as in opportunistic networking the intention is to utilize the public hotspots, the method of Open System authentication will be used in the thesis work. Once authenticated, the station sends an Association Request to the access point to associate itself with the access point. After having associated to the access point the station acquires an IP address. We discuss on DHCP procedure as it is used as the IP allocation mechanism in our work. After the link layer connection establishment phase, we focus on service discovery phase. In this section, we discuss Avahi which is a Zero Configuration Networking (Zeroconf) implementation for Linux and BSDs, including a system for mDNS/ DNS-SD. We also discuss on Avahi architecture. From the architectural overview we can conclude that a client node goes through name reservation, service browsing and service resolution steps during the service discovery phase. After the service discovery phase, while discussing on data transmission over TCP in wireless medium, we see that TCP yields poor performance in wireless networks as it interprets all losses in wireless links as congestion and consequently invokes congestion control mechanism.

3 Testbed Design

We are interested in characterizing the opportunistic contacts between two stations when they communicate via an access point in an indoor wireless environment. Understanding the indoor wireless characteristics is also a key concern in our work. A wireless testbed, when designed and administered properly, can provide profound insights on the deployed wireless networks as well as on the experiments that have been carried out on it. Due to the dynamic and complex characteristics of indoor wireless environment (as discussed in section 2.2) many basic assumptions made for analytical and simulation studies are likely to be invalid in a real wireless network deployment. In order to serve its purpose, the wireless testbed should be examined for its correct functionality. The hardware components must be thoroughly tested and the software must be properly configured. In addition, nodes need to be deployed taking into account the environmental variability. Here we introduce our testbed setup in details, describing the equipment, software and tools used during the experiments, followed by the description of the measurement environment.

3.1 Hardware and Software Components

In this section, we describe the hardware and software components that have been used during the different phases of our experiments. First we discuss the stations in section 3.1.1 and the access point in 3.1.2. Later in section 3.1.3, we discuss the application that we build to use in our experiments.

3.1.1 Stations

Here we discuss the specifications of the stations that are part of the deployed networks in our experiments.

Nokia N810

Nokia N810 is a wireless internet tablet based on the Maemo 4.1 platform (OS: Linux based OS2008, Kernel version: 2.6.21-omap1, Firmware version: 2.2007.50-2). Maemo is a modified version of the Debian GNU/Linux distribution. The processor is a TI OMAP 2420 (ARM architecture) running at 400 MHz. Nokia N810 provides 2 GB of internal memory and 128MB of RAM. The device supports 802.11b/g Wi-Fi standards. It has the STLC4550 chipset for wireless LAN connectivity [38]. The device supports both the 802.11 operational modes: Ad-hoc and Infrastructure. Two radio transmission power levels supported Nokia N810 are 10 mW and 100 mW. The device is featured with the power saving mode. When enabled, this mode can temporarily disable the network during the periods of no network activity to reduce the amount of battery drain. The Wi-Fi security mechanisms available in Nokia N810 are Open System Authentication and WEP.

Dell Latitude D830 Laptop

Dell Latitude D830 Laptop runs Ubuntu 11-04 (Natty Narwhal, Kernel Version: 2.6.35-24-generic-pae). The processor is Intel Core 2 duo (T7500, 2.20GHz). It has a 4 GB of RAM. The laptop supports 802.11a/b/g Wi-Fi standards and it is equipped with Intel PRO/Wireless 3945ABG wireless adapter. The laptop can operate in Ad-hoc mode as well as in Infrastructure mode. The default setting it has for the wireless transmit power is 15 dBm. The laptop has the Open System Authentication security mechanism along with the different versions of WEP and WPA.

3.1.2 Access Point

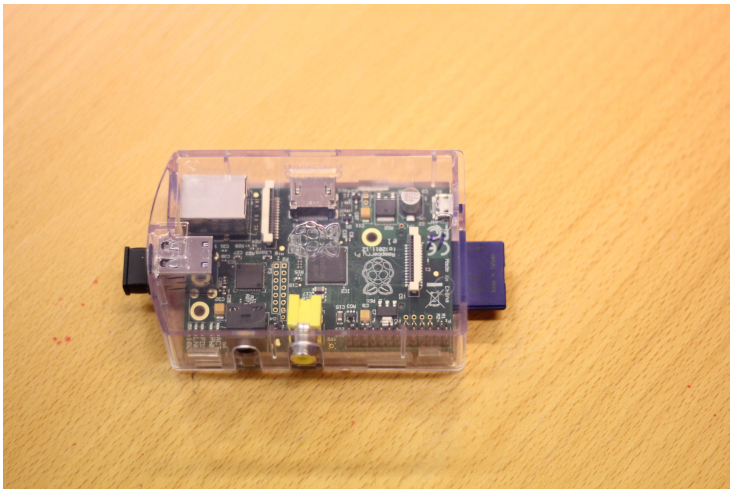


Figure 12: Access point (Raspberry Pi)

In our experiments, a Raspberry Pi is used as the access point (Figure 12) [67]. We use the standard Raspbian distribution on it. Raspbian is an operating system based on Debian and optimized for the Raspberry Pi hardware [69]. Hostapd runnnig on the Raspberry Pi enables the device to operate as an access point and the dnsmasq program it has can provide DHCP service [43, 68]. We equip the Raspberry Pi with the NETWJORK PICO 150 Mbps WLAN USB adapter. The default setting it has for the transmission power is 20 dBm. The wireless adapter can support both Ad-hoc and Infrastructure mode and is compatible with IEEE 802.11b/g/n standards. The encryption methods it has are WEP, WPA, WPA2, WPS and 802.1x.authentication.

Set-up at the Access Point End

As discussed in section 2.2, radio propagation in indoor systems is very complex to calculate due to the various radio wave propagation phenomena and that makes it difficult to define the edge of the access point coverage area. As our aim is to characterize the opportunistic contacts, we require a controlled RF testbed where we can have the possibilities to study the impact of different indoor wireless characteristics

(a) (b)

Figure 13: (a) Rohde&Schwarz Step Attenuator. Picture adopted from [73]. (b) Rohde&Schwarz Shield Box. Picture adopted from [72].

on the opportunistic contacts. That is why we attenuate the signal transmitted by the access point in a way that the maximum range of the Wi-Fi network falls within the testbed area and thus to increase the possibility of studying the behaviour of the opportunistic contacts under different wireless conditions within the short range of the Testbed area. To have the controlled testbed environment, we impose an attenuation of 17.5 dB on the access point signal using a Rohde&Schwarz step attenuator (Figure 13a) [73]. Connecting an access point to an attenuator directly through cable and connector may raise an issue of unwanted signal leak. The leak (i.e., radiation) can occur from the body of the access point or due to the connector/cable. To reduce the radiation to a minimum we use a RF shield box (Rohde&Schwarz CMS-Z10 Model 1204.7008K02) (Figure 13b) [72]. The purpose of RF shielding is to reduce the level of electromagnetic emission.

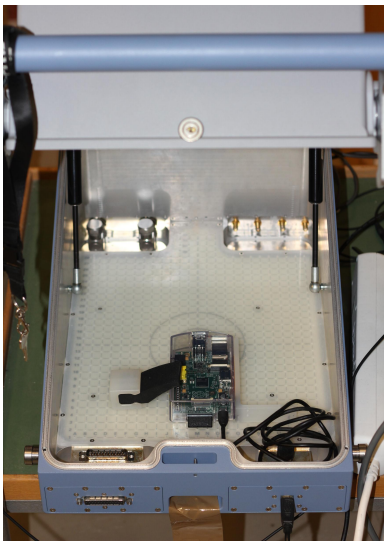


Figure 14: Access point inside the Rohde&Schwarz shield box.

In our set-up, the access point is kept inside the RF shield box in order to reduce

the radiation (Figure 14). The RF output connector of the shield box is connected with the step attenuator through a coaxial cable. Another coaxial cable connects the step attenuator to an omnidirectional antenna (gain 3 dBi) that is mounted at the height of 160 cm (Figure 15).

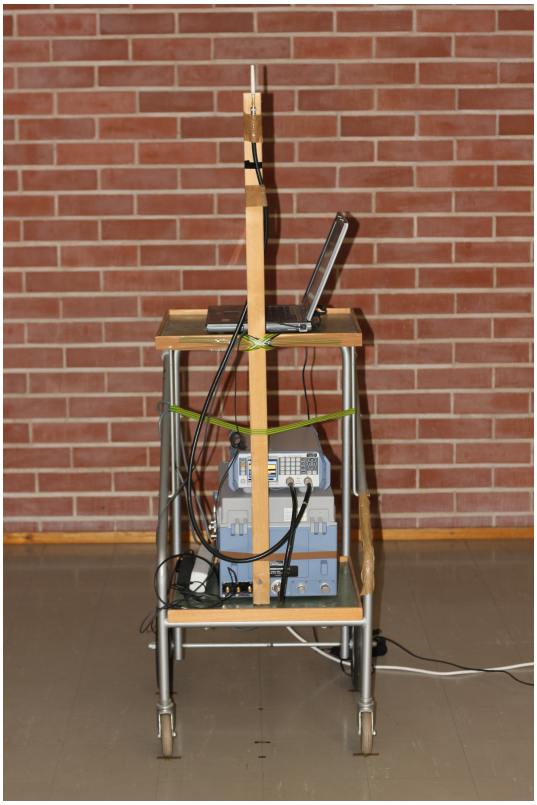


Figure 15: Set-up at the access point end (The figure also shows a station that is connected to the AP).

3.1.3 Application (Service Browser and Service Publisher)

In order to enable the service provisioning feature in our experiments, we build a Service Browser and Service Publisher application which is based on Avahi-client API [82]. The API does three main tasks: i) Registering a service, ii) Browsing for services and iii) Resolving service names to host names. Details on Avahi service discovery mechanism are in section 2.5. Here we discuss the basic functionality of the application and provide a brief overview on the Avahi-client API architecture.

Basic Functionality of the Application

The station (i.e., the server) running the Service Publisher application offers the service named "DtnUpload" in the network. The Service Browser application helps the client node (i.e. the other station of the network) to discover the desired service through service query. After receiving the query response from the server, the

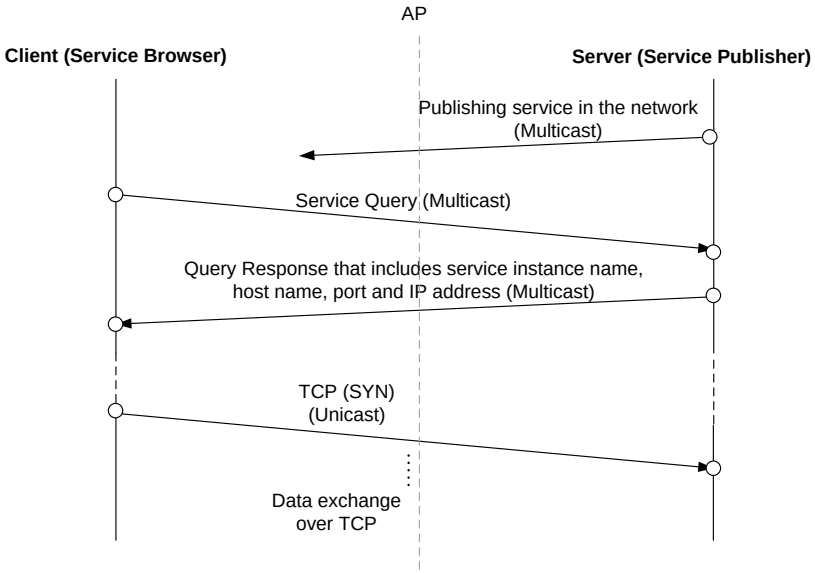


Figure 16: Functionality of the Service Browser and Service Publisher Application.

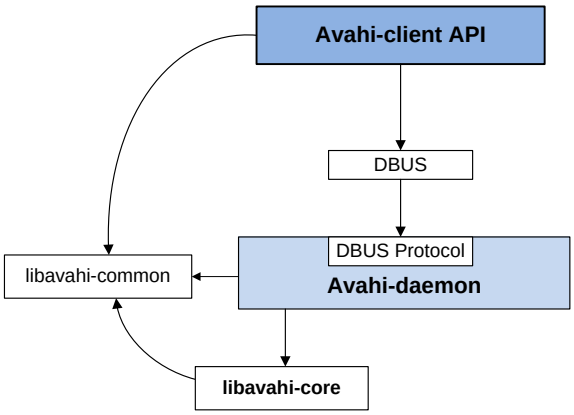


Figure 17: Avahi-client API Architecture.

Service Browser application resolves the service information (i.e., host name, port, IP address) regarding the offered service. Using the resolved port and IP address the client(i.e. the Service Browser application) now establishes TCP connection with the server to data transfer (i.e., to upload data). Figure 16 shows the basic operation between the Service Browser and and Service Publisher mentioned above. To have an understanding on the different events that take place at the application layer during the service discovery phase we feature our application with timestamp capability.

Avahi-client API Architecture

Avahi-client API uses DBUS to communicate with the Avahi-daemon (Figure 17). Avahi-daemon makes use of libavahi-core to implement an mDNS/DNS-SD stack (mDNS and DNS-SD are discussed in section 2.5). The daemon is also used to co-ordinate application efforts in caching replies, necessary to minimize the traffic imposed on networks. The functions of the library 'libavahi-common' are used both by clients to the Avahi-daemon and the mDNS stack itself [81]. While carrying out the experiments in our work, both the Service Publisher and Service Browser require the Avahi-daemon to run all the time at the client and server nodes.

3.2 Measurement and Analysis Tools

The most important source of information to characterize or analyze the wireless networks is the statistics inferred from packets sent on the channel(s). They are obtained using packet sniffing tools that passively listen the packets on the wireless medium. While many tools exist, no single tool provides all the functionalities required to achieve the goals of our work. So in our experiments, we have used a number of tools including Tcpdump, Wireshark, and AirPcap Nx.

3.2.1 Tcpdump

Tcpdump is a open-source packet analyzer that runs under the command line [5]. It allows the user to display TCP/IP and other packets being transmitted or received over a network to which the station is attached. Tcpdump works on most Unix-like operating systems: Linux, Solaris, BSD, OS X, HP-UX, Android and AIX among others.

3.2.2 Wireshark

Wireshark is a open-source network protocol analyzer that allows the user to interactively browse as well as to capture packets from a live network [87]. It is also used to analyze the previously captured packets. Wireshark is very similar to tcpdump but has a Graphical User Interface (GUI). Wireshark's native capture file format is libpcap format, which is also the format used by tcpdump and various other tools(e.g., AirPcap). It runs on Linux, OS X, BSD, Solaris, some other Unix-like operating systems and Microsoft Windows.

3.2.3 AirPcap Nx

Figure 18: AirPcap devices attached with the Windows Laptop.

AirPcap Nx is a USB-based adapter with two external antenna connectors that captures low-level 802.11a/b/g/n wireless traffic (including control frames, management frames and power information) and delivers the data to the Wireshark platform [80]. Once AirPcap Nx is installed, Wireshark displays a special toolbar that provides direct control of the adapter during the capturing session. The Wireshark UI is then employed to perform analysis on the captured packets. AirPcap Nx works on the Windows platform. That is why, an HP EliteBook laptop (running Windows 7) attached with two AirPcap Nx devices is used to capture the wireless traffics during the experiments (Figure 18).

3.3 Testbed Environment

Our wireless testbed is deployed at one of the corridors (corridor H203) of the Electrical Engineering building in the Aalto University School of Electrical Engineering. The length of the corridor is 37 m. Figure 19 depicts the testbed area where we can see the location of the AP (i.e., the location of the trolley equipped with the whole access point end set-up (Figure 15)) marked as red dot and this position is kept fixed for the AP throughout our work. There is another fixed position in Figure 19 that is marked as a green dot and the position is also very close to the AP. Whenever there are two stations used in our network, one of the stations is placed on this particular position. Actually the station is placed on the same trolley that is equipped with the access point end accessories (shown in Figure 15). Then in Figure 19, we can see 17 yellow dots that are marked at a 2 m interval along the line (34 m long) that is originated from the base of the access point trolley. These 17 marks defines the position of a station during the different stages of our experimental work. The corridor has doors and windows that are all closed during the experiments. There is a glass partition (framed with wooden and metal parts) equipped with a door in the corridor but to ensure the clear LOS (Line-of-sight) between the station and the

Figure 19: Testbed area including the location(s) of the AP (red dot) and the stations (green and yellow dots). (Figure not drawn to scale).

Figure 20: A view of the deployed Wi-Fi network when distance between the AP and one station is 16 m while another station is within half a meter range of the the AP (Windows laptop attached with the AirPcap devices are also visible in the picture (right side of the picture)).

access point, we always keep that door open during the experiments. A view of one of the deployed networks during our experiments is shown in Figure 20 where we can also see the sniffing laptop equipped with the AirPcap devices.

All the experiments are carried out either at night or during the weekends to avoid interference from moving people and to facilitate reproducibility (though reproducing is not possible always due to the complex indoor radio wave propagation characteristics). Another reason is to have the minimal impact by external interferences, such as traffic from wireless nodes associated with the university networks/ different research group networks, cell phones, microwaves etc. The indoor RF environment can be highly dynamic and it is all but impossible to ensure that the RF environment across the experiments is identical. So this is one of the key concerns we take under consideration while scheduling the different phases of the experiments. We also notice that overheating of the devices (i.e., stations and AP) can severely degrade the wireless performance. When the heat gets reduced, the devices are back to the state where they start performing normally again. We make sure that devices are properly placed on a stand/trolley so that the vents on the bottom of the devices are not blocked. We keep the AP inside the RF shield box which obviously has no ventilation system. So we pause our experiments in case we find the AP getting overheated and after cooling it down resume the experiments again.

Scenarios of Network Deployment

The network topologies deployed to carry out the experiments in our work belong to one of the following scenarios,

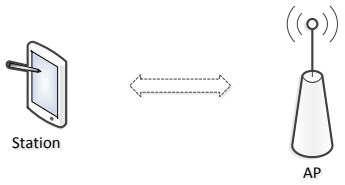


Figure 21: Experimental set-up (Scenario 1).

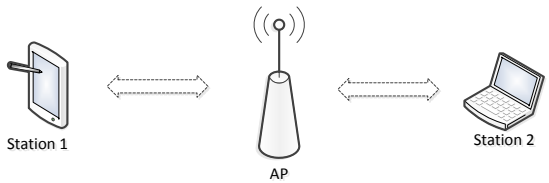


Figure 22: Experimental set-up (Scenario 2).

Scenario 1: This scenario comprises a station and an AP (Figure 21). In an Infrastructure-assisted Wi-Fi network, a station needs to connect to the AP and

acquire an IP address before it can start communicating with the other station(s) in the network. So this scenario serves the purpose that is required to carry out the experiments related with the connection establishment and IP acquisition phases between a station and an AP. The same scenario will be used during the testbed site survey experiments as well.

Scenario 2: This scenario consists of two stations and the AP (Figure 22). In this scenario, both station 1 and station 2 are connected to the AP and they are now ready to communicate with each other. So this is a scenario appropriate for the experiments that we conduct to characterize the service discovery phase as well as the data transmission over TCP phase.

Settings of Wireless Parameters

The experiments in our work are conducted in IEEE 802.11b Infrastructure mode. The access point is configured to operate on channel 6. In the testbed area, using the Android Wifi Analyzer tool we find that, channel 6 is comparatively less crowded [1]. An overcrowded wireless channel results in decreased bandwidth and connectivity issues for the network. So we set channel 6 as the operating channel of the access point as we find it less crowded.

Transmission power at Nokia N810 is set to 100mW and for the Dell Latitude D830 Laptop we keep the default setting, i.e., 15 dBm. In case of the access point (Raspberry Pi), the signal is attenuated by 17.5 dB as discussed in section 3.1.2. While carrying out the experiments, the power saving mode at the stations is kept off. The reason behind such choice is the service discovery phase of our experiments that involves multicasting in the network. If there is one or more stations in power saving mode, the access point buffers all multicast packets and sends them only after the next DTIM (Delivery Traffic Indication Message) beacon, which may be every one, two or three beacons. If no stations within the network are in power save mode, multicast packets are sent immediately when they arrive [41]. That is the reason why we keep the power save mode option disabled in the stations.

The RTS/CTS (Request to Sent/ Clear to Send) is kept off in the stations as well as in the access point. The reason of turning the RTS/CTS off is to reduce the number of control frames in the network. Those control frames are not required in achieving the goals of our work and thus setting the RTS/CTS off we avoid unnecessary flow of traffic (control frames) in the network.

3.4 Summary

In the Testbed Design section, we discuss on different hardware and software components that are used in our experiments. We can see that mostly Linux based systems we use in our experiments and the reason behind such choice is mainly due to the controlability they provide to the users. Also the platforms are suitable for the application (i.e., the Service Browser and Service Publisher application) we build to conduct our research though we need to use the Scratchbox cross compilation tool to make the application runnable on Nokia N810 (Platform: Maemo 4.1). To achieve

a controlled testbed environment (to a certain extent) we have an access point end set-up configured with a step attenuator and a RF shield box. Different sniffing tools we have used in our work including the AirPcap Nx device. We choose AirPcap Nx mainly due to its feature of capturing low level IEEE 802.11 frames. Small changes in position and orientation of the devices can impact the RF behaviour. Also the time of the experiments when they are conducted and thermal state of the devices can have great impact on the experimental results. We consider all these issues while planning and conducting our experiments. Not a single scenario is enough to achieve the goals of our work, so we use different scenarios for different phases of the experiments. We use channel 6 as the operating channel of the AP as we find it less crowded. Though power saving mode is kept on usually in wireless stations in order to save the battery life, in our work we decide to keep it off as keeping it on slow down the multicast traffic in the network.

4 Measurement and Analysis of Indoor Wireless Testbed Characteristics

Our experiments in this section are intended to assess the characteristics of the indoor wireless environment. The goal of the Thesis is to carry out experiments in order to quantify device-to-device data transfer in 802.11 Infrastructure-assisted wireless networks in an indoor environment. Investigating the behaviour of the indoor wireless network environment is crucial as it will be useful in evaluating the reasoning behind certain characteristics of device-to-device communications. In this site survey experiment, we observe how the connectivity and packet transfer between the station and the access point vary on different spots of the testbed area though there is clear LOS between them. The observation helps us to mark the spots with different wireless characteristics and thus we gain an understanding on the indoor wireless testbed environment.

4.1 Experimental Setup

The experiments are carried out in the testbed environment that is discussed in section 3.3. The AP end set-up (explained in section 3.1.2) is placed in a fixed position marked as red dot in Figure 23.

Figure 23: Testbed area. (Figure not drawn to scale).

We place the station (Nokia N810) on the 17 yellow dots marked in Figure 23 but at two different levels of heights (i.e., 80 cm (low) and 110 cm (high)). The reason of doing so is to check how the link performance varies due to the change in height of the station. These 17 marks and on each mark 2 levels of height (i.e., 80 cm (low) and 110 cm (high)) for the station conclude that there are in total 34 test points where the experiments are carried out.

In this site survey experiment, we use number of connection failure(s) and packet loss rate as the two performance metrics. For the connectivity part, we try to establish connection between the station and the access point from all those 34 spots mentioned above. In each position, 10 attempts have been made to connect

the station to the access point and we keep record of the failed attempts of connection establishment.

For the packet transfer phase, a series of ping packets (25 packets and each packet was of size 64 bytes) have been sent from the station to the access point. The pinging sequence is run 20 times from each position. We run these pinging sequence when we have successful connection between the access point and the station. Most of the time all the ping sequence (20 sets) are carried out after one successful connection establishment between the node and the access point but sometimes it is not possible as the station gets disconnected from the access point after several sets of successful pings. Sometimes the disconnection takes place when a set of ping is partially complete (i.e., just after the successful transfer of a small number of ping packets of a set). There are even occasions where there is no successful ping at all even the station can get connected to the access point. So at times instead of running them altogether, 20 ping sequence (i.e., 20 sets) are needed to be split out among the different attempts (maximum attempts of connection establishment: 10) of connection establishment. The pinging method is used to measure the packet loss rate and the data then are used to calculate the average packet loss rate on each test point.

4.2 Results and Findings

In this experiment, *number of failed connection attempts* and *the packet loss rate* are used as the *performance metrics* in order to assess the indoor wireless environment.

In Figure 24, we observe all the ***connection attempts*** between the station (Nokia N810) and the access point being successful till the 12 m mark when the station is kept in high position (height: 110 cm). For the same height of the station, we see that from 14 m mark onwards almost every test point (except the 20m mark) experiences failed connection attempt(s).

For 14m and 16m marks the failure rate are 2 out of 10 and 1 out of 10 respectively but after the 18m marks the failure rate is very high at most of the test points as we can see in Figure 24. 50% to 100% connection attempts fail from those test points. So it can be said that the connectivity between the node and the access point is poor at those test points (i.e., between 18 m and 34m marks). But there are few exceptions as well: the previously mentioned 20m mark where there is no connection failure at all and the 28 m mark where the number of failure is only 1. We also notice in Figure 24 that, there are dead spots at the 24 m and 30 m marks but beyond those marks there are spots where the connection establishment between the station and the access point is possible. From these kind of characteristics we can assume that the success or failure of connection attempts in an indoor wireless environment doesn't only depend on the distance between the node and the access point but also depends on the local interference at the receiver and non-distance based channel propagation effects as discussed in section 2.2. In the testbed area, there are obviously interference sources as multiple university wireless networks are active in that area. The non-distance based channel propagation effects include signal loss due to obstacles and variability due to shadowing and multipath fading. The

SL	Distance between the AP and the station (m)	Number of connection failure (Out of 10 attempts)	
		Height of the station: 80 cm (Low)	Height of the station: 120 cm (High)
1	2	0	0
2	4	0	0
3	6	0	0
4	8	0	0
5	10	2	0
6	12	0	0
7	14	0	2
8	16	0	1
9	18	10	5
10	20	7	0
11	22	10	6
12	24	2	10
13	26	7	7
14	28	10	1
15	30	10	10
16	32	10	8
17	34	10	9

Table 1: Distance Vs. Connection failure

presence of walls, floor, ceiling, doors, windows, furniture at the corridor can reflect, diffract, scatter or absorb signals and thus contributing to the signal attenuation have some significant impact on the success/failure of connection establishment at different spots of the testbed area.

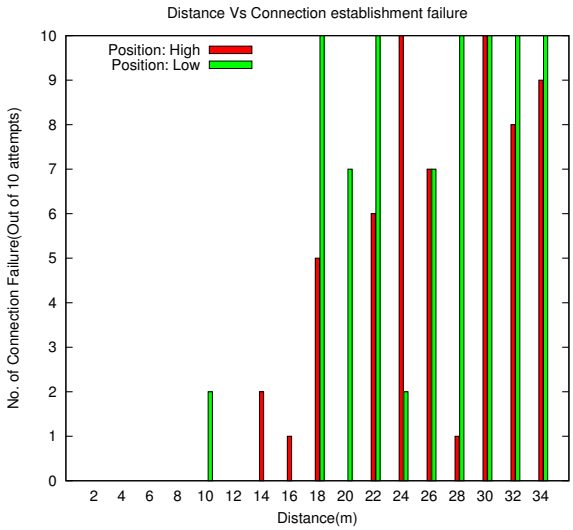


Figure 24: Distance Vs. Connection failure

Now if we observe Figure 24 for the low position (height: 80 cm) of the station, we see further interesting results. In case of low position, mostly from 18 m marks onwards there are connection failures as it is in case of high position of the station. But we see difference in characteristics for the two heights though the distance from the access point is the same. For instance, in case of low height of the station, there are 100% connection failure at the 18 m, 22 m, 28 m, 30 m, 32 m and 34 m marks. whereas in case of 110 cm height there are only 2 such marks (i.e., 24 m and 30 m marks). Here also noticeable that, in case of high position when there is no successful connection attempts possible from the 24 m mark, 8 connection attempts are successful when the station is kept in low position. So here we see that small changes in location have great impact on the wireless connectivity due to the characteristics of indoor radio wave propagation (discussed in section 2.2) . During the experiments we notice that the change in antenna orientation can influence the wireless connectivity as well. For instance, at the 22 m distance when the stationis in usual horizontal position at a height of 80 cm, there is no success in connecting the station to the access point. But when we change the angular position of the station (by making the front side of the mobile up around 45 degree) so as to change the orientation of the antenna, we can connect the node to the access point from the same position.

We gain an understanding on the link reliability of the indoor wireless environment through the ***packet loss rate*** phase of experiment. In Figure 25, we can see that, the link is very stable till 16m mark and it is true for both the heights (i.e., Low: 80 cm and High: 110 cm) of the station. Till the 16 m mark, the maximum average packet loss rate the station observes at a height of 110 cm is 8% and for

the low position it is 4.8%. From the 18 m mark onwards, we see that the average packet loss rate is no longer constant but has rather high degree variability that signifies unstable link performance. There are marks where 100% packet loss takes place (Figure 25). These packet loss phenomena are also closely linked with the characteristics of the indoor wireless environment as discussed above.

SL	Distance between the AP and the station (m)	Height of the station: 80 cm (Low)		Height of the station: 100 cm (High)	
		Average Packet Loss (%)	Standard Deviation of packet loss from 20 sets of ping	Average Packet Loss (%)	Standard Deviation of packet loss from 20 sets of ping
1	2	2.20	2.96	1.80	2.68
2	4	3.80	4.28	3.60	4.72
3	6	4.00	4.00	3.60	4.18
4	8	2.20	2.04	3.60	3.87
5	10	4.60	2.98	4.40	4.84
6	12	1.80	2.42	5.20	7.69
7	14	0.80	1.64	8.00	7.46
8	16	4.80	4.02	2.80	2.63
9	18	100.00	X	30.00	13.07
10	20	26.00	24.47	8.80	5.89
11	22	100.00	X	41.80	17.24
12	24	21.20	19.03	100.00	X
13	26	100.00	X	19.40	14.64
14	28	100.00	X	31.20	26.14
15	30	100.00	X	100.00	X
16	32	100.00	X	89.80	25.55
17	34	100.00	X	92.00	19.60

Table 2: Distance Vs. Packet loss

During the experiments we notice that there are certain spots (e.g., 26 m, Height: 80cm) from where it is possible to connect the station to the access point but the transfer of ping packet is not possible at all (i.e., 100% packet loss) (Table 1, Table 2). Also there are spots (e.g., 18 m, 28 m marks and Height: 110 cm) where we can run all the 20 sets of ping but the pinging tests are often interrupted as the link is getting disconnected frequently while running the series of ping. Another interesting mark we notice is 26 m at a height of 110 cm. At this position, 7 out of 10 attempts of connecting the station to the access point fail. Only 3 times there are success but the average packet loss rate is not high (i.e., 19.40%). This scenario of packet loss can be explained with adaptive modulation scheme feature of 802.11b standard. Systems using 802.11b are allowed to adaptively choose modulations for maximum data rates of 1, 2, 5.5 or 11 Mbps. Wireless cards can adjust their encoding rate to the quality of the wireless channel, i.e., "autorate". The autorate functionality is implemented on wireless cards such that if a high transmission rate cannot be effectively supported, the card can fall back to a lower more robust (in terms of encoding) transmission

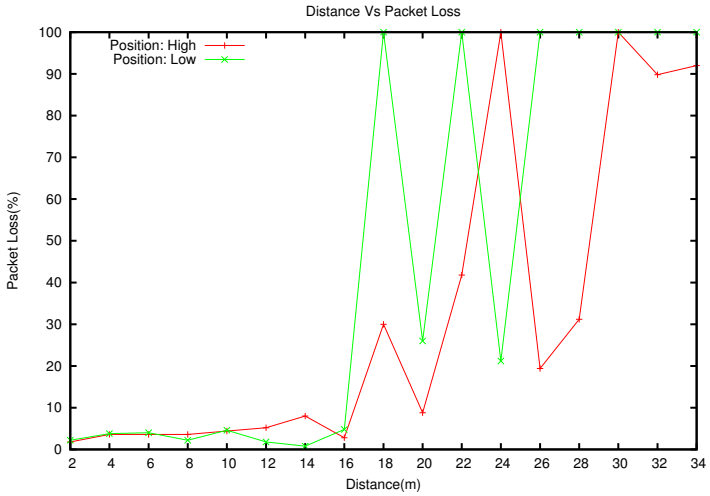


Figure 25: Distance Vs. Packet loss

rate [57]. Thus the system sacrifices performance over connectivity and choose lower data rates to maintain connectivity. But the system's decision of choosing modulation scheme cannot always keep the link connectivity alive or maximize the link performance.

4.3 Summary

In this chapter, we have have assessed the the characteristics of the indoor wireless testbed area. From the experimental results we have gained valuable as well as interesting insights on the indoor RF wave propagation characteristics. For example, we observe dead spots (i.e., no connectivity) at the 24 m and 30 m marks (when the height of the station is 110 cm) but beyond those marks there are spots where the connection establishment between the station and the access point is possible. This kind of characteristics shows that the success or failure of connection attempts in an indoor wireless environment does not only depend on the distance between the node and the access point but also depends on the local interference at the receiver end as well as on the non-distance based channel propagation effects. Another interesting observation is that due to the change in station's height (the distance from the AP is same) the rate of connection failure can significantly change. We also notice that antenna orientation plays an important role on the success/failure of connection attempts. During the experiments we notice certain spot where the connection failure rate (7 out of 10 connection attempts failed) is very high but the average packet loss rate is not very high (i.e., 19.40%). This scenario can be explained with the adaptive modulation scheme feature of 802.11b standard. Under this scheme, the system sacrifices performance over connectivity and choose lower data rates to maintain connectivity.

5 Measurement and Analysis of Opportunistic Contacts in Infrastructure-assisted Wireless Network

The focus of our work is to study the characteristics of the opportunistic contacts in Infrastructure-assisted Wi-Fi networks. Opportunistic contact occurs when two stations come within the range of the same access point and establish communications. In Infrastructure-assisted Wi-Fi networks the stations communicate via access point. So during an opportunistic communication, a station has to establish connection with the access point first (i.e., associating to the access point). This connection establishment phase involves *scanning (network discovery)*, *joining the network*, *authentication* and *association*. Then the station has to *acquire an IP address* for further communication. Measurement and analysis of these steps that we present in section 5.1 give us an idea on the overall procedure of the connection establishment and IP acquisition in Infrastructure-assisted Wi-Fi networks and that enables us to gain an understanding of the time required for each of the steps.

Next is to deal with the *service discovery* phase. The station (i.e., the client node) that needs a service, performs a discovery step, which typically initiates queries in the network for the appropriate service offered by the other station (i.e., the server) in the network. In section 5.2, through the measurement and analysis we define the timeline of the events that take place during this service discovery mechanism.

Then in section 5.3, we see that the application at the client end starts uploading data to the server using the service offered by the server. The *data transmission* between the nodes is done over *TCP*. The experiments of this phase are carried out to gain an understanding of how TCP behaves as a transport layer protocol in Wi-Fi networks and thus to have an understanding on its suitability in opportunistic networking.

5.1 Measurement and Analysis of Connection Establishment and IP Address Acquisition phases

In this measurement and analysis section, we deal with the IEEE 802.11 link layer connection establishment procedure and IP acquisition procedure through DHCP. During the ***link layer connection establishment*** experiments, we measure the *scanning* period required for the station to discover the desired network (i.e., the access point with desired SSID). As the access point is set to operate on a particular channel (in our work, channel 6), we calculate the scanning time that the station takes on that particular channel to discover the access point. It is the time the station waits to receive the Probe Response (with desired SSID) from the access point for the first Probe Request (with broadcast SSID) it sends on that particular channel. So the scanning delay we calculate here is channel specific, i.e., in our work channel 6 which is the operating channel of the desired access point (Figure 26). But scanning process can start on any channel [33] and the operating channel of the access point may not be the first channel where the station starts probing (Probe Request with broadcast SSID). In our work, we always find the station probing

channel 1 before channel 6 when it initiates the scanning process. Though the observation in [33] suggests that channel scanning might not be sequential always, in our measurements, we observe it is always the channel 1 before channel 6. Due to this scenario, we measure the time difference between the 1st Probe Requests (with broadcast SSID) sent by the station on channel 1 and channel 6 as well. This measured delay is the delay a station goes through before it starts scanning the operating channel of the desired network (Figure 26). There is a possibility that channel 1 may not be the first channel where the station starts scanning[33]. Still this probing delay between channel 1 and channel 6 helps us to have an insight on the overall delay a station experiences before discovering the desired network.

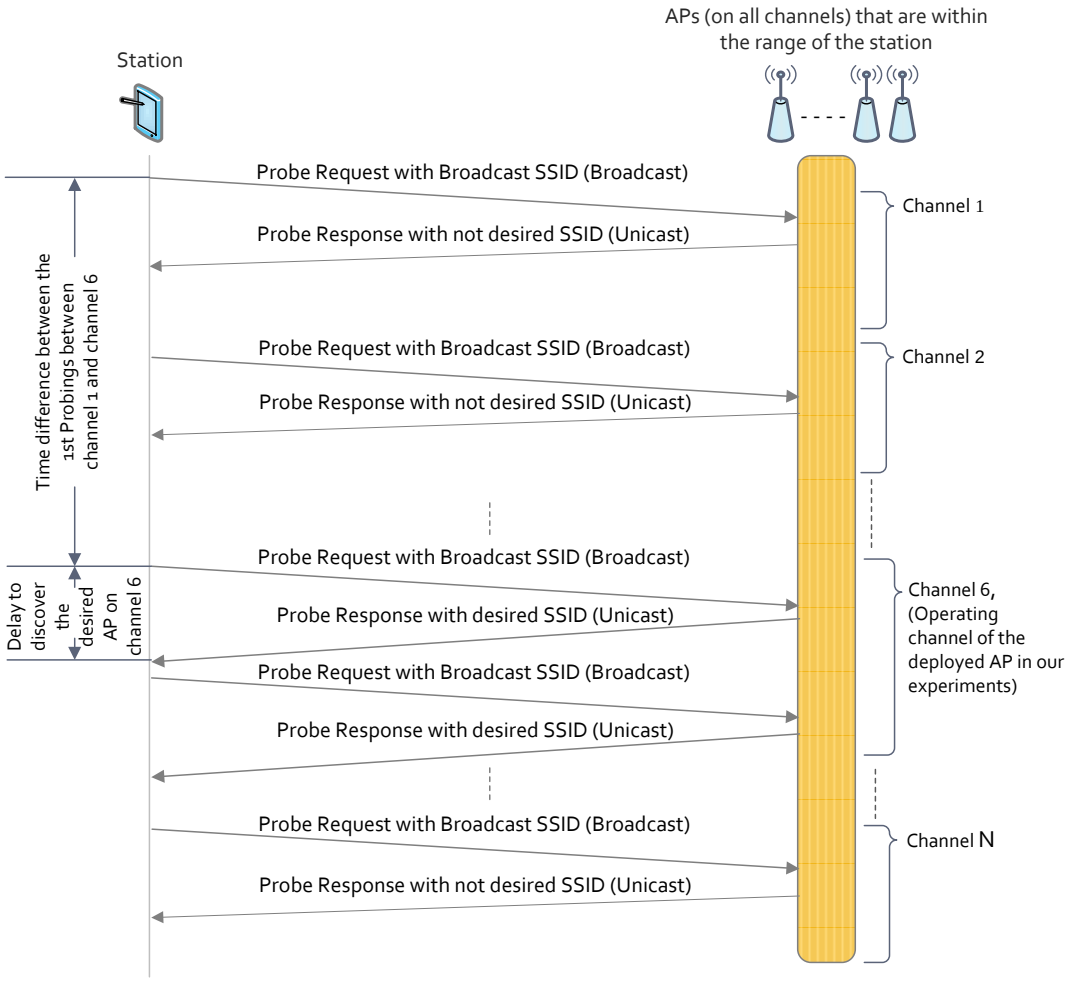


Figure 26: Scanning delay measurement

After the scanning phase, the *joining phase* starts when the station elects to join one of the networks. Choosing which network (i.e., BSS) to join is an implementation-specific decision or it can be done with user intervention. In our work, the task has been accomplished with user intervention. In section 2.3.2, we have discussed that it

cannot be told exactly when the station has joined the network because the joining process is internal to a node and it involves matching local parameters (BSSID, timing parameters, PHY parameters etc.) to the parameters required by the selected BSS. Station getting synchronized with the network (selected BSS) is one of the most important tasks of this phase. To have an understanding on the duration of the joining phase we calculate two time periods. First, we calculate the time the station waits to receive the Probe Response for the Probe Request (with specific SSID) it has sent when the SSID of the desired network is manually selected by the user. But as mentioned earlier receiving the Probe Response from the access point of the selected BSS does not ensure that the station has joined the network. To have an indication on the station joining the network, we give a look on one of the synchronization parameters: BSSID. We notice that, after receiving the Probe Response from the access point, the next frame that is sent by the the station is an Authentication frame. In that frame, the BSSID of the selected BSS is used instead of broadcast BSSID that the station used in earlier frames. This adoption of new BSSID indicates that the station has successfully joined the network, i.e., the transmission are directed to the correct set of stations and ignored by stations in another BSS. So we calculate the time gap between the Probe Response and the Authentication frame as well to have a complete timeline of the joining phase.

Next the *authentication phase* starts where the station transmits an Authentication frame to authenticate itself with the access point. Open System authentication that is used in our experiments includes two message transactions between the station and the access point as discussed in section 2.3.2. We measure the time the station waits to have the response for the Authentication frame from the access point.

Once authenticated, the station sends an *Association* Request to the access point to associate itself with the access point (details in section 2.3.2). We calculate the time the station takes to have the Associate Response from the access point. To be sure about the successful association we check the status value of the response frame as well as the Association ID that is assigned to the station by the access point.

With successful association to the access point, the link layer connection establishment procedure is now complete. But to communicate with the other device(s) in the network the station needs to ***acquire an IP address***. In our work, the station acquires the IP address through DHCP and in section 2.4 the overall procedure of DHCP has been discussed. Here, in the measurement and analysis part, we calculate the time difference between the DHCPDISCOVER and DHCPOFFER packets as well as the time difference between the DHCPREQUEST and DHCPACK packets. From this part we can see how much time is required for a station to acquire an IP address through DHCP. We also calculate the time the station takes before sending the first packet using the newly acquired IP address. In our work, that first packet with the newly allocated IP is an IGMPv3 Membership Report packet.

5.1.1 Experimental Setup

Experiments in this phase are carried out in the indoor testbed environment that is discussed in section 3.3. Our measurement study in this part mainly uses an access point (Raspberry Pi) and a station (Nokia N810). To have a view of an opportunistic networking scenario, we assume that a node (e.g., Dell Latitude D830 laptop) is already connected to the access point. The station (Nokia N810) has to establish connection with the access point and acquire an IP address before it can start communicating with the other node (i.e., the Dell Latitude D830 laptop) in the network. So here the experiments mainly deal with the connection establishment and IP acquisition procedures that take place between the station (Nokia N810) and the AP. IEEE 802.11b is the underlying wireless technology used through out our work and channel 6 is the operating channel of the AP. AP signal is attenuated by 17.5 dBm using a step attenuator. The details of the hardware and software configurations and testbed setup are available in section 3.

Figure 27: Testbed area. (Figure not drawn to scale).

The experiments are carried out keeping the station in two positions (i.e., at 2 m and 16 m apart from the access point (Figure 27)) while the access point is always kept in a fixed position (AP position: red dot in Figure 27). The reason of keeping the station in two different positions is to see the impact of indoor wave propagation characteristics (discussed in section 2.2) on connection establishment and IP acquisition procedures between the station and the AP.

First, the station scans to discover the networks in the testbed area. When the SSID of the desired network (BSS) is found in the scan list of the station, it (SSID) is selected manually so that the station can get itself associated to the access point of the selected BSS as well as can proceed with the IP address acquisition phase. While this whole procedure of this connection establishment and IP acquisition is going on, 20 samples (captured through AirPcap devices) are collected when the distance between the station and the access point is 2 m and 17 samples (captured through AirPcap devices) when the distance is 16m . In both cases, the station is always kept stationary at a height of 110 cm.

The laptop (HP EliteBook2560p running Windows 7) equipped with two AirPcap

sniffing devices (discussed in section 3.2) is used to capture the link layer connection establishment related frames as well as the packets related with the IP acquisition procedure. The sniffing laptop is placed between the station and the AP in the testbed area. The AirPcap devices are configured to capture packets on channel 1 and channel 6 as we measure the scanning delay between the first probings on channel 1 and channel 6 and the network discovery delay on channel 6 discussed in section 5.1. It would be possible to check how much time in total the station dwells on all the channels if we had sufficient number of AirPcap devices to capture traffics on all the channels.

5.1.2 Results and Findings

a. Connection establishment and IP acquisition at close range (2 m distance)

SL.	Time difference between the 1st Probe Requests on Channel 1 and Channel 6 (sec)	Time to discover the desired SSID (i.e., desired network) on channel 6 (sec)	Time between 1st probe request with specific SSID (i.e., selecting the specific SSID) and its response (sec)	After the probe response gap before getting the indication of successful joining (sec)	Total time to have the indication of successful joining after the selection of specific SSID (sec)	Authentication time (sec)	Gap between Authentication phase and Association phase (sec)	Association time (sec)	Total time Authentication and Association phases take (including the gap between the two phases) (sec)
1	0.513956	0.006000	0.001253	3.685720	3.686973	0.001292	0.014235	0.004249	0.019776
2	0.586729	-	0.003632	3.760555	3.764187	0.001276	0.014371	0.001154	0.016801
3	0.515029	-	0.004977	3.704503	3.709480	0.000924	0.014551	0.002508	0.017983
4	0.523478	0.002999	0.001257	3.685688	3.686945	0.001122	0.014354	0.001130	0.016606
5	0.585952	0.002119	0.001132	3.107085	3.108217	0.001132	0.014492	0.001124	0.016748
6	0.585906	0.001529	-	-	-	0.001113	0.014505	0.001120	0.016738
7	0.585982	0.001220	0.001124	3.091931	3.093055	0.001113	0.014498	0.001130	0.016741
8	0.586058	0.001973	0.003825	3.112834	3.116659	0.001125	0.014358	0.001267	0.016750
9	0.586000	0.001227	0.001249	3.133358	3.134607	0.001142	0.012132	0.000969	0.014243
10	0.585807	0.006612	0.001268	3.045070	3.046338	0.001071	0.015161	0.001239	0.017471
11	0.586110	0.001251	0.001123	3.162385	3.163508	0.000973	0.015147	0.003500	0.019620
12	0.585760	0.006753	0.001277	3.185756	3.187033	0.000985	0.014638	0.001169	0.016792
13	0.586252	0.006468	0.001259	3.139576	3.140835	0.001114	0.013626	0.001122	0.015862
14	0.585981	0.003468	0.001125	3.116661	3.117786	0.001104	0.013626	0.004131	0.018861
15	0.585853	0.001478	0.001283	3.084260	3.085543	0.001108	0.014386	0.001255	0.016749
16	0.585998	0.006846	0.001037	3.052969	3.054006	0.001108	0.014646	0.001597	0.017351
17	0.585976	0.001246	0.001023	3.123279	3.124302	0.002123	0.013752	0.001372	0.017247
18	0.585335	0.001122	0.001265	3.201187	3.202452	0.002867	0.012874	0.001126	0.016867
19	0.587867	0.003503	0.005004	3.197519	3.202523	0.001126	0.015130	0.001102	0.017358
20	0.515547	0.001374	0.001370	3.091949	3.093319	0.001115	0.014472	0.000998	0.016585
Avg.	0.572279	0.003177	0.001868	3.246436	3.248304	0.001247	0.014248	0.001663	0.017157
Std. dev.	0.028413	0.002272	0.001358	0.249527	0.250088	0.000451	0.000744	0.001050	0.001229

Table 3: Measured time of the steps during the connection establishment procedure between the station and the AP (distance 2 m)

Figure 28 shows the *timeline view* (based on the average delays from Table 3) of the different steps that take place during the ***IEEE 802.11 link layer connection establishment procedure*** between the station and the AP (distance: 2m). In the figure, we can see that the station starts *scanning* on channel 6 (the operating channel of the AP) 572.279 ms (average time) after the first Probe Request it sent on channel 1 (ECDF graph in Figure 29a). Then the station takes 3.177 ms on an

average to discover the desired SSID on channel 6 (ECDF graph in Figure 29b). We can see that during the Active scanning phase the delay between probings (i.e., between 1st Probe Requests) on channel 1 and channel 6 highly dominates over the network discovery delay on channel 6. The scanning process requires wireless station to switch between and scan each channel independently before scan report is generated (discussed in section 2.3.2). That is why, we see that the station takes on an average 572.279 ms to scan some other channels though the operating channel of the desired access point is channel 6. As suggested by the authors in [54], total scanning delay can be reduced substantially by reducing the number of scanned channels to a subset where APs are known to exist. In section 2.3.2 we also discussed two timers, namely the MinChannelTime and MaxChannelTime that determine the time a station needs to wait on a channel. Though the scanning process is defined in the 802.11 standard [37], the duration of the timers is vendor specific [54]. Scanning delays can be reduced by optimizing these timers [54].

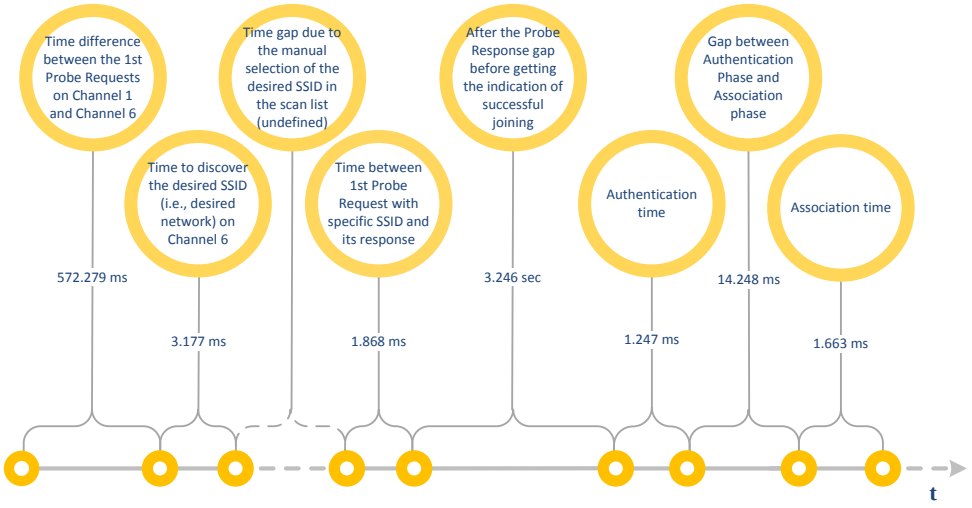
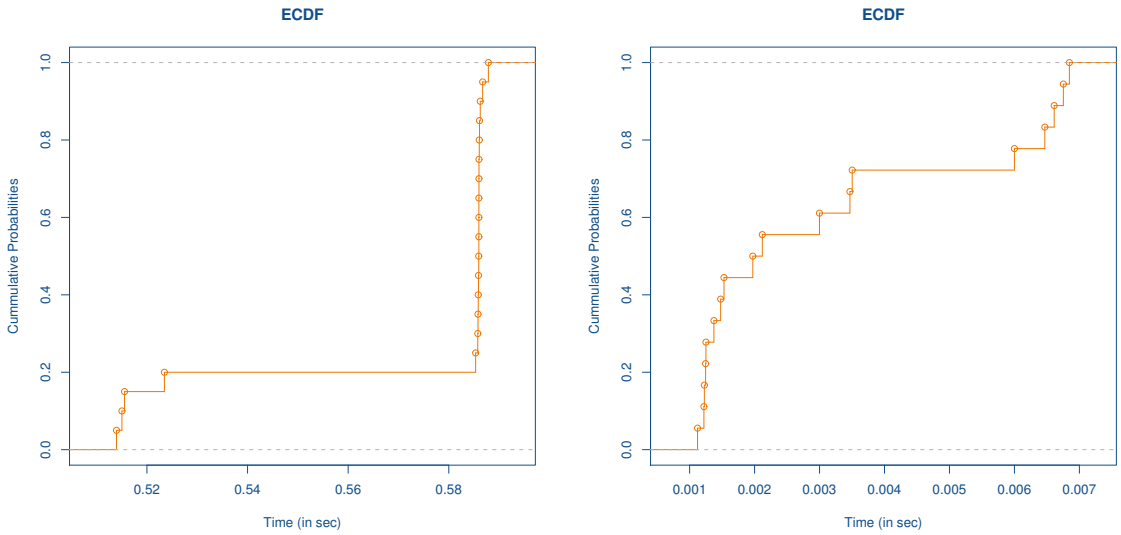


Figure 28: Timeline view of connection establishment procedure when the distance between the station and the AP is 2m (Figure not drawn to scale).

When the SSID of the desired network is displayed in the scan list of the station (i.e., in the "Select Connection" utility of Nokia N810), the selection of that specific SSID is done with user intervention. That manual selection added some delay between the scanning and joining phases. As it is a delay due to the manual selection of the desired SSID, it is not required to measure that delay. That is why, in Figure 28, we can see an undefined time gap between the scanning and joining phases.

When the *specific SSID* is selected, the station sends Probe Request with that specific SSID and it takes on an average 1.868 ms for the station to get the Probe Response from the AP of the selected BSS. After 3.246 sec (average time) of getting this Probe Response, the station sends the Authentication frame (using the BSSID of the desired network instead of broadcast BSSID) which indicates the station's *successful joining* to the selected BSS (Figure 28).

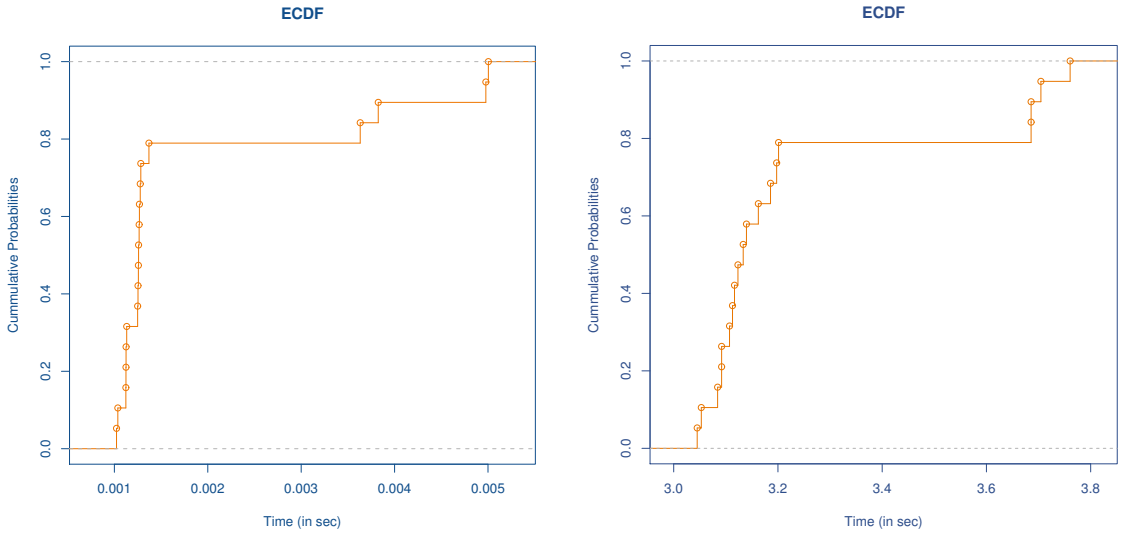


(a) (b)

Figure 29: (a) ECDF of time difference between the first probings on channel 1 and channel 6 (distance between the station and AP: 2 m),
(b) ECDF of desired network discovery time on channel 6 (distance between the station and AP: 2 m).

Figure 31 shows the ECDF of total delay during the joining phase, namely the sum of Probe Request(with specific SSID)-Probe Response time (ECDF in Figure 30a) and the time to have the indication of successful joining (ECDF in Figure 30b). If we compare Figure 31 with the Figures 30a and 30b, we can see clearly that the time gaps before having the successful joining indication dominates significantly over the probing times. In other words, the higher delays in joining phase are due to the time gaps that the station experiences before having the joining indication. During these gaps, we always find the station and the AP exchanging another set of Probe Request(with specific SSID) and Probe Response frames. But as discussed in 2.3.2 and 5.1, station *getting synchronized with the selected BSS* is the important task during the joining phase and that could be key reason behind these higher time gaps before having the joining indication.

While working with the data points of scanning phase and joining phase we can see in Table 3 that, some of the of entries are missing and are marked with "-". The reason behind this is the missing packet(s) in the captured data (the data that is captured using AirPcap). For example, in these particular cases, we have not seen any Probe Response (in the captured data) for the 1st Probe Request sent by the station. Probe Response packets are sent using normal frame transmission rules of IEEE 802.11 standard [37]. So there will be acknowledgement packet from the station to the AP when it receives the Probe Response. If the station does not receive the Probe Response, there will be no acknowledgement from the station and the AP will retransmit the Probe Response packet. During the measurements of



(a) (b)

Figure 30: (a) ECDF of time difference between the first Probe Request with specific SSID and its response (distance between the station and AP: 2 m),
(b) ECDF of time gap between the Probe Response and getting the indication of successful joining (distance between the station and AP: 2 m).

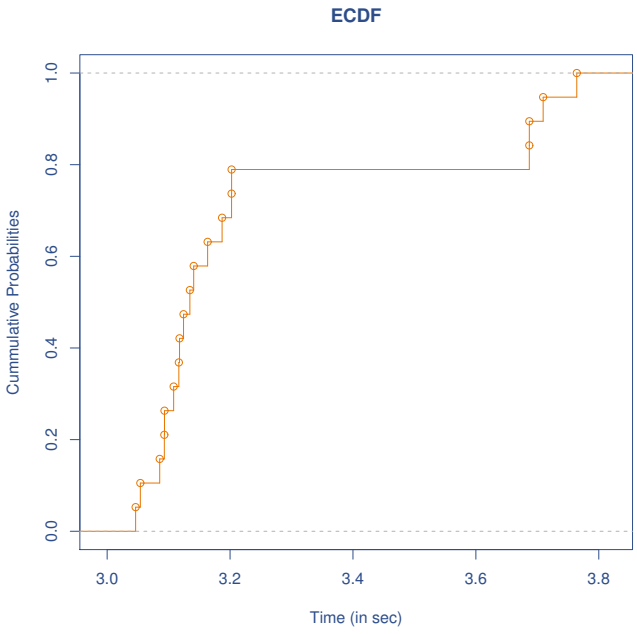
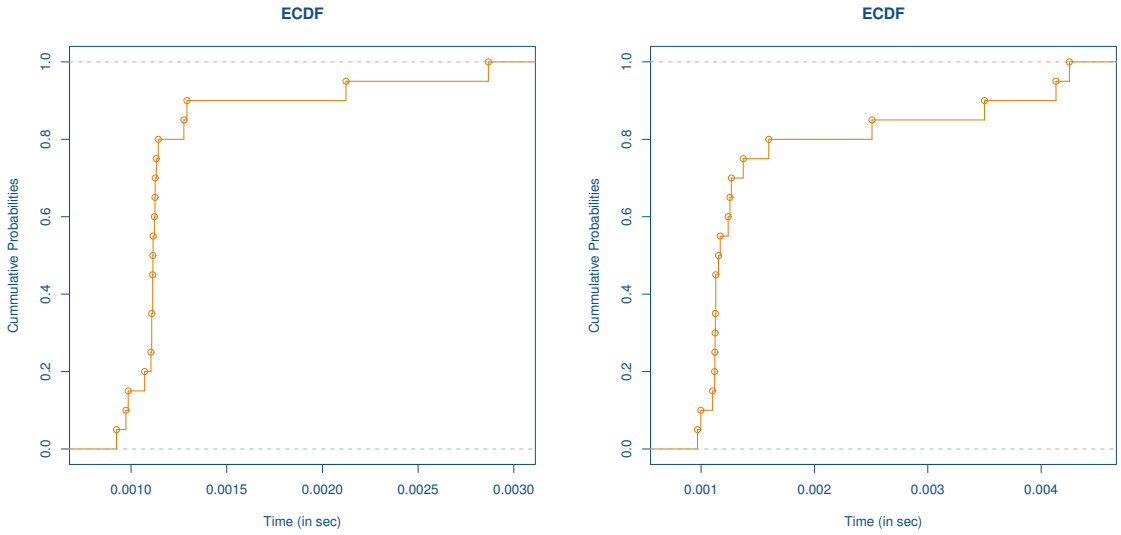


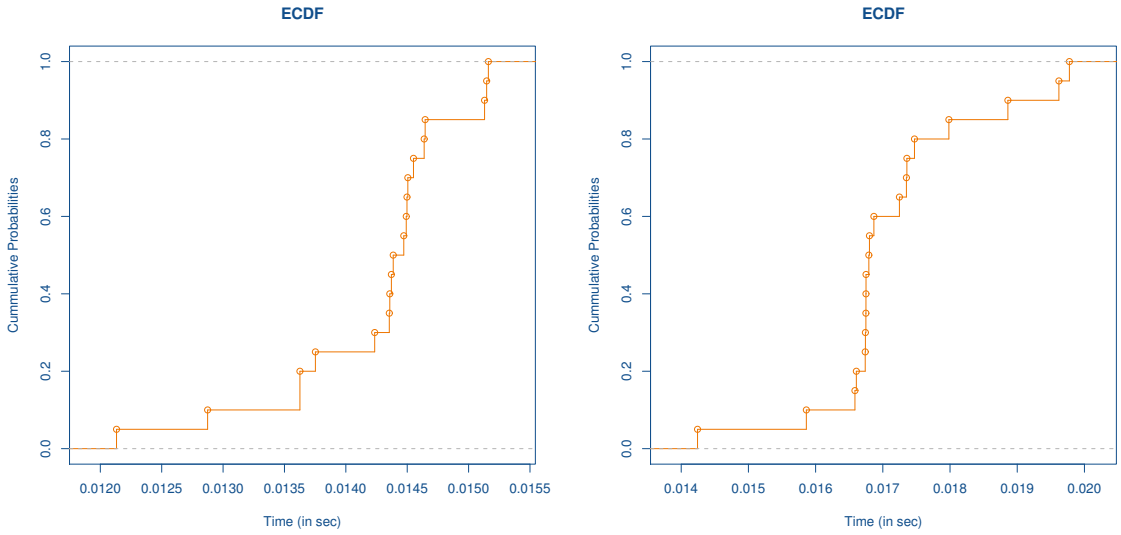
Figure 31: ECDF of total time to have indication of successful joining after the selection of specific SSID (i.e., SSID of the desired network)

.



(a) (b)

Figure 32: (a) ECDF of Authentication time (distance between the station and AP: 2 m),
(b) ECDF of Association time (distance between the station and AP: 2 m).



(a) (b)

Figure 33: (a) ECDF of time gap between the Authentication phase and Association phase (distance between the station and AP: 2 m),
(b) ECDF of total Authentication-Association time (distance between the station and AP: 2 m).

this phase, we capture the packets using AirPcap that is attached to a windows laptop. No capture of management frames are done in the station nor in the access point. In the AirPcap captured data, missing of a Probe Response packet does not necessarily mean that the station has not received the Probe Response. There is a possibility that AirPcap might have failed to capture the packet. On couple of occasions we notice malformed retransmitted packets and acknowledgement packets in the captured data but the source and destination are not recognizable there. It is also possible that the station or the AP might have missed the Probing packets and that is why we do not see the Probe Response packet in the captured data. Due to these limitations as well as the uncertainty, we could not calculate the delays on those particular cases. The same reasoning applies to all the missing entries (marked with "-") in the tables throughout our work unless otherwise mentioned.

After the joining phase the station took 1.247 ms (average time) to *authenticate* itself to the access point and the *association* phase took on an average 1.663 ms (Figure 28). But in Table 3, we can see that in total the Authentication-Association phase took 17.157 ms (average time). The gap between the Authentication phase and Association phase causes the significant jump in the total Authentication-Association time that will be well understood from the analysis of the ECDF graphs. Figure 32a and Figure 32b are the ECDFs of Authentication time and Association time respectively. But from the ECDF of time gap between Authentication and Association phase (Figure 33a) and the ECDF of total Authentication-Association time (Figure 33b), we can see that the time gap between the two phases is the dominant component of the total Authentication-Association time.

The *total link layer connection set-up time* we measure in our work is 3.841 sec. The *scanning delay* we measure contributes 14.98% of the total connection set-up time. But the most dominant component is the delay during the joining phase (including the delay to have the successful joining indication). This *joining delay* contributes 84.57% of the total connection set-up time, whereas the contribution of the *Authentication-Association phase* is 0.45%.

After 189.851 ms (average time) of getting associated to the access point, the station starts ***acquiring an IP address*** for itself by broadcasting DHCPDISCOVER packet. From the *timeline view of the IP acquisition phase* (Figure 34; based on the data from Table 4) we can see that, the average time to receive the DHCPOFFER from the access point (i.e., DHCP server) is 14.680 ms. Then after 49.043 ms of receiving the DHCPOFFER, the station broadcasts a DHCPREQUEST message requesting the offered IP Address. The average delay to receive the DHCPACK from the access point is 19.976 ms. In total, the station takes on an average 83.698 ms (excluding the gap between the association phase and DHCP phase) to acquire the IP address through DHCP. From the ECDF graphs (Figures 35 and 36), we can see that the gaps between the DHCPOFFER and DHCPREQUEST packets contributes the most in the total IP acquisition time (i.e., 58.60% of the total IP acquisition time). The contributions of DHCPDISCOVER-DHCPOFFER and DHCEPREQUEST-DHCPACK to the total DHCP time are 17.54% and 23.87% respectively. When the IP acquisition is over, the station takes on an average 68.425 ms (average time) to send the first packet (i.e., IGMPv3 Membership Report packet)

using the acquired IP address.

SL.	Time to start DHCP process after Association (sec)	Time between DHCPDIS-COVER and DHCPOFFER (sec)	Gap before sending the DHCPRE-QUEST (sec)	Time between DHCPRE-QUEST and DHCPACK (sec)	Gap before sending the first packet using IP address after the DHCP process (sec)
1	0.334227	0.008888	0.098603	0.017111	0.114409
2	0.468625	0.008748	0.123433	0.018148	0.114374
3	0.294477	0.012027	0.089355	0.016657	0.068390
4	0.335106	0.012027	0.042355	0.018514	0.082481
5	0.147934	2.760302 *	0.036478	0.019264	0.058014
6	0.147741	0.008709	0.038115	0.018497	0.059018
7	0.147813	0.008787	0.038210	0.019893	0.057308
8	0.155466	0.008715	0.037840	0.017237	0.083958
9	0.147875	0.008768	0.037984	0.016770	0.060803
10	0.147071	0.008642	0.038284	0.019709	0.057748
11	0.144726	0.021769	0.040996	0.017308	0.060697
12	0.148055	0.011898	0.051277	0.018760	0.057631
13	0.147933	0.008948	0.037671	0.019746	0.073500
14	0.144367	0.008642	0.047630	0.016885	0.059105
15	0.147592	0.080821	0.028472	0.032858	0.060257
16	0.147139	0.015256	0.047040	0.020101	0.059762
17	0.147309	0.011518	0.036796	0.018743	0.057238
18	0.156166	0.008708	0.037428	0.034392	0.058747
19	0.147187	0.011361	0.035585	0.017374	0.061006
20	0.140216	2.266961 *	0.037307	0.021543	0.064053
Average	0.189851	0.014680	0.049043	0.019976	0.068425
Standard deviation	0.091515	0.016837	0.024762	0.004856	0.017593

Table 4: Measured time of the steps during the station's IP Acquisition through DHCP when the distance with the AP is 2 m
* Outlier and omitted while calculating the Average and Standard Deviation

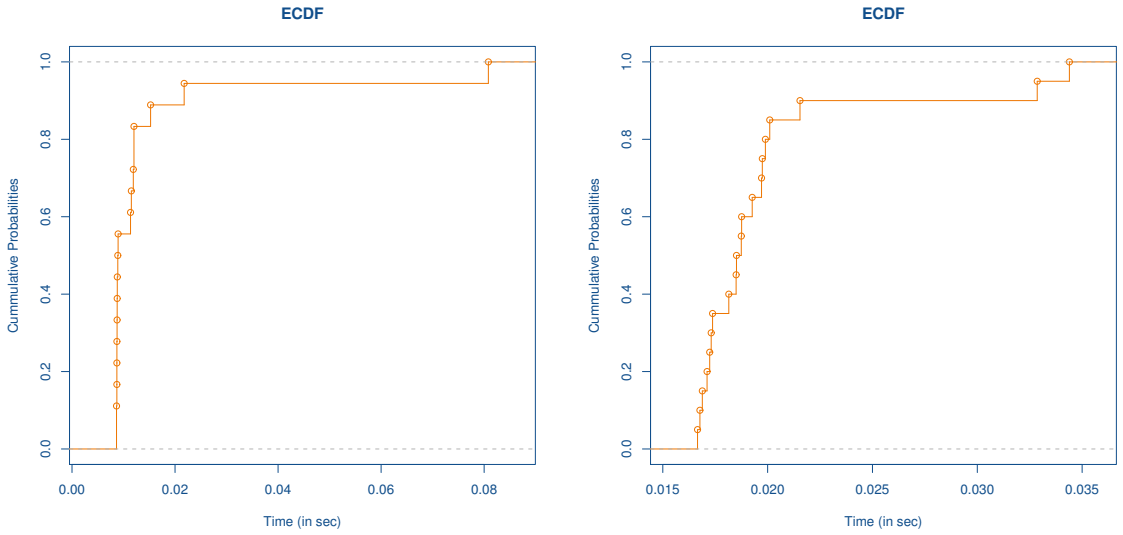
Figure 34: Timeline view of IP acquisition (through DHCP) procedure when the distance between the station and the AP is 2m (Figure not drawn to scale).

In Table 4, we mark some of the entries as outliers (marked with *). These entries are significantly higher in values than the others of the corresponding data point. When we analyze the captured data for those entries, we don't find any

valid reasoning behind such behaviour. For example, in case of the outliers for the DHCPDISCOVER-DHCPOFFER data point, there is no malformed or retransmitted packets are seen in the captured data that might cause the higher delays. As only 2 out of 20 samples have this kind of characteristics which differs significantly from other entries of the data set as well as the absence of valid reasoning behind such characteristics, we mark them as outliers. Those outliers are not used while calculating the average time, standard deviation as well as during the generation of ECDFs. The same reasoning applies to all the outliers (marked with "*") in the tables throughout our work unless otherwise mentioned.

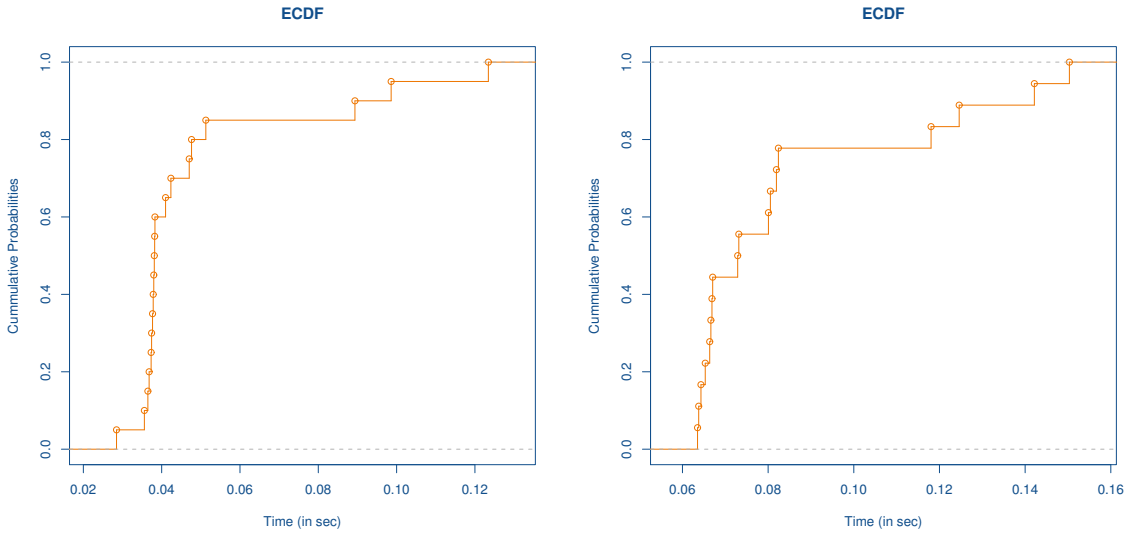
Though there were outliers, there were couple of occasions (i.e., data points) during the IP acquisition phase where we notice retransmission of DHCPDISCOVER and DHCPREQUEST packets by the station (i.e., the DHCP client) that cause extra delays. We find the station retransmitting packets due to the following reason that is a feature of DHCP: DHCP uses UDP (User Datagram Protocol) as a transport layer protocol [29, 64]. Since UDP is unreliable, there is no guarantee that messages will get to their destination. This can lead to potential confusion on the part of a client. For example, a client sends a DHCPDISCOVER packet and waits for DHCPOFFER packet in reply. If it gets no response, the reason could be either no DHCP server is willing to offer it service, or simply that its DHCPDISCOVER packet is lost and the server has never received it. To avoid such confusion the DHCP client takes the responsibility, since the client initiates the contact and can most easily keep track of packets sent and retransmit them when needed. A server can not know when a client's request is lost, but a client can react to a server's reply being lost. During the request/reply message exchange, the DHCP client uses a retransmission timer that is set to a period of time that represents how long it is reasonable for it to wait for a response. If no reply is received by the time the timer expires, the client assumes that either its request or the response coming back is lost [45]. The client must adopt a retransmission strategy that incorporates a randomized exponential backoff algorithm to determine the delay between retransmissions. The delay between retransmissions should be chosen to allow sufficient time for replies from the server to be delivered based on the characteristics (e.g., the speed) of the underlying network between the client and the server. For example, the retransmission delay should be doubled with subsequent retransmissions up to a maximum of 64 seconds in case of a 10 Mbps ethernet network [29].

In our measurements, we also notice the DHCP server (i.e. access point) retransmitting DHCPOFFER packets on couple of occasions when it has not received any acknowledgement packet from the client. To deal with the retransmission of request/response packets observed in our measurement data, we always consider the worst case scenario i.e., calculating the delay between the first request packet and the last retransmitted packet.



(a) (b)

Figure 35: (a) ECDF of DHCPDISCOVER-DHCPOFFER time (distance between the station and AP: 2 m),
(b) ECDF of DHCPREQUEST-DHCPACK time (distance between the station and AP: 2 m).



(a) (b)

Figure 36: (a) ECDF of time gap between DHCPOFFER and DHCPREQ (distance between the station and AP: 2 m),
(b) ECDF of total DHCP time (distance between the station and AP: 2 m).

b. Connection establishment and IP acquisition at longer range (16 m distance)

Figure 37 shows the *timeline view* (drawn based on the average delays from Table 5) of the different steps that take place during the ***IEEE 802.11 link layer connection establishment procedure*** between the station and the AP (distance: 16m). In that Figure, we can see that the station starts *scanning* on channel 6 (the operating channel of the AP) 585.659 ms (average time) after sending the first Probe Request on channel 1 (ECDF graph in Figure 29a). Then the station takes 6.072 ms on an average to discover the desired SSID on channel 6 (ECDF graph in Figure 29b). As of 2 m distance, in this case we also see that the time spent to discover the desired network on channel 6 is very short compared to the time the station spends scanning on some other channels.

When the desired SSID is selected in the scan list, in total it takes on an average 3.108 sec to have the indication of successful *joining* to the selected BSS (ECDF in Figure 39a. Then the *Authentication-Association* phase takes on an average 17.214 ms (ECDF in Figure 39b). *The total link layer connection set-up time* we calculate in our measurements (for 16 m distance) is 3.717 sec. The joining delay is the most significant contributor in the total connection set-up time with a contribution of 83.62%, whereas the scanning delay and the Authentication-Association delay contribute 15.92% and 0.46% of the total delay respectively.

SL.	Time difference between the 1st Probe Requests on Channel 1 and Channel 6 (sec)	Time to discover the desired SSID (i.e., desired network) on channel 6 (sec)	Total time to have the indication of successful joining after the selection of specific SSID (sec)	Total time Authentication and Association phases take (the gap between the phases included) (sec)
1	0.585826	-	3.108779	0.019905
2	0.585657	0.003499	3.155907	0.016653
3	0.585837	0.006859	3.116180	0.018690
4	0.585937	-	3.155647	0.016622
5	0.585908	-	3.038449	0.016620
6	0.587761	0.006604	3.109105	0.017026
7	0.585946	0.005376	3.062062	0.018220
8	0.586008	0.006850	3.034049	0.016609
9	0.585784	0.006053	3.163495	0.016777
10	0.579801	0.005854	3.140618	0.017019
11	0.586067	0.006107	3.139989	0.016894
12	0.585963	-	3.155651	0.016621
13	0.585910	0.006987	3.098522	0.018771
14	0.585985	0.006100	3.064777	0.015777
15	0.585953	0.007028	3.054250	0.016878
16	0.585921	0.006266	3.085406	0.016676
17	0.585935	0.005351	3.147958	0.016886
Average	0.585659	0.006072	3.107697	0.017214
Std. Dev.	0.001577	0.000956	0.044329	0.001016

Table 5: Measured time of the steps during the connection establishment procedure between the station and the AP (distance 16 m)

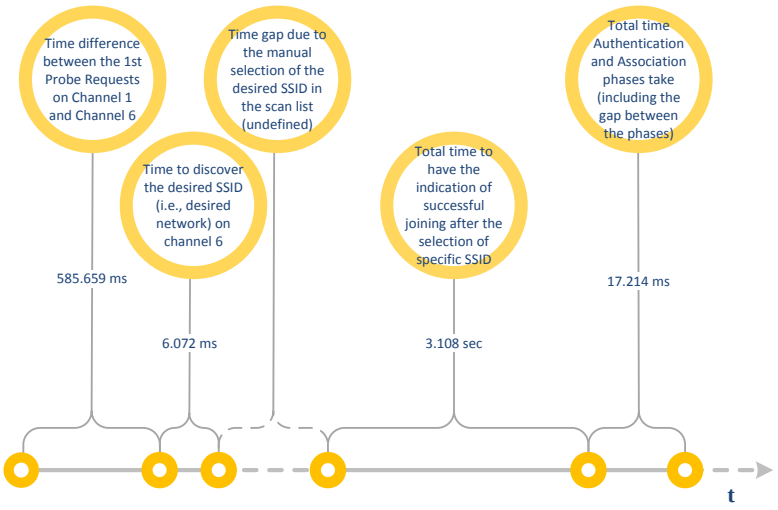
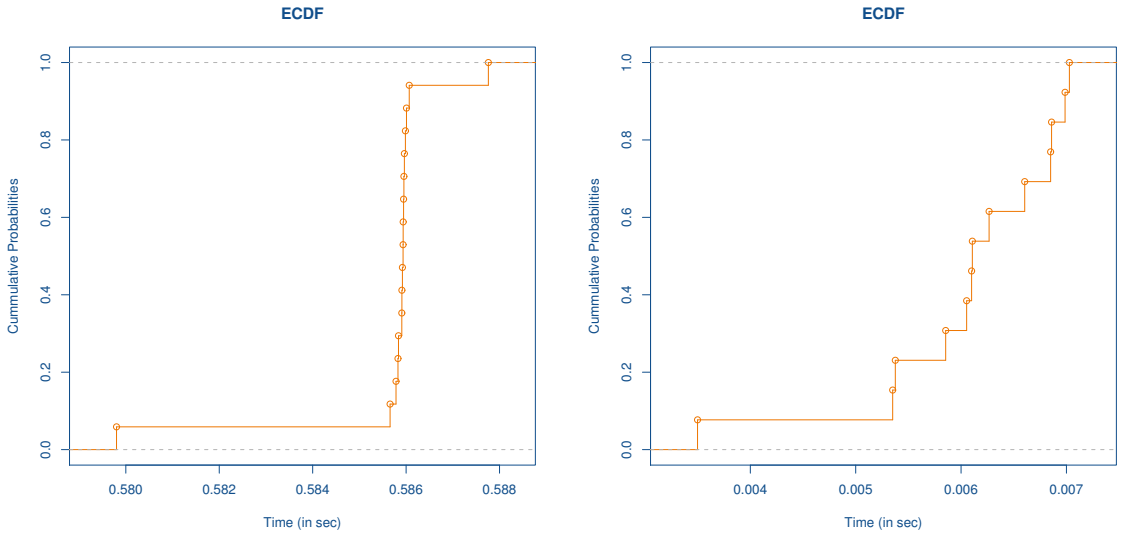
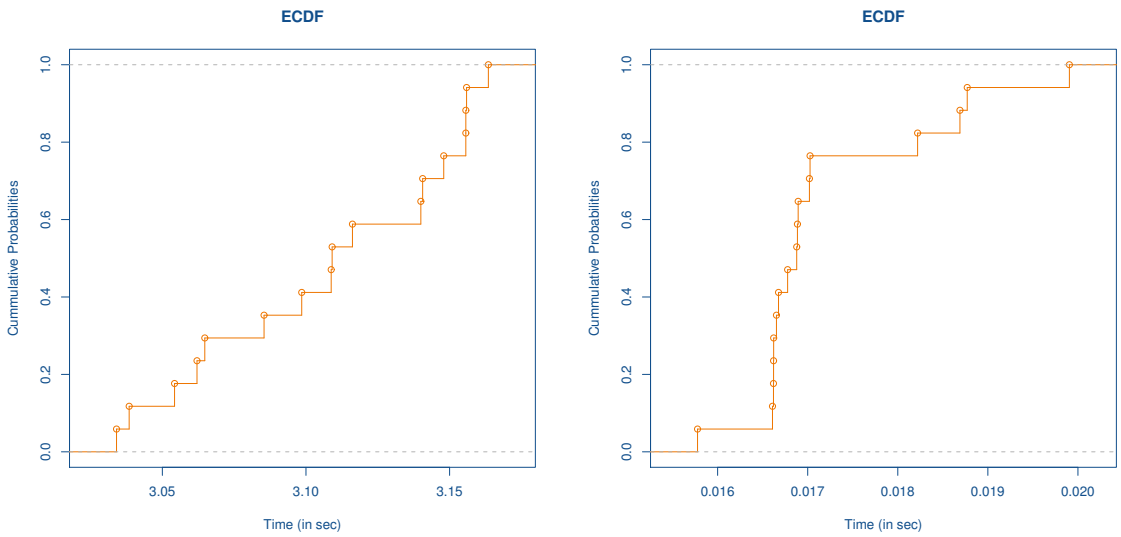


Figure 37: Timeline view of connection establishment procedure when the distance between the station and the AP is 16m (Figure not drawn to scale).



(a) (b)

Figure 38: (a) ECDF of time difference between the first probings on channel 1 and channel 6 (distance between the station and AP: 16 m),
(b) ECDF of desired network discovery time on channel 6 (distance between the station and AP: 16 m).



(a) (b)

Figure 39: (a) ECDF of total time to have indication of successful joining after the selection of specific SSID (distance between the station and AP: 16 m),
(b) ECDF of total Authentication-Association time (distance between the station and AP: 16 m).

After 149.352 ms (average time) of getting associated to the access point, the station starts ***acquiring an IP address*** by broadcasting DHCPDISCOVER packet. From the *timeline view of the IP acquisition phase* (Figure 40; based on the average delays from Table 6) we can see that, in total the station takes on an average 99.504 ms to acquire the IP address through DHCP (ECDF graph is in Figure 41). When the IP acquisition is over, the station takes on an average 60.167 ms (average time) to send the first packet (i.e., IGMPv3 Membership Report packet) using the acquired IP address.

While presenting the connection establishment and IP acquisition related data in this section we mostly show the total duration of the phases as can be seen in the Tables 5, 6 as well as in the Figures 37, 40. Due to the longer distance (i.e., 16 m) between the station and the access point, the position of the AirPcap devices are not close enough from both the devices (i.e., the station and the access point). That causes the AirPcap to drop some of the packets and consequently restrains us from deriving the timeline for the smaller steps that take place within a phase. But we have sufficient data sets to derive the total duration of a phase and that enables us to gain an understanding on the overall delays that the different phases go through during the connection establishment and IP acquisition procedures.

SL.	Time to start DHCP process after Association (sec)	Total DHCP time (sec)	Gap before sending the first packet using IP address after the DHCP process (sec)
1	0.144604	3.057618*	0.061204
2	0.163278	0.132064	0.054774
3	0.145900	0.104488	0.074371
4	0.155627	0.099631	0.056134
5	0.147816	0.085866	0.061828
6	0.147042	0.090698	0.058113
7	0.154270	0.072007	0.067637
8	0.163402	0.097105	0.058451
9	0.163312	0.104172	0.043904
10	0.147103	0.142995	0.044516
11	0.147585	0.071144	0.061086
12	0.147874	0.102531	0.060714
13	0.137735	0.095391	0.060240
14	0.145948	0.105143	0.058372
15	0.139862	0.090627	0.080657
16	0.140012	0.101067	0.062359
17	0.147618	0.097139	0.058482
Average	0.149352	0.099504	0.060167
Std. Dev.	0.008020	0.018207	0.008857

Table 6: Measured time of the steps during the station's IP Acquisition through DHCP when the distance with the AP is 16 m
* Outlier and omitted while calculating the Average and Standard Deviation

We run our measurements keeping the station 2 m and 16 m apart from the access point to study the impact of indoor wireless signal characteristics (discussed in 2.2) on the delays the station experiences during the connection establishment and IP acquisition procedures. In the results (i.e., the delays) obtained from the 2 m and 16 m measurements, we can see that the delays for the different phases are almost identical between these two data sets (comparing the timeline figures, i.e., Figure 28 with Figure 37 and Figure 34 with Figure 40). The indoor wireless signal

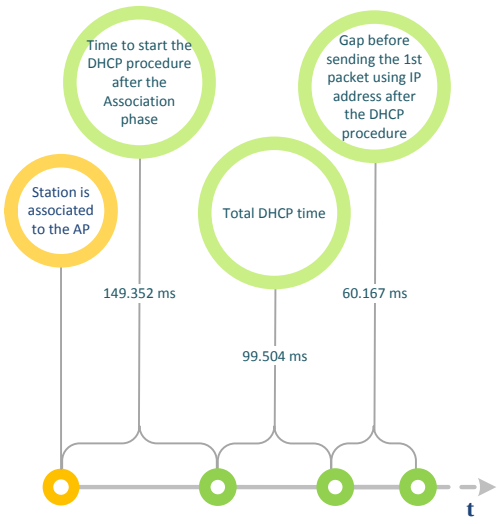


Figure 40: Timeline view of IP acquisition (through DHCP) procedure when the distance between the station and the AP is 16m (Figure not drawn to scale).

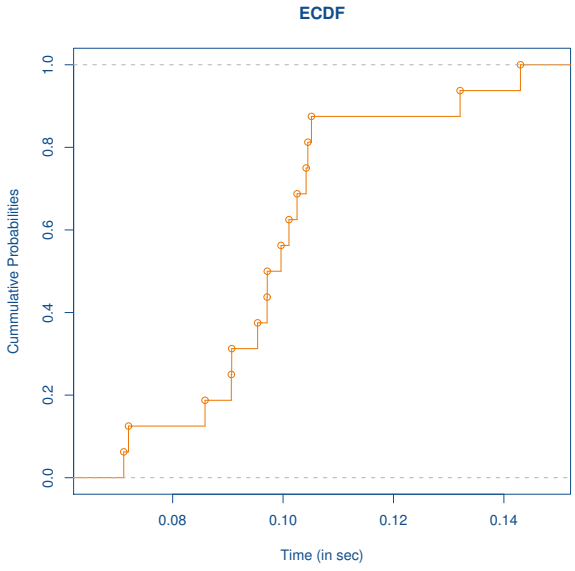


Figure 41: ECDF of total IP acquisition through DHCP when the distance between the station and the AP is 16m.

characteristics are very dynamic in nature and varies from location to location. Also the characteristics of the wireless signal on a particular location gets changed over time. Our understanding is, it would be better to conduct more sets of experiments changing the location of the station to have an understanding on the impact of indoor wireless characteristics on the delays. But due to nature of the experiments (i.e., very time consuming) as well as due to scope of the Master's Thesis work we restrain ourselves from conducting more experiments.

5.2 Measurement and Analysis of Service Discovery Phase

In this section, we present our analysis on the measured data of the steps that a station (i.e., the client node) goes through during the service discovery phase (details on service discovery are in section 2.5) .

The station joins the Wi-Fi network where another node (i.e., the server node), has published a service named "DtnUpload". As discussed in section 2.5.2, after getting connected to the AP and acquiring an IP address, the client node reports (i.e., by sending IGMPv3 Membership Report packet) to its neighboring multicast node(s) in the network about its multicast group membership and then it goes through the name reservation procedure. The client first claims a domain name and if there is no conflict, it announces the uniqueness of its name in the network. Here, we calculate the time difference between the 1st IGMPv3 packet and the 1st mDNS query packet that is sent for name claiming. Then we measure how much time the name reservation procedure takes by calculating the time difference between the name claiming (1st claim) and the announcement (1st announcement) (Figure 42). This name reservation phase is accomplished by the Avahi-daemon running at the client node.

After the name reservation phase, the Service Browser application at the client node is started with user intervention and the application starts querying for the service (details on the application are in section 3.1.3). The application creates the service browser and then the Avahi-daemon running at the client node sends query for service instance name of a particular service type (_afs3-fileserver._tcp.local.). The client then receives response from the server (Service Publisher) with the service instance name (DtnUpload._afs3-fileserver._tcp.local.). After that the steps of service resolution take place. In section 2.5.4, we have discussed in detail on the service browsing and resolution procedures. But in our experiments, we have found that the DNS-SD has followed the traffic reduction mechanism (that is also discussed in section 2.5.4) during the service discovery and resolution phases. So the query response from the server includes all the necessary information (i.e., service instance name, host name, port and IP address) (Figure 42). At this point, we measure the query response time at the client node.

Service Browser application at the client node uses the resolved information (i.e., port and IP address) for the next phase of our work that involves data transmission to the server over TCP. At this stage, ARP (Address Resolution Protocol) is used for the resolution of network layer address (IP address) into link layer address (MAC address) [61]. We measure the ARP Request-ARP Response time at the client node

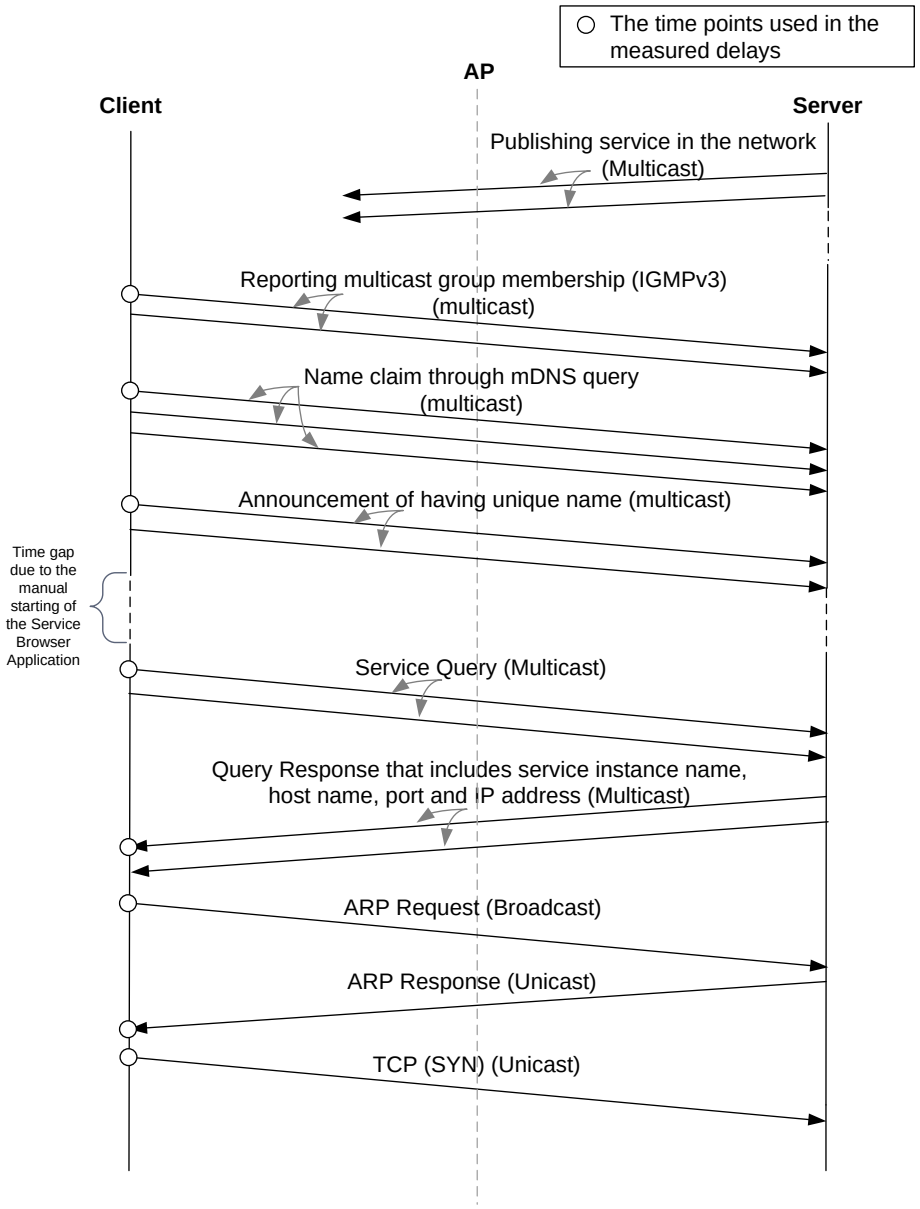


Figure 42: Time capturing points during the service discovery phase.

as well as the time gap before the client node starts sending the TCP SYN packet as part of a TCP handshaking process (Figure 42) [63].

5.2.1 Experimental Setup

Experiments in this phase are carried out in the indoor testbed environment that is discussed in section 3.3. Our measurement study in this part involves two stations (i.e., Nokia N810 and Dell Latitude D830 laptop) and an access point (Raspberry Pi).

To conduct the experiments in this phase, we create a scenario where a station (server node; Dell Latitude D830 laptop) is already connected to the AP. In our experiments the server node is placed on the same trolley as of the AP. So it has the same position as the AP but with different height. The locations of the AP (red dot) and the server node (green dot) are marked in Figure 43. First we make sure that Avahi-daemon is running in the server node and then run the Service Publisher application on it. Now the server node starts advertising the service "DtnUpload" in the network.

At this phase of the experiment, another station (i.e., client node; Nokia N810) starts trying to establish connection with the AP. When the connection is established and the client node acquires an IP address through DHCP, we run the Service Browser application in the client node. It is necessary to make sure that Avahi-daemon is already running in the client node when we start the application. The daemon will accomplish the task of name reservation at the client node before we run the Service Browser application on it.

When we run the Service Browser application, the client starts working as a service browser for the desired service type and after some time it discovers the service instance name of its desired service type that is offered by the server node in the network. Then the service resolution takes place and using the resolved information (i.e., port and IP address), the client establishes TCP connection with the server to transfer data.

Figure 43: Testbed area. (Figure not drawn to scale).

While carrying out the experiments, the client node is kept in two positions, i.e.,

at 2 m and 16 m apart from the access point but we keep the height same (i.e., 110 cm) (Figure 43). The reason of keeping the station in two different positions is to see the impact of indoor wave propagation characteristics (discussed in section 2.2) on the delays measured during the service discovery phase. For both of these positions, we repeat the whole process (that is mentioned above) of the experiment 20 times and each time we use an AirPcap device to capture the packets (on channel 6) related with the naming phase and for other parts of the experiments (i.e., to measure the delays of service query-response, ARP request-reply and gap before TCP SYN) we run tcpdump at the client node. We use AirPcap for capturing packets during the naming phase as it takes sometime to run the tcpdump at the client node (after it gets connected to the AP) and this delay makes the tcpdump unable to capture the naming related packets. To check the link reliability (i.e., success/failure of packet transfer) we capture packets at the server end as well using Wireshark. The time capturing feature of the application helps us to have an understanding on the different events that take place at the application layer (time capturing feature is discussed in section 3.1.3).

5.2.2 Results and Findings

In this section, we give emphasis on the experimental results that are measured when the client node goes through the different steps of service discovery (i.e., naming, service browsing and service resolution etc.) phase. We carry out the experiments in this part keeping the client node at two different locations (i.e., 2 m and 16 m apart from the AP) to see the impact of indoor wireless characteristics on the measured delays. Below are the discussions on the experimental data for those two different positions of the client node.

a. Service Discovery phase at 2 m distance (close range)

Figure 44 shows the *timeline view* (based on the average delays from Table 7 and Table 8) of the different steps that take place during the *service discovery* phase, namely, naming, service browsing and service resolution. The timeline view also shows the delay between ARP Request and ARP Reply as well as the time gap before the TCP handshaking.

In the timeline figure, we can see that after 150.687 ms (average time) of multicast group membership reporting, the client node starts the name reservation procedure. The average time between the domain name claim and the first announcement of its uniqueness is 700.228 ms (ECDF graph is shown in Figure 45).

The Service Browser application at the client node starts at this stage and in the timeline figure, we can see some gap (undefined) due to the manual starting of the service browser application. When the application starts, the client node sends a service query. The average time it takes for the client to receive the response is 292.709 ms. The ECDF graph (Figure 46) based on the delays during the query-response phase shows that on several occasions the response time is around 1 second that is significantly higher compared to the average delay of 292.709 ms.

SL.	After the 1st IGMPv3 Membership Report packet, time gap before the name reservation process starts (sec)	Time required to have the confirmation of no name conflict (sec)
1	0.132684	0.713657
2	0.257811	0.693275
3	0.125594	0.688618
4	0.133695	0.682843
5	0.226636	0.706380
6	0.164328	0.697234
7	0.070783	0.704724
8	0.109945	0.703713
9	0.141141	0.696954
10	0.109816	0.689846
11	0.229602	0.694212
12	0.265403	0.697058
13	0.172199	0.698842
14	0.051372	0.700848
15	0.273854	0.708611
16	0.065475	0.705094
17	0.076466	0.698720
18	0.211712	0.705846
19	0.055230	0.714399
20	0.139996	0.703695
Average	0.150687	0.700228
Std. Dev.	0.072214	0.008119

Table 7: Measured time during the name reservation (distance 2 m)

SL.	Time to receive the Service Query Response (sec)	After Service Query Response, time gap before ARP Request (sec)	Time between ARP Request and ARP Reply (sec)	After ARP process, time gap before the TCP handshaking (i.e., before the data transmission over TCP) (sec)
1	0.082001	0.019074	0.005462	0.000092
2	0.056396	0.026917	0.005981	0.000061
3	0.934631	0.157623*	0.004303	0.000061
4	0.109619	0.022400	0.004395	0.000061
5	0.094879	0.021118	0.004120	0.000092
6	0.103485	0.020080	0.004364	0.000061
7	1.071777	0.020844	0.004974	0.000061
8	0.130920	0.024323	0.006012	0.000061
9	0.044036	0.024903	0.004394	0.000061
10	1.131440	0.025482	0.004059	0.000061
11	0.081574	0.026641	0.006317	0.000092
12	0.088806	0.028351	0.004455	0.000061
13	3.058716*	0.028137	0.004333	0.000061
14	0.084534	0.141083*	0.020447	0.000061
15	1.165588	0.026306	0.112763*	0.000091
16	0.033966	0.074798	0.004181	0.000092
17	0.084748	0.029693	0.005249	0.000061
18	0.102570	0.028259	0.004578	0.000061
19	0.069153	0.023437	0.010010	0.000061
20	0.091339	0.026764	0.007263	0.000061
Average	0.292709	0.027640	0.006047	0.000069
Std. Dev.	0.418163	0.012177	0.003778	0.000013

Table 8: Measured time of the steps during the service discovery phase (distance 2 m).
* Outlier and omitted while calculating the Average and Standard Deviation.

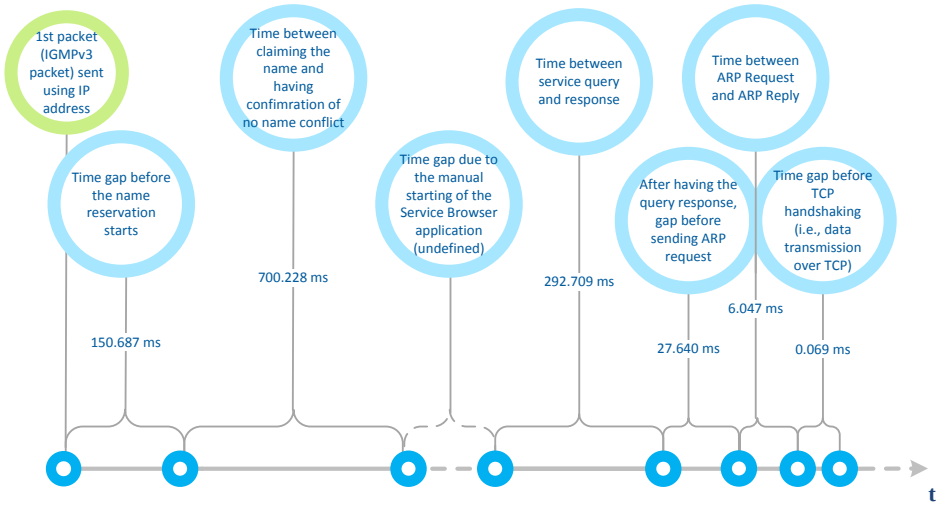


Figure 44: Timeline view of the name reservation and service discovery phases between the two stations when the distance between one of the stations (i.e., the client node) and the AP is 2 m (Figure not drawn to scale).

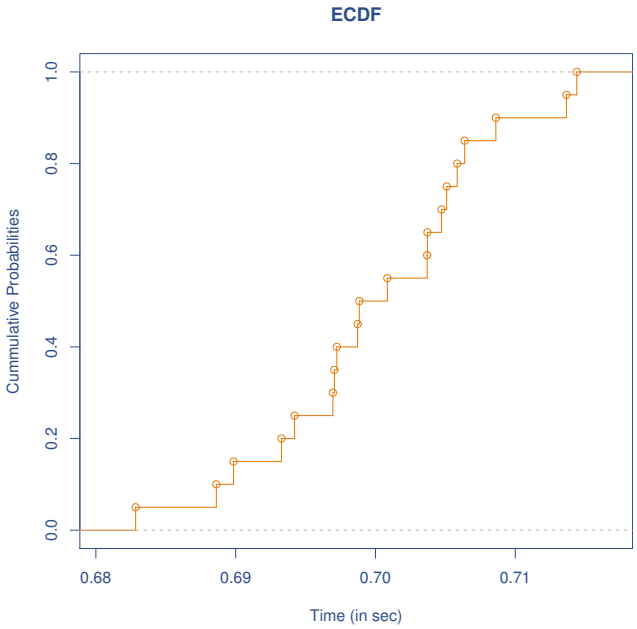


Figure 45: ECDF of name reservation time when the distance between the client node and the AP is 2 m.

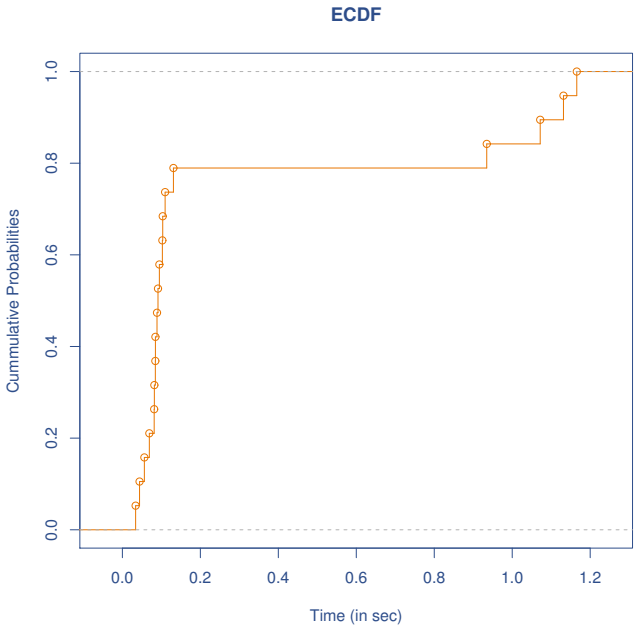


Figure 46: ECDF of service query response time when the distance between the client node and the AP is 2 m.

While investigating the reasoning behind this, we notice the client sending multiple multicast queries before it receives the query response. The reason of multiple query packets could be the retransmission due to the loss of query packets in the wireless medium. But in the internet draft on DNS-SD/mDNS extensions [52], we can see that, though to improve transmission reliability, the IEEE 802.11 MAC requires positive acknowledgement of unicast frames, it does not, however, support positive acknowledgement of multicast frames. So there is no retransmission mechanism based on the acknowledgement packets in case of multicast frames in IEEE 802.11 wireless medium. The reason of multiple query packets is a mechanism called 'Continuous Multicast DNS Querying' that is mentioned in RFC 6762 [48]. In this mechanism, having received one response is not necessarily an indication that there will be no more relevant responses and the querying operation continues until no further responses are required. Determining when no further responses are required depends on the type of operation being performed. For example, if the operation is looking up the IPv4 and IPv6 addresses of another host, then no further responses are required once a successful connection has been made to one of those IPv4 or IPv6 addresses. In those cases of higher query-response delays, when we analyze the server side data, we notice two scenarios. In one scenario, we find the server receiving multiple queries before it starts responding. So the higher delay in that scenario is just because of the Avahi-daemon (at the server node) taking longer time to respond though it has received the earlier query packets. In the other scenario, we see the server responding just after receiving the first query packet but the response packet cannot reach its destination (i.e., the client node).

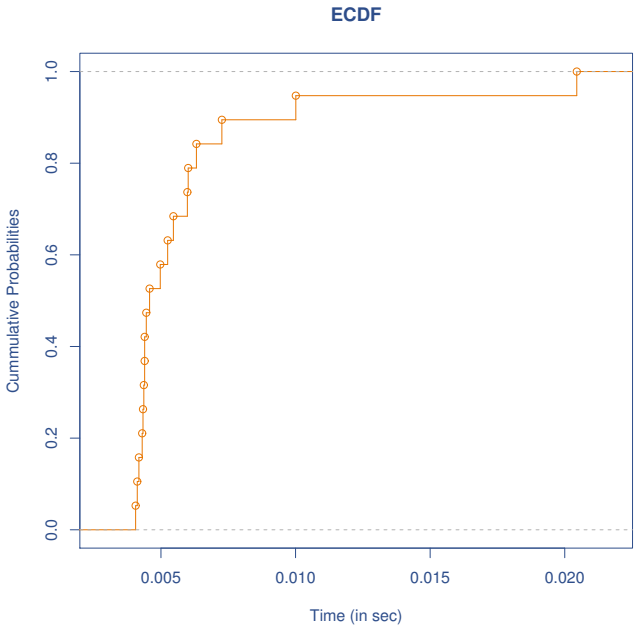


Figure 47: ECDF of ARP Request-ARP Reply time when the distance between the client node and the AP is 2 m.

Using the resolved information (i.e., port and IP address) from the query response, the application at the client node starts transmitting data to the server over TCP. But before establishing the TCP connection, the client has to go through the ARP Request-Reply phase. In our experiments, after 27.640 ms (average time) of receiving the service query response, the client sends ARP Request to the server (Figure 44). Through the time capturing mechanism of the Service Browser application, we come to know about the events that take place at application layer during this 27.640 ms gap before ARP. We see that the application gets the service instance name first and then the resolved port and IP address. When the ARP Request is sent, it takes on an average 6.047 ms for the client node to have an ARP Reply from the server node (ECDF graph in Figure 47). After the ARP phase, the client takes 0.069 ms to send the TCP SYN packet to the server in order to establish TCP connection for transferring data.

b. Service discovery phase at 16 m distance (longer range)

Figure 48 shows the *timeline view* (based on the average delays from Table 9 and Table 10) of the different steps that take place during the *service discovery* phase (i.e., naming, service browsing and service resolution) when the distance between the client node and the AP is 16 m. The timeline view also shows the delay between ARP Request and ARP Reply as well as the time gap before the TCP handshaking.

In the timeline figure, we can see that after 154.015 ms (average time) of multicast group membership reporting, the client node starts the name reservation procedure. The average time between the domain name claim and the first announcement of

SL.	After the 1st IGMPv3 Membership Report packet, time gap before the name reservation process starts (sec)	Time required to have the confirmation of no name conflict (sec)
1	0.107294	0.683926
2	0.087194	0.696472
3	0.062378	0.704946
4	0.086258	0.714491
5	0.165596	0.689109
6	0.174771	0.695499
7	0.055607	0.706186
8	0.141921	0.712599
9	0.338492	0.508586
10	0.070999	0.696767
11	0.111542	0.696292
12	0.295113	0.697171
13	0.119029	0.689365
14	0.234669	0.736379
15	0.251585	0.710537
16	0.226782	0.705588
17	0.118374	0.683016
18	0.133986	0.689204
19	0.102998	0.696854
20	0.195712	0.690125
Average	0.154015	0.690156
Std. Dev.	0.077854	0.044524

Table 9: Measured time during the name reservation (distance 16 m)

SL.	Time to receive the Service Query Response (sec)	After Service Query Response, time gap before ARP Request (sec)	Time between ARP Request and ARP Reply (sec)	After ARP process, time gap before the TCP handshaking (i.e., before the data transmission over TCP) (sec)
1	0.124665	0.029235	0.004364	0.000061
2	1.090149	0.026794	0.008026	0.000062
3	0.058533	0.024963	0.010590	0.000061
4	0.052185	0.033569	0.121521*	0.000061
5	3.116333*	0.085327	0.005371	0.000061
6	1.138580	0.024963	0.004609	0.000061
7	1.116272	0.023255	0.004822	0.000061
8	1.092712	0.022461	0.006744	0.000061
9	0.049988	0.027496	0.005920	0.000061
10	0.062683	0.030487	0.005310	0.000091
11	3.129059*	0.026916	0.005005	0.000061
12	0.080842	0.028289	0.004395	0.000061
13	0.119721	0.020202	0.008637	0.000061
14	0.126801	0.020538	0.004792	0.000061
15	0.113525	0.026764	0.004822	0.000092
16	0.047760	0.024109	0.006836	0.000061
17	1.079620	0.021362	0.005249	0.000061
18	0.060363	0.032929	1.022125*	0.000061
19	0.093750	0.023224	0.008362	0.000061
20	0.123931	0.031189	0.005829	0.000061
Average	0.368449	0.029204	0.006093	0.000064
Std. Dev.	0.469975	0.013775	0.001760	0.000009

Table 10: Measured time of the steps during the service discovery phase (distance 16 m).
* Outlier and omitted while calculating the Average and Standard Deviation.

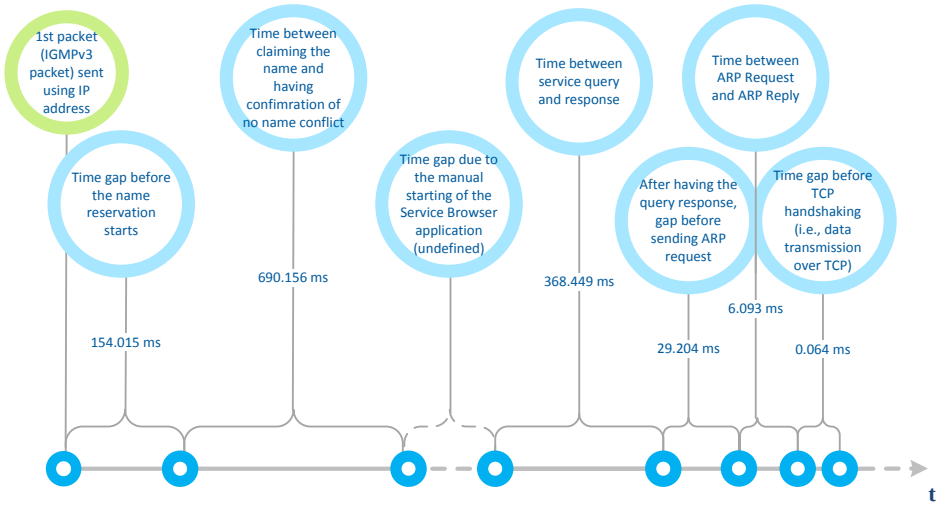


Figure 48: Timeline view of the name reservation and service discovery phases between two stations when the distance between one of the stations (i.e., the client node) and the AP is 16 m (Figure not drawn to scale).

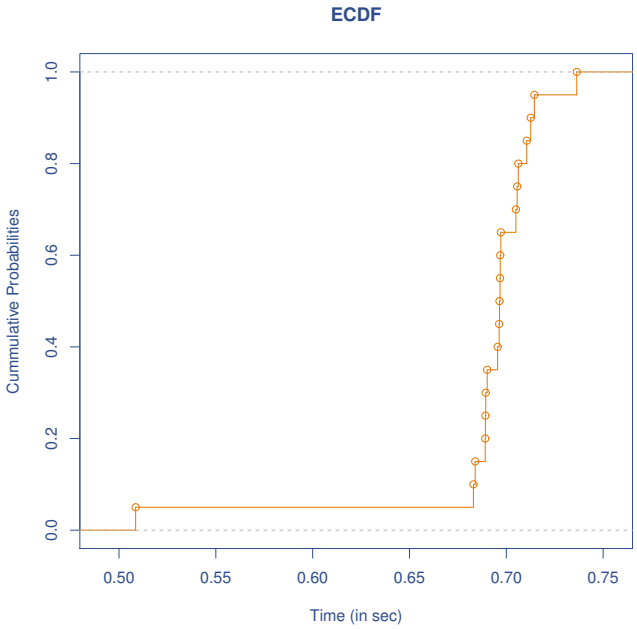


Figure 49: ECDF of name reservation time when the distance between the client node and the AP is 16 m.

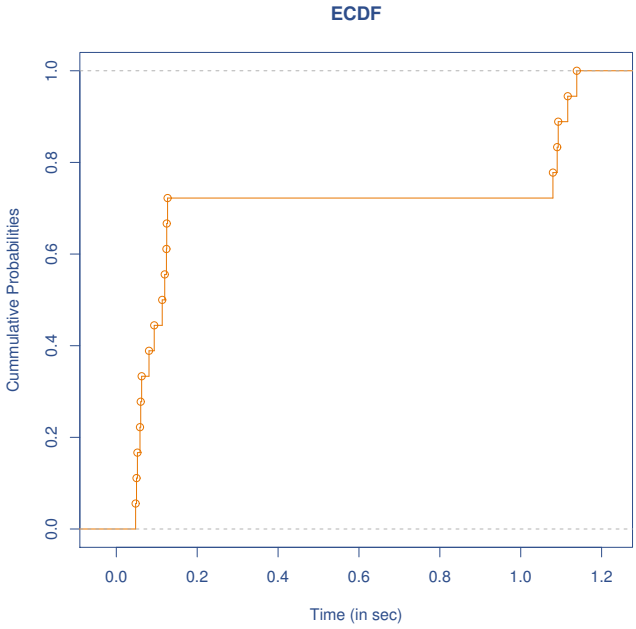


Figure 50: ECDF of service query response time when the distance between the client node and the AP is 16 m.

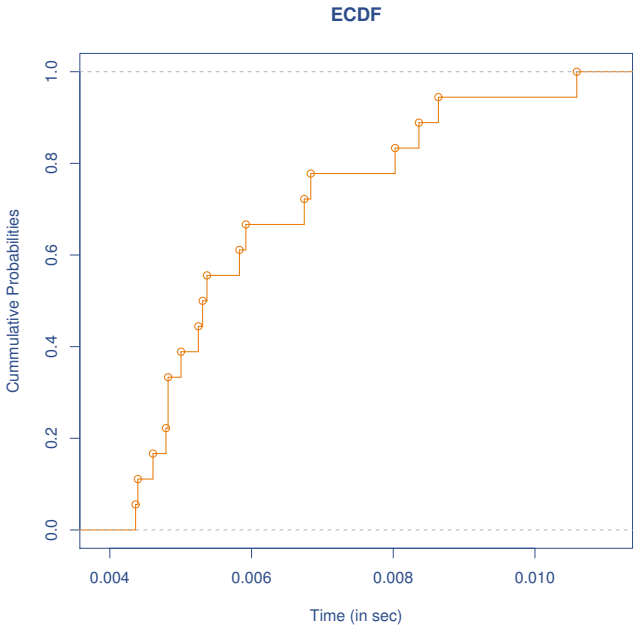


Figure 51: ECDF of ARP Request-ARP Reply time when the distance between the client node and the AP is 16 m.

its uniqueness is 690.156 ms (ECDF graph in Figure 49). Then the service browser starts at the client node and the average time between the service query and response is 368.449 ms. As of the 2 m case, here (ECDF graph: Figure 50) we also notice higher query-response delays (i.e., around 1 sec) on some occasions. The reason is the same as discussed in the 2 m case.

After 29.204 ms (average time) of receiving the service query response, the client sends ARP Request to the server. It takes on an average 6.093 ms to have the ARP Reply from the server node (ECDF graph in Figure 51). After the ARP phase, the client takes 0.064 ms to send the TCP SYN packet to the server in order to establish TCP connection for transferring data.

We run our measurements keeping the client node 2m and 16 m apart from the access point during the service discovery phase in order to study the impact of indoor wireless signal characteristics (discussed in 2.2) on the measured delays. In the results (i.e., the delays) obtained from the 2m and 16 m measurements (Figure 44 and Figure 48), we can see that there is no significant difference between these two data sets. Conducting more sets of experiments changing the location of the client node would be better to have more in depth understanding on the impact of indoor wireless characteristics on the delays. But due to nature of the experiments (i.e., very time consuming) as well as due to scope of the Master's Thesis work we restrain ourselves from conducting more experiments.

5.3 Measurement and Analysis of Data Transmission over TCP

In the last phase of our work, the Service Browser application at the client end starts uploading data to the server using the "DtnUpload" service offered by the server. The experiments of this phase are carried out keeping the client node in different positions to study the impact of indoor wireless environment on data throughput when TCP is used as the transport layer protocol. We use TCP in our experiments as one of the goals of the Thesis is to check the suitability of TCP as a transport layer protocol in opportunistic networks.

5.3.1 Experimental Setup

Experiments in this phase are carried out in the indoor testbed environment that is discussed in section 3.3. As of the service discovery phase, our measurement study in this part also involves two stations (i.e., Nokia N810 and Dell Latitude D830 laptop) and an access point (Raspberry Pi). The server node is placed on the same trolley as of the AP. So it has the same position as the AP but with different height. The locations of the AP (red dot) and the server node (green dot) are marked in Figure 52. The client node is kept in four different positions, i.e., at 2 m, 8 m, 16 m and 32 m apart from the access point while keeping the height always same (i.e., 110 cm) (Figure 52). The client node is kept stationary on those positions and a clear LOS between the client node and the AP is always maintained during the experiments.

Figure 52: Testbed area. (Figure not drawn to scale).

While carrying out the experiments in this phase, we focus on the data transmission part, though the station has to go through the connection establishment, IP address acquisition and service discovery phases before it can establish TCP connection with the server for data transmission. After the service discovery phase, the service browser application at the client node establishes TCP connection with the server node using the resolved port and IP address information in order to transfer 100 MB of data to the server. The reason of transferring large volume of data in the experiments of opportunistic contacts is to monitor the throughput behaviour of the TCP connection for a longer period of time. We perform single run of the experiment on each location (i.e., at 2 m, 8 m, 16 m and 32 m) and capture the traffic during the data transmission using Wireshark at the server node. Later we use tcptrace for analyzing the captured traffic [6]. We also use AirPcap device to capture traffic on channel 6 (operating channel of the selected BSS).

5.3.2 Results and Findings

Figure 53, 54, 55 and 56 show the throughput of the data transmission over TCP between the client node and the server node when the distance between the client node and the AP is 2 m, 8 m, 16 m and 32 m respectively.

In the throughput graphs, we can see that the throughput is greatly affected by distance. The throughput for the 2 m, 8 m, 16 m and 32 m distances is 56.709 KBps, 24.934 KBps, 3.674 KBps and 531 Bps respectively. In our experiments, we observe that, while 100 MB data transfer is possible at 2 m and 8 m distances, the connection between the client node and the AP gets disconnected after 6.875 MB of data transfer when they are located 16 m apart. In case of 32 m distance, i.e., when the client is located almost at the edge of the AP coverage area, we find the client node getting disconnected from the AP just after 112.944 KB of data transfer.

Though 802.11b has a maximum data rate of 11 Mbps, it is important to remember that, due to the CSMA/CA protocol overheads, in practice the maximum achievable 802.11b throughput over TCP is about 5.9 Mbps [31]. In our experiments, though at the close range (i.e., 2 m apart from the AP) the client node could

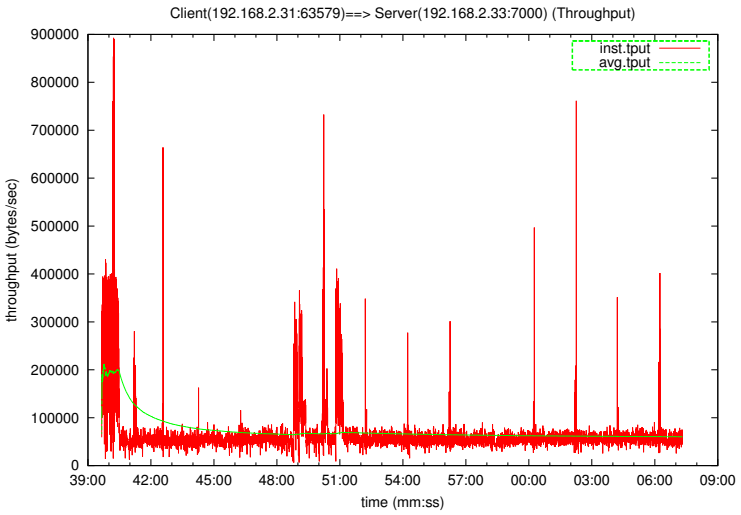


Figure 53: Throughput graph when the station (i.e., the client node) is 2 m apart from the AP.

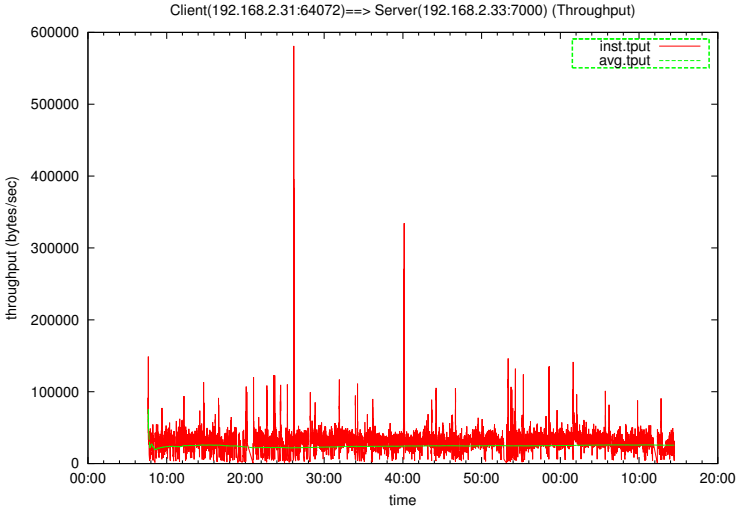


Figure 54: Throughput graph when the station (i.e., the client node) is 8 m apart from the AP.

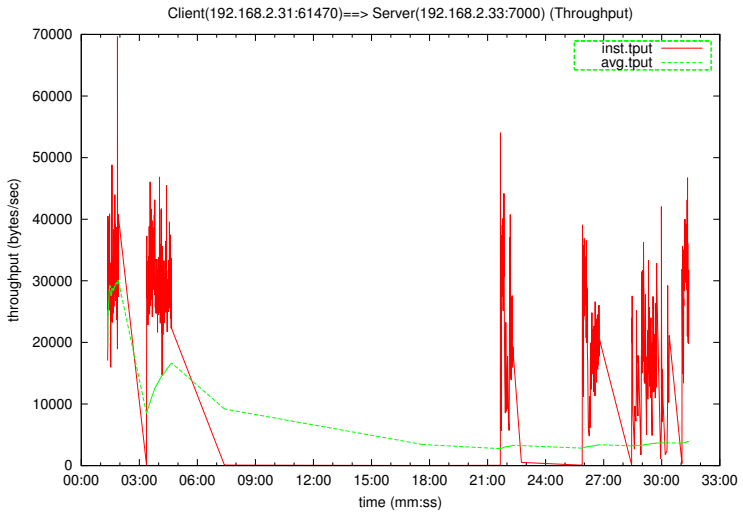


Figure 55: Throughput graph when the station (i.e., the client node) is 16 m apart from the AP.

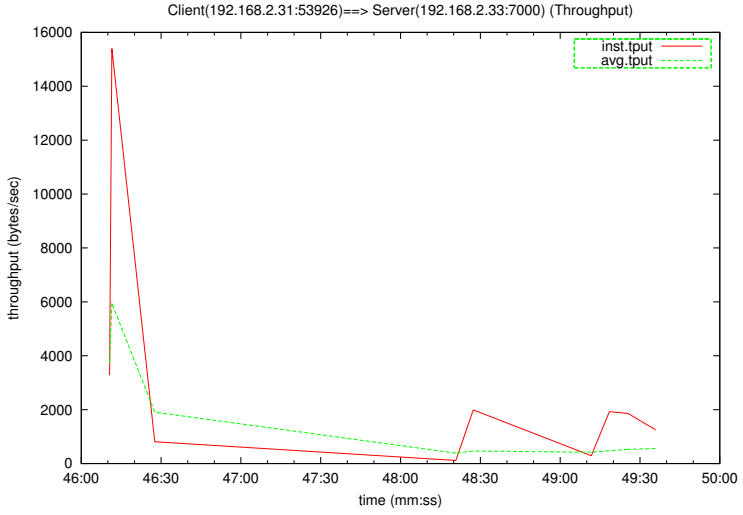


Figure 56: Throughput graph when the station (i.e., the client node) is 32 m apart from the AP.

send 100 MB data to the server node, the throughput (i.e., 56.709 KBps) achieved in that position fall short significantly even to match with the 5.9 Mbps standard.

In Figure 53, we can see that at the beginning of the data transfer the throughput is around 400 KBps but it drops sharply after couple of minutes. There are several spikes on some occasions, but the higher level of throughput could never sustain for longer period. The reason behind sharp drop in throughput as well as the low throughput when the client node is at close range of the AP, could be the neighbouring wireless devices (that are part of the University Wi-Fi networks) active in the testbed area. When we analyze the traffic captured by the AirPcap device, we find that some of those neighbouring wireless devices are operating on channel 6 as of the stations and the AP of our experiments. So the reason of low throughput and sharp drop in throughput could be the client node's sharing of channel capacity with the neighbouring wireless devices. Another reason could be the indoor RF wave propagation characteristics (e.g., path loss/attenuation due to multipath propagation) that might have contributed in the degradation of signal quality and thus in lowering the throughput. When we analyze the captured traffic, we notice that the data rate keeps on changing during the data transmission phase. 802.11b network cards can operate at 11 Mbps but can scale back to 5.5 Mbps, 2 Mbps, then 1 Mbps depending on signal quality. Since the lower data rates use less complex and more redundant methods of encoding the data, they are less susceptible to corruption due to interference and signal attenuation [31]. So the frequent change of data rates in our experimental data validates the reasoning of co-channel interference and/ or signal attenuation behind the low throughput.

From the close range (i.e., 2 m distance) experimental data we can see that, even though the client node is positioned very close to the AP and having a clear LOS to the AP, the achieved throughput is far lower than the satisfactory level and that shows TCP's inability to perform well in opportunistic networks.

5.4 Summary

In this chapter, we have analyzed the whole protocol stack performance. First, in section 5.1 we analyzed the link layer connection establishment process followed by layer-3 IP address acquisition. In this section we showed that the time it takes to establish the link layer connection and acquire an IP address is in the order of few seconds. This delay is dominated by link layer joining time followed by the network scanning time. Second, in section 5.2 we looked at the service discovery delay for an IP multicast based service discovery mechanism. We found that this phase took about a second, due to round trips needed by the service discovery mechanism. In an optimal service discovery mechanism it could be in the order of round trip time (10s of ms). Finally, we looked at the data transmission over TCP in section 5.3. As expected, we found that TCP performs poorly as a transport layer protocol in wireless networks, especially as the link quality drops as a function of the distance from the access point. This implies that other transport layer protocols should be evaluated for opportunistic networks. In next chapter, we conclude our work and discuss the possible future work.

6 Conclusions and Future Work

In this thesis, we characterized the opportunistic contacts in infrastructure-assisted Wi-Fi networks. We first carefully designed the testbed in order to have profound insights on the experiments that we carried out to achieve the goals of the thesis. We conducted a site survey at the indoor wireless testbed area to have an understanding on the the characteristics of the indoor RF wave propagation characteristics. The survey helped us in evaluating the reasoning behind certain characteristics of the opportunistic contacts. Then we measured the delays that a station goes through during the different phases of a contact, namely, link layer connection establishment, layer-3 IP address acquisition and service discovery etc. While characterizing the opportunistic contacts, we also investigate the throughput performance of data transmission between two stations when TCP is used as the transport layer protocol.

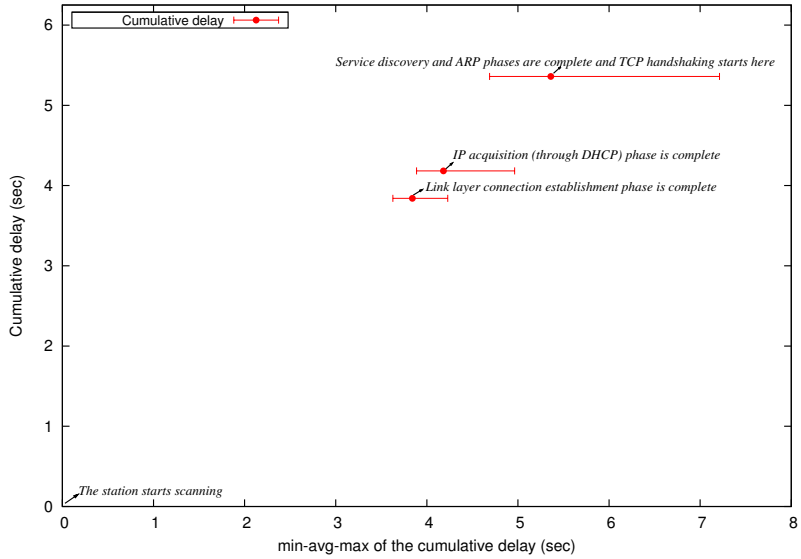


Figure 57: Minimum, average and maximum of cumulative delay (phases included: connection establishment, IP address acquisition, service discovery, ARP) when the distance between the station and the AP is 2 m.

During the indoor wireless site survey, we observed that the link quality did not depend only on the distance between the wireless devices but also on the non-distance based channel propagation effects (e.g., multipath propagation, path loss/ attenuation, fading, interference etc.). We also noticed that small changes in position and orientation of the wireless devices could impact the RF behaviour though there was clear LOS between the devices. We could spot certain location where number of connection failures was very high (e.g., 7 out of 10 connection attempts failed), but the packet loss rate was not high once the station could establish the connection with the access point. We found the adaptive modulations scheme of 802.11b very handy in handling such situation as this scheme makes the system to sacrifice performance over connectivity and choose lower data rates to maintain connectivity.

We carried out our delay measurement (i.e., delays during the different phases

of opportunistic contacts) related experiments keeping the station in two different locations (i.e., 2 m and 16 m apart from the AP) in order to study the impact of indoor RF wave propagation characteristics on the measured delays. In Figure 57, we can see the cumulative delay a station (position: 2 m apart from the AP) experienced over the different phases (i.e., connection establishment phase, IP acquisition phase, service discovery phase and ARP phase) during the contacts.

In the Figure, we can see that it took on an average 5.36 sec for the station before it could start transferring data using TCP. Contact duration directly influences the capacity of opportunistic networks because it limits the amount of data that can be transferred between the stations. In the context of short contact duration in opportunistic contacts, it can be said from Figure 57 that, link layer connection establishment delay (3.8 sec) contributed the most on the overall delay. From our experimental results, we noticed that the scanning delay (i.e., 575 ms) contributed 15% of the total connection set-up time. But the most dominant component was the delay during the joining phase (i.e., 3.2 sec; including the delay to have the successful joining indication). This joining delay contributed 84% of the total connection set-up time, whereas the contribution of the Authentication-Association phase was only 0.45%. The station gets synchronized with the access point during the joining phase and that could be the key reason behind the higher delay during this phase. Another major contributor during the connection establishment phase was scanning delay. In our experiments, the station took on an average 572 ms to scan some other channels, though to discover the desired network on channel 6 (i.e., the the operating channel of the desired access point) it took only 3 ms. This result suggests that total scanning delay can be reduced substantially by reducing the number of scanned channels to a subset where APs are known to exist.

In Figure 57, we can see that, after 3.8 sec of connection establishment phase, the IP address acquisition phase starts and it lasts till 4.2 sec. The delay during the IP address acquisition phase (total DHCP time 84 ms; including the gaps before and after DHCP procedure the delay was around 342 ms) was insignificant compared to the delay during the connection establishment phase. The next significant delay the station experienced was during the service discovery phase. Excluding the gaps the naming and service browsing/resolution steps during this phase contributed 993 ms to the overall delay. Due to traffic reduction mechanism of DNS-SD, the service browsing/resolution step took less time, i.e., 293 ms, whereas the name reservation step took 700 ms. Having ended with the service discovery phase the station took only 6 ms for the ARP phase. After 0.07 ms of the ARP phase it started the TCP handshaking in order to transfer data over TCP.

When compared with the 2 m experimental data, the results (i.e., data sets of the delays) obtained from the 16 m experiments do not have a considerable difference in performance in terms of the measured delays, even though from the wireless testbed site survey experiments we noticed that, the indoor wireless signal characteristics could be very dynamic in nature and vary from location to location. During the site survey, another observation we had that, the characteristics of the wireless signal on a particular location could get changed over time. Our understanding is it would be better, if we could conduct more sets of experiments changing the location of the

station. It would have provide us more in depth understanding on the impact of indoor RF wave propagation characteristics on the delays. But due to nature of the experiments (i.e., very time consuming) as well as due to the scope of the Master's thesis work we did not conduct further experiments.

The experimental results (i.e., throughput) of the data transmission phase over TCP showed clearly that the performance of TCP as a transport layer protocol in wireless networks is not satisfactory. The throughput for the 2 m, 8 m, 16 m and 32 m distances was 56.709 KBps, 24.934 KBps, 3.674 KBps and 531 Bps respectively. Analyzing the throughput we can see that, the throughput was greatly affected by distance. In our experiments, at the close range (i.e., 2 m apart from the AP) the station (i.e., the client node) could send 100 MB data to the server node but the throughput (i.e., 56.709 KBps only) achieved in that position was not satisfactory considering the maximum capacity (i.e., 11 Mbps) of 802.11b standard. During the close range experiment, we also observed that initially the throughput was around 400 KBps but it drops sharply after couple of minutes of data transfer. The reason of low throughput and sharp drop in throughput could be the station's sharing of channel capacity with the neighbouring wireless devices. Another reason could be the indoor RF wave propagation characteristics (e.g., path loss/attenuation due to multipath propagation) that might have contributed in the degradation of link quality. Both of these phenomena (i.e., co-channel interference, path loss/attenuation) could have caused packet losses over the wireless link. The drawback of of TCP as transport layer protocol in wireless networks is that it interprets such losses as congestion and invokes congestion control mechanism, whereas the losses were not due to congestion. Thus TCP created significant drop in throughput. The experimental data from the throughput experiments helped us to conclude that TCP is not a good choice for the opportunistic networks because of its lack of efficiency in handling the packet loss over wireless links.

The experiments, that we conducted in order to characterize the opportunistic contacts, have provided us in depth understanding on the issues that can limit the amount of data transfer during the short connection window of an opportunistic contact. One possible direction of future work is to characterize opportunistic contacts in different environments (e.g., RF shielded interference free room, outdoor etc.) with more varied scenarios (e.g., non-line-of-sight, mobility etc.). We also plan to use recent variants of 802.11 (e.g., 802.11a/g/n) in the experiments. Another possible direction of future work is to characterize the opportunistic contacts using a real world opportunistic networking deployment (e.g., Liberouter) so as to compare the real world system with the one we used in this thesis. It would also be interesting to work on the mechanisms that will reduce the scanning and joining latencies during the 802.11 connection establishment phase.

References

[1] Android Apps on Google Play. https://play.google.com/store/apps/details?id=cz.webprovider.wifianaly%zer. [Accessed on: 25 April, 2015].

[2] IEEE 802.11b-1999. http://en.wikipedia.org/wiki/IEEE_802.11b-1999. [Accessed on: 18 April, 2015].

[3] Indoor wireless path loss – security today. http://security-today.com/articles/2012/04/01/indoor-wireless-path-loss.aspx. Accessed: 28.01.2015.

[4] Physical radio channel models | Wireless & Cable. http://www.wica.intec.ugent.be/research/propagation/physical-radio-channel-models. [Accessed on: 02 May, 2015].

[5] TCPDUMP/LIBPCAP public repository. http://www.tcpdump.org/. [Accessed on: 25 April, 2015].

[6] tcptrace-Official Hompage. http://www.tcptrace.org/index.html. [Accessed on: 11 May, 2015].

[7] A tutorial on indoor radio propagation. http://www.sss-mag.com/indoor.html. Accessed: 18.01.2015.

[8] The wireless channel: Propagation and fading. http://media.johnwiley.com.au/product_data/excerpt/18/04708256/0470825618.pdf. Accessed: 28.01.2015.

[9] U. S. Robotics . Wireless LAN Networking. http://support.usr.com/download/whitepapers/wireless-wp.pdf. [Accessed on: 18 April, 2015].

[10] Adrio Communications Ltd. IEEE 802.11 Standards | WiFi Specifications | Radio-Electronics.com. http://www.radio-electronics.com/info/wireless/wi-fi/ieee-802-11-standards-tutorial.php. [Accessed on: 26 March, 2015].

[11] Adrio Communications Ltd. WiFi Channels | WiFi Frequency Bands list | Radio-Electronics.com. http://www.radio-electronics.com/info/wireless/wi-fi/80211-channels-number-frequencies-bandwidth.php. [Accessed on: 26 March, 2015].

[12] Alcatel-Lucent. Exploring Traffic Geo-Location as a Key Strategy to Off-load Macro Networks with Small Cells Solutions. http://www.tmcnet.com/tmc/whitepapers/documents/whitepapers/2013/6680-exploring-traffic-geo-location-as-key-strategy-off.pdf. [Accessed on: 28 September, 2014].

[13] Andrew Burger. Report: Carriers Deployed More Than 7 Million Wi-Fi Access Points. http://www.telecompetitor.com/report-carriers-deployed-more-than-7-million-wi-fi-access-points/. [Accessed on: 02 May, 2015].

[14] Apple Inc. Bonjour Operations. https://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/NetServices/Articles/NetServicesArchitecture.html. [Accessed on: 03 May, 2015].

[15] Apple Inc. Dynamic Address Assignment. https://developer.apple.com/library/ios/documentation/NetworkingInternet/Conceptual/NetworkingConcepts/NetworkingBasics/NetworkingBasics.html. [Accessed on: 10 April, 2015].

[16] Huseyin Arslan, Zhi Ning Chen, and Maria-Gabriella Di Benedetto. *Ultra wideband wireless communication*. John Wiley & Sons, 2006.

[17] Hossein Bidgoli. *Handbook of Information Security, Threats, Vulnerabilities, Prevention, Detection, and Management*, volume 3. John Wiley & Sons, 2006.

[18] Scott Burleigh, Adrian Hooke, Leigh Torgerson, Kevin Fall, Vint Cerf, Bob Durst, Keith Scott, and Howard Weiss. Delay-Tolerant Networking: An Approach to Interplanetary Internet. *Communications Magazine, IEEE*, 41(6):128–136, 2003.

[19] B. Cain, S. Deering, I. Kouvelas, B. Fenner, and A. Thyagarajan. RFC 3376: Internet Group Management Protocol, Version 3. 2002.

[20] Augustin Chaintreau, Pan Hui, Jon Crowcroft, Christophe Diot, Richard Gass, and James Scott. Pocket switched networks: Real-world mobility and its consequences for opportunistic forwarding. Technical report, University of Cambridge, 2005.

[21] Xiang Chen, Hongqiang Zhai, Jianfeng Wang, and Yuguang Fang. TCP performance over mobile ad hoc networks. *Electrical and Computer Engineering, Canadian Journal of*, 29(1/2):129–134, 2004.

[22] S. Cheshire, B. Aboba, and E. Guttman. RFC 3927: Dynamic configuration of IPv4 link-local addresses. *IETF standard*, 2005.

[23] Stuart Cheshire and Daniel H. Steinberg. *Zero Configuration Networking: The Definitive Guide*. O'Reilly Media, 2006.

[24] Dah-Ming Chiu and Raj Jain. Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. *Computer Networks and ISDN systems*, 17(1):1–14, 1989.

[25] Cisco Systems, Inc. Antenna Patterns and Their Meaning - Cisco. http://www.cisco.com/c/en/us/products/collateral/wireless/aironet-antennas-accessories/prod_white_paper0900aecd806a1a3e.html. [Accessed on: 04 May, 2015].

[26] Cisco Systems, Inc., Americas Headquarters, San Jose, CA. *IP Addressing Guide*, 2010.

[27] Moonblink Communications. 802.11b WiFi Frequency Channels. http://www.moonblink.com/store/2point4freq.cfm. [Accessed on: 26 March, 2015].

[28] Dr. Ranjan Bose. Wireless communications. http://textofvideo.nptel.iitm.ac.in/117102062/lec9.pdf. [Accessed on: 02 May, 2015].

[29] Ralph Droms. RFC 2131: Dynamic host configuration protocol. 1997.

[30] Electronic Communications Committee (ECC) within the European Conference of Postal and Telecommunications Administrations (CEPT). ercrep025. http://www.erodocdb.dk/Docs/doc98/official/pdf/ercrep025.pdf. [Accessed on: 17 April, 2015].

[31] ETHW. Wireless LAN 802.11 Wi-Fi - Engineering and Technology History Wiki. http://ethw.org/Wireless_LAN_802.11_Wi-Fi. [Accessed on: 12 May, 2015].

[32] M. Gast. *802.11 Wireless Networks: The Definitive Guide*. O'Reilly Media, 2005.

[33] Vaibhav Gupta, Raheem Beyah, and Cherita Corbett. A Characterization of Wireless NIC Active Scanning Algorithms. In *Wireless Communications and Networking Conference, 2007. WCNC 2007. IEEE*, pages 2385–2390. IEEE, 2007.

[34] E. Hamadani and V. Rakocevic. Evaluating and improving TCP performance against contention losses in multihop Ad Hoc Networks. In *IFIP International Conference (MWCN), Marrakech, Morocco*, pages 923–934, 2005.

[35] Hewlett Packard. Wireless Overview - The radio modem. http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Linux.Wireless.modem.html. [Accessed on: 04 May, 2015].

[36] IEEE. IEEE 802.11, The Working Group Setting the Standards for Wireless LANs . http://www.ieee802.org/11/. [Accessed on: 15 April, 2015].

[37] IEEE Computer Society. 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. 2007.

[38] James F. Carter. Nokia N810 and 770 Internet Tablets: Hardware. http://www.math.ucla.edu/~jimc/hardware/nokia770/physical.shtml#cpu. [Accessed on: 23 April, 2015].

[39] Wang Jian-qiang and Wu Chen-wen. A Novel Opportunistic Routing Protocol Applied to Vehicular Ad hoc Networks. In *2010 5th International Conference on Computer Science and Education (ICCSE)*, pages 1005–1009. IEEE, 2010.

[40] Jim Geier. How to: Assign 802.11b/g Access Point Channels. http://www.wireless-nets.com/resources/tutorials/assign_ap_channels.html. [Accessed on: 18 April, 2015].

[41] Jim Geier. How to: Get the most from 802.11 multicasting. http://www.wireless-nets.com/resources/tutorials/802.11_multicasting.html. [Accessed on: 25 April, 2015].

[42] Jim Geier. Understanding 802.11 Frame Types. http://www.wi-fiplanet.com/tutorials/article.php/1447501. [Accessed on: 18 April, 2015].

[43] Jouni Malinen. hostapd: IEEE 802.11 AP, IEEE 802.1X/WPA/WPA2/EAP/RADIUS Authenticator. http://w1.fi/hostapd/. [Accessed on: 24 April, 2015].

[44] Philo Juang, Hidekazu Oki, Yong Wang, Margaret Martonosi, Li Shiuan Peh, and Daniel Rubenstein. Energy-efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet. *SIGARCH Comput. Archit. News*, 30(5):96–107, 2002.

[45] Charles M. Kozierok. The TCP/IP Guide - DHCP Message Generation, Addressing, Transport and Retransmission . http://www.tcpipguide.com/free/t_DHCPMessageGenerationAddressingTransportandRetrans-3.htm. [Accessed on: 21 April, 2015].

[46] Charles M. Kozierok. The TCP/IP Guide - IP Datagram Encapsulation. http://www.tcpipguide.com/free/t_IPDatagramEncapsulation.htm. [Accessed on: 10 April, 2015].

[47] Charles M. Kozierok. The TCP/IP Guide - Network Layer (Layer 3). http://www.tcpipguide.com/free/t_NetworkLayerLayer3.htm. [Accessed on: 18 April, 2015].

[48] M. Krochmal and S. Cheshire. RFC 6762: Multicast DNS. 2013.

[49] M. Krochmal and S. Cheshire. RFC 6763: DNS-Based Service Discovery. 2013.

[50] David M Lambeth. Design Considerations for an Indoor Location Service Using 802.11 Wireless Signal Strength. Master's thesis, Massachusetts Institude of Technology, 2009.

[51] Ka-Cheong Leung and Victor O. K. Li. Transmission Control Protocol (TCP) in wireless networks: issues, approaches, and challenges. *Communications Surveys & Tutorials, IEEE*, 8(4):64–79, 2006.

[52] K. Lynn, S. Cheshire, M. Blanchet, and D. Migault. Requirements for Scalable DNS-SD/mDNS Extensions. 2015.

[53] Mario Marchese. Interplanetary and pervasive communications. *Aerospace and Electronic Systems Magazine, IEEE*, 26(2):12–18, 2011.

[54] David Murray, Terry Koziniec, and Michael Dixon. An analysis of handoff in multi-band 802.11 networks. In *Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE Internatonal Conference on*, pages 1–10. IEEE, 2007.

[55] T. Narten, T. Jinmei, and S. Thomson. RFC 4862: IPv6 Stateless Address Autoconfiguration. 2007.

[56] Nokia Siemens Network. Wi-Fi integration with Cellular Networks enhances the customer experience (White Paper). [Accessed on: 28 September, 2014].

[57] Konstantina Papagiannaki, Mark D Yarvis, and W Steven Conner. Experimental characterization of home wireless networks and design implications. In *INFOCOM*, 2006.

[58] Alex Pentland, Richard Fletcher, and Amir Hasson. Daknet: Rethinking connectivity in developing nations. *Computer*, 37(1):78–83, 2004.

[59] Anna Kaisa Pietiläinen and Christophe Diot. Experimenting with Opportunistic Networking. In *Proc. of the ACM MobiArch Workshop*, 2009.

[60] Mikko Pitkanen, Teemu Karkkainen, and Jörg Ott. Mobility and Service Discovery in Opportunistic Networks. In *2012 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pages 204–210. IEEE, 2012.

[61] D.C. Plummer. RFC 826: An Ethernet Address Resolution Protocol –or– Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware. 1982.

[62] Tobias Pögel. Optimized DTN-routing for urban public transport systems. In *OASIcs-OpenAccess Series in Informatics*, volume 17. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2011.

[63] J. Postel. RFC 793: Transmission Control Protocol. 1981.

[64] Postel, J. RFC 768: User Datagram Protocol. 1980.

[65] Md Abdur Rahim. *Interference Mitigation Techniques to Support Coexistence of Ultra-WideBand Systems*. Jörg Vogt Verlag, 2010.

[66] Theodore Rappaport. *Wireless Communications: Principles and Practice.* Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 2001.

[67] Raspberry Pi Foundation. Raspberry Pi. `https://www.raspberrypi.org/`. [Accessed on: 24 April, 2015].

[68] Raspberry Pi Foundation. Raspberry Pi Learning Resources. `https://www.raspberrypi.org/learning/networking-lessons/lesson-3/README.md`. [Accessed on: 24 April, 2015].

[69] Raspbian Community. Raspbian. `http://www.raspbian.org/`. [Accessed on: 24 April, 2015].

[70] Ahmed Riadh Rebai and Saïd Hanafi. An Adaptive Multimedia-Oriented Handoff Scheme for IEEE 802.11 WLANs. *International Journal of Wireless & Mobile Networks (IJWMN)*, 3(1), 2011.

[71] Riverbed Technology, 199 Fremont Street, San Francisco, CA 94105. *AirPcap User's Guide*, 2013.

[72] Rohde&Schwarz. R&S®CMW-Z10 RF Shield Box - Overview - Rohde&Schwarz International. `http://www.rohde-schwarz.com/en/product/cmwz10-productstartpage_63493-10816.html`. [Accessed on: 24 April, 2015].

[73] Rohde&Schwarz. R&S®RSC Step Attenuator - Overview - Rohde&Schwarz International. `http://www.rohde-schwarz.com/en/product/rsc-productstartpage_63493-11395.html`. [Accessed on: 24 April, 2015].

[74] Sandra Sendra, Miguel Garcia, Carlos Turro, and Jaime Lloret. WLAN IEEE 802.11 a/b/g/n Indoor Coverage and Interference Performance Study. *International Journal on Advances in Networks and Services*, 4(1 and 2):209–222, 2011.

[75] Suranga Seneviratne, Aruna Seneviratne, Prasant Mohapatra, and Pierre-Ugo Tournoux. Characterizing WiFi Connection and Its Impact on Mobile Users: Practical Insights. In *Proceedings of the 8th ACM international workshop on Wireless network testbeds, experimental evaluation & characterization*, pages 81–88. ACM, 2013.

[76] John S Seybold. *Introduction to RF propagation.* John Wiley & Sons, 2005.

[77] Sangho Shin, Andrea G Forte, and Henning Schulzrinne. Seamless Layer-2 Handoff using Two Radios in IEEE 802.11 Wireless Networks. 2006.

[78] Tara Small and Zygmunt J. Haas. The Shared Wireless Infostation Model: A New Ad Hoc Networking Paradigm (or Where There is a Whale, There is a Way). In *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing*, MobiHoc '03, pages 233–244. ACM, 2003.

[79] William Stallings. *Wireless communications & networks*. Pearson Education India, 2009.

[80] Riverbed Technology. SteelCentral Network Performance Management | Riverbed AirPcap | Riverbed. `http://www.riverbed.com/products/performance-management-control/networ%k-performance-management/wireless-packet-capture.html`. [Accessed on: 25 April, 2015].

[81] The Avahi team. ArchitecturalOverview - Avahi. `http://www.avahi.org/wiki/ArchitecturalOverview`. [Accessed on: 27 April, 2015].

[82] The Avahi team. Avahi. `http://www.avahi.org`. [Accessed on: 03 May, 2015].

[83] The IETF Zeroconf Working Group. Zero configuration networking. `http://www.zeroconf.org`. [Accessed on: 03 May, 2015].

[84] Ye Tian, Kai Xu, and Nirwan Ansari. Tcp in wireless environments: problems and solutions. *Communications Magazine, IEEE*, 43(3):S27–S32, 2005.

[85] Sacha Trifunovic, Bernhard Distl, Dominik Schatzmann, and Franck Legendre. WiFi-Opp: Ad-Hoc-less Opportunistic Networking. In *Proceedings of the 6th ACM workshop on Challenged networks*, pages 37–42. ACM, 2011.

[86] Md Yusuf S. Uddin, David M. Nicol, Tarek F. Abdelzaher, and Robin H. Kravets. A Post-disaster Mobility Model for Delay Tolerant Networking. In *Winter Simulation Conference*, WSC '09, pages 2785–2796. Winter Simulation Conference, 2009.

[87] Wireshark Foundation. Wireshark - About. `https://www.wireshark.org/about.html`. [Accessed on: 25 April, 2015].

[88] Lunan Zhao, Fan Li, Chao Zhang, and Yu Wang. Routing with multi-level social groups in mobile opportunistic networks. In *Global Communications Conference (GLOBECOM), 2012 IEEE*, pages 5290–5295. IEEE, 2012.