Politecnico di Milano

Faculty of Information Engineering
Master of Science in Telecommunications Engineering
Electronics, Information and Bioengineering Department

Opportunistic Wi-Fi Direct networking with channel state based routing algorithm

Supervisor: Prof. Antonio CAPONE
Advisor: Prof. Vincenzo MANCUSO

Graduation Thesis of:
Giulia RESMINI
Student ID 803566

Academic Year 2014-2015

Acknowledgments

This thesis documents the work done in IMDEA Networks, Leganes (Madrid). I would like to thank my supervisor Antonio Capone, who allows me to have a new working experience abroad. In addition, I would like to thank my advisor Vincenzo Mancuso for his kind help and guide during all my permanence in IMDEA Networks. A particular thank goes also to Arash Asadi from IMDEA Networks, who gave me support and explanations in this project. Without them this would not have been possible.

Lastly, I would like to thank my family, friends and boyfriend for their emotional support during the master course and more in particular during the work of thesis.

Giulia

Contents

List of Figures

List of Tables

Abstract

Today, most of the people in the world use Wi-Fi to stay connected. This technology is widely available and it is used in many types of devices.
Introducing Wi-Fi Direct technology, users are able to exchange data through efficient device-to-device (D2D) communications. Wi-Fi P2P technology builds on traditional Wi-Fi strengths like performance, security, ease of use and ubiquity, and adds features that optimize it for consumer uses that do not require access to an infrastructure network.
This thesis is part of a wider project in collaboration between Politecnico of Milan and IMDEA Networks of Madrid. The aim of the work is to study the potentiality of Wi-Fi Direct technology: exploiting all the new features introduced by Wi-Fi Direct technology, we reproduce P2P communications through an experimental testbed composed by four Wi-Fi Direct enabled devices. First we enable connections among clusters of P2P groups; moreover we connect one of the Wi-Fi Direct device to the external network for the sake of exchanging packets with Internet. This topology is used to reproduce a real scenario with an opportunistic relay system, in which users can communicate with an external network even if they are not directly connected to it.
We also present a new model to improve packets exchange in an opportunistic network: the combination of this new topology with a specific innovative routing algorithm let people be free to move and exchange data without loosing connectivity.

Abstract (Italiano)

Al giorno d'oggi, la maggior parte delle persone nel mondo utilizza il Wi-Fi per essere sempre connessa. Questa tecnologia è molto diffusa e utilizzata nella gran parte dei dispositivi.
Con l'avvento della tecnologia Wi-Fi Direct, gli utenti hanno la possibilità di scambiarsi dati tramite comunicazioni D2D (Device-to-Device). La tecnologia P2P (Peer-to-Peer) da la possibilità di costruire, sulla tradizionale Wi-Fi, migliori performance, sicurezza, facilità di utilizzo e ubiquità, aggiungendo caratteristiche che ottimizzano la tecnologia, permettendo agli utenti di non avere necessità di una infrastruttura di rete fissa.
Questa tesi è frutto di una collaborazione tra il Politecnico di Milano e l'istituto di ricerca IMDEA Networks di Madrid atto a valutare le potenzialità del WI-FI Direct: sfruttando le nuove funzionalità introdotte dal Wi-Fi Direct, abbiamo riprodotto comunicazioni P2P per mezzo di un testbed sperimentale composto da quattro Wi-Fi Direct devices(Alix Boards). Per prima cosa si abilitano le comunicazioni tra cluster di gruppi P2P; in seguito viene poi connesso un dispositivo alla rete esterna, in modo tale da poter scambiare pacchetti e dati con Internet. La topologia appena citata è usata per riprodurre uno scenario reale con un sistema opportunistico di relay dei pacchetti, per mezzo del quale gli utenti possono comunicare tra loro o con la rete esterna anche se non direttamente connessi.
Viene inoltre presentato un nuovo modello per migliorare lo scambio di pacchetti tramite l'utilizzo di un algoritmo innovativo di routing che permette agli utenti di muoversi liberamente scambiando dati senza mai perdere connettività.

Acronyms and Definitions

Acronyms

AP: Access Point

CCMP: Counter Cipher Mode with block chaining message authentication code Protocol

DHCP: Dynamic Host Configuration Protocol

D2D: Device to Device

IEEE: Institute of Electrical and Electronics Engineers

GO: P2P Group Owner

MAC: Media Access Control

NoA: Notice of Absence

P2P: Peer-to-Peer

PBC: Push Button Configuration

STA: Non-AP Station

TKIP: Temporal Key Integrity Protocol

UUID: Universally unique identifier

WLAN: Wireless Local Area Network

WPA: Wi-Fi Protected Access

WPS: Wi-Fi Protected Setup

Definitions

Client: a P2P Client or a Legacy Client that is connected to a P2P Group Owner.

Legacy Client: a STA that is Wi-Fi CERTIFIED, but not P2P compliant.

Listen Channel: the channel chosen from the set of Social Channels, which is used by a P2P Device to be discoverable.

P2P Client: a P2P Device that is connected to a P2P Group Owner.

P2P Device: WFA P2P certified device that is capable of acting as both a P2P Group Owner and a P2P Client.

P2P Discovery: a capability that provides a set of functions to allow a device to easily and quickly identify and connect to a device and its services in its vicinity.

P2P Group: a set of devices consisting of one P2P Group Owner and zero or more Clients.

P2P Group Owner: an "AP-like" entity that may provide and use connectivity between Clients.

P2P Interface Address: the MAC address of the P2P interface, an address used to identify a P2P Device within a P2P Group.

Persistent P2P Group: a P2P Group for which credentials are stored and may be made available for reuse after the initial use completes. Such a P2P Group has a lifetime that may extend over a number of distinct sessions beyond the initial use until the group is deliberately dissolved.

Scan Phase: the process in P2P Discovery to collect information about surrounding devices or networks by scanning all supported channels.

Social Channels: a subset of commonly available channels in the 2.4 GHz band (1, 6 and 11 channels).

Temporary P2P Group: a P2P Groups that is formed only when required and ceases to exit after the initial use competes. Such a P2P Group has a lifetime consisting of a single use.

Topology: the arrangement in which the nodes of a network are connected to each other and (in some cases) to other networks.

Chapter 1

Introduction

The massive and growing use of mobile devices in the last decade has challenged previous technologies for information sharing. Nowadays people need to connect devices for the sake of storing, sharing and sending information all the time and everywhere. Industry has faced these challenges as developing smart solutions to fulfil these needs. One of the most successful technologies to cover information sharing is Wi-Fi Direct. It is a new technology derived from Wi-Fi, based on P2P architecture and created to make connections simpler and more convenient for users compared to standard Wi-Fi.

The main concept of Wi-Fi Direct is the creation of "direct" connections among Wi-Fi Direct devices without requiring the presence of a traditional Wi-Fi infrastructure network. Instead of having a fixed AP, as the traditional Wi-Fi architecture, the AP-like functionality is implemented dynamically on a P2P device. The communication among P2P enabled devices is implemented by creating "P2P groups" composed by two or more devices.

The main idea of Wi-Fi Direct is to create groups composed by multiple devices; differently from ad-hoc networks, a single device must be capable of two different roles:

- P2P Group Owner (GO): it has an "AP-like" capability that controls a Wi-Fi P2P group and enables P2P devices connectivity

- P2P Client: it is a Wi-Fi P2P-compliant device that may connect to a P2P Group Owner

A simple topology composed by three devices is proposed in Figure 1.1 .

There are several ways in which two devices can establish a P2P Group, de-

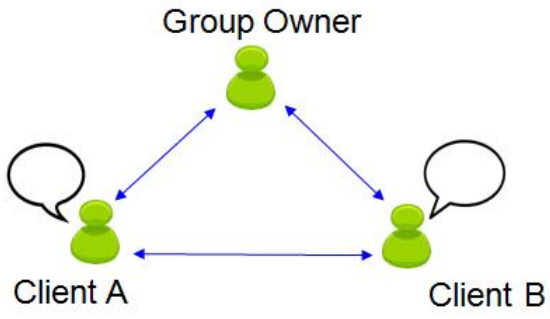

Figure 1.1: *P2P group.*

pending on, e.g., whether they have to negotiate the role of P2P GO, or if there is some pre-shared security information available. From these possible configurations, three methods are defined to create P2P groups: Standard, Autonomous and Persistent[8].

In this work we mainly utilize persistent group formation to establish groups of P2P devices: this method helps us to use a specific group more than once after its previous creation, thanks to the automatic updated specifications of the group in the configuration file of the device.

This technology considers also the creation of multiple groups, allowing the peers to be in common for more than one group simultaneously.

All the features and methods of Wi-Fi Direct group formation are studied and analyzed. Then, all the mechanisms are explored in a real scenario, taking into account that every device can act either as client or GO at the same time. One device can be both client and GO of two different groups. In contrast, it is not possible that a device acts as GO for more than one group. Every interface on a single device can have a large number of virtual interfaces but only one of them

can cover the role of GO in one group, whereas the others act as clients.
Once the groups are created and Wi-Fi Direct communications among devices are implemented, it is possible to create different network typologies in order to design several scenarios. More in particular, this work is focused on a typical scenario that could be a source for future studies. An example is shown in Figure 1.2.
The work of connections enabling is elaborated by Linux OS. More in particular,

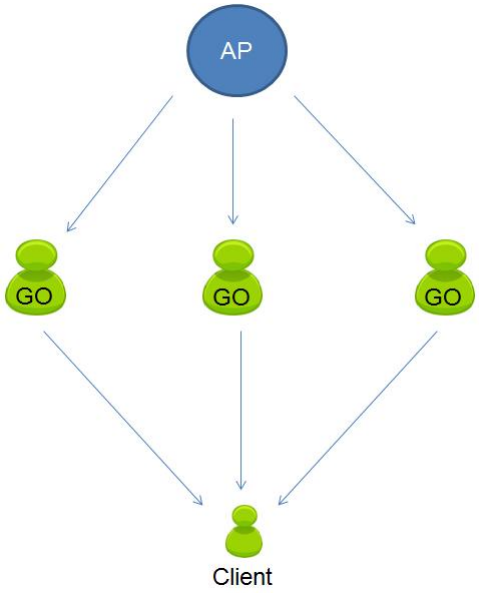

Figure 1.2: *Example of an implemented scenario.*

"wpa supplicant" and "wpa cli" modules of Linux are used to manage features and commands utilized for setting devices connections. The work is conducted on PC-Engine Alix system Board: this type of machines is used as P2P device to establish Wi-Fi Direct communications among users.
After the analysis of theoretical mechanisms, we focus on the evaluation of practical performances of Wi-Fi Direct. More in particular, we test connections between devices according to theoretical definitions. Furthermore, a specific and opportunistic routing algorithm is formulated and presented to offer mobility to users without loosing the possibility to communicate exchanging packets.

1.1 Outlines

This thesis is structured in six sections.

The first one contains a brief explanation about all the concepts analyzed in the thesis.

The second section of the work is dedicated to the related works on Wi-Fi Direct and the thesis objectives.

In the third chapter it is given a complete picture of the whole topic, explaining in particular how Wi-Fi Direct is structured and how it works.

After background chapters, all the physical and virtual tools are presented to apply theoretical concepts to real appliances.

The fifth chapter presents the thesis work, with performance evaluations and final results.

The last chapter concludes the thesis, discussing briefly the aim of the project and the main results.

An appendix section follows the last chapter to present some important method used and tested during the work.

Chapter 2

Related Work

The past decade has witnessed the prosperity of online social networks and pervasive computing. The market of mobile devices, including smartphones and other portable wireless devices, has also been growing rapidly. More and more new social networking applications are developed for mobile platform and exploit proximity-based interaction [6].

There are several obvious disadvantages with the centralized service model. On one hand, it risks users privacy and some users are reluctant to provide their location information to the server. On the other hand, the central server can be the point-of-failure or performance bottleneck. If the server is out of service or overloaded, users cannot get the results even if their friends are actually in vicinity. In fact, most of the location-based services only need local information and the help from a powerful central server is unnecessary if direct user-to-user communication is available.

After a tremendous success whereby Wi-Fi has become a predominant way to access the Internet wirelessly, it is now embracing the challenge of becoming pervasive also in direct D2D communications. In this respect, the Wi-Fi Alliance has recently developed the Wi-Fi Direct technology that builds upon the Wi-Fi infrastructure mode to enable direct device to device connectivity [8].

Wi-Fi CERTIFIED Wi-Fi Direct provided by Wi-Fi alliance allows Wi-Fi devices connect to each other directly in a new convenient way without the need of a

wireless access point (AP) [8]. In traditional wireless networks, all wireless devices connect to the AP or wireless router to communicate with multiple peers. In contrast, devices with Wi-Fi Direct capability can form a flexible and secure temporary network to communicate without an AP.

Wi-Fi CERTIFIED Wi-Fi DirectTM devices can connect in a way that makes it simpler and more convenient than ever for users to print, share, sync and display. Wi-Fi Direct devices can connect directly to one another without access to a traditional network, so mobile phones, cameras, printers, PCs, and gaming devices can connect to each other directly to transfer content and share applications anytime and anywhere. Devices can make a one-to-one connection, or a group of several devices can connect simultaneously. They can connect for a single exchange, or they can retain the memory of the connection and link together each time they are in proximity.

The introduction of the Wi-Fi Alliance Peer-to-Peer Specification and certification testing program dramatically expands peer-to-peer connectivity for consumers by introducing an interoperable technology with distinctive new features. In a recent study, US consumers revealed that they would most want to use direct connections for instant messaging, sharing pictures with friends and family, displaying those pictures from a portable device to a monitor or TV screen, video chatting, and playing video games with others while not at home, such as when using public transit [14].

Another useful paradigm is to share the cellular connection of a mobile phone as shown in Figure 2.1 [6]. In the first scenario on the left, devices are directly connected to a PC: they also have the possibility to exchange data with one or more devices belonging to the same "group". In the second scenario, a mobile phone is sharing cellular connection with other devices, such as computers and tablets.

Wi-Fi Direct is a novel technology that introduces new opportunities to deploy

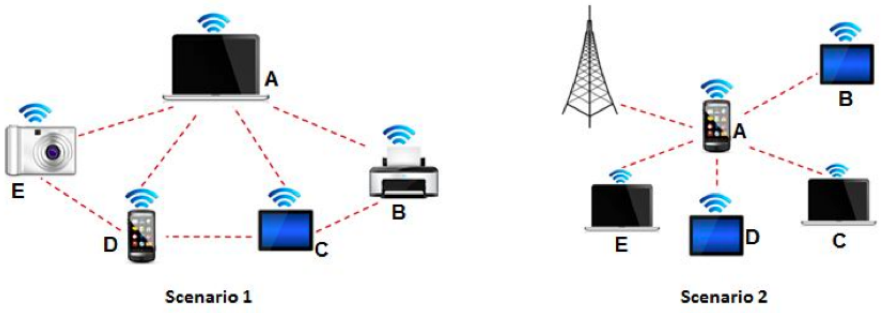

Figure 2.1: *Typical Scenarios of Wi-Fi Direct Enabled D2D Networks.*

real opportunistic networks through mobile devices. It overtakes all the limitations that a initial Wi-Fi architecture had, by supporting P2P and AP communications. This is a great opportunity to let users exchange contents taking advantage of the human mobility.

D2D communication in cellular networks is defined as direct communication between two mobile users without traversing the Base Station (BS) or core network [4]. Several studies and experiments were performed in order to analyze the capabilities of Wi-Fi Direct technology.

Many studies on general Wi-Fi features were performed: more in particular, experimental evaluations ware realized, quantifying group formation delays to be expected in real-life scenarios [6]. In addition to this analyses, also the time required to form a P2P group was widely studied in literature: all the measurements were performed in all the three cases described in the theory section (Standard, Autonomous, Persistent). Moreover this kind of study wants to quantify the achievable performance trade-offs using this new technology, and comparing them against legacy operation [8].

A really interesting topic that is analyzed in many studies concerns measurements and analyses of energy consumption of today's device-to-device communication technologies: Wi-Fi Direct, Bluetooth and WLAN-Opp (a solution based on the WLAN access point mode). Energy-efficient operation is a key prerequisite for

user acceptance of opportunistic D2D communication. About peer discovery, it was found that Bluetooth consumes less than half the energy of WLAN-Opp, which in turn consumes only half of Wi-Fi Direct [13]. Further, each technology is potentially unfair as the different roles of the devices required to maintain a connection, such as being a master versus being a slave, show different energy consumption footprints [13].

Energy consumption and power saving are two of the main features studied in P2P communications. An important study was made on two power saving protocols defined in Wi-Fi Direct allowing APs to save power, Opportunistic Power Save (OPS) and Notice of Ab- sence (NoA) [9]. Moreover two algorithms were designed to efficiently use the two power saving protocols: Adaptive Single Presence Period (ASPP) and Adaptive Multiple Presence Periods (AMPP). ASPP and AMPP successfully manage to significantly reduce the power consumption of Wi-Fi Direct devices acting as access points without introducing a major user experience degradation; the NoA protocol in combination with the AMPP algorithm delivers a close to optimal user experience and energy efficiency. Moreover, AMPP's tuning parameters can be configured to prioritize either energy saving or user experience according to the device manufaturer preferences [9].

One of the studies done on Wi-Fi Direct technology investigates the feasibility of creating opportunistic networks on top of Wi-Fi Direct framework by analyzing the protocol's performances in real scenarios with a variable number of mobile devices[12]. The situation shown in Figure 2.2 describes a real scenario in which

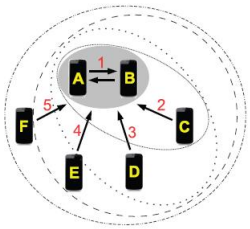

Figure 2.2: *Six nodes: incremental join.*

three or four people meet and their devices try to establish simultaneously an opportunistic network in order to exchange contents. A starting standard group was created between two devices: then, other nodes sequentially request to join the group after receiving a bacon messages. After the experiment, it is possible to notice that higher delays were experienced after adding more than 5 nodes to the same Wi-Fi Direct network. These results reflect the complexity of Wi-Fi Direct to manage a variable number of nodes joining the same group and the need to introduce additional policies at the application/middleware layer to manage additional constraints of opportunistic networks [12]. All experimental analyses of Wi-Fi Direct through real testbeds done in [12] allow to analyze advantages and constraints of using specific protocols in order to deploy opportunistic networks on a large scale. The work described in [12] represents one of the first works on Wi-Fi Direct. It is a very important work that helps people to have a first idea of how this technology behaves. So, performing opportunistic networks, we have to take into account different parameters, in terms of technical requirements (e.g., available resources and connectivities) but also of users characteristics and profiles, which can heavily influence the system's performances and devices' interactions [12].

As presented before, many studies about Wi-Fi Direct technical performances were already performed. More in details, the main studies were dedicated to energy consumption, power savings and time required to complete some important functionalities of this new technology. In this work of thesis, it is not planned to continue studies about previously presented works, but the goal is to analyze the technology from different points of view, such as group joining and data exchanging in opportunistic Wi-Fi Direct networking with channel state based routing algorithm.

2.1 Thesis Objectives

This thesis reports exploration on Wi-Fi Direct characteristics, as working and technical features. More in particular, the objective of this thesis is finding an opportunistic network strategy in which devices can exchange packets via Wi-Fi Direct using an adaptive routing algorithm. Opportunistic networks represent the natural evolution of mobile ad hoc networks (MANETs), overtaking limitations and constraints of this paradigm related to the continuous update of highly dynamic network topologies. Opportunistic networks takes advantage of the human mobility and the consequent network dynamism by defining new opportunities of communication generated by the occasional encounter of users and their personal devices [12].

We want to create a network topology in which devices are free to join or leave P2P groups without damaging the flowing packets. This aim will be reached fist by studying all the technical features of Wi-Fi and Wi-Fi Direct connections: so, all the possible strategies and methods to form clusters of Wi-Fi Direct devices will be evaluated. Later, through some experiments, some more precise analysis will be performed on behaviors and interactions among devices and external contents. More in particular, a more complete study will be performed on a particular opportunistic network with different types of packets relay methods. Finally, a new channel state based routing algorithm will be presented and tested to guarantee excellent performance even if connections are not stable.

Chapter 3

Wi-Fi Direct

Wi-Fi is a technology and standard of wireless local area network, which is defined in IEEE 802.11. Wi-Fi CERTIFIED Wi-Fi Direct, provided by Wi-Fi alliance and initially called Wi-Fi P2P, was introduced in October 2010 and it allows Wi-Fi devices connect to each other directly in a new convenient way without the need of a wireless access point (AP) . It is the reference standard to support device-to-device (D2D) communications on Wi-Fi channels[12].

Wi-Fi Peer-to-Peer technology adds, on traditional Wi-Fi, features and capabilities that optimize technology and consumer uses: this is made possible because it allows the creation of direct connections between enabled Wi-Fi Direct devices without requiring the presence of a traditional Wi-Fi infrastructure network (i.e., AP). Rather than connecting first to an infrastructure network and then connecting to another networked device, users can connect directly to those devices offering the services they need.

Wi-Fi Direct-certified devices support connection with existing legacy Wi-Fi devices. In this way, a direct connection capability is possible on the hundreds of millions of legacy Wi-Fi CERTIFIED devices (802.11 a/g/n) already deployed.[14] Devices can establish a one-to-one connection, or a group of several devices can connect simultaneously. This new technology presents some benefits to consumers:

- Mobility and Portability: since a Wi-Fi AP is not required, Wi-Fi Direct

devices can connect each other anytime and anywhere.

- Immediate Utility: P2P technology plays a role anywhere a quick connection is desired. Users have the possibility to create direct connections with any Wi-Fi Direct enable device.

- Facility of use: Wi-Fi Direct technology allows user to identify available devices and services before establishing a connection.

- Simple secure connections: Wi-Fi Direct devices use Wi-Fi Protected Setup™ to make it simple to create secure connections between devices. Users either press a button on both devices or type in a PIN (i.e., displayed by a device) to easily create a secure connection.

By definition, a Wi-Fi Direct device is capable of a peer-to-peer connection, and can support either an infrastructure or a P2P connection. Wi-Fi Direct devices have the ability to join infrastructure networks as typical stations (STAs), and must support Wi-Fi Protected Setup enrollee functionality. Wi-Fi certified Wi-Fi Direct devices work just like any Wi-Fi device, with ranges up to 200 meters. They can connect from just a few meters away, but also across a home. This means that making a Wi-Fi Direct group connection will be convenient, even when devices are not in immediate proximity to one another [1].

3.1 Key mechanisms and important capabilities

In the Wi-Fi Alliance P2P Specification some important capabilities and mechanisms are introduced. All the key mechanisms that will be presented are mandatory for a Wi-Fi Direct Devices; the other capabilities are optional but not less important.
Key mechanisms:

- Device Discovery: it is the phase in which P2P devices have to find and identify each other exchanging device information before any group formation. It is used to identify other P2P devices if they are near the considered station.

- Service Discovery: this mechanism facilitates discovery and it is not mandatory. It is used before a group establishment and it describes the service that devices provide.

- Group Formation: it includes all the phases that enable communications between devices: more in particular, when two devices want to communicate, they have to connect one to each other sending P2P invitation request in order to ask to a specific device to join a group. Otherwise, a device can discover another one through Client Discovery mechanism and they can negotiate the roles to take(Negotiation Phase). There are different types of group formation mechanism that will be explained more in detail at the end of this chapter.

- Provisioning and Encryption: security provisioning starts after discovery has taken place and, if required, the respective roles have been negotiated. Wi-Fi Direct devices are required to implement Wi-Fi Protected Setup (WPS) to support a secure connection with minimal user intervention. In particular, WPS allows to establish a secure connection by, e.g., introducing a PIN in the P2P Client, or pushing a button in the two P2P Devices. WPS is based on WPA-2 security and uses Advanced Encryption Standard (AES)-CCMP as cypher, and a randomly generated PreShared Key (PSK) for mutual authentication [5][8].

- Power Management: usually, efficient use of power is critical for portable devices. The P2P Specification includes power management mechanisms that can reduce power consumption for devices, regardless of role within a P2P

group: so, realization of power savings depends on settings and interaction between devices in a given environment. More in particular, there are two new power savings mechanisms: Opportunistic Power Save and Notice of Absence. The first allows to a particular P2P device, which is in charge of a group, to save power by entering in sleep state when also all the other connected P2P devices are in sleep period. The second mechanism makes possible to signal a planned absence. This two types of power management are available only for use in P2P groups in which only Wi-Fi Direct devices are associated. Indeed, if legacy devices are present, both power saving mechanism analyzed before cannot be employed.

Important capabilities:

- Creation of Persistent Groups: this first capability is a mechanism that allows a previously established P2P group to be re-invoked at a future time without a previous performed provisioning phase. This important feature will be explain more in detail in Chapter 4 in P2P Persistent Group Formation section.

- Enable Concurrent Connections: it is an important Wi-Fi Direct capacity that maintains multiple connections simultaneously. Connections are intended to be between P2P groups, traditional WLANs or between both of them together. In this case it is possible to recognized two types of concurrent connections:

  - Multiple Groups: P2P devices maintains membership in multiple groups simultaneously.

  - Cross-Connection: it allows Wi-Fi Direct devices, leading a P2P group, to provide infrastructure access to other devices in that specific group.

- Manage Device: this is a mechanism that allows a Wi-Fi Direct device to

respond to management direction from an AP regarding coexistence, channel selection or power limitations.

3.2 P2P Topology: the Wi-Fi Direct Architecture

Unlike the previous technologies, the Wi-Fi Direct technology takes a different approach to enhance D2D connectivity. Instead of leveraging the ad-hoc mode of operation, Wi-Fi Direct builds upon the successful IEEE 802.11 infrastructure mode and lets devices negotiate who will take over the AP-like functionalities[8]. In a traditional Wi-Fi network, devices acting as clients discover and associate to WLANs, which are created and announced by APs. Wi-Fi Direct architecture is different: a device behaves either as an Access Point or as a client, and the roles of the devices are specified as dynamic. This means that a device implements both the role of a client and the role of an AP.

3.2.1 Components of Wi-Fi Direct network

A P2P network topology can be configured as one-to-one or one-to-many as shown in Figure 3.1 below. The principal topology of Wi-Fi Direct is composed by groups

Figure 3.1: *P2P Network topologies.*

of multiple devices acting as clients connected to one device, who is having the role of Access Point (Group Owner). A Wi-Fi Direct group is called P2P group. More in general, Wi-Fi P2P networks may include two types of devices:

- P2P Device: it is a Wi-Fi Certified device that is compliant with the Wi-Fi P2P specification. It supports both P2P Group Owner and P2P Client roles.

Furthermore it supports Wi-Fi Protected Setup (WPS) and P2P Discovery mechanism.

- Legacy Device: it is a Wi-Fi Certified device that is not compliant with the Wi-Fi P2P specification.

P2P devices must be capable of two different roles in a P2P group:

- P2P Group Owner (GO): it is an "AP-like" capability that controls a specific Wi-Fi P2P group and enables P2P devices connectivity. This type of role makes it possible to work as an AP in the Wi-Fi infrastructure mode allowing the other devices to join groups as clients. GO also provides WPS Internal Registrar functionality, giving wireless settings to enrollees (new devices that join the network).

- P2P Client: it is a Wi-Fi P2P-compliant device that may connect to a P2P GO and join P2P Groups. It provides WPS Enrolee functionality.

  - Legacy devices may only operate as client in a P2P Group

3.3 Wi-Fi Protected Setup (WPS)

The Wi-Fi Protected Setup (WPS) standard is introduced to facilitate the deployment of secure Wi-Fi networks without any complicated procedures and the insertion of additional devices over time.

3.3.1 WPS methods

WPS, developed by the Wi-Fi Alliance, is a standard used for creation and instauration of secure connections on a Wi-Fi network: it helps users configure WPA/WPA2-Personal (PSK) security on wireless routers and clients.
WPS supports two main methods:

- Personal Identification Number (PIN)
- Push Button Configuration (PBC)

The PIN method allows users to use a PIN to establish a connection. It is the mandatory baseline mode; every Wi-Fi Protected Setup certified product must support it. In all Wi-Fi Protected Setup networks, each device requires a unique PIN in order to join the network. PIN is used to make sure the intended device is added to the network being set up and will help to avoid accidental or malicious attempts to add unintended devices to the network [2].
PBC method allows users to connect multiple devices to the network and enable data encryption by simply pushing a button, either a physical or a virtual one. This is an optional method used in some WPS networks. PBC is not as secure as PIN method: after the setup period unintended devices could join the network if they are in range. In both methods, the whole operation often takes a few seconds and it does not exceed a two-minute period.

3.3.2 WPS Components

There are three logical components involved in WPS:

- Registrar: a device with the authority to issue and revoke credentials to a network. A Registrar may be integrated into a wireless AP, or it may be separate from the AP. A Registrar device can be detected when a new Wi-Fi device is in range; then it prompts the user to enter the PIN if the new device wants to join the network.
- Enrollee: a device seeking to join an AP or a wireless network such as a laptop or a cell phone.
- AP: an Access Point functioning as a proxy between a registrar and an enrollee.

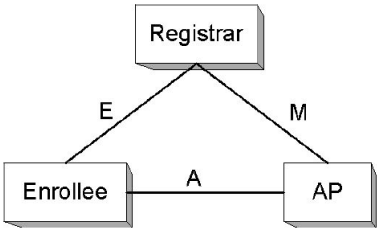

Figure 3.2: *WPS Components.*

As shown in Figure 3.2, interface E is logically located between the Enrollee and the Registrar. The purpose of Interface E is to enable the Registrar to discover and issue WLAN Credentials to the Enrollee. Interface M is the interface between the AP and the Registrar. It enables an external Registrar to manage a WPS AP. WPS uses the same protocol for setting up the AP Management interface as for issuing Credentials to Enrollee devices. Interface A is between the Enrollee and the AP. The function of Interface A is to enable discovery of the WPS WLAN and to enable communication between the Enrollee and IP-only Registrars.

3.3.3 WPS vulnerability

The PIN method (mandatory for certification) requires only the knowledge of the PIN making the WPS potentially vulnerable to brute force attacks. Discovering

the PIN exposes the WPA/WPA2 PSK found in ConfigData which contains the WLAN settings and Credentials for the Enrollee. Basically, these attacks rely on discovering the PIN much quicker than brute forcing the PSK and work as follow:

- If the WPS Registration Protocol fails at some point, the Registrar will send a NACK message.
- If the attacker receives a NACK message after a particular EAP message exchange, he knows that the first or the second half of the PIN was incorrect, depending on the EAP message.

3.3.4 Provisioning and Encryption in real case

Analyzing more in detail the real case study, the provisioning phase begins when the client connects to the GO to exchange credentials with the WPS protocol: this procedure is an exchange of eight EAP messages.
To allow the connection, the user normally has to enter a PIN code or push a button on the device. When devices have to join an existing group or to speed up the provisioning phase later, they can send Provision Discovery request/response frames before starting the group negotiation. If not, the GO Negotiation may fail and have to be restarted when the user has taken more time than expected. After that, the normal RSN (WPA2) 4-way handshake begins to exchange the encryption keys, where the GO assumes the role of authenticator and the client is the supplicant. Then the client will request a IPv4 address from the GO, which is required to implement an DHCP server, and finally actual data transfer can happen.
To avoid users having to enter a PIN code every time when a group is formed regularly between some devices, the group can be made persistent, in which the devices store credentials and can automatically re-connect when required. A persistent group may use a different channel and device MAC addresses for each session. The persistent group gives the possibility to reuse the same credentials to

join the group more than once. Moreover, every time a persistent group has to be activated, it can work on a different channel than the previous session. This important feature allows to enable communications on the best channel, depending on the actual network status.

3.4 Group Formation

Wi-Fi Direct devices connect by forming groups (in a one-to-one or one-to-many topology) that work in a manner similar to an infrastructure BSS. More in particular, every single P2P device can be in charge of a group, including controlling which devices are allowed to join and when that specific group is started and terminated.

All Wi-Fi Direct devices have to be capable of acting as Group Owner in P2P groups, and they must be able to negotiate which devices have to take the role of Group Owner during a group creation. Communications can be established between Wi-Fi Direct and legacy devices (Figure 3.3). There are several ways in

Figure 3.3: *Group composed by Wi-Fi Direct devices and Legacy Client.*

which two devices can establish a P2P Group, depending on, e.g., if they have to negotiate the role of P2P GO, or if there is some pre-shared security information available. There are three cases that will be analyzed: Standard, Autonomous and Persistent.

3.4.1 P2P Standard Group Formation

Using this first method, P2P devices have first to discover each other, and then negotiate which device will act as P2P GO. Four phases are defined in order to establish a P2P Standard Group:

1. Discovery
2. GO Negotiation
3. WPS Provisioning
4. Address Configuration

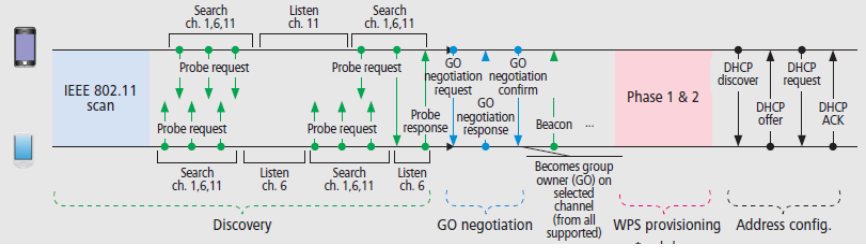

Figure 3.4: *Phases of P2P Standard Group Formation.*

3.4.1.1 Discovery phase

This is one of the key mechanisms defined in the Wi-Fi Alliance P2P specification. Indeed, it is used to find Wi-Fi Direct devices, exchanging device information. At the beginning of each group formation procedure, Wi-Fi Direct devices perform an active or passive scan, by which they can discover existent P2P Groups and Wi-Fi networks in the near area. After this initial scan, a P2P Device selects one of the Social channels (1, 6 or 11 in the 2.4 Ghz band) as its Listen channel. Then, it alternates between two states: search and listen. When the device is in the first state, it performs active scanning by sending Probe Requests in each of the social channels. The latter state is used by the device for listen to Probe Requests in its listen channel, so it can then reply with Probe Responses. The amount of time that a P2P Device spends on each state is randomly distributed, typically between 100 ms and 300 ms, but it is up to the implementation to decide on the actual mechanism to, e.g., trade-off discovery time with energy savings by interleaving sleeping cycles in the discovery process.

3.4.1.2 GO Negotiation phase

Once the two P2P Devices have found each other, they start the GO Negotiation phase. This is implemented using a three-way handshake, namely GO Negotiation Request/Response/Confirmation, in which the two devices agree on which device will act as P2P GO and on which channel the group will operate, which can be in the 2.4 Ghz or 5 Ghz bands. In order to agree on the device that will act as P2P GO, P2P devices send a numerical parameter between 1 and 15 (GO Intent value), within the three-way hand-shake, and the device declaring the highest value becomes the P2P GO. In order to prevent conflicts when two devices declare the same GO Intent, a tie-breaker bit is included in the GO Negotiation Request, which is randomly set every time a GO Negotiation Request is sent. The whole procedure is simply described in the Figure 3.5 below.

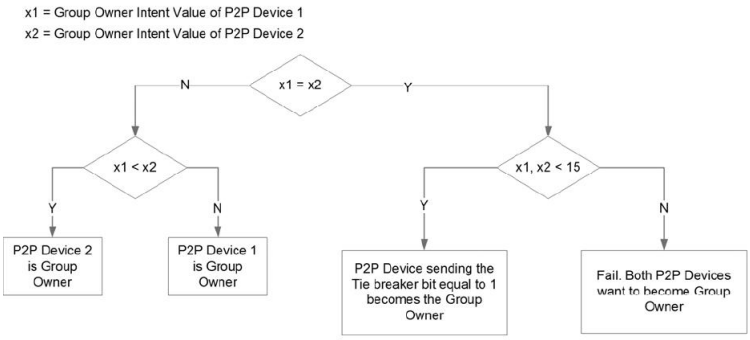

Figure 3.5: *Group Owner determination flowchart.*

3.4.1.3 WPS Provisioning phase

Once the devices have discovered each other and agreed on the respective roles, the next phase is the establishment of a secure communication using WPS, which we denote as WPS Provisioning phase (shown in Figure 3.6). Here, P2P devices perform mutual authentication and share encryption key. More in detail, the WPS Provisioning is composed by two phases: in the first one P2P devices generate and share Master key. Then in the second phase, using 4-way handshake, they generate

link key from the shared Master key [7].

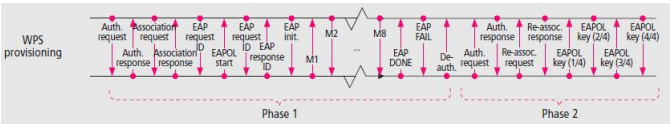

Figure 3.6: *WPS Provisioning Phase.*

3.4.1.4 Address Configuration phase

The last part of the P2P group formation is dedicated to the address configuration phase, in which a DHCP exchange is done in order to set up the IP configuration. Once the GO role is defined, the device that acts as group owner becomes also a DHCP server: it has the capability of provide IP address to each of the clients of the group. In DHCP server, the following procedures are occurred:

- Configuration of the udhcpd configuration file
- Running of the DHCP server
- Definition of the IP tables to enable the NAT (Network Address Translation)

3.4.2 P2P Autonomous Group Formation

A P2P Device may autonomously create a P2P Group, where it immediately becomes the P2P GO, by sitting on a channel and starting to send beacons. Other devices can discover the established group using traditional scanning mechanisms, and then directly proceed with the WPS Provisioning and Address Configuration phases. Compared to the previous case, the Discovery phase is simplified in this case as the device establishing the group does not alternate between states, and indeed no GO Negotiation phase is required.

3.4.2.1 Phases

There are three principal phases in order to create a P2P Autonomous Group:

1. Discovery
2. WPS Provisioning
3. Address Configuration

The flow of commands in this method is illustrate as follows: In order to configure

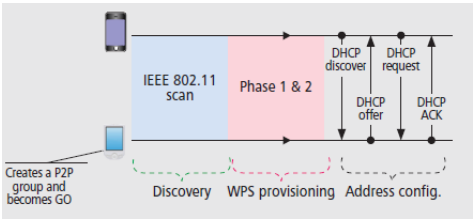

Figure 3.7: *P2P autonomous group formation.*

the IP address of the GO and the clients there are 2 methods:

1. Static IP
2. Enable the DHCP server at the GO side, which will provide IP address to the clients using DHCP session

3.4.3 P2P Persistent Group Formation

A persistent group can be re-invoked for additional sessions after initial formation. So, it is possible to restart this kind of groups without provisioning, eliminating some phases such as entering a WPS PIN.
Persistent groups are accomplished by Wi-Fi Direct devices storing the group information and credentials in each device that is part of the group. Since Persistent Groups are invoked using Invitation signaling, it is clear that all the P2P devices that have the capability of supporting Persistent Groups, have also to support Invitation mechanism [14].
Four main phases characterize this type of formation:

- Discovery

- Invitation
- WPS provisioning
- Address configuration

As the previous case, the negotiation phase is not present because the group is already created and the device that want to establish a group becomes immediately GO as shown in Figure 3.8. So, every device can enable a group already created in which it act as GO of that group. After the activation of the group, all the clients already participating at the group can join in it again just asking for the IP address to the GO.

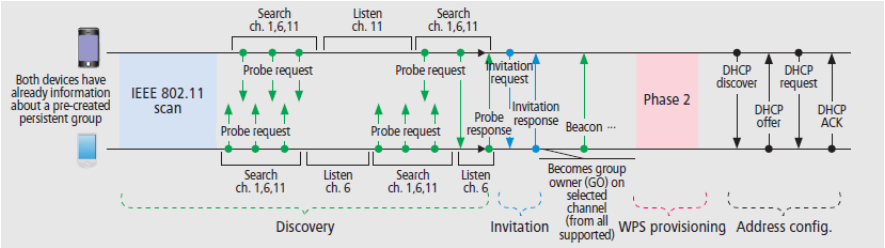

Figure 3.8: *P2P Persistent Group Formation.*

Chapter 4

Materials and Methods

In this chapter the used tools and programs are described in order to better understand how the work was made possible.

4.1 PC-Engine Alix system Board

The main role was covered by the devices involved in the Wi-Fi Direct communication: PC-Engine Alix system Board. The specifications of the machines can be listed as follows:

- CPU: 500 MHz AMD Geode LX800
- DRAM: 256 MB DDR DRAM
- Storage: CompactFlash socket, 44 pin IDE header
- Power: DC jack or passive POE, min. 7V to max. 20V
- Three front panel LEDs, pushbutton
- Expansion: 2 miniPCI slots, LPC bus
- Connectivity: 2 Ethernet channels (Via VT6105M 10/100), 2 Wireless Atheros cards

- I/O: DB9 serial port, dual USB port
- Board size: 6 x 6" (152.4 x 152.4 mm)
- Firmware: tinyBIOS

These machines were used as Wi-Fi Direct devices in order to establish P2P communications among them. An example of Alix Board is shown in Figure 4.1 below.
Each board is equipped with Compact Flash cards, 4Gb each: on each of them

Figure 4.1: *PC Engine Alix Board.*

there was initially installed Ubuntu 12.04.4 (Precise Version). The optimal procedure and steps of installation are proposed in the appendix of the thesis.
All the boards can be connected among each other in order to communicate and to exchange data. The most important part for managing communications is covered by the usage of two fundamental programs: wpa_supplicant and wpa_cli.

4.2 WPA

Wi-Fi Protected Access (WPA) is a security protocol developed by Wi-Fi Alliance. WPA offers both a shared secret and support for real user authentication, usually through a RADIUS server on the backend. While packet encryption options are

a bit limited, it does offer extended key lengths over WEP, and provides a key rotation procedure (called TKIP or Temporal Key Integrity Protocol) to avoid weaknesses inherent in both WEP's crypto and its implementation. It's generally considered to be a good enough solution for now, and should last until well after WPA2 support starts appearing in commercial products.
Like WEP, WPA requires hardware support to help with the packet encryption. In this work Atheros cards were used because they were perfecty compatible with WPA. We needed a Linux driver for WPA-enabled wireless card that support the WPA features. 802.11 driver was used for the Atheros cards. In addiction, we need a supplicant, that is a daemon program that authenticates the ethernet or wireless connection. The supplicant will set up all aspects of an authentication connection, assuring that the network port is valid. After authenticating a DHCP will be needed for a dynamic assignment of IP addresses.

4.2.1 wpa_supplicant

wpa_supplicant is a free software implementation of an IEEE 802.11 supplicant for many operative systems, like Linux, Mac OS X and Windows; it implements WPA, WPA2 and other wireless LAN security protocols.
It is designed to be a "daemon" program that runs in the background controlling the wireless connection that will be establish. wpa_supplicant implements a control interface that can be used by external programs to control the operations of the wpa_supplicant daemon and to get status information and event notifications. wpa_supplicant supports many front-end programs like wpa_cli and GUI (wpa_gui): these two are example programs using a small C library, that provides helper functions to facilitate the use of the control interface [3].
wpa_supplicant uses the control interface for two types of communication: commands and unsolicited event messages. The former are a pair of messages, a request from external program and a response from wpa_supplicant. The latter are sent by wpa_supplicant to the control interface connection without specific

request from the external program for receiving each message. However, the external program needs to attach to the control interface to receive these unsolicited messages [3].
In order to open and close one or more control interfaces, two specific commands are used: "wpa_ctrl_open()" and "wpa_ctrl_close()". This software needs a configuration file previously defined to store descriptions of all wireless networks with all their credentials. This configuration file can be used to allow automatic connection once the WPA Supplicant process is running. Wireless networks are listed in that file in order to having immediate informations when needed.
An example of configuration file used in this thesis work is the following:
$ctrl\_interface = /var/run/wpa\_supplicant$
$update\_config = 1$
$p2p\_go\_ht40 = 1$
The second line of the code is in charge of enabling multiple connections. The third line is set in order to support 802.11 for GO.

4.2.2 wpa_cli

wpa_cli is a text-based front-end program for interacting with wpa_supplicant. It is fundamental for managing Wi-Fi Direct communications. More in particular wpa_cli program is used to allow the interaction between devices through some fundamental commands:

- *list_networks*: list of configured networks

  - network id / ssid / bssid / flags
    0 DIRECT-xy MAC_GO [DISABLED][P2P PERSISTENT]
    1 DIRECT-dX any

- *status*: current WPA/EAPOL/EAP status information of the current network. The output of this command is a text block with each line in variable=value format. An example is shown below:

  - Selected interface 'p2p-wlan0-0'
    bssid=06:f0:21:06:1d:1f
    ssid=DIRECT-xl
    id=0
    mode=P2P GO
    pairwise_cipher=CCMP
    group_cipher=CCMP
    key_mgmt=WPA2PSK
    wpa_state=COMPLETED
    p2p_device_address=04:f0:21:06:1d:1f
    address=06:f0:21:06:1d:1f
    uuid=0207cb47-7e8f-5e85-83fc-0726eb3b802b

- *p2p_group_add*: set up a P2P Group Owner manually (without GO negotiation with a specific peer). This is also known as autonomous GO.
- *p2p_group_add persistent=<network_id>* : command used to specify restart of a persistent group
- *p2p_connect MAC_GO pbc join*: start P2P group formation with a discovered P2P peer. "pbc" string starts pushbutton method. "join" indicates that this is a command to join an existing group as a client. It skips the GO Negotiation part
- *p2p_find*: start P2P device discovery
- *scan*: request a new BSS scan
- *scan_results*: get the latest scan results

  - bssid / frequency / signal level / flags / ssid
    b4:74:9f:dc:2d:aa 2412 -60 [WPA-PSK-TKIP][WPA2-PSK-CCMP][ESS] giulia
    b6:74:9f:dc:20:13 2462 -29 [WPA2-PSK-CCMP][WPS][ESS][P2P] DIRECT-He
    b6:74:9f:dc:25:87 2462 -61 [WPA2-PSK-CCMP][WPS][ESS][P2P] DIRECT-Jn

- *wps_pbc*: complete the WPS negotiation that generate a new WPA PSK
- *remove_network <network_id>*: remove a network. "network_id" can be received from the LIST_NETWORKS command output

- *terminate*: terminate wpa_supplicant process

4.3 Additional Programs

In addition to the programs already mentioned, in order to generate traffic and study the flow of packets among devices, Wireshark, tcpdump and iperf were used.

- Wireshark: this first one is a network packet analyzer and it is used for capturing and analyzing network packets in many levels of details. Thank to Wireshark, it has been possible to see the components of each packet: this made possible the study of each packet to understand how the system works in each phase of the connection.

- Tcpdump: is a command-line tool for monitoring and sniffing network traffic. Using this tool, flows of packets were checked and monitored.

- Iperf: it is a tool for active measurements of the maximum achievable bandwidth on IP networks. It is possible to set each device as a client or as a server. The client peer decides how to send packets (TCP or UDP setting), the quantity and all the settings about the traffic to send. On the server side, the device gets ready to receive packets, setting only the TCP or UDP function chosen by the client. This tool were used to study the time and the way of packets arrival in each device.

Chapter 5

Experimental Results and Performance Evaluation

This chapter has the aim of describing how Wi-Fi Direct connections are established and how these are used in different scenarios. Moreover, a new opportunistic routing algorithm is presented and tested in a real scenario. Its efficiency is proved thanks to many experiments that will be presented in this chapter.
First, we decided to study and test all the group establishment typologies, choosing the one that can be used in the following experiments. When groups of devices are formed, we use more than one device to establish connections, for the sake of understand connection behaviors in a cluster scenario. The next step of the work is focused on establishing more than one group using the same devices, to be sure about GO and client's features, already studied in theory. With this experiment, we are able to test IP assignment methods and study the final status of involved devices. Thanks to this study, we have also the possibility to analyze physical and virtual interfaces topic, discovering their working principles and how they behave in a specific group establishment instant.
Routing packets in Wi-Fi Direct is one of the main topic of this study: starting from a simple topology, we want to present a real scenario in which devices exchange packets with external networks, using two different Wi-Fi technologies simultaneously: standard and Direct.

In the last part of the chapter, it is presented an experimental testbed with four wireless nodes, built on solid requirements tested in the previous studies. This testbed is useful in order to test and use a new particular opportunistic routing algorithm thought for relaying packets from a source node to a destination node localized in two different groups. This innovative algorithm is studied and tested to guarantee excellent performance even if connections are not stable. Specific throughput tests are also illustrated to show how the algorithm improves traffic performance.

5.1 Cluster formation

The first part of the experimental work is dedicated to establish connections among Wi-Fi Direct devices. At the beginning, we tested all types of Wi-Fi Direct connections between two Alix Boards; subsequently, the trials were made with more devices, up to eight components. The first experiments described in this section are useful to understand Wi-Fi Direct basis, in order to apply good results to future experiments.

As explained in previous chapters, in Wi-Fi Direct Groups there are two fundamental roles that devices can cover in a P2P Group: P2P Group Owner (P2P GO), that implements AP-like functionality, and client (known as P2P Client) attached to one or multiple GOs.

Knowing that all the communications in a group have to pass through the GO, once the GO disconnects, the routing channel is killed and also the group is stopped. In this situation, connections between devices have to be reestablished and one of the remaining clients will take the role of GO.

In order to create groups of devices connected among each other, as shown in the example of Figure 5.1, we used machines composed by two physical wireless interfaces each. Every interface can support multiple virtual interfaces: which

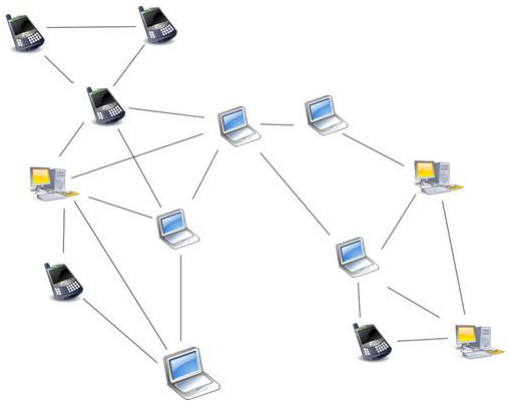

Figure 5.1: *P2P devices connected through Wi-Fi Direct Groups*

are automatically created everytime a device uses one interface for taking part to more than one P2P group.

5.1.1 Group establishment

There are several ways to create P2P groups. The most generic one is the Standard Group Formation, composed by several steps:

- Discovery
- GO Negotiation
- WPS Provisioning
- Address Configuration

Every connection is established between two devices: each of them scans the channels to find other groups and devices in range. Using the specific commands *wpa_cli -i interface scan* and *wpa_cli -i interface scan_results*, it is possible to have scanning results like the ones shown in the example presented below:

```
bssid / frequency / signal level / flags / ssid
b4:74:9f:dc:2d:aa 2412 -60 [WPA-PSK-TKIP][WPA2-PSK-CCMP][ESS] giulia
b6:74:9f:dc:20:13 2462 -29 [WPA2-PSK-CCMP][WPS][ESS][P2P] DIRECT-He
```

```
b6:74:9f:dc:25:87 2462 -61 [WPA2-PSK-CCMP][WPS][ESS][P2P] DIRECT-Jn
18:64:72:0d:74:72 5500 -76 [WPA2-PSK-CCMP][ESS] IMDEANetworks5G
18:64:72:0d:81:32 5260 -76 [WPA2-PSK-CCMP][ESS] IMDEANetworks5G
18:64:72:0d:74:61 2462 -72 [WPA2-PSK-CCMP][ESS] INGuest
18:64:72:0d:74:60 2462 -74 [WPA2-PSK-CCMP][ESS] IMDEANetworks2G
18:64:72:0d:81:20 2437 -75 [WPA2-PSK-CCMP][ESS] IMDEANetworks2G
```

From the example above, we recognize P2P groups from the flags and the name attached to each network: the name of a Wi-Fi Direct group is usually composed by "DIRECT-" and one of the flag of the specific network is [P2P]. Therefore it is possible to notice that the second and the third group are Wi-Fi Direct groups. Once a device has found an active P2P group, it can ask to the GO (marked in the table above as "bssid") to join the group with a specific request. In this case the group is already formed and the GO is previously established, so the Negotiation phase is not necessary and the device is asking to join though a specific command:

```
wpa_cli -i INTERFACE p2p_connect MAC_GO_ADDRESS pbc join
```

When the GO receives the request, it replies accepting or rejecting the join request. Next, the GO initiates the Wi-Fi security setup using WPS. After the security setup is complete, the Address Configuration phase begins. In particular, the GO device runs the DHCP protocol and it assigns IPs to the device asking for joining the group. The specific procedure is proposed in the appendix of this thesis. At the end of the procedure, any device can control its own status using this specific command: *wpa_ cli -i interface status*. An example of the result of this command is composed as follow.

```
bssid=b4:74:9f:dc:2d:aa
ssid=DIRECT-xy
id=0
mode=station
```

```
pairwise_cipher=CCMP
group_cipher=TKIP
key_mgmt=WPA2-PSK
wpa_state=COMPLETED
ip_address=192.168.0.21
p2p_device_address=b4:74:9f:dc:20:27
address=b4:74:9f:dc:20:27
uuid=99c36f0d-ef4e-5d21-bb8c-209c59f695ff
```

In this particular case, this device covers the role of client and it is part of a P2P group called "DIRECT-xy". It is possible to notice that it has already an IP address because of the DHCP phase operated in the previous group establishment. In addition, each device has its own MAC address and a UUID (universally unique identifier). Every device that joined a group keeps also saved, in the configuration file, all the P2P group features, i.e. the MAC address of the GO (identified by "bssid"), the name of the group or, in case the device is a GO, the clients MACs of the group. An example of configuration file is presented below.

```
ctrl_interface=/var/run/wpa_supplicant
driver_param=use_p2p_group_interface=1
update_config=1
device_name=machine119
device_type=1-0050F204-1
p2p_go_ht40=1

network={
        ssid="DIRECT-wv"
        bssid=04:f0:21:06:1d:1f
        psk=b1704a8ad003a8f1d7bb183c8d90cb5c2a9e6819099618b31449aa4690748e49
        proto=RSN
```

```
        key_mgmt=WPA-PSK
        pairwise=CCMP
        auth_alg=OPEN
        disabled=2
}
network={
        ssid="DIRECT-Up"
        bssid=b4:74:9f:dc:20:27
        psk="E2Aqw0NE"
        proto=RSN
        key_mgmt=WPA-PSK
        pairwise=CCMP
        auth_alg=OPEN
        mode=3
        disabled=2
        p2p_client_list=b4:74:9f:dc:2d:aa
}
```

In this case, this device is client of "DIRECT-wv" group and GO of "DIRECT-Up" because we can notice that in the latter it is also present the "p2p_client_list" attribute. When a device has saved the client list of its group, it is able to update the routing table when one arriving packet has the destination MAC equal to a P2P client address. Here the group is composed by only two members, so the GO has one client in the client list.

Every device, after saving all the features mentioned before, knows who is its own GO. Moreover this is important when the device wants to send a packet to other machines: it has to send everything to the GO, which acts as AP, relaying packets based on its own routing table.

Thanks to the methods and features described, there is the possibility to create

a cluster of devices that communicate among each other with Direct techniques. If a device, belonging to a group, wants to send packets to another device that takes part to another group, there is the possibility to route the packet through other devices in order to deliver the packet correctly. In this case, the client device sends the packet to his GO, which forwards the data to other devices till the right receiver. Each device that wants to communicate with others has to pass through the GO of the group it belongs to. If the receiver device is not belonging to the same group of the client, the "bridge" device which is taking part to more than one group, will forward the packet to the right group, and consequently to the right device.

In order to have an idea on how the groups work, four devices were used to create multiple groups among them. Each device has two wireless physical interfaces. Figure 5.2 below gives an idea of the topology.

The black group is composed by A and B: A is the GO and B is the client.

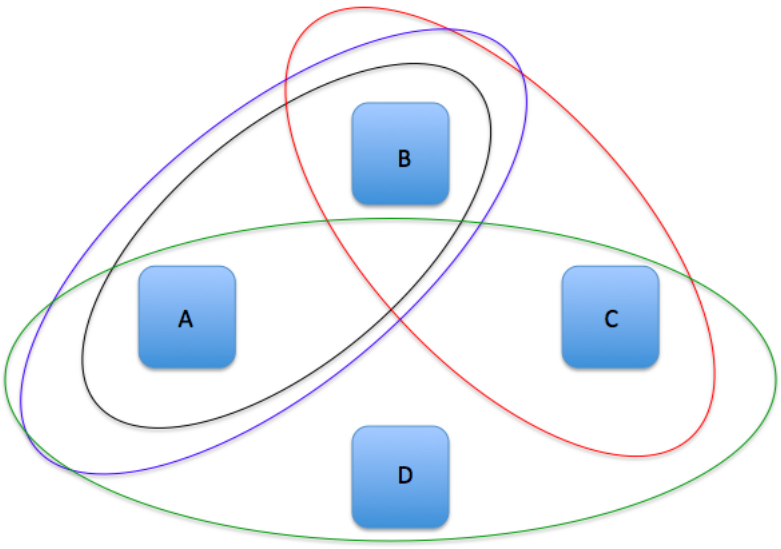

Figure 5.2: *Basic topology composed by four devices and four P2P groups.*

Both devices have automatically created virtual interfaces in order to support the creation of the P2P group. The blue group involves the same devices of the

previous group: this trial is made to study the behavior of one device involved in two different groups. In this case, A is covering the role of GO in the black group: so, it was tried to enable A to be GO also for the second group (blue). It was observed that one device cannot cover the role of GO in more than one group simultaneously.
The third group of the topology is formed by B, already client of the black group, and C. B can cover the role of GO, because it is already client in another group but not GO elsewhere. Therefore, a device can be client of many groups simultaneously: each role covered by the device is placed on a different virtual interface. Unfortunately, a device cannot cover the GO role in more than one group simultaneously: this is an important feature that has to be taken into account when more than one group among the same devices has to be formed.
The last group is composed by three devices: two of them are already part of two P2P groups mentioned before, the third is new. The GO of this last group is C: this is possible because C is already part of another group, but as client, so it can cover the role of GO or client. The new client joins the group with a join request.
The topology just presented is the first work performed in order to understand how multiple P2P groups can interact, analyzing the specific behavior of each device that takes part of multiple group simultaneously.

5.1.1.1 First results

The work presented in the previous part of the chapter allows us to show some positive results, that will be presented in this section. Moreover, it will be presented the concept of multiple virtual interface, fundamental for establishing many groups with the same joining devices. It possible to assign IP addresses to all the devices using two main methods:

1. Fixed Assignment

2. Dynamic Assignment (DHCP)

In the first case IP addresses were manually set in fixed mode to P2P devices after establishing the group. The second method allows the devices to have an IP address assigned by the GO. In this situation, the GO dynamically assigns the addresses to all devices of his group.
It is also possible to create P2P groups with multiple devices (more than 2). After the creation of the Persistent Group, the client devices can exchange packets between each other without using the GO as AP.
Every device can act as client and GO at the same time: one device can be both client and GO of two different groups. It is not possible that a device covers the position of GO for more than one group.
Every created group is strictly correlated to one virtual interface. An example is show in Table 5.1: this is the output of the "status" command of three devices belonging to the same group. Peer 2 has the role of GO and his MAC address is the BSSID saved in each device. The name of the group (DIRECT-1M) is saved in SSID content. Moreover, each peer is connected to the group though a specific interface, specified in the first field of the table: the interface is selected when the group is established and it is never chosen two times for two different active groups.

If the group already created is a Persistent Group, every time a device wants to reconnect to this group, the peer can connect easily to the GO because every component of the group has saved the group credentials inside each own device, as shown in the table.
In order to increase the potentiality of the created network of devices, another goal to reach is related to create a group with dynamic GO. This means that a Persistent Group is created by a GO that has not a role already predefined.
The creation of Persistent Groups can be used in future studies about dynamic clusters. A cluster is typically formed in order to collect two or more devices

Peer 1	Peer 2	Peer 3
Selected interface 'p2p-wlan0-3'	Selected interface 'wlan0'	Selected interface 'wlan0'
bssid=04:f0:21:06:1d:14	bssid=04:f0:21:06:1d:14	bssid=04:f0:21:06:1d:14
ssid=DIRECT-1M	ssid=DIRECT-1M	ssid=DIRECT-1M
mode=**station**	mode=**P2P GO**	mode=**station**
pairwise_cipher=CCMP	pairwise_cipher=CCMP	pairwise_cipher=CCMP
group_cipher=CCMP	group_cipher=CCMP	group_cipher=CCMP
key_mgmt=WPA2-PSK	key_mgmt=WPA2-PSK	key_mgmt=WPA2-PSK
wpa_state=COMPLETED	wpa_state=COMPLETED	wpa_state=COMPLETED
p2p_device_address= 04:f0:21:06:1d:1f	p2p_device_address= 04:f0:21:06:1d:14	p2p_device_address= b4:74:9f:dc:20:12
uuid= 0207cb47-7e8f-5e85-83fc-0726eb3b802b	uuid= 94f8bf7f-bad8-5fc8-87c2-05c3afbed265	uuid= f4591d12-95b5-5ff8-8f68-8a02249f4227

Table 5.1: *3 devices join the same group DIRECT-1M.*

together that have common purposes or similar properties. The creation of these groups will be managed by algorithms based on specific parameters. The dynamic cluster procedure will be implemented in order to relay packets between devices involved in the groups.

5.1.2 Multiple Interfaces

As previously mentioned on the important feature of having multiple virtual interface for each physical once, this section will explain how the interfaces work.
An example of Alix Board with two wireless interfaces is shown in Figure 5.3. The possibility of having more than one virtual interface active in the same time is fundamental: thanks to this, each Alix Board can be connected to more than one group simultaneously.
Every time a single device creates or joins a group, this is established on a specific interface. Each physical interface is enabled to create multiple virtual interfaces on the same physical one.

Figure 5.3: *Alix Board composed by two wireless interfaces.*

5.2 Routing packets in Wi-Fi Direct

The previous experiments were performed to know clearly how Wi-Fi Direct works, checking that group of devices can be formed following different methods and communications among machines can be established in few seconds.

Going on with analyses and observation, the next experiment, presented in this section, is focused on routing of packets. Starting from a simple topology, we want to prove that when a client device wants to send packets to another client of a P2P group, it sends automatically all the data to its own GO: furthermore, each packet passes through GOs, given that the latter is working in a similar way respect to AP in Wi-Fi architecture.

First of all, starting from a simple topology shown in Figure 5.4, two clients are able to communicate, sending packets from one device to the other passing through the GO.

Referring to Figure 5.4, we wanted to send packets from A to B, knowing that the peer on the top is the GO (set during the previous phases of group formation).

In order to generate traffic and analyze the flowing of packets, we use Iperf:

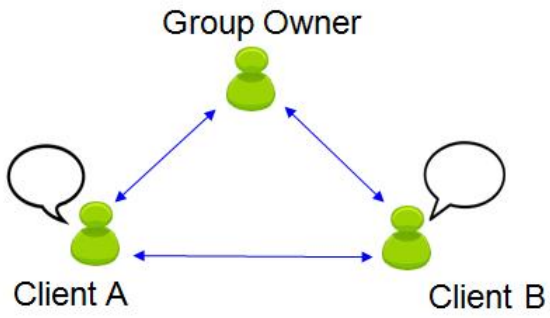

Figure 5.4: *Simple P2P Group with one GO and two clients.*

device A is set as client and device B as server. The principal goal that we want to obtain is to find out how packets are relayed and how routing is changing due to the role of devices. While packets are flowing in the network, it is possible to study all the traffic through Wireshark. Thank to this protocol analyzer, we have the opportunity to study all the specific components of flowing packets.
In particular, we are interested in three packets fields:

- Source
- Transmission
- Destination

These addresses correspond respectively to the IP address of the client machine, the IP of the intermediate device in which packets pass through and server address (client B in this case). Analyzing these three fields, it is possible to notice that the GO is always a transmitter device when the packets destination is different from its address. Knowing that the group owner owns the AP-like role, we can confirm that the packets sent by A arrive at B after passing through the GO.
With the above described experiment, it is possible to understand how Wi-Fi Direct works: regarding relay of packets from source to destination, there are no differences between Wi-Fi and the new technology because both of them are

using the same method to route packets among devices. It is clear, from the IP address contained in the packet's transmission field, that GOs play the same role of APs. Therefore, in Wi-Fi Direct there are no "direct" connections between devices: every packet sent by a client is forwarded directly to its own GO because each machine knows only the address of its GO and GO knows the addresses of its clients.

The routing through the GO is clearly described by the routing tables. Once the group is formed, the devices' routing tables automatically change and adapt to the new topology.

More in details, it is possible to present a simple example to understand how routing table works in a simple scenario (Figure 5.5). In Figure 5.5 three devices

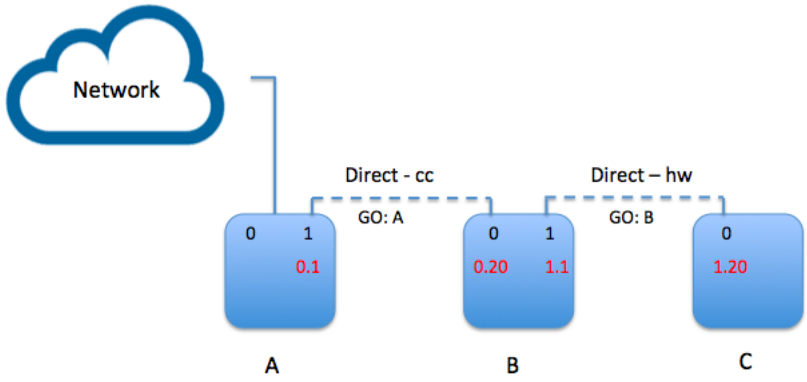

Figure 5.5: *Small topology composed by three Alix Board. One of them is connected to the external network. The communication between Alix C and Internet is possible thanks to routing packets through the other devices.*

are connected together through two different P2P Groups: Direct-cc and Direct-hw. The number "0" and "1" represent interfaces numbers for each machine. The red numbers identify the end of the IP address of each device on each interface. Machine B has two IPs because it is connected to two groups simultaneously.

If machine C wants to exchange data with the external network, packets have to pass through A and B. Since the GO of the left group (Direct-cc) is A and the GO of the right one is B, each client knows that if it wants to send data to another device, it had to send it to the GO of its group. In this case, machine A

does not know anything about connections between machine B and C because its routing table has information only about the IP of B (as client) concerning the same group. So, we have to add in the routing table of A:

```
route add -net 192.168.1.0/24 gw 192.168.0.20
```

This enable machine A to send packets with destination in this net 192.168.1.0/24 to device B. Without this command, this could not be possible because device A is the GO, and it would never send a packet to its clients without a specific cause. Also in device B, we have to add:

```
route add -net 192.168.1.0/24 gw 192.168.1.1
```

for the same reason as the previous case.
Now the system works in both direction, whether from external network to device C or in the other way around.
Each client of the group knows that it has to send packets to the GO whatever is the destination. Moreover, also GO knows something more about the neighbor network, so they can send and forward packets to clients, even if the destination device is not in the same group.

5.3 Real topology

In this section, many scenarios in real topology will be presented, to study the behavior of connections depending on the particular experiment.
Now that all the working principles of Wi-Fi Direct connections are clear, it is possible to present a real scenario in which more devices communicate among each others. The first experiment is focused on one single Alix Board (Figure 5.6): the goal is to use both physical wireless interfaces to connect the device to a PC and to a specific network. More in particular, with the first interface (wlan0), the device can connect to a PC through Wi-Fi Direct technology. With the other

interface, wlan1, it connects to a specific network using a Wi-Fi technology, as shown in Figure 5.6.

This brief experiment let us be sure about some very important working mech-

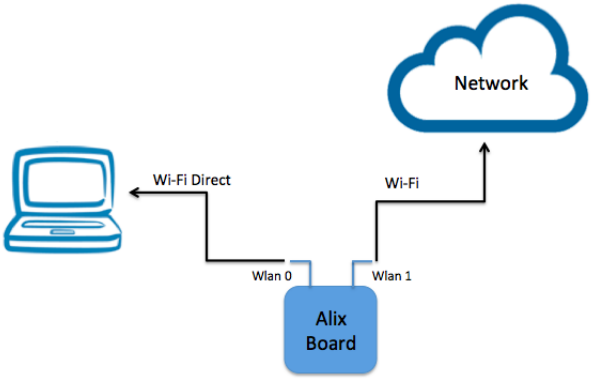

Figure 5.6: *One device topology.*

anisms: one Alix can be connected simultaneously to other devices or networks using both Wi-Fi technologies: standard and Direct. The approach presented above works perfectly: this experiment allows us to prove that one device can use at the same time two technologies combined in a perfect way. therefore, this simple system can be use in bigger architectures, knowing that this has solid and tested working principles. We have always to take into account that a device has to use the two technologies on two different physic interfaces, in order to establish two different connections. The flow of packets is perfectly fluent from the source to destination, passing through the middle device: the PC in Figure 5.6 can reach contents from the external network to which the device is direcly connected.

The evaluation of the performances goes on with a more complicated topology, composed by more than two devices. As shown in Figure 5.7, three Alix Boards are used to study the behaviors of connections. In this topology, the connection between Alix A and the external network is Wi-Fi, instead, the other, on interface 1, is a Wi-Fi Direct connection. All the other connections among the devices are Wi-Fi Direct. After creating groups of devices, all the machines can communicate

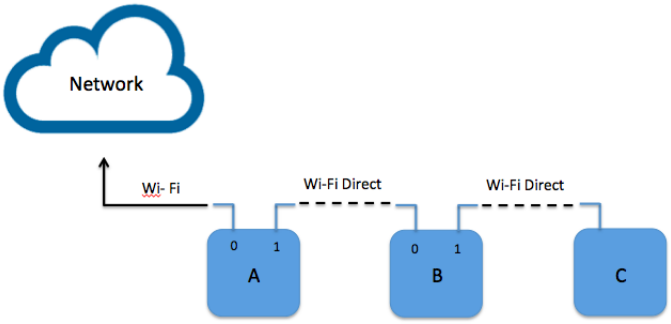

Figure 5.7: *Single device topology.*

among each other and between them and the external network. More in particular, what is important in this experiment is the routing of the requests in all the topology: device C can communicate with B or with A. In this second case, the packet from C to A will be routed by B to A. About the network access, A, B and C can reach the network easily: indeed, if B or C wants to connect to the network, it has to pass through A, in order to route the request to the network. So, if C want to access to the network, but it is connected only to B, it sends a request to interface 1 of B (see Figure 5.7): the request is routed in the other interface and sent again through interface 0. When the request is in A, there is a transition between the two interface, and finally the request is forwarded by interface 0 of device A.

In this topology there are two P2P groups: one is composed by A and B, connected through A1 and B0 with A as GO. The other is formed by B and C, with the role of GO covered by B. So B is client of the first group and GO of the second. A is connected through two different technologies in order to be able to exchange data with both an external network and another device.

These two kind of topologies help us to create network of devices more and more complicated depending on the needs. The topology that will be used in many studies in this thesis is composed by four Alix Boards. A representative image is shown below in Figure 5.8. One device is connected to an external network;

Figure 5.8: *Four alix connected through three different P2P groups.*

the others are connected together through Wi-Fi Direct technology. Three P2P groups are present: one is formed by A, B and C, another by B and D and the last one is composed by C and D. Device A is the GO of his P2P group; B is the group owner of the second group and C is the group owner of the third. So, D is a client of two groups. Each device is capable of being part of multiple groups because of its two physical interfaces: each interface can cover the role of GO or client. This way, each Alix Board can take part at least of two groups: each physical interface can create multiple virtual interfaces in order to enable the device to be part of more than one group per physical interface. In this case, we have four devices: each device has two physical interface and each interface is taking part of only one group.

All the devices can reach the external network, just sending its request or packet to the GO of the group it belongs to. The routing will be execute automatically thanks to the routing tables in each device.

5.4 Channel diversity in Wi-Fi Direct

Taking in consideration the topology already delineated, we want to analyze the channels on which the communications are distributed.
In order to understand the differences between channels and frequencies, a brief scheme of channels with respective frequencies is presented in Table 5.2. In this

Channel	Frequency(MHz)
1	2412
2	2417
3	2422
4	2427
5	2432
6	2437
7	2442
8	2447
9	2452
10	2457
11	2462
12	2467
13	2472
14	2484

Table 5.2: *Channels and Frequencies.*

study we are using the same topology presented before: it is formed by four Alix Boards, each of them composed by two physical wireless interfaces (wlan0, wlan1). The first part of the analysis was focused on differentiating communications among devices using different channels. Taking in consideration the topology presented in the last paragraph, the relay of the packets is working perfectly using groups set on the same channel. Unlike the previous situation, in WiFi-Direct two different physical interfaces located in the same device cannot use two different channels. We want to use different channels for every P2P groups in the topology. Enabling new groups in a sequential mode, we notice that in Wi-Fi Direct two physical interfaces on the same machine cannot use different channels if one of the two

interfaces act as GO. This is demonstrated by creating a P2P group on a specific interface (wlan0) and specifying the frequency of the chosen channel (channel 6) as:

```
wpa_cli p2p_group_add persistent freq=2437
```

Using this command, the chosen channel is assigned to the correspondent interface of the created group. Creating the P2P group without negotiation, we have that this last device acts as GO on this specific interface.
After the creation of the group, we enable another device in order to make it join the group (as client) through his physical interface (wlan0). So, in device A we have:

```
p2p-wlan1-0  IEEE 802.11abgn  Mode:Auto  Frequency:2.437 GHz  Tx-Power=27 dBm
  Retry  long limit:7   RTS thr:off   Fragment thr:off
  Power Management:on
```

and in device B:

```
  wlan0     IEEE 802.11abgn  ESSID:"DIRECT-9M"
    Mode:Managed  Frequency:2.437 GHz  Access Point: B6:74:9F:DC:20:2C
    Bit Rate=54 Mb/s   Tx-Power=27 dBm
    Retry  long limit:7   RTS thr:off   Fragment thr:off
    Encryption key:off
    Power Management:on
    Link Quality=70/70  Signal level=-22 dBm
    Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
    Tx excessive retries:0  Invalid misc:2   Missed beacon:0
```

Now, activating the second physical interface on device B (wlan1), we create a new P2P group specifying a different frequency: enabling a new group without GO Negotiation allows interface wlan1 to act as GO. The command used in this situation is shown below:

```
wpa_cli -i wlan1 p2p_group_add persistent freq=2462
Selected interface 'wlan1'
FAIL
```

The creation of the new group fails because we inserted a different frequency (2462= channel 11) with respect to the one already set in the other interface. Unlike the situation just described above, it is possible to have different frequencies for each of the two physical interfaces only in these two following cases:

- The device is connected via Wi-Fi Direct in one interface and via Wi-Fi in the other interface.
- The device acts as client of a P2P group in both of the interfaces.

About the first case, it is possible to assign different channels to two different physical interfaces belonging to the same device if and only if one of the two channels is working in Wi-Fi. As a matter of fact, in the previous case both of the channels were used in Wi-Fi Direct.
If we try now to connect one interface (wlan0) of device A to an internet network, we have:

```
wlan0     IEEE 802.11abgn  ESSID:"IMDEANetworks2G"
  Mode:Managed  Frequency:2.437 GHz  Access Point: 18:64:72:0D:74:60
  Bit Rate=52 Mb/s   Tx-Power=20 dBm
  Retry  long limit:7   RTS thr:off   Fragment thr:off
  Encryption key:off
  Power Management:on
```

```
Link Quality=35/70  Signal level=-75 dBm
Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
Tx excessive retries:4  Invalid misc:3   Missed beacon:0
```

Now we create a P2P group in the other physical available interface (wlan1), setting a different channel respect to the one already set up in the first interface. Due to the results of the specific command:

```
wpa_cli p2p_group_add persistent freq=2462
Selected interface 'wlan1'
OK
```

It is so possible to create a P2P group. It has been created a new group with frequency equal to 2,462 GHz (setting the channel to 11). We can observe that in interface wlan0, channel 6 is set. So, it is possible to understand that it is feasible to have two channels on two different physical interfaces if and only if one channel works in Wi-Fi and the other in Wi-Fi Direct.
In the second case, we have a topology shown in Figure 5.9. In this scenario, device

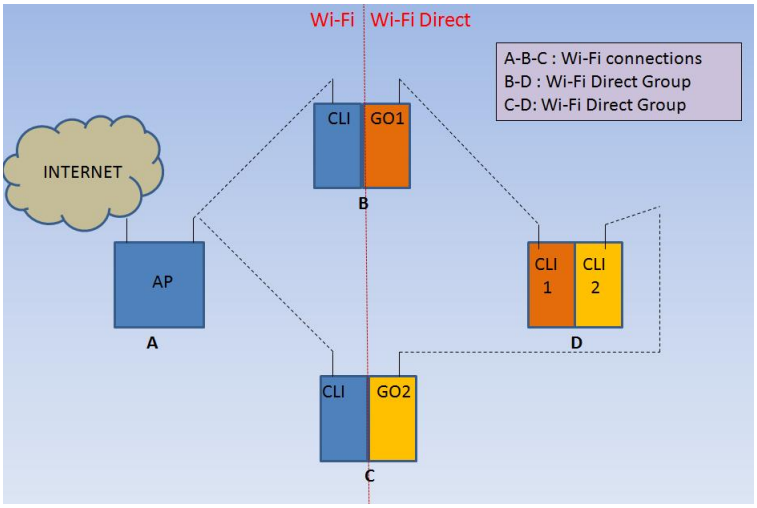

Figure 5.9: *New topology with two different type of connections.*

A is set as AP; devices B and C access to the network through Wi-Fi connections

respectively with the first available interface on the left part (wlan0 in both of the machines). On each wlan1 of the two devices it is created a persistent P2P group with predefined GO. Now we have one P2P group on each wlan1 of both devices. These latter act as GO for their own group.
The next step is to introduce a new device (D): it joins each of the two P2P groups created by B and C, acting as client in both of them. It has two active physical interfaces (wlan0, wlan1): joining the two groups it will have both of the interfaces dedicated to two different groups. Knowing this, we can notice that each group is working normally on two different channels:

```
wlan1     IEEE 802.11abgn  ESSID:"DIRECT-gF"
  Mode:Managed  Frequency:2.447 GHz  Access Point: B6:74:9F:DC:20:13
  Bit Rate=54 Mb/s   Tx-Power=27 dBm
  Retry  long limit:7   RTS thr:off   Fragment thr:off
  Encryption key:off
  Power Management:on
  Link Quality=70/70  Signal level=-20 dBm
  Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
  Tx excessive retries:0  Invalid misc:2   Missed beacon:0

wlan0     IEEE 802.11abgn  ESSID:"DIRECT-EA"
  Mode:Managed  Frequency:2.462 GHz  Access Point: B6:74:9F:DC:25:87
  Bit Rate=54 Mb/s   Tx-Power=27 dBm
  Retry  long limit:7   RTS thr:off   Fragment thr:off
  Encryption key:off
  Power Management:on
  Link Quality=70/70  Signal level=2 dBm
  Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
  Tx excessive retries:0  Invalid misc:7   Missed beacon:0
```

As a matter of fact, we can confirm the thesis presented beforehand: it was specified the possibility of having one device connected with two different channels, both of them in Wi-Fi Direct. The only limitation of this is about the devices roles: in order to be able to connect to different channels simultaneously, devices have to act as clients in each group they belong to.

5.5 Relay of packets with opportunistic method

The last part of this work is focused on Wi-Fi Direct packets relay. More in particular, this new technology can be used in very different ways to set opportunistic network of devices. The main idea is related to connect many devices to the same external network through a specific device. So, if a single device is able to connect to internet, we want to give the same possibility to other devices without being directly connected to the external network. This is possible using Wi-Fi Direct technology: each device that is part of a P2P group can communicate with other devices of other groups if the topology is connected. Each machine can also establish a connection with external network if at least one device of the topology is connected. This is a great opportunity that offers the possibility to connect to internet even if the device is not directly connected.

We consider the scenario already described, with four Alix Boards. In order to study an opportunistic model to relay packets, we use four machines, each of them is composed by two physical interfaces. The topology is composed by a first Alix Board connected to an external wireless network (internet) through one of the two physical interface. The other interface is connected to other two devices through a specific Wi-Fi Direct group. The two devices in the middle of the architecture are connected to three different P2P group: one is formed by the device on the left and the two devices in the middle through their left physical interface. The other two groups are composed by three devices, with one in common, as shown

in Figure 5.10.

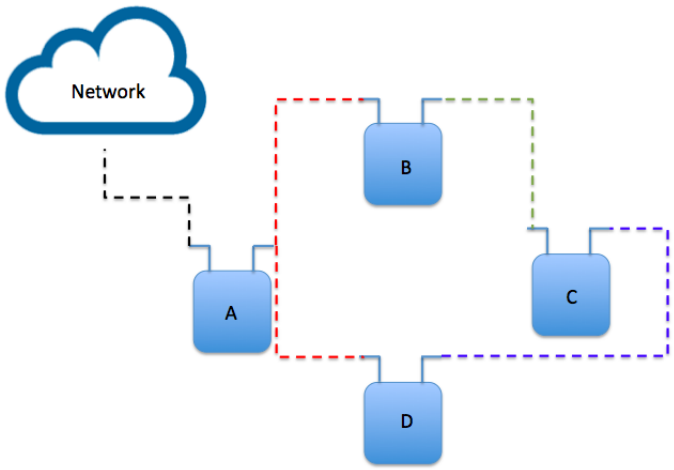

Figure 5.10: *Main topology.*

The goal of this scenario is to have the capability to reach an external network from each device in this topology. The first device on the left can directly reach internet from his physical interface. The other machines have to pass through Alix A in order to reach the external network. So, each device send the packets to its GO, and this latter is in charge of routing the received requests. With this mechanism, all devices can communicate with all the others; moreover, since one device is connected to the external network, all the machines can establish connections with external devices attached to the network. All the connections of this scenario are Wi-Fi Direct connections, so each device can communicate with the others without passing through an AP like in the standard Wi-Fi once. This technology enables any connection with any device and networks in the simplest way.

Once the experiment is done with Wi-Fi Direct connections, we want to try to use two types of connections: Wi-Fi and Wi-Fi Direct. This is possible by using one of the Alix as AP and the others as normal P2P devices. So the structure now is shown in Figure 5.11. The two machines in the middle are directly connected through Wi-Fi connections to the AP in order to be connected to the external net-

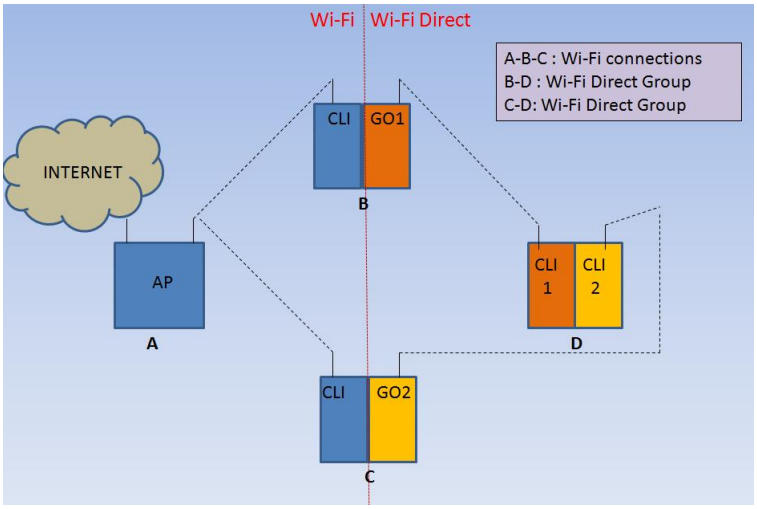

Figure 5.11: *New topology with two different type of connections.*

work. So, since these two have the possibility to connect to internet, also the third P2P device can reach the network by requesting connections to the two central GOs: since the last device is a client for both groups, it can ask the connection to both GOs. This topology recalls the initial idea of a particular scenario in which one device, covering the role of AP, can share internet connection to other devices through the new Wi-Fi Direct technology (Figure 5.12). This is a great idea to be applied in some specific scenarios: one user can have the free access to the external network and it can give this opportunity also to other devices that cannot be directly connected to the network. So, other devices can have the possibility to reach the network without paying a service or by directly connected to it. Indeed, our experiments prove that Wi-Fi Direct is a great technology that can be used to share contents among P2P devices.

5.6 Routing algorithm in opportunistic topology

In this section a channel state based routing algorithm is presented to work in an opportunistic Wi-Fi Direct networking. The algorithm is applied to a specific topology, presented in Figure 5.12 and it is tested to evaluate performance and

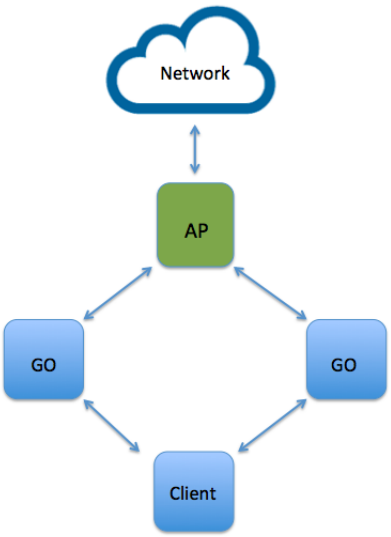

Figure 5.12: *New scenario.*

limits of the approach. Considering the topology in Figure 5.12, the initial idea was to send packets from an AP to a P2P device.

The access point has the possibility to reach the external network. Each device of a specific network or group can ask to the AP a specific content from internet. The request can be done in a direct way if the device is directly connected to the AP, or through other machines if the device is part of a group, but not connected to the AP. If the origin of the request can reach only near devices but not the AP, the data can be asked through GOs, till reaching the AP and finally the network. Taking in consideration the simple topology with four devices (one AP and three P2P devices), we have to think about all the possible problems that can happen while packets are flowing in the network. One of the main common issue is related to the connection of a node in the network. If a node disconnects during the relay of packets there could be some problems reaching the destination. This problem can be caused by breaking of antennas or bad signals among devices. If one of the two antennas of the device disconnects, the signal becomes lower powerful, and a break of connections can rises.

When a device asks for some contents from the network, it has to be sure that the data will arrive to destination. In order to guarantee this goal, a stable connection has to be assured and maintained.
To study and prove all the possible cases of failures many experiments were performed. The first experiment has the goal to evaluate connections by asking contents from a P2P device in the Wi-Fi Direct group to the external network. The network returns the needed data and the packets flow through the AP and the GO. The relay of the packets is working perfectly, so, if the network is stable enough, all the data arrives to destination. Now that is clear that everything is working perfectly, an good experiment to evaluate the stability of the signals is based on deteriorating the signal between two machines. When the signal is low, the communication is worst and the packets take more time to reach destination. When the signal is completely lost, packets cannot arrive to destination and the receiver device won't reach any data.
In order to solve the problem of losses due to low quality connections, an alternative routing algorithm was thought. Since in our topology there is a single device that has the role of client for two different groups, and it belongs to two different Group Owners, all the data transferred to this machine can be sent from two different paths. More in particular it is possible to distinguish two cases in which the packet's path has to be change according to the network status:

- Low signal, bad connection
- Losses of middle devices

According to the first case, the connection between two devices is bad: this can be due to long distances between devices or antennas malfunctioning (Figure 5.13). The second case referred to broken machines in the middle of the network or devices that decide to leave the group in which they belong, as shown in Figure 5.14. In both of the cases, an algorithm is implemented in order to improve the performances of the entire system: the main goal is to send packets to the destination

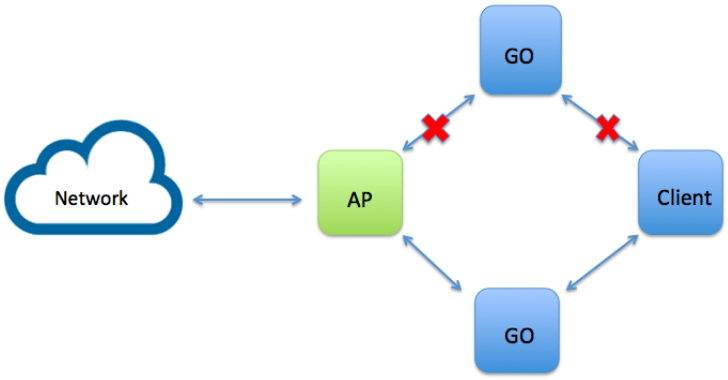

Figure 5.13: *Connections between devices are damaged.*

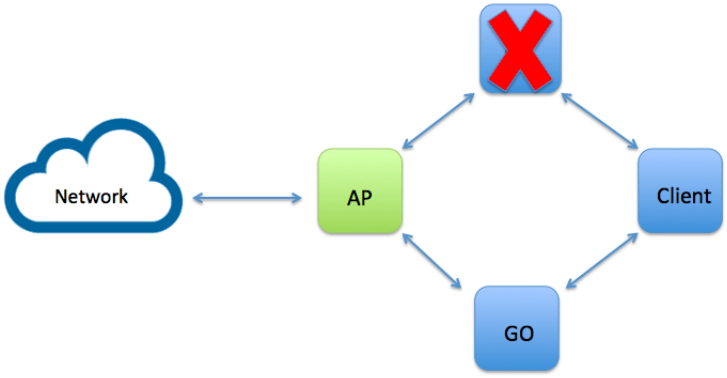

Figure 5.14: *The device breaks or goes out from the group.*

even if some devices in the middle are no more part of the topology or if there are some technical connection problems. Moreover, it gives the possibility to send data through the path with better quality of connection. This feature allows packets to flow through the network always with the best connection. Each time a packet has to be send from a device to the closest one, the network is measured and the quality of connection is evaluated.

Lets taking the topology shown in Figure 5.15 as example for a brief explanation of the algorithm working principles. The first device has to send packets to the client: first of all, it checks if the middle devices are active. It ping both the machines: if both reply, it can measure the quality of the channels. Every few seconds the algorithm is run in the device from which the data have to be transmitted.

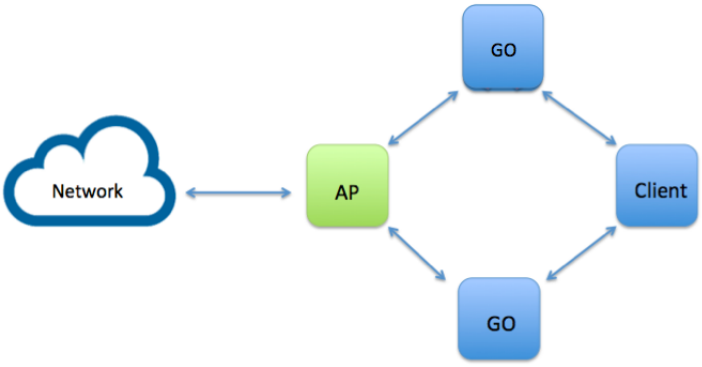

Figure 5.15: *Working topology.*

Thanks to this running periodicity the device can monitor the connection status with a really small gap of delay: so, if a device stops working or the considered connection is less efficient, the packets have to change path. So, if the AP evaluate the upper connection as a bad one, it send data to the lowest one.

In all the experiments, in order to get the signal worst, the antennas are modified and sometimes taken out from the device: in this way the signal begins worst in few seconds and sometimes it gets lost. The algorithm measures the quality of the signal and it verifies the presence of the neighbor machine: if the connection is no more present, the routing table inside the device has to be changed.

In order to change path, the routing tables have to be changed according to the working devices and to the best chosen channel. When the transmitting device reach this two kind of information, the routing table of the specific machine is changed by the algorithm. This method requires some additional milliseconds, nevertheless the algorithm has very good performances and it is very reliable. This new channel state based routing algorithm proves that it is fundamental for avoiding packets losses. Moreover, it improves quality of transmissions, guaranteeing high rate of packet arrival. The innovation of this algorithm is distinguished by choosing the best channel and scanning always the network before a transfer of data has to take place. Scanning near devices, lets the device have information

about near machines: this can really help the algorithm to choose the most efficient path to improve transmission performances.

5.6.1 Throughput analyses

For the sake of evaluating the behavior and the quality of the algorithm, the traffic was studied in a more accurate way, through "tcpdump" program. To have a complete evaluation of the algorithm, two studies were conducted: one on traffic flowing through the network avoiding to use the routing algorithm. The other study analyzes traffic of packets flowing in the network with the support of the new provided algorithm. It is possible to have an idea of algorithm performances looking at the graphical representation of the the two studies. Figure 5.16

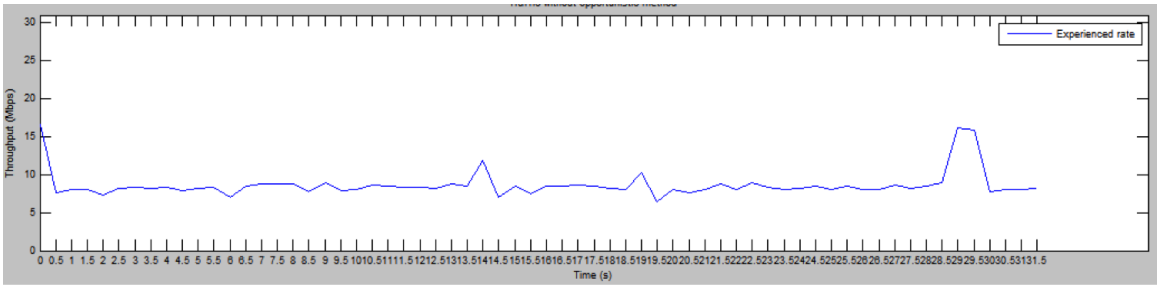

Figure 5.16: *Graphical representation of traffic behavior without opportunistic method.*

shows graphically the experienced rate in relation to time (measured in seconds) and throughput (measured in Mbps). The evolution of the rate is almost stable between 5 and 18 Mbps. The behavior has no relevant changes during all the observed time. Many experiments with different time intervals were tested and they always have shown similar results.

In Figure 5.17 the experienced rate is represented in relation to seconds and Mbps, as the other graph. The results are totally different respect to the other. During the same interval of time the throughput goes in a wide range from 0 to 30 Mbps. This trend shows the efficiency of the algorithm: indeed, the maximum peak occurs when the algorithm find the best channel through which a device can

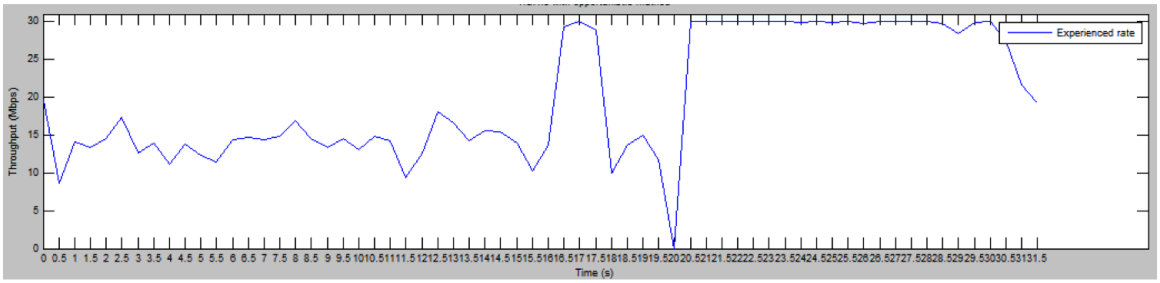

Figure 5.17: *Graphical representation of traffic behavior using the new efficient routing algorithm.*

transmit, so the traffic throughput is the highest. On the contrary, the rate reach zero value when the connection is temporarily lost and the algorithm is looking for the best solution depending on network status.

Comparing the two results shown in Figure 5.16 and Figure 5.17 is clear that traffic analysis with innovative routing algorithm offers best results against the initial throughput.

This traffic study is proving that the algorithm really improves network performances, scanning the network every time a device has to relay packets, for the sake of transmitting always in the best network conditions.

5.7 Results

The experimental results of the working principle of Wi-Fi Direct are very important to improve future studies. More in particular, what was studied and described in this thesis can be used to develop bigger and more complicated architectures.

The initial study on Wi-Fi Direct working principles was done to understand how this new technology works in details: this topic is proposed in many papers and related works but few of them analyzed the details of the Direct communications. The steps of our study were few, but very detailed: first of all we want to understand how the P2P groups can be composed, so the three types of group formations were implemented and tested with different number of Wi-Fi Direct devices. Later, we used persistent group formation to create P2P groups because

it is the most used and with an easy approach. Since we have to study how devices and connections react in different situations, we tried to enable communications among different types of devices. All the trials were performed among more than two devices, to study the behavior of the technology in different situations. When communications were established and the working principles were totally clear and tested, we started to create different topologies in order to study the best one for a specific opportunistic network. The main goal of this study is based on a real scenario with a topology composed by four Alix Board, an external network and a PC. The network scheme is shown in Figure 5.18. The aim of this work

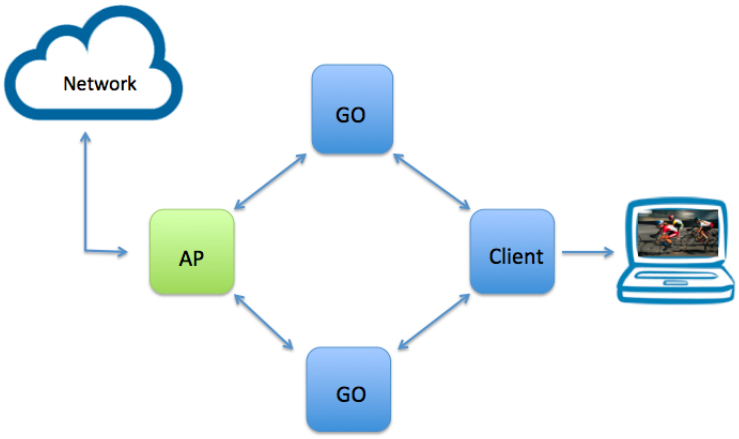

Figure 5.18: *Opportunistic network with video streaming test*

is to create an opportunistic network combined with our new routing algorithm: this enables each device to send, relay and receive packets without loosing any of them. The main goal of the routing algorithm has the purpose of sending packets to destination without loosing data: there could arise many problems related to connections and broken devices, but, thanks to the algorithm, this has no impact on the final aim. Every time a packet has to be sent, the involved device performs a scan of the near devices and connections and it reorganizes the programmed path. The great efficiency of this algorithm overcomes all the possible problems. So, combining an opportunistic network with the algorithm previously presented, and to be sure that these can work perfectly combined together, we tried to ask

contents to the external network from the last device on the right (client of two P2P groups). This device is connected to a PC to have the possibility of visualize on screen the asked contents. In order to be sure that everything is working well, the perfect example to see how the communications work is related to ask to the external network to watch a streaming video, as presented in Figure 5.18. This way, it is possible to monitor the flowing of packets from the origin (internet) to the destination (PC through client device). Combining together this type of topology and the routing algorithm, we had great results: the video is played on the screen without big interruptions and gaps. All the packets arrives to the destination without losses. If one of the middle device stops working, the path is reconfigured and the packets are rerouted through another machine. The reconfiguration of the routing table of the devices is fundamental to avoid losses and leaks on played video.

It was tested the working principles of all the experiments previously presented; this last opportunistic topology can be used in a typical scenario in which a user establishes a contract with a telephone company and it can share the access to the network with other devices, using this new great technology called Wi-Fi Direct. This latter can be use in many situations because it has the novelty of exchanging data among devices without passing through an AP. This new technology, combined with different types of devices topologies, can let arise new opportunistic scenarios very useful to users that want to exchange data and access to internet through other devices in a very simpler way.

The scenario presented before can be reuse in many situations, changing number of devices or typology of connections. All the experiments done in this thesis were made in order to be free to combine different devices and different connections being sure that the opportunistic scenario works every time the users need it.

Chapter 6

Conclusions

Wi-Fi Direct is a new technology defined by the Wi-Fi Alliance aimed to enhance direct device to device communications in Wi-Fi. The work of thesis was focused on develop a wireless network using IEEE 802.11g standard and Wi-Fi Direct technology to manage dynamic groups of users which communicate using Device-to-Device(D2D) paradigm. P2P groups are created and modified depending on how transmission channels change; this can be caused i.e. by users mobility.

The first part of this work was dedicated to study protocols and features of D2D communications. In the second part of the work it was created a specific network in which mobile users can decide to join one or more groups simultaneously. Each device has the capability of sending, forwarding or receiving packets; thanks to an opportunistic network combined with a specific routing algorithm, data flows through the network depending on some associated mechanisms. Every time a packet has to be sent from a device to another, the channel quality and the near devices are checked: thanks to this periodic study of the network, the packets can change paths according to the network updated behavior. This mechanism was performed in order to be sure that connections are stable and channels have always the best quality.

Moreover, to manage traffic exchanged among P2P devices and an external network (Internet), we developed tools and scripts to control the flowing of the packets; in addition there were created a routing algorithm to avoid packets losses and

to change routing tables on each device belonging to the network.
The work was made possible by using Alix Board as Wi-Fi Direct devices. Thanks to these tools we were able to develop an experimental testbed with four wireless nodes. This topology was used to reproduce a real scenario with an opportunistic relay system, through which data were sent from a source node to a destination, situated in a D2D group. This kind of experiment gave us the possibility to understand the real behavior of Wi-Fi Direct connections. Indeed, when a device wants to receive data from an external network, the packets pass through many devices before arriving to destination. All the packets arrive to destination without losses: this is possible thanks to our new channel state based routing algorithm combined with the opportunistic network. The algorithm is installed in all the P2P machines and it lets packets flowing through the channel with the best quality and through safe forwarding nodes.
This work gives us the possibility to adapt the opportunistic network, created and tested in a testbed system, to a real scenario in which people is free to move and exchange data without loosing part of it. Devices can communicate in a direct way thanks to Wi-Fi Direct technology. Moreover they can communicate with an external network even if they are not directly connected to it. This is made possible by the routing of the packets through near P2P devices belonging to the same or to a different D2D group. In addition to these important features, this opportunistic network could be used in future works to enlarge topologies related to mobile internet contracts: users could have the possibility to share connections with other enabled devices.
This study, although being designed and considered as a starting point for future developments, is already able to provide valuable practical advice for future works in this field.

Chapter 7

Appendix

7.1 SSH login without password

Therefore you need an automatic login from computer acting as user A to Alix board with the role of user b. The requirements of executing this ssh within a shell script force this operation to be without passwords. In order to be able to enter in the machines without passwords, the following steps are needed:

- log in on A as user a and generate a pair of authentication keys without enter a passphrase:

```
a@A:~> ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/a/.ssh/id_rsa):
Created directory '/home/a/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/a/.ssh/id_rsa.
Your public key has been saved in /home/a/.ssh/id_rsa.pub.
The key fingerprint is:
3e:4f:05:79:3a:9f:96:7c:3b:ad:e9:58:37:bc:37:e4 a@A
```

- Now use ssh to create a directory /.ssh as user b on B. (The directory may already exist):

```
a@A:~> ssh b@B mkdir -p .ssh
b@B's password:
```

- Finally append as new public key to b@B:.ssh/authorized_keys and enter b's password one last time:

```
a@A:~> cat .ssh/id_rsa.pub | ssh b@B 'cat >> .ssh/authorized_keys'
b@B's password:
```

- From now on it is possible to log into B as b from A as a without password:

```
a@A:~> ssh b@B
```

Depending on your version of SSH you might also have to do the following changes:

- Put the public key in .ssh/authorized_keys2
- Change the permissions of .ssh to 700
- Change the permissions of .ssh/authorized_keys2 to 640

contensources: http://www.linuxproblem.org/art_9.htmlt...

7.2 Opportunistic Method

Referring to the specific topology composed by four devices and two types of connections (Wi-Fi and Wi-Fi Direct), it is possible to send packets from device A to device D, passing through B and D following a simple routing scheme.
Instead of changing routes when a routing-device is broken or it is out of the network, we want to implement an opportunistic system in which the AP (device A) obtains channels quality values from devices attached to it. This is possible because both of peer B and C scan the channel between them and the AP. From this type of scan, we can obtain the information shown below:

```
bssid / frequency / signal level / flags / ssid
b4:74:9f:dc:2d:aa 2422 -71 [WPA-PSK-TKIP][WPA2-PSK-CCMP][ESS] giulia
```

Each of these two devices saves the signal level of the channel and it sends it to the AP which compares the values and evaluates the best channel among them. Choosing the best channel means that the machine A is setting a specific device for the relay of the packets.
In order to choose on which channel is better to transmit, it is possible to implement an opportunistic method:

CONTROL CHANNEL QUALITY ON 120

```
x=0
echo machine 120
while [ "100" -gt $x ]
do
#scp root@192.168.97.111:/root/final_signal_level1.txt /root/
#scp root@192.168.97.115:/root/final_signal_level2.txt /root/
#v1=$(cat final_signal_level1.txt | grep -oP '\d+')
#v2=$(cat final_signal_level2.txt | grep -oP '\d+')
```

```
vv1=$(cat final_signal_level1.txt)
sleep 1
vv2=$(cat final_signal_level2.txt)
v1=$(echo "$vv1" | bc -l)
v2=$(echo "$vv2" | bc -l)
echo Values of 111 and 115: $v1 $v2
if [ $(echo "$v1>$v2" | bc ) == 1 ] ; then
        echo "higher channel quality : 111"
        route del -net 192.168.2.0/24 gw 192.168.0.21
        route add -net 192.168.2.0/24 gw 192.168.0.20
        route del -net 192.168.1.0/24 gw 192.168.0.21
        route add -net 192.168.1.0/24 gw 192.168.0.20

else
        echo "higher channel quality : 115"
        route del -net 192.168.1.0/24 gw 192.168.0.20
        route add -net 192.168.1.0/24 gw 192.168.0.21
        route del -net 192.168.2.0/24 gw 192.168.0.20
        route add -net 192.168.2.0/24 gw 192.168.0.21
fi
done
```

PRINCIPAL TEST ON COMPUTER

```
#!/bin/bash
x=0
a=0.9
b=$(echo "1-$a" |bc -l)
y=0
z=0
```

```
#timestamp="$(date +"%T")"
while [ "15" -gt $x ]
do
     #while [ "5" -gt $x ]
     #do
         #MACHINE 111
         ssh root@192.168.97.111 ./check_quality_channel.sh &

         #ssh root@192.168.97.111 wpa_cli -i wlan0 scan
         #ssh root@192.168.97.111 wpa_cli -i wlan0 scan_results | grep giulia
          | awk '{print($3)}' > signal_level1.txt
         #ssh root@192.168.97.111 value1=$(cat signal_level1.txt
         | grep -oP '\d+')
         #y=($a*$value) + (1-$a)*$y
         #ssh root@192.168.97.111 s=$(echo "$a*$value1" |bc)
         #ssh root@192.168.97.111 r=$(echo "$b*$y" |bc)
         #ssh root@192.168.97.111 y=$(echo "$r+$s" |bc)
         #ssh root@192.168.97.111 echo $y > final_signal_level1.txt
         #MACHINE 115
         ssh root@192.168.97.115 ./check_quality_channel.sh
         #ssh root@192.168.97.115        wpa_cli -i wlan0 scan
         #ssh root@192.168.97.115 wpa_cli -i wlan0 scan_results | grep giulia
         | awk '{print($3)}' > signal_level2.txt
         #ssh root@192.168.97.115 value2=$(cat signal_level2.txt |
         grep -oP '\d+')
            #y=($a*$value) + (1-$a)*$y
            #ssh root@192.168.97.115 s=$(echo "$a*$value2" |bc)
            #ssh root@192.168.97.115 r=$(echo "$b*$y" |bc)
```

```
            #ssh root@192.168.97.115 y=$(echo "$r+$s" |bc)
            #ssh root@192.168.97.115 echo $y > final_signal_level2.txt
        #done
#ssh root@192.168.97.120 scp root@192.168.97.111:/root/
  final_signal_level1.txt /root/
#ssh root@192.168.97.120 scp root@192.168.97.115:/root/
  final_signal_level2.txt /root/
#ssh root@192.168.97.120 v1=$(cat final_signal_level1.txt | grep -oP '\d+')
#ssh root@192.168.97.120 v2=$(cat final_signal_level2.txt | grep -oP '\d+')

scp root@192.168.97.111:/root/final_signal_level1.txt /home/imdea/Desktop &
scp root@192.168.97.115:/root/final_signal_level2.txt /home/imdea/Desktop
scp /home/imdea/Desktop/final_signal_level1.txt root@192.168.97.120:/root/
scp /home/imdea/Desktop/final_signal_level2.txt root@192.168.97.120:/root/
#ssh root@192.168.97.120 ./control_channel_function.sh
x=$(($x + 1))
done
```

HOSTAPD.CONF in AP 120

```
#sets the wifi interface to use.
interface=wlan0
#driver to use, nl80211 works in most cases
driver=nl80211
#sets the ssid of the virtual wifi access point
ssid=giulia
#sets the mode of wifi, depends upon the devices
you will be using. It can be a,b,g,n. Setting to g ensures backwar
hw_mode=g
#sets the channel for your wifi
```

```
channel=1
#macaddr_acl sets options for mac address filtering.
0 means "accept unless in deny list"
macaddr_acl=0
#setting ignore_broadcast_ssid to 1 will disable the broadcasting of ssid
ignore_broadcast_ssid=0
#Sets authentication algorithm
#1 - only open system authentication
#2 - both open system authentication and shared key authentication
auth_algs=1
#####Sets WPA and WPA2 authentication#####
#wpa option sets which wpa implementation to use
#1 - wpa only
#2 - wpa2 only
#3 - both
wpa=3
#sets wpa passphrase required by the clients to authenticate
themselves on the network
wpa_passphrase=password
#sets wpa key management
wpa_key_mgmt=WPA-PSK
#sets encryption used by WPA
wpa_pairwise=TKIP
#sets encryption used by WPA2
rsn_pairwise=CCMP
```

7.3 Setting up an Alix Board

7.3.1 The easy way

The easiest way would be to find a working machine and copy one CF card into another using *dd* command. The format of the command is:
 # dd if=/dev/sdb1 of=image.img
 # dd if=image.img of=/dev/sdb1

7.3.2 Everything from scratch

To setup an ubuntu on alix from scratch we use the following tutorials:

- `https://wiki.wsartori.com/wiki/Installing_Ubuntu_Linux_on_an_ALIX_2D13`
- `http://thindot.blogspot.com.es/2011/01/howto-install-ubuntu-on-cf-card-for.html`
- `http://www.youtube.com/watch?v=6VPsgR4pMik`

In what follows we describe the steps in order to setup Ubuntu 12.04 LTS on an alix pc-engine.

7.3.2.1 Partition the CF card

- fdisk /dev/sdc
  Commands:
  - In the first step, digit : d
  - In the second step, digit: n
    - Select: p
    - Select: 1
  - Choose: a

        * Than: 1

    - Finally, digit: w

- sudo mkfs.ext2 -L root /dev/sdc1

7.3.2.2 Format the CF card

7.3.2.3 Creation of a directory used for mounting

- sudo mkdir /mnt/alix

7.3.2.4 Mount: attach the file system on the device (/dev/sdc1) at the directory created

- sudo mount /dev/sdc1 /mnt/alix

7.3.2.5 Install Ubuntu (Precise Version)

- sudo debootstrap –arch=i386 –variant=buildd precise /mnt/alix http:// archive.ubuntu.com/ubuntu/

7.3.2.6 Mount....

- sudo mount -o bind /proc /mnt/alix/proc
- sudo mount -o bind /sys /mnt/alix/sys
- sudo mount -o bind /dev /mnt/alix/dev

7.3.2.7 Copy resolv.conf and enter as root

- sudo cp /etc/resolv.conf /mnt/alix/etc/resolv.conf
- sudo chroot /mnt/alix /bin/bash

7.3.2.8 Install vim

apt-get install vim

7.3.2.9 Modify source.list file

- source /etc/profile
- vim /etc/apt/source.list

  deb http://archive.ubuntu.com/ubuntu precise universe multiverse

  deb-src http:// archive.ubuntu.com/ubuntu precise universe multiverse

7.3.2.10 Install packages

- apt-get install language-pack-en-base
- apt-get install linux-image-generic
  - more:
  - GRUB install devices:2

7.3.2.11 Middle passages

- echo proc /proc proc nodev,noexec,nosuid 0 0 > /etc/fstab
- echo UUID=‘dumpe2fs /dev/sdc1 | grep UUID | awk ’print $3’‘ / ext2 defaults,noatime 0 1 » /etc/fstab

7.3.2.12 Modify grub file

- vim /etc/default/grub

  modify: GRUB_CMDLINE_LINUX_DEFAULT="verbose console=ttyS0,138400n8 rebo

  add: GRUB_SERIAL_COMMAND="serial –unit=0 –speed=38400"

  modify: GRUB_TERMINAL=serial

7.3.2.13 Remove and create file configuration

- rm /etc/init/tty?.conf

- vim /etc/init/ttyS0.conf

put inside:

```
# ttyS0 - getty
#
# This service maintains a getty on ttyS0 from the
point the system is
# started until it is shut down again.

start on stopped rc RUNLEVEL=[2345]
stop on runlevel [!2345]

respawn
exec /sbin/getty -8 38400 -L ttyS0
```

10cm

7.3.2.14 Modify hostname and hosts

- vim /etc/hostname
- vim /etc/hosts

7.3.2.15 Install packages

- apt-get install rsyslog
- apt-get install sudo

7.3.2.16 Change password of the root

- passwd root

7.3.2.17 Install and update grub

- grub-install /dev/sdc
- update-grub

7.3.2.18 Enter in grub.cfg and modify

- chmod 644 /boot/grub/grub.cfg
- vim boot/grub/grub.cfg Substitution of (hd1,msdos1) in (hd0,0):

- :%s/hd1,msdos1/hd0,0/gc

- chmod 444 /boot/grub/grub.cfg

7.3.2.19 Install packages

- apt-get install ssh
- apt-get install iputils-arping
- apt-get install ubuntu-minimal

7.3.2.20 Update

- update-rc.d ssh defaults

7.3.2.21 Exit from the root

- exit

7.3.2.22 Umount the directory mounted at the beginning

- umount -l /mnt/alix

7.4 Setting up *wpa-supplicant.*

Install wpa-supplicant first as follows:

```
# apt-get install wpasupplicant
```

Then, from here we can follow two methods:

7.4.1 Method I

- Create the configuration file (say wpa_supplicant.conf) in **/etc/wpa_supplicant**. The content of the file for wpa-personal is as follows:

```
ctrl_interface=/var/run/wpa_supplicant
network={
        ssid="melocoton"
        proto=RSN
        key_mgmt=WPA-PSK
        pairwise=CCMP TKIP
        group=CCMP TKIP
        psk="password"
}
```

- Modify the *interface* in **/etc/network** ,which is the configuration file for different network interfaces, as follows:

```
auto wlan0
iface wlan0 inet dhcp
wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

7.4.2 Method II

- we can include the following lines in **/etc/rc.local**

```
wpa_supplicant -B -i wlan0 -c /etc/wpa_supplicant/wpa_supplicant.conf
sleep 0.5s
dhclient wlan0
```

7.5 Configuration file setup

7.5.1 Using multiple virtual interfaces for concurrent usage

If the driver advertises support, wpa_supplicant will automatically create secondary P2P interfaces. To force this without the driver advertising support, one can add the following to the config file:

driver_param=use_p2p_group_interface=1

When this is added, start the supplicant normally on wlan0 like above. Then, when P2P negotiation finishes, it will create a new interface for the group (called "p2p-wlan0-0") and put it into the appropriate mode (GO or P2P client).

7.5.2 Something about persistent groups

wpa_cli uses a configure file in which the persistent group information will be recorded. We have an example of what kind of information are recorded in the file:

```
network={
        ssid="DIRECT-BU"
        bssid=02:90:4c:02:3b:9f
        psk="cZpEMHRO"
        proto=RSN
        key_mgmt=WPA-PSK
        pairwise=CCMP
        auth_alg=OPEN
        mode=3
        disabled=2
        p2p_client_list=da:50:e6:03:86:20
}
```

When wpa_cli start up, it will load it automatically to get ready for re-invoking a persistent group. we also can use wpa_cli list_networks to find that, and use remove_network to remove it.

7.6 Setup a dhcp server.

- First install the dhcp3-server package.

```
apt-get install dhcp3-server
```

- Edit the configuration file in **/etc/dhcp3/dhcpd.conf** as follows:

```
ddns-update-style none;
ignore client-updates;
authoritative;
option local-wpad code 252 = text;

subnet
10.0.0.0 netmask 255.255.255.0 {
# --- default gateway
option routers
10.0.0.1;
# --- Netmask
option subnet-mask
255.255.255.0;
# --- Broadcast Address
option broadcast-address
10.0.0.255;
# --- Domain name servers, tells the clients which DNS servers to use.
option domain-name-servers
10.0.0.1, 8.8.8.8, 8.8.4.4;
option time-offset
0;
range 10.0.0.3 10.0.0.13;
default-lease-time 1209600;
max-lease-time 1814400;
}
```

- We should also make some changes in the interface configuration file in /**etc/network/interface**.

```
auto wlan1
iface wlan1 inet static
address 10.0.0.1
netmask 255.255.255.0
hostapd /root/hostapd.conf
```

- The last line causes the hostapd to run automatically on startup.
- If the dhcp is not up, the following will start the dhcp service.

```
/etc/init.d/dhcp3-server start
```

7.7 Hostapd setup.

The first step is installing hostapd. (Do not leave empty lines and spaces).

```
apt-get install hostapd
```

After installing hostapd, create a configuration file as below:

```
#sets the wifi interface to use.
interface=wlan1
#driver to use, nl80211 works in most cases
driver=nl80211
#sets the ssid of the virtual wifi access point
ssid=melocoton
#sets the mode of wifi, depends upon the devices you will be using.
It can be a,b,g,n. Setting to g ensures backward compatiblity.
```

```
hw_mode=g
#sets the channel for your wifi
channel=6
#macaddr_acl sets options for mac address filtering.
0 means "accept unless in deny list"
macaddr_acl=0
#setting ignore_broadcast_ssid to 1 will disable the broadcasting of ssid
ignore_broadcast_ssid=0
#Sets authentication algorithm
#1 - only open system authentication
#2 - both open system authentication and shared key authentication
auth_algs=1
#####Sets WPA and WPA2 authentication#####
#wpa option sets which wpa implementation to use
#1 - wpa only
#2 - wpa2 only
#3 - both
wpa=3
#sets wpa passphrase required by the clients to authenticate
themselves on the network
wpa_passphrase=password
#sets wpa key management
wpa_key_mgmt=WPA-PSK
#sets encryption used by WPA
wpa_pairwise=TKIP
#sets encryption used by WPA2
rsn_pairwise=CCMP
```

Bibliography

[1] Wi-Fi Direct. http://www.wi-fi.org/knowledge-center/faq/how-far-does-a-wi- fi-direct-connection-travel.

[2] Wi-Fi Direct setup work. http://www.wi-fi.org/knowledge-center/faq/how-does-wi-fi-protected-setup-work.

[3] wpa_supplicant/ hostapd. http://w1.fi/wpa_supplicant/devel/index.html.

[4] A survey on device-to-device communication in cellular networks. *Communications Surveys & Tutorials, IEEE (Volume:16 , Issue: 4)*, 2014.

[5] Wi-Fi Alliance. Wi-fi protected setup specification. Technical report, December 2006.

[6] Feng Li Bingwei Liu Ari Hadiks, Yu Chen. A study of stealthy denial-of-service attacks in wi-fi direct device-to-device networks. 2012.

[7] Zhipeng Cai, Chaokun Wang, Siyao Cheng, Hongzhi Wang, and Hong Gao, editors. *Wireless Algorithms, Systems, and Applications - 9th International Conference, WASA 2014, Harbin, China, June 23-25, 2014. Proceedings*, volume 8491 of *Lecture Notes in Computer Science*. Springer, 2014.

[8] ANDRES GARCIA-SAAVEDRA Daniel Camps-Mur, Nec Network LABORATORIES and PABLO SERRANO. Device-to-device communications with wifi direct: Overview and experimentation. 2013.

[9] Sebastia Sallent-Ribes Daniel Camps-Mur, Xavier Perez-Costa. Designing energy efficient access points with wi-fi direct. 2011.

[10] A. Farrel, A. Satyanarayana, A. Iwata, N. Fujita, and G. Ash. Crankback Signaling Extensions for MPLS and GMPLS RSVP-TE. RFC 4920 (Proposed Standard), July 2007.

[11] R. H. Kravets. A study of stealthy denial-of-service attacks in wi-fi direct device-to-device networks.

[12] Giovanni Minutiello Roberta Paris Marco Conti, Franca Delmastro. Experimenting opportunistic networks with wifi direct. 2013.

[13] Theus Hossmann Karin Anna Hummel Sascha Trifunovic, Andreea Picu. Slicing the battery pie: Fair and efficient energy usage in device-to-device communication via role switching.

[14] Wi-Fi Alliance. Wi-Fi CERTIFIED wi-fi direct. Technical report, October 2010.