Device to device communications with WiFi Direct: overview and experimentation

Daniel Camps-Mur, Andres Garcia-Saavedra and Pablo Serrano

***Abstract*—Wi-Fi Direct is a new technology defined by the Wi-Fi Alliance aimed at enhancing direct device to device communications in Wi-Fi. Thus, given the wide base of devices with Wi-Fi capabilities, and the fact that it can be entirely implemented in software over traditional Wi-Fi radios, this technology is expected to have a significant impact. In this paper we provide a thorough overview of the novel functionalities defined in Wi-Fi Direct, and present an experimental evaluation that portrays the performance to be expected in real scenarios. In particular, our results quantify the delays to be expected in practice when Wi-Fi Direct devices discover each other and establish a connection, and the performance of its novel power saving protocols. To the best of the authors' knowledge this is the first paper in the state of the art that provides a wide overview and experimental evaluation of Wi-Fi Direct.**

***Index Terms*—Wi-Fi Direct, Group Formation, Power Saving**

I. Introduction

More than a decade after its initial design, the IEEE 802.11 standard [1], has become one of the most common ways to access the Internet. However, to continue with its striking success the Wi-Fi technology needs to evolve and embrace a larger set of use cases. Given the wide adoption of Wi-Fi in many kinds of devices, a natural way for the technology to progress is to target device to device connectivity, i.e. without requiring the presence of an Access Point (AP), traditionally provided by other technologies [3]. This is the purpose of the Wi-Fi Direct technology that has been recently developed by the Wi-Fi Alliance [4].

Direct device to device connectivity was already possible in the original IEEE 802.11 standard by means of the *ad-hoc* mode of operation. However this never became widely deployed in the market and hence presents several drawbacks when facing nowadays requirements, e.g. lack of efficient power saving support or extended QoS capabilities. Another relevant technology in the Wi-Fi device to device communications space is 802.11z, also known as Tunneled Direct Link Setup (TDLS) [2], which enables direct device to device communication but requires stations to be associated with the same AP.

Unlike the previous technologies, the Wi-Fi Direct technology takes a different approach to enhance device to device 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. Thus, legacy Wi-Fi devices may seamlessly connect to Wi-Fi Direct devices (as explained in detail in Section II). By taking this decision, Wi-Fi Direct immediately inherits all the enhanced QoS, power saving, and security mechanisms, e.g. [5] and [6], developed for the Wi-Fi infrastructure mode in the past years.

Wi-Fi Direct being a recent specification, its deployment is still on a very early stage. There is however an initial open source implementation, available in [7], which we have used to evaluate experimentally the performance of this technology in realistic scenarios. In the following, we review the main contributions of the paper.

Firstly, we provide an overview of the Wi-Fi Direct specification, focusing on its novel functionalities and illustrating three representative group formation procedures. We then present an experimental evaluation of the delay associated with these group formations, based on a popular open source implementation of Wi-Fi Direct [7], and compare its performance against the expected figures from simulations, discussing the observed discrepancies. Finally, we implement a novel power saving protocol (the "Notice of Absence") and assess the resulting performance trade-offs in various realistic environments.

This paper is organized as follows. Section II provides a detailed overview of Wi-Fi Direct. Section III presents an experimental evaluation of Wi-Fi Direct that analyses the performance of its group formation procedures and of its power saving protocols. Finally, Section IV concludes the paper.

II. Wi-Fi Direct: A Technical Overview

In a typical Wi-Fi network, clients discover and associate to WLANs, which are created and announced by Access Points (APs). In this way, a device unambiguously behaves either as an AP or as a client, each of these roles involving a different set of functionality. A major novelty of Wi-Fi Direct is that these roles are specified as *dynamic*, and hence a Wi-Fi Direct device has to implement both the role of a client and the role of an AP (sometimes referred to as *Soft-AP*). These roles are therefore *logical* roles that could even be executed simultaneously by the same device, for instance by using different frequencies (if the device has multiple physical radios) or time-sharing the channel through virtualization techniques (similarly to [14] or [15]). In order to

D. Camps-Mur is with NEC Network Laboratories in Heidelberg, Germany.

A. Garcia-Saavedra and Pablo Serrano are with Univ. Carlos III de Madrid, Spain.

establish a communication, then, P2P devices have to agree on the role that each device will assume. In the following we describe how this communication is configured, by first introducing the general architecture and then summarizing the main specified procedures, namely device discovery, role negotiation, service discovery, security provisioning and power saving.

A. Architecture

Wi-Fi Direct devices, formally known as *P2P Devices*, communicate by establishing *P2P Groups*, which are functionally equivalent to traditional Wi-Fi infrastructure networks. The device implementing AP-like functionality in the P2P Group is referred to as the *P2P Group Owner (P2P GO)*, and devices acting as clients are known as *P2P Clients*. Given that these roles are not static, when two P2P devices discover each other they *negotiate*[1] their roles (P2P Client and P2P GO) to establish a P2P Group. Once the P2P Group is established, other P2P Clients can join the group as in a traditional Wi-Fi network. Legacy clients can also communicate with the P2P GO, as long as they are not 802.11b-only devices and support the required security mechanisms (see Section II-D) . In this way, legacy devices do not formally belong to the P2P Group and do not support the enhanced functionalities defined in Wi-Fi Direct, but they simply "see" the P2P GO as a traditional AP.[2]

The logical nature of the P2P roles supports different architectural deployments, two of them illustrated in Figure 1. The upper part of the figure represents a scenario with two P2P groups. The first group is created by a mobile phone sharing its 3G connection with two laptops; for this first group, the phone is acting as P2P GO while the two laptops behave as P2P Clients. In order to extend the network, one of the laptops establishes a second P2P Group with a printer; for this second group, the laptop acts as P2P GO. In order to act both as P2P Client and as P2P GO the laptop will typically alternate between the two roles by time-sharing the Wi-Fi interface; in Section II-E we will introduce the NoA protocol that can be used for this purpose. The lower part of Figure 1 illustrates the case of a laptop accessing the Internet through a legacy infrastructure AP while at the same time streaming content to a TV set by establishing a P2P Group, where the laptop acts as P2P GO (see [13] for more illustrative examples).

Like a traditional AP, a P2P GO announces itself through beacons, and has to support power saving services for its associated clients. The P2P GO is also required to run a Dynamic Host Configuration Protocol (DHCP) server to provide P2P Clients with IP addresses. In addition, only the P2P GO is allowed to cross-connect the devices in its P2P Group to an external network (e.g., a 3G network or an infrastructure WLAN as shown in Figure 1), and for this cross-connection bridging is not allowed. Therefore,

[1]This step might be avoided in some cases as it will be explained in Section II-B.

[2]Most P2P functionality is deployed through the P2P Information Element included in management frames and through novel action frames. Legacy devices ignore these information elements and action frames.

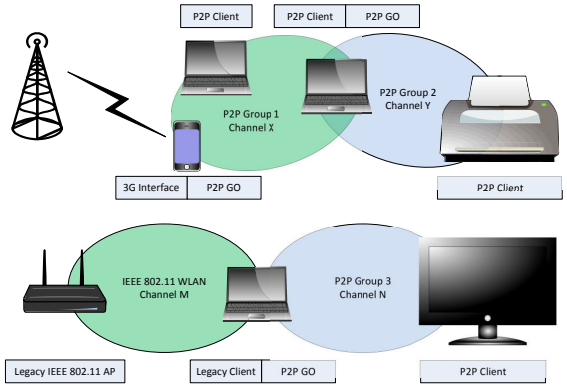


Fig. 1. Example of Wi-Fi Direct supported topologies and use cases.

the connection must be done at the network layer, typically implemented using Network Address Translation (NAT).

Finally, Wi-Fi Direct does not allow *transferring* the role of P2P GO within a P2P Group. In this way, if the P2P GO leaves the P2P Group then the group is torn down, and has to be re-established using some of the specified procedures.

B. Group Formation

There are several ways in 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. Here we first describe the most complex case, which we denote as the *Standard* case, to afterwards highlight a couple of simplified cases which we denote as the *Autonomous* and *Persistent* cases. These three group formation cases are illustrated in Figure 2.

1) Standard: In this case the P2P devices have first to discover each other, and then negotiate which device will act as P2P GO. Wi-Fi Direct devices usually start by performing a traditional Wi-Fi scan (active or passive), by means of which they can discover existent P2P Groups[3] and Wi-Fi networks. After this scan, a new **Discovery** algorithm is executed, which we describe next. First, a P2P Device selects one of the so-called *Social* channels, namely channels 1, 6 or 11 in the 2.4 Ghz band, as its *Listen* channel. Then, it alternates between two states: a *search* state, in which the device performs active scanning by sending Probe Requests in each of the social channels; and a *listen* state, in which the device listens for Probe Requests in its listen channel to respond 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 each 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. An example operation of this discovery algorithm is illustrated in the first part of Figure 2.

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*

[3]Passive scanning should be used to discover a P2P GO that is sleeping (see Section II-E).

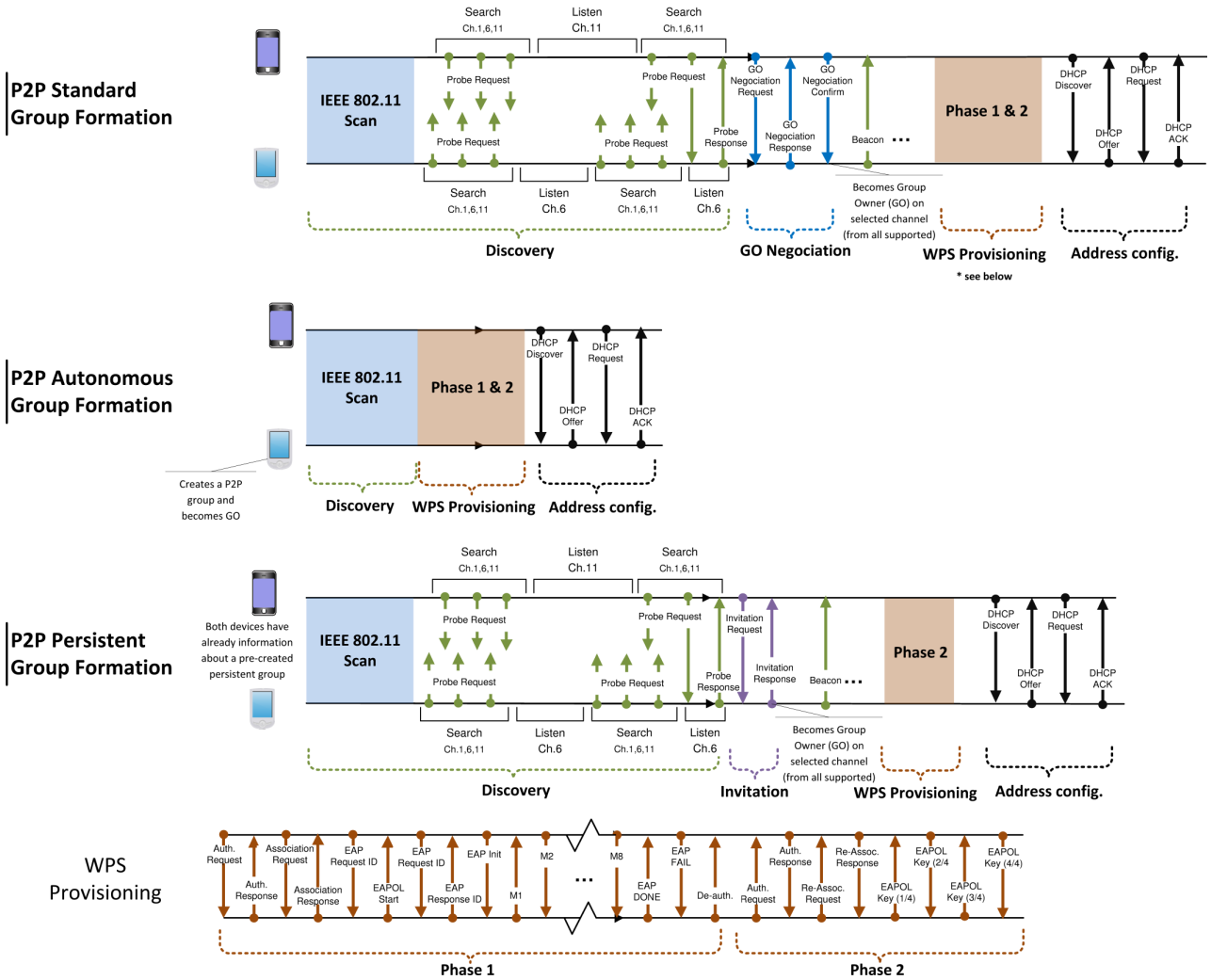


Fig. 2. Typical frame exchange sequences in the *Standard*, *Autonomous* and *Persistent* group formation procedures.

Request/Response/Confirmation, whereby the two devices agree on which device will act as P2P GO and on the channel where 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, the *GO Intent* value, within the three-way handshake, and the device declaring the highest value becomes the P2P GO. 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.

Once the devices have discovered each other and agreed on the respective roles, the next phase is the establishment of a secure communication using *Wi-Fi Protected Setup*, which we denote as **WPS Provisioning** phase and will be described in Section II-D, and finally a DHCP exchange to set up the IP configuration (the **Address config.** phase in the figure).

2) Autonomous: A P2P Device may autonomously create a P2P Group, where it immediately becomes the P2P GO, by sitting on a channel and starting to beacon. 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, then, 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. An exemplary frame exchange for this case is illustrated in the middle part of Figure 2.

3) Persistent: During the formation process, P2P devices can declare a group as *persistent*, by using a flag in the P2P Capabilities attribute present in Beacon frames, Probe Responses and GO negotiation frames. In this way, the devices forming the group store network credentials and the assigned P2P GO and Client roles for subsequent re-instantiations of the P2P group. Specifically, after the Discovery phase, if a P2P Device recognizes to have formed a persistent group with the corresponding peer in the past, any of the two P2P devices can use the *Invitation Procedure* (a two-way handshake) to quickly re-instantiate the group. This is illustrated in the lower part of Figure 2, where the *Standard* case is assumed as baseline, and the GO Negotiation phase is replaced by the invitation exchange,

and the WPS Provisioning phase is significantly reduced because the stored network credentials can be reused.

C. *Layer Two Service Discovery*

A salient feature of Wi-Fi Direct is the ability to support service discovery at the link layer. In this way, *prior* to the establishment of a P2P Group, P2P Devices can exchange queries to discover the set of available services and, based on this, decide whether to continue the group formation or not. Notice that this represents a significant shift from traditional Wi-Fi networks, where it is assumed that the only service clients are interested in is Internet connectivity.

In order to implement the above, service discovery queries generated by a higher layer protocol, e.g., UPnP or Bonjour [8], are transported at the link layer using the *Generic Advertisement Protocol (GAS)* specified by 802.11u [9]. GAS is a layer two query/response protocol implemented through the use of public action frames, that allows two non-associated 802.11 devices to exchange queries belonging to a higher layer protocol (e.g. a service discovery protocol). GAS is implemented by means of a generic container that provides fragmentation and re-assembly, and allows the recipient device to identify the higher layer protocol being transported. The interested reader is referred to [4] for a detailed description.

D. *Security*

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)* [6] 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.

Following WPS terminology, the P2P GO is required to implement an internal *Registrar*, and the P2P Client is required to implement an *Enrollee* [6]. The operation of WPS is composed of two parts. In the first part, denoted as "Phase 1" in the lower part of Figure 2, the internal Registrar is in charge of generating and issuing the network credentials, i.e., security keys, to the Enrollee. WPS is based on WPA-2 security and uses Advanced Encryption Standard (AES)-CCMP as cypher, and a randomly generated Pre-Shared Key (PSK) for mutual authentication. In the second part, depicted as "Phase 2" in Figure 2, the Enrollee (P2P Client) disassociates and reconnects using its new authentication credentials. In this way, if two devices already have the required network credentials (this is the case in the Persistent group formation), there is no need to trigger the first phase, and they can directly perform the authentication.

E. *Power Saving*

Using Wi-Fi Direct, battery-constrained devices may typically act as P2P GO (soft-AP), and therefore energy efficiency is of capital importance. However, power saving mechanisms in current Wi-Fi networks are not defined for APs but only for clients. Notice that with Wi-Fi Direct, a P2P Client can benefit from the existing Wi-Fi power saving protocols, i.e. legacy power save mode [1] or U-APSD [5]. In order to support energy savings for the AP, Wi-Fi Direct defines two new power saving mechanisms: the *Opportunistic Power Save* protocol and the *Notice of Absence (NoA)* protocol.

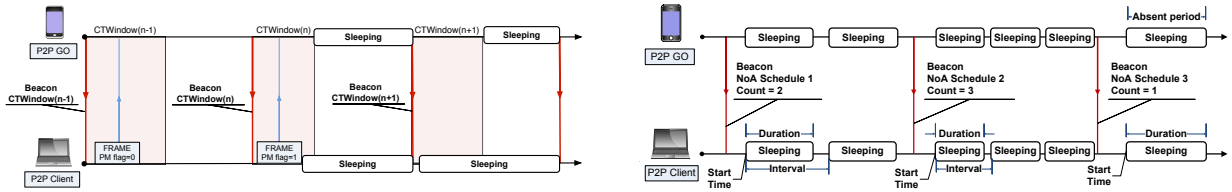
1) Opportunistic Power Save: The basic idea of Opportunistic Power Save is to leverage the sleeping periods of P2P Clients. The mechanism assumes the existence of a legacy power saving protocol, and works as follows. The P2P GO advertises a time window, denoted as *CTWindow*, within each Beacon and Probe Response frames. This window specifies the minimum amount of time after the reception of a Beacon during which the P2P GO will stay awake and therefore P2P Clients in power saving can send their frames. If after the *CTWindow* the P2P GO determines that all connected clients are in doze state, either because they announced a switch to that state by sending a frame with the Power Management (PM) bit set to 1, or because they were already in the doze state during the previous beacon interval, the P2P GO can enter sleep mode until the next Beacon is scheduled; otherwise, if a P2P Client leaves the power saving mode mode (which is announced by sending a frame with the PM bit set to 0) the P2P GO is forced to stay awake until all P2P Clients return to power saving mode. Figure 3 provides an example of the operation of the Opportunistic Power Save protocol for a scenario consisting of one P2P GO and one P2P Client.

Notice that, using this mechanism, a P2P GO does not have the final decision on whether to switch to sleep mode or not, as this depends on the activity of the associated P2P Clients. To give a P2P GO higher control on its own energy consumption Wi-Fi Direct specifies the *Notice of Absence* protocol, which is described next.

2) Notice of Absence: The Notice of Absence (NoA) protocol allows a P2P GO to announce time intervals, referred to as *absence periods*, where P2P Clients are not allowed to access the channel, regardless of whether they are in power save or in active mode. In this way, a P2P GO can autonomously decide to power down its radio to save energy.

Like in the Opportunistic Power Save protocol, in the case of NoA the P2P GO defines absence periods with a signaling element included in Beacon frames and Probe Responses. In particular, a P2P GO defines a *NoA schedule* using four parameters: (*i*) *duration* that specifies the length of each absence period, (*ii*) *interval* that specifies the time between consecutive absence periods, (*iii*) *start time* that specifies the start time of the first absence period after the current Beacon frame, and (*iv*) *count* that specifies how many absence periods will be scheduled during the current NoA schedule[4]. A P2P GO can either cancel or update the current NoA schedule at any time by respectively omitting or modifying the signaling element. P2P Clients

[4]If *count* is set to 255 the current NoA schedule runs permanently until it is explicitly cancelled.



(a) Example of Opportunistic Power Save

(b) Example of NoA

Fig. 3. P2P GO power saving protocols in Wi-Fi Direct.

always adhere to the most recently received NoA schedule. Figure 3 depicts an example operation of the NoA protocol.

In order to foster vendor differentiation, the Wi-Fi Direct specification does not define any mechanism to compute the *CTWindow* in the Opportunistic Power Save protocol or the schedule of *absence periods* in the Notice of Absence protocol. In Section III we provide some experimental results comparing the impact of different power saving policies that can be used to configure the NoA protocol.

III. Experimental Evaluation

In this section we report experimental results using Wi-Fi Direct in a testbed composed of commercial, off-the-shelf devices. Our goal is two-fold: first, to analyze the time required to form a P2P group for the three cases described in the previous section; second, to quantify the achievable performance trade-offs using Wi-Fi Direct and compare them against legacy operation.

A. Testbed Setup

We deploy a testbed consisting of two nodes in an office environment, where different WLANs could be detected, and perform our experiments during working hours to guarantee that our numerical figures correspond to realistic settings. Our nodes are two laptops equipped with an 802.11a/b/g D-Link PCMCIA card with an Atheros chipset, running Linux and the *mac80211/ath5k* driver.

We use an open source implementation of Wi-Fi Direct [7], which builds upon the widely deployed `wpa_supplicant` software, and extend it to implement the Notice of Absence (NoA) protocol. We decided to implement NoA as this mechanism provides the AP with the tightest control and therefore should provide the maximum energy savings. Throughout our experiments we always pre-provision the devices with a WPS PIN, to support automatized execution.

B. Group Formation Delays

We first analyze the time needed to establish a P2P Group for the three group formation cases introduced in Section II-B, namely *Standard*, *Autonomous* and *Persistent*. In order to gain insight on the total time required to form a P2P group, we measured two delays:

- *Discovery delay*, this being the time required for the two P2P devices to find each other.
- *Formation delay*, this being the time required to agree on the roles, establish a secure communication and perform the DHCP exchange.

With the above, the *total delay* is obtained by adding the two previous delays. To compute these delays, we process the log files provided by `wpa_supplicant` to obtain the time instants when a device (i) starts the discovery phase, (ii) discovers the other device, and (iii) obtains an IP address from the P2P GO. We repeat each group establishment 250 times and report the experimental cumulative distribution function (CDF) of the the above delays in Figure 4. In order to understand the behavior of the experimental implementation, we also plot in the figure using dashed lines the results from an event-driven simulator which closely follows the exchanges illustrated in Figure 2.[5]

The CDF of the *discovery delay* (top subplot) shows that the initial scans delay all procedures by at least 3 seconds, and that the *Standard* and *Persistent* cases behave very similarly, an expected result as they use the same discovery algorithm. The figure also quantifies the randomness introduced by the novel Wi-Fi Direct discovery algorithm, which results in additional delays between approximately one and seven seconds. The *Autonomous* case, in contrast, shows an approximately constant delay of three seconds, which is caused by the implementation of its discovery mechanism: the list of P2P GO candidates is returned only after an active scanning in all available channels has been completed (note that our card operates in both the 2.4 and the 5 GHz band). The simulation results show, for the Autonomous case, that there is room for significant improvement if the procedure stops immediately after the P2P GO of interest is discovered; for the Standard and Persistent cases, simulations provide the lower bound for the discovery delay in case we were not experiencing interference and traffic from neighboring WLANs (which is the reason for the differences between the simulation and the experimental results). Given the observed *discovery* times, we conclude that continuously trying to discover other Wi-Fi Direct devices in a battery constrained device may be a challenging task. For this purpose, smart algorithms would be required that interleave sleeping and discovery cycles in order to trade-off discovery time and battery life.

VV IMP. Note

[5] Note that our simulator uses an *optimized* discovery mechanism for the autonomous group formation, in which the process stops as soon as the GO is found.

We next analyze the CDF of the *formation delay* (middle subplot), i.e., the time required to negotiate roles (if needed), to establish a secure communication and to provide the IP configuration. It should be first noted that, for the case of the Autonomous group formation, the open source implementation we are using [7] triggers an additional scan right before the WPS phase, something not specified in the standard. This scan increased the delay by approximately three seconds, thus we decided to remove its effect to present a clear overview of the Wi-Fi Direct performance. The figure shows that the three group formations present similar behavior, with the differences over simulations being caused by channel interference and the generation of cryptographic information, and that the simplified signaling of the *Persistent* case reduces delays in about half a second. It should also be noted that despite the fact that the Autonomous frame exchange is simpler than the Standard one, in practice they result in almost the same delay. This is because the GO Negotiation phase takes very little time as compared to the WPS provisioning phase, which requires a long frame exchange that is affected by the interference from other WLANs.

The addition of the above delays results in the *total delay* depicted in Figure 4, bottom. The figure shows that 80% of the time the Autonomous group formation took less than five seconds to form the group, while for the Persistent and Standard case this value is eight and nine seconds, respectively, an order that follows the relative complexity of the signaling procedures illustrated in Figure 2. It is also worth remarking that, on average, there are a few seconds of difference between simulations and the use of our real-life testbed, caused by interferences from neighboring WLANs.

VV IMP. Note

The above non-negligible delays prevent the use of opportunistic P2P Group establishments to support energy-efficient operation, e.g., setting up a group when traffic is detected, and powering down wireless interfaces when there is no traffic. Instead, in order to amortize the overhead of creating a P2P Group, we expect P2P Groups to have a considerable life time. Thus, in that case, energy efficient operation within a P2P Group, for both P2P Client and P2P GO, becomes a critical feature, and motivates the use of the power saving mechanisms introduced in Section II-E.

C. Energy Efficient Operation

Many Wi-Fi Direct devices will run on batteries, therefore energy efficiency is of paramount importance. In order to quantify the achievable energy savings using Wi-Fi Direct we consider the case of *tethering*, i.e., a mobile device acting as a soft-AP and sharing its 3G connection with a client (see Figure 1). Note that in legacy Wi-Fi networks the AP is forced to remain active at all times. In contrast, the Notice of Absence (NoA) protocol lets a P2P GO announce sleep periods and turn its interface(s) to doze in order to save power. Thus, a key challenge with the use of NoA is how to compute the length of the absence periods in order to trade-off the energy saving achieved in the P2P GO with the performance of the traffic in the P2P Group. We analyze this question next.

VV IMP. Note

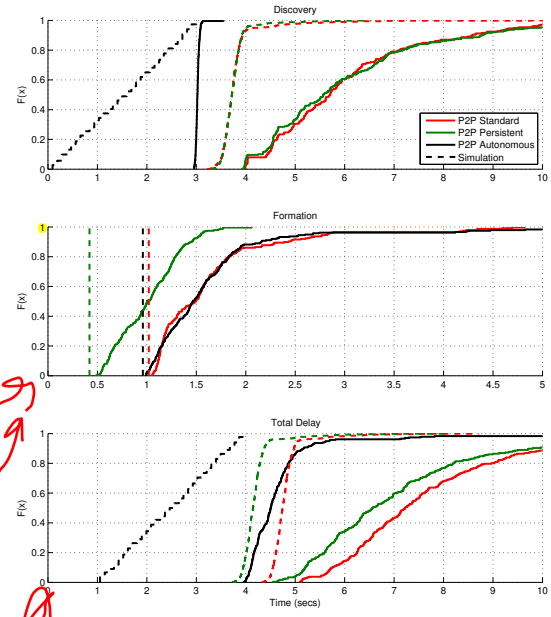


Fig. 4. Evaluation of P2P Group Formation considering *Standard*, *Autonomous* and *Persistent* group formation procedures.

In order to quantify the energy savings supported by Wi-Fi Direct, we fix the Beacon period to 100 ms and analyze three representative policies to implement the NoA protocol: (i) an *Active* policy where the P2P GO remains always active, thus mimicking the legacy operation. Notice that this policy is optimal in terms of traffic performance, but should result in a worst case in terms of energy efficiency; (ii) a *Static* policy, where the P2P GO advertises a fixed presence window of 25 ms right after each Beacon frame, and (iii) a *Dynamic* policy based on the Adaptive Single Presence Period (ASPP) algorithm, which adjusts the presence window based on the estimated traffic activity (utilization) in the channel. Specifically, ASPP estimates the time the channel is busy using an exponentially weighted moving average, and based on this estimation uses a proportional controller to drive the WLAN to the desired point of operation (for more details we refer the interested reader to [11]).

To understand the impact of these policies, we first equip one laptop with a 3G USB dongle and configure it as P2P GO, running NoA with the Dynamic policy; the other laptop is configured as P2P Client and mimics a web-browsing session during ten minutes. We measure the resulting channel activity and the presence window advertised by the P2P GO, and depict both in Figure 5. We also mark in the figure the 25 ms window that would be announced with the Static policy; note that the *Active* operation would correspond to a fixed setting of 100 ms. The figure illustrates how an Active policy would disregard any chance to save energy, despite the frequent periods of time when there is little traffic. In contrast, the use of a Static setting of 25 ms would help to significantly

reduce the energy wastage; however, not only this policy requires to manually tune the announced window for each application, but also it would suffer from performance issues due to the accidental traffic bursts. Finally, the figure illustrates how the use of the Dynamic policy is effective in adapting the presence window to the traffic demand in the network.

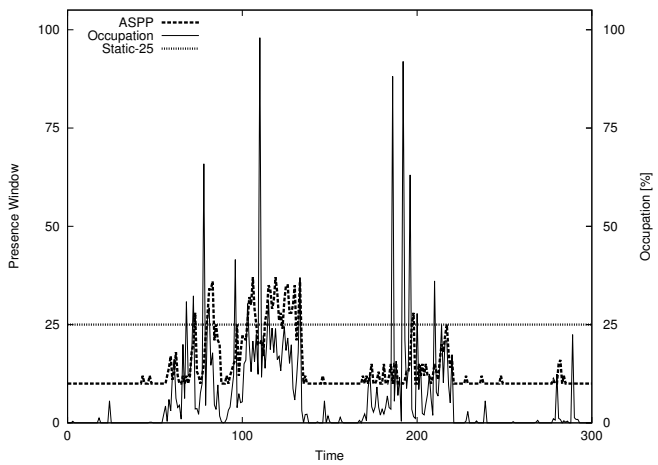


Fig. 5. *ASPP* presence window adaptation with Web traffic.

In order to quantitatively compare these policies in a fair manner, we substitute the 3G dongle by an NS3-based emulator [10], thus ensuring control over the 3G channel conditions and the repeatability of the experiments. We implemented three HSDPA models defined by the Eurane project [12], namely a *pedestrian* channel with good radio conditions, an *urban* channel with average conditions, and a *vehicular* channel with poor conditions. For each scenario and policy, we measure the throughput obtained by a TCP download using `iperf`, and the time that the Wi-Fi interface of the P2P GO spends in the active state and in the sleep state. To provide absolute figures in terms of energy consumption, we translate these times into Joules using the power consumption characteristics of the AR6001 Atheros chipset. In order to have figures with good statistical meaning, each experiment is run for 100 seconds and repeated 5 times. The obtained results for throughput and energy cost are depicted in Figure 6, in which we also add the results for the case of an intra-group communication (i.e., without 3G channel), namely, a TCP download from the P2P GO to the P2P Client.

Figure 6 confirms the results observed in Figure 5. For the case of 3G channels, the *Active* policy achieves the highest throughput in all cases, but at the highest energy cost, increasing this energy cost as the available bandwidth decreases (e.g., *Vehicular* channel). In contrast, the *Static* policy is able to achieve a good performance both in terms of throughput and energy cost when the bandwidth is relatively small, but the 25 ms window becomes a bottleneck when the achievable bandwidth is large. This poor performance is exacerbated in the intra-group communication, where the throughput is less than one fourth of the other mechanisms, this being caused by the overly small presence window, while the energy cost per unit of transferred information is the highest, due to the non-negligible energy consumption of the device during the sleep state. Finally, the *Dynamic* policy is able to achieve throughput figures very close to those of the Active case, while maintaining a small energy cost across all considered scenarios. For the case of 3G channels, then, the energy costs of the communication are reduced by up to 66% as compared to legacy operation (Active case), with almost no price paid in terms of performance. Therefore we conclude that the use of NoA together with an adaptive policy would significantly extend the lifetime of battery-powered devices using Wi-Fi Direct at no significant performance penalty.

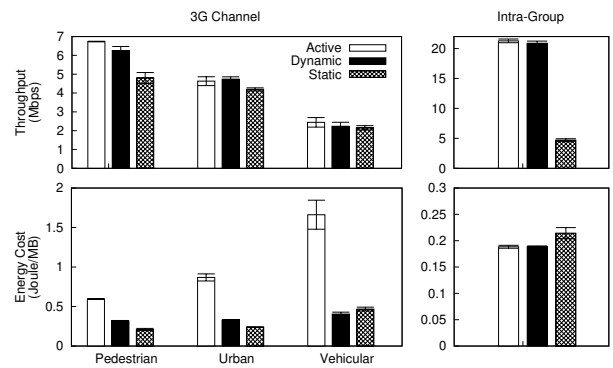


Fig. 6. Perfomance of the *Dynamic*, *Active* and *Static* power saving policies for different scenarios.

IV. Conclusions

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 device to device 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.

In this paper we presented a thorough overview of the novel technical features specified in Wi-Fi Direct, following by an experimental evaluation that quantifies the group formation delays to be expected in real-life scenarios. Finally we have analyzed the performance-energy trade-offs of the novel NoA power saving protocol defined in Wi-Fi Direct through extensive experimentation. To the best of the authors' knowledge this is the first paper in the state of the art that provides a wide overview and experimental evaluation of Wi-Fi Direct.

Regarding future research directions. First, the NoA protocol could also be re-used to virtualize the roles of P2P GO/Client over multiple concurrent P2P Groups. Second, concurrent operation together with the dynamic nature of the P2P GO/Client roles could be used to improve performance in dense environments, for instance by means of dynamic relays. Finally, if Wi-Fi Direct becomes a widespread technology as expected, it faces the challenge of improving coexistence and reducing interference with other unlicensed devices.

ACKNOWLEDGMENT

The authors are grateful to the anonymous referees for their valuable comments which greatly helped in improving the paper.

REFERENCES

[1] IEEE 802.11-2007 Standard, *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, 2007.
[2] IEEE 802.11z-2010 - *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 7: Extensions to Direct-Link Setup (DLS)*.
[3] Jin-Shyan Lee; Yu-Wei Su; Chung-Chou Shen, *A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi*, Industrial Electronics Society, 2007. IECON 2007.
[4] Wi-Fi Alliance, P2P Technical Group, *Wi-Fi Peer-to-Peer (P2P) Technical Specification v1.0*, December 2009.
[5] Wi-Fi Alliance, Quality of Service (QoS) Task Group, *Wi-Fi Multimedia (including WMM PowerSave) Specification v1.1*, 2005.
[6] Wi-Fi Alliance,*Wi-Fi Protected Setup Specification v1.0h*, Dec. 2006.
[7] Wi-Fi Direct in Linux, http://linuxwireless.org/en/developers/p2p/
[8] Edwards, W. K., *Discovery systems in ubiquitous computing*, Pervasive Computing, IEEE, vol. 5, no. 2, pp. 70–77, April-June 2006.
[9] 802.11u-2011 - *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 9: Interworking with External Networks.*
[10] ns-3 simulator, http://www.nsnam.org
[11] D. Camps-Mur, X. Perez-Costa, S. Sallent-Ribes, *Designing energy efficient Access Points with Wi-Fi Direct*, Computer Networks, September 2011.
[12] *Enhanced UMTS Radio Access Network Extensions for NS2 (EURANE)*, http://eurane.ti-wmc.nl/eurane.
[13] Lochan Verma and Scott Seongwook Lee, *Proliferation of Wi-Fi: Opportunities in CE Ecosystem*, in Proc. PerNets 2011, Las Vegas, USA, Jan 9 12, 2011.
[14] Nicholson, A.J.; Wolchok, S.; Noble, B.D.; *Juggler: Virtual Networks for Fun and Profit*, Mobile Computing, IEEE Transactions on, vol.9, no.1, pp.31-43, Jan. 2010
[15] Microsoft's Virtual WiFi Homepage, http://research.microsoft.com/en-us/projects/virtualwifi/