# Designing Energy Efficient Access Points with Wi-Fi Direct

Daniel Camps-Mur[a], Xavier Pérez-Costa[a], Sebastià Sallent-Ribes[b]

[a]*NEC Network Laboratories, Kurfürsten-Anlage 36, Heidelberg, Germany, Email: {camps,perez}@netlab.nec.de*
[b]*Fundació i2cat, c/ Gran Capita 2-4 , Barcelona, Spain, Email: sebastia.sallent@i2cat.net*

**Abstract**

Wi-Fi Direct is a new technology defined by the Wi-Fi Alliance in order to enable efficient device-to-device communication. Portable devices are a main target of this technology and hence, power saving is a key objective. In this paper we analyze the power saving protocols defined in Wi-Fi Direct and design two algorithms to efficiently use them: Adaptive Single Presence Period (ASPP) and Adaptive Multiple Presence Periods (AMPP). Operating only with layer two information, our proposed algorithms allow Wi-Fi Direct mobile devices to address the trade-off between performance and energy consumption in a configurable manner when offering access to an external network. The two algorithms presented, ASPP and AMPP, are evaluated by OPNET simulations.

*Keywords:* Wi-Fi Direct, AP Power Saving, QoS

## 1. Introduction

The Wi-Fi technology[1] has enjoyed since its conception a tremendous market success. Most of this success has been built around the *infrastructure* mode of operation defined in the IEEE 802.11 standard [1], which allows a set of clients to share an Internet connection through an Access Point (AP). Following this success, the infrastructure mode has continuously evolved to address an increasing set of requirements covering an ever growing number of use cases and devices. Examples of this evolution are the addition of QoS and advanced power saving capabilities through the Wi-Fi Multimedia (WMM) and WMM-PowerSave (WMM-PS) technologies [2], or advances in security and ease of use through the Wi-Fi Protected Setup technology [3].

However, unlike infrastructure mode, the *ad-hoc* mode defined in the 802.11 standard, aimed at providing device-to-device connectivity, has not enjoyed a similar market success. In order to address this market void, the Wi-Fi Alliance (WFA) [4] has recently developed the *Wi-Fi Direct* technology, which is however based on a different approach than the 802.11 ad-hoc mode. Instead of having a network of equally capable devices, in Wi-Fi Direct devices negotiate the roles of AP and client to set up an infrastructure-like network. Hence, Wi-Fi Direct can immediately benefit from the technologies defined for infrastructure mode, and can seamlessly integrate all the existent basis of Wi-Fi devices in the market, which will simply see a Wi-Fi Direct device as a traditional AP.

Targeting a successful device-to-device connectivity experience, the Wi-Fi Direct technology goes beyond having a device acting as a traditional Wi-Fi AP and introduces the following new functionalities: i) AP power saving capabilities, critical in portable devices like mobile phones, ii) Concurrent Operation, allowing devices to maintain simultaneous connections to different Wi-Fi Direct networks, and iii) Layer two service discovery, allowing devices to discover available services prior to establishing a connection. For a detailed description of each of these features the interested reader is referred to [5].

Our work in this paper focuses on the AP power saving protocols defined by Wi-Fi Direct. Specifically, we target the case where the Wi-Fi Direct device acting as AP is a battery limited device (e.g. a mobile phone) and offers its connected clients access to an external network (e.g. a cellular network). Figure 1 depicts an example of our target scenario. In this context, we design and evaluate algorithms that address the trade-off between the power consumed by the Wi-Fi Direct AP and the performance experienced by its connected clients.

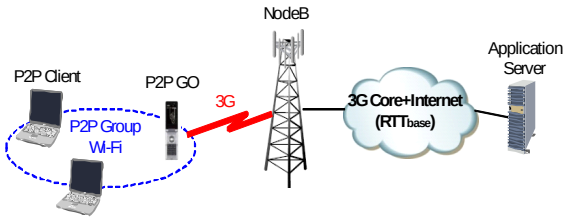



Figure 1: Our target scenario. A mobile phone sharing access to a 3G network with a set of connected devices using Wi-Fi Direct.

Previous work considering methods enabling AP power saving can be divided in two categories depending on whether they can be applied to the current 802.11 standard or require protocol extensions.

The IEEE 802.11-2007 standard [1] by means of the *Quiet element* already defines a method allowing an AP in the 5Ghz band to silence its associated clients. Although originally defined in order to scan for interfering radars it can also be applied to reduce the power consumption of APs. An alternative solu-

[1] In this paper we refer indistinctly to the set of technologies derived from the work of the IEEE 802.11 group as Wi-Fi or 802.11.

tion designed to allow APs to save power even in the presence of legacy 802.11 devices is described in [20]. The approach taken in this case is to block the medium by signaling a longer network allocation vector (NAV) than actually required. In this way the AP can be sure that no Wi-Fi device will try to access the channel during the defined period. Notice though that this mechanism has the drawback of also blocking access to the channel to neighboring Wi-Fi networks. Solutions as the ones previously described allow for some AP power saving but due to the fact that the protocols used were not specifically designed for this purpose they are not flexible enough to accommodate the needs of different use cases.

On the other hand, there have been proposals which define extensions to the current 802.11 standard to achieve AP power saving. [6] and [7] are examples of such efforts which defined new signaling mechanisms for solar/battery powered Wi-Fi APs.

As a result of the research and market interest in this area during the last years, the Wi-Fi Alliance has defined the Wi-Fi Direct protocol which is expected to become the predominant solution in practice. Therefore, in this paper we focus on the design of AP power saving algorithms for Wi-Fi Direct. To the best of the authors' knowledge there is no previous work designing AP power saving algorithms based on the Wi-Fi Direct specification.

The rest of the paper is organized as follows. Section 2 provides an overview of the AP power management protocols defined by Wi-Fi Direct. Sections 3 and 4 design and evaluate, respectively, our proposed Wi-Fi Direct Adaptive Single Presence Period (ASPP) and Adaptive Multiple Presence Periods (AMPP) algorithms. Finally, Section 5 concludes this paper.

## 2. AP Power Management in Wi-Fi Direct

Wi-Fi Direct devices, named Peer to Peer (P2P) devices [5], must be able to act both as a Wi-Fi AP or as a Wi-Fi Client. In particular Wi-Fi Direct defines the concept of a P2P Group, where a P2P Group Owner (P2P GO) acts as an AP for a set of connected P2P Clients. There are two possible ways to decide which P2P device will act as P2P Group Owner:

1. A P2P device autonomously initiates a P2P Group.
2. Two P2P devices run a negotiation protocol after discovering each other.

Once the P2P Group is established new P2P devices can discover and join the group using active or passive scanning mechanisms like the ones used in traditional Wi-Fi networks. Acting as P2P Group Owner provides certain advantages. For instance, a P2P Group Owner is allowed to cross-connect a P2P Group with an external network, e.g. a cellular network if the P2P Group Owner has a 3G interface. However, becoming a P2P Group Owner requires performing certain functions (e.g. beaconing, forwarding) that will result in a higher power consumption than the one of a P2P Client, which can benefit from the power saving protocols already defined in IEEE 802.11. In order to address the power consumption imbalance between a P2P Group Owner and a P2P Client, and to allow battery powered devices to efficiently operate as P2P Group Owners, the Wi-Fi Direct specification defines two new protocols that can be used by a P2P Group Owner: i) the *Opportunistic Power Save* protocol, and ii) the *Notice of Absence* (NoA) protocol.

### 2.1. Opportunistic Power Save Protocol

The Opportunistic Power Save protocol (OPS) allows a P2P Group Owner to *opportunistically* save power when all its associated clients are sleeping. This protocol has a low implementation complexity but, given the fact that the P2P Group Owner can only save power when all its clients are sleeping, the power savings that can be achieved by the P2P Group Owner are limited. OPS is based on the design of the traditional power save mode used by clients in an infrastructure network. A P2P Group Owner can save power by defining a limited *presence period* after every Beacon transmission, known as *CTWindow*, where P2P Clients are allowed to transmit. If at the end of the CTWindow all associated P2P Clients are sleeping, the P2P Group Owner is allowed to sleep until the next Beacon time. However, if any P2P Client stays in active mode at the end of the CTWindow the P2P Group Owner is forced to remain awake until the next Beacon time. The operation of the Opportunistic Power Save protocol is depicted in Figure 2(a).

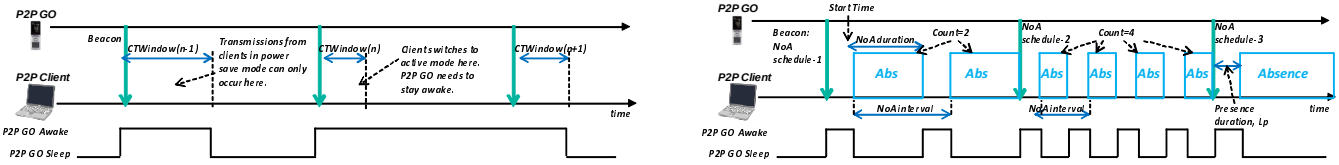
### 2.2. Notice of Absence Protocol

Unlike Opportunistic Power Save, the Notice of Absence (NoA) protocol can be used by a P2P Group Owner to save power regardless of the power state of its associated clients. The NoA protocol requires a higher implementation complexity than Opportunistic Power Save but delivers to a P2P Group Owner a higher control on its power consumption. The idea behind the NoA protocol is to let a P2P Group Owner advertise a set of *absence periods* where its associated P2P clients are not allowed to transmit. Thus, a P2P Group Owner may sleep during these absence periods in order to save power.

The NoA protocol provides a P2P Group Owner the means to signal a flexible *absence schedule*. In particular, a NoA absence schedule is defined by a 4-tuple: *start time*, *duration*, *interval* and *count*.

- *Start time* defines the start time of the next absence period.
- *Duration* indicates the duration of an absence period in the schedule.
- *Interval* indicates the time between consecutive absence periods.
- *Count* indicates the number of absence period occurrences until the advertised schedule expires. If *count* is set to 255 the advertised schedule repeats until is explicitely cancelled.

In order to keep complexity low, the Wi-Fi Direct specification allows a P2P Group Owner to advertise at any point in time a single NoA schedule in Beacon frames and Probe Responses. A P2P Group Owner can update the current NoA




(a) Example of Opportunistic Power Save operation.

(b) Example of Notice of Absence operation.

Figure 2: Example operation of the Wi-Fi Direct power saving protocols for a P2P Group Owner.

schedule simply by modifying the correspondent signaling element in Beacon frames and Probe Responses, or can cancel it by omitting the correspondent signaling element. A P2P Client always adheres to the most recently observed NoA schedule advertised by the P2P Group Owner. Finally, the NoA protocol includes a mechanism, the P2P Presence Request/Response handshake, that allows a P2P Client to request a P2P Group Owner to be present at certain intervals. Although not mandatory, such a request mechanism is useful when a P2P Client runs applications that require QoS guarantees, like VoIP. Figure 2(b) depicts an exemplary NoA schedule and illustrates the function of each parameter.

In order to allow for product differentiation, the Wi-Fi Direct specification does not define how a P2P Group Owner has to build a Notice of Absence schedule, or how *CTWindow* has to be adjusted in case of Opportunistic Power Save. Therefore, the focus of this paper is to define algorithms that a P2P Group Owner can use to reduce energy consumption while minimizing any effect on the performance of its associated P2P Clients. In particular we designed and evaluated two novel algorithms:

- *Adaptive Single Presence Period* (ASPP) which can be used with both the Opportunistic Power Save and the Notice of Absence protocols.
- *Adaptive Multiple Presence Periods* (AMPP) which can be used only with the Notice of Absence protocol but improves the performance of ASPP.

## 3. ASPP: Adaptive Single Presence Period

In this section we present the design of the *Adaptive Single Presence Period* (ASPP) algorithm. This algorithm is applicable to both power saving protocols described in the previous section, i.e. Opportunistic Power Save and Notice of Absence. Its basic idea is to adaptively adjust the size of the *Presence Period* that a P2P Group Owner advertises at every Beacon frame, as illustrated in Figure 2(a), based on the consideration of the trade-off between service quality of experience and device power saving.

### *3.1. ASPP Algorithm Design*

We start the design of our ASPP algorithm by establishing the architectural constraint that ASPP should operate using only information available at layer two, i.e. running in the Wi-Fi driver of the P2P Group Owner, without requiring any interaction with higher layers or other network interfaces. Note that this design decision generalizes the applicability of the developed algorithm since it can be then applied even if new network interfaces are incorporated at the P2P Group Owner or even if the P2P Group Owner is not directly connected to the external network. An example of the previous case would be for instance a laptop acting as P2P Group Owner where a 3G card is used to provide access to the Internet.

The main challenge to be solved by ASPP is hence how to *dimension* the advertised presence periods, $L_P$, every Beacon interval, in order to efficiently balance the energy consumed by the P2P Group Owner and the performance experienced by the associated P2P Clients.

We study how to address the previous objective in the context of our scenario of interest depicted in Figure 1, which consists of a mobile phone acting as P2P Group Owner and sharing access to a 3G network among its associated P2P Clients. It is fair to assume in this scenario that the bandwidth available in a 3G link will normally be lower than the bandwidth available in the P2P Group. Note that current Wi-Fi networks provide peak rates above 54Mbps (up to 300Mbps with 802.11n), while the majority of deployed 3G networks with HSDPA have peak rates of up to 7.2Mbps in the downlink [18]. Thus, if the previous assumption holds, the maximum data rates achieved in the connections established by the P2P Clients connecting through the mobile phone in Figure 1 will be limited by the bandwidth available in the 3G network. Therefore, an ideal dynamic algorithm would advertise the smallest presence periods that can deliver in the P2P Group the bandwidth available in the 3G link. In this way a P2P Group Owner would maximize its sleep periods without the P2P Clients noticing any performance degradation.

However, in order to implement this ASPP dynamic algorithm two challenges need to be addressed:

1. The bandwidth available in the 3G link can be highly variable and, given our Layer 2 information only constraint, is not known by the P2P Group Owner.
2. Even if the bandwidth in the 3G link was perfectly known to the P2P Group Owner, applications may not offer enough load to saturate the 3G link. Therefore, a P2P Group Owner should avoid over-dimensioning its advertised presence periods and instead, size them in order to satisfy the minimum between the application's required data rate, and the bandwidth available in the 3G link.

In order to adress the aforementioned challenges, ASPP will adapt the length of the presence periods based on the amount of traffic flowing between the P2P Group and the 3G network. The assumption here is that TCP connections will saturate in the 3G link, and thus reacting to the amount of traffic flowing between the P2P Group and the 3G network, ASPP will indirectly follow the variations in the available 3G bandwidth. The implications of this assumption will be carefully analyzed in the ASPP algorithm evaluation presented in Section 3.2.

Thus, we want to design an algorithm that dimensions presence periods in the following way:

- First, if the bandwidth available in the 3G link increases and the applications offer enough load, the P2P Group Owner will measure an increase of traffic in the P2P Group and will increase $L_P$ accordingly.
- Second, if either the bandwidth available in the 3G link or the load offered by the applications decreases, the P2P Group Owner will measure a decrease of traffic in the P2P Group and will in turn decrease $L_P$.

The previous behavior can be implemented using the following proportional controller (a similar controller was used in [6] in the context of solar powered APs):

$$L_P(n+1) = L_P(n) + K(U_{last} - U_{target})L_P(n) \quad (1)$$

Where $K$ is a constant used to trade-off convergence speed and stability, $U_{last}$ is the utilization measured in the P2P Group during the last presence period $L_P(n)$, i.e. $U_{last} \sim \frac{used\ time}{L_P}$, and $0 \leq U_{target} \leq 1$ is the algorithm's target utilization.

It can be observed that the previous controller adjusts $L_P$ according to the measured utilization, $U_{last}$, in order to maintain future utilizations around $U_{target}$. For instance, a small value of $U_{target}$ in Equation 1 will lead to high presence intervals even when the network utilization is light, which can result in a good traffic performance but in an increased energy consumption in the P2P Group Owner. In the next section we will study how to appropriately set the $U_{target}$ and $K$ parameters. A P2P Group Owner will execute Equation 1 prior to Beacon transmissions in order to decide on the presence duration to be advertised.

A critical aspect in the previous controller is the way a P2P Group Owner measures utilization in the P2P Group. This is indeed a non trivial issue in contention based networks like Wi-Fi. For instance, if low priority contention settings are used, a P2P Group Owner would sense less transmissions than if higher priority contention settings are used, although in both cases a P2P Client might have a queue full of packets. Therefore, we propose to compute $U_{last}$ in Equation 1 in the following way:

$$U_{last} = \frac{tx\_time + contention\_time}{L_P(n)} \quad (2)$$

Where a P2P Group Owner accumulates the duration of transmitted and received frames during a presence period in the variable *tx_time*. In order to capture the effect of contention, an estimate of the average access delay, which is defined as the time since a packet is first in the transmission queue until it is successfully transmitted, is kept in the variable $Acc\_Del[AC]$ for each Access Category (AC). The $Acc\_Del[AC]$ estimates are updated by means of an Exponentially Weighted Moving Average (EWMA) updated every time the P2P Group Owner transmits a frame through an AC:

$$Acc\_Del[AC](n) = \alpha \cdot Acc\_Del[AC](n-1) + (1-\alpha) Last\_Acc\_Del[AC] \quad (3)$$

where we empirically set $\alpha = 0.9$. Thus, for each transmitted or successfully received frame[2], the P2P Group Owner accumulates in the variable *contention_time* the correspondent $Access\_Delay[AC]$.

Finally, in addition to the previous steps, presence periods are limited between $L_{P_{max}}$, the size of the Beacon interval, and $L_{P_{min}}$, a minimum configurable by the device manufacturer. Thus, when there is no traffic, small presence periods are advertised achieving a significant energy reduction. As soon as some traffic appears in the network the size of the presence periods is adjusted in order to accomodate it until $L_{P_{max}}$ is reached.

| | Data/Control/Mgmt Rate | AIFS/CWmin/CWmax/TXOP |
|---|---|---|
| ***WiFi AC_BK*** | 54/24/1 Mbps | 7/31/1023/0ms |
| ***WiFi AC_VI*** | 54/24/1 Mbps | 2/7/15/3ms |

Table 1: Wi-Fi Configurations under study

### 3.2. ASPP: Algorithm Evaluation

In this section we evaluate the performance of our proposed ASPP algorithm by means of packet level simulations. We divide this evaluation in two stages, a first stage where we illustrate the algorithm dynamics, and a second stage where we extensively evaluate the performance of the algorithm with popular data applications like file transfers and Web traffic.

#### 3.2.1. Simulation Framework

Our performance evaluation is carried out by means of packet level simulations using OPNET [8]. We implemented in OPNET the Opportunistic Power Save and the Notice of Absence protocols defined in [5] and all the relevant Wi-Fi protocols required in Wi-Fi Direct for QoS and power saving, i.e. WMM and WMM-PS [2]. Table 1 contains all the Wi-Fi related simulation parameters employed in our evaluation.

Our simulations reproduce the scenario depicted in Figure 1, where in order to model the 3G link we ported to OPNET the 3G simulation framework defined by the Eurane project [9], which simulates a HSDPA link. For the purpose of our evaluation we use three baseline 3G channel models, which capture a wide spectrum of possible 3G channels:

- *Pedestrian-A* channel model which represents a scenario of reduced mobility and *good* radio conditions
- *Typical Urban* channel model which represents a scenario of moderate mobility and *average* radio conditions

[2]Notice that a P2P Group Owner can discover the Access Category of a received frame looking at the User Priority field present in the Wi-Fi header.

- *Vehicular-A* channel model which represents a scenario with high mobility and *poor* radio conditions.

Figure 3 depicts an example of how the available PHY data rate varies in time with each of the previous channel models. Table 3 contains all the 3G related parameters used in our evaluation. In addition, the core network connecting the NodeB to the application servers in Figure 1 is modeled using a node that introduces a configurable delay, hereafter referred to as $RTT_{base}$.

We consider Web browsing and File Transfers, e.g. Video streaming, as the most relevant applications to be evaluated in our scenario of interest, i.e. a mobile phone providing 3G access through Wi-Fi Direct. TCP New Reno is the transport protocol used in our evaluation.

In order to evaluate the energy consumed by the Wi-Fi interface of the P2P Group Owner, we make use of a well known model that captures the energy consumed by a Wi-Fi chipset. This model consists of four basic states: Sleep, Listen, Reception and Transmission. Energy is computed by integrating the power that a Wi-Fi device spends in each of the previous states over a certain target time. In our evaluation this target time will be the time to transfer a file or a web page. The power values used were obtained from a known embedded Wi-Fi chipset [10] and are shown in Table 2.

| **Wi-Fi Chipset** | **Sleep** | **Listen** | **Rx** | **Tx** |
|---|---|---|---|---|
| **Power (mW)** | 0.3 | 432 | 432 | 640 |

Table 2: Power Consumption levels in the Wi-Fi chipset

Finally, in order to gain statistical confidence, we run every simulation with 15 independent seeds and plot the 95% confidence intervals on the obtained average values. Notice though that sometimes these confidence intervals are too small to be clearly observed.

### *3.2.2. Algorithm dynamics*

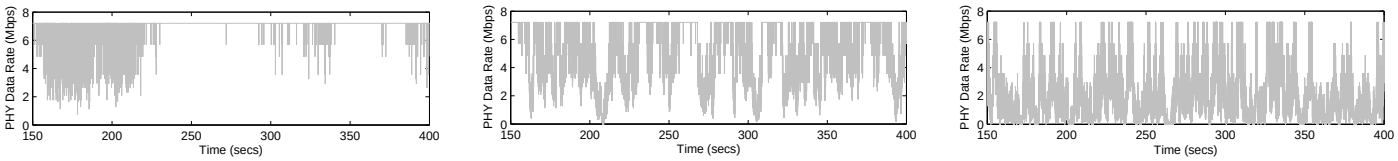
In this section we study the dynamics of the ASPP algorithm and provide a deeper understanding on the effect of its configuration parameters, i.e. $K$ and $U_{target}$ in Equation 1. In addition, the maximum and minimum allowed presence durations are set respectively to $L_{P_{max}} = 100ms$ and $L_{P_{min}} = 10ms$.

To illustrate the dynamics of ASPP we start performing an experiment where three P2P Clients connect to a P2P Group Owner offering access to a 3G network. The first P2P Client downloads a 50MB file (close to the median video file size in the Internet [11]) from the 3G network, and the other two P2P Clients exchange another 50MB file over the P2P Group. The Typical Urban channel model is used in this example. Figure 4(a) depicts the results of this experiment, where a double y-axis is used. The presence durations advertised by the P2P Group Owner are plotted against the left y-axis, while the instantaneous PHY data rate offered in the 3G channel and the throughput being forwarded by the P2P Group Owner are plotted against the right y-axis. As observed in the figure, when only the File Transfer through the 3G link is active (e.g. after 370 seconds), the P2P Group Owner dimensions the presence periods according to the bandwidth variations in the 3G link, in such a way that the P2P Group Owner is only awake the minimum time required to deliver the bandwidth available in the 3G link. However, when the intra-group transfer takes place (330-370 secs), this transfer saturates the Wi-Fi network and the P2P Group Owner stays constantly awake (advertises a presence period of 100ms). Notice that this approach, besides not slowing down the File Transfer, is energy efficient because, if the Wi-Fi network is saturated, the P2P Group Owner spends all its awake time transmitting useful data.

| | |
|---|---|
| **Uplink** | 384Kbps CBR Bearer |
| **Downlink** | HSDPA (UE Category=7, max. PHY rate=7.2Mbps) |
| **TTI** | 2ms |
| **HARQ Feedback Delay** | 3 TTI |
| **HARQ Retransmission Delay** | 6 TTI |
| **HARQ Max Retransmissions** | 3 |
| **Reordering Buffer Size** | 32 PDUs |
| **T1 Timer** | 100ms |
| **Channel Models** | Pedestrian-A<br>Typical Urban<br>Vehicular-A |
| **Distance from BS** | 100m<br>250m<br>500m |
| **Terminal speed** | 3 Km/h<br>50 Km/h<br>120 Km/h |

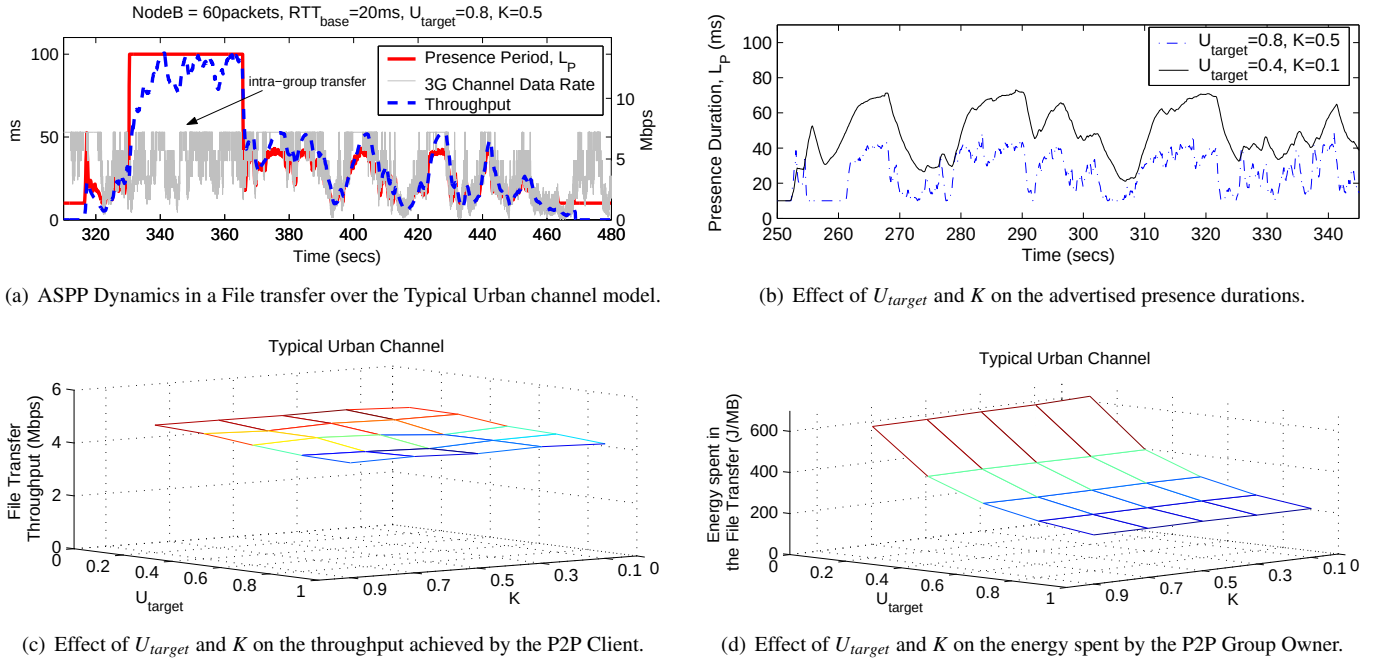
Table 3: Default 3G Simulation Parameters

We can see in Figure 4(a) that ASPP is indeed able to follow variations in the bandwidth available in the 3G link. In a general setting though this will depend on the characteristics of the 3G channel and on the parameters used to configure ASPP, i.e. $K$ and $U_{target}$. To illustrate the effect of these parameters Figure 4(b) depicts the dynamics of the presence periods advertised by the P2P Group Owner, in a setting where a P2P Client retrieves a 50MB file through the 3G network characterized again using the Typical Urban channel model. This time though, for the sake of clarity, the 3G PHY data rate or the achieved throughput are not depicted. Figure 4(b) depicts the result of using two different parameter configurations: i) $U_{target} = 0.8/K = 0.5$ and ii) $U_{target} = 0.4/K = 0.1$. Although the throughput experienced by the P2P Client was very similar in both configurations, it is clear in the figure that the advertised presence durations in each configuration significantly differ. The effect of each parameter is the following. On the one hand decreasing $U_{target}$ increases the advertised presence periods, hence increasing power consumption. The reason is that given a certain amount of load, if $U_{target}$ decreases, higher presence periods are needed to maintain the utilization around $U_{target}$ (note that the duration of presence periods is proportional to $\frac{1}{U_{target}}$). On the other hand, $K$ controls how fast ASPP adapts to variations in the traffic flowing between the P2P Group and the 3G network. Clearly, a small $K$ decreases the ability of ASPP to adapt to changes and a too high $K$ might result in a unwanted large presence periods that




(a) Pedestrian-A, 3Km/h, 100m, *good* channel. (b) Typical Urban, 50Km/h, 250m, *average* channel. (c) Vehicular-A, 120km/h, 500m, *poor* channel.

Figure 3: Sample of PHY rate variation in the HSDPA Channels considered in our evaluation.




(a) ASPP Dynamics in a File transfer over the Typical Urban channel model. (b) Effect of $U_{target}$ and $K$ on the advertised presence durations.

(c) Effect of $U_{target}$ and $K$ on the throughput achieved by the P2P Client. (d) Effect of $U_{target}$ and $K$ on the energy spent by the P2P Group Owner.

Figure 4: Dynamics of the *ASPP* algorithm.

penalize power consumption.

In order to configure ASPP, Figures 4(c) and 4(d) illustrate respectively the throughput achieved by the P2P Client while transferring the 50MB file used in our previous example, and the energy spent by the P2P Group Owner during this file transfer. In these figures we vary $U_{target}$ and $K$ between 0 and 1, and report the results obtained with the Typical Urban channel model. Similar results were obtained with the other channel models. Looking at Figure 4(d), we can see how a small $U_{target}$ penalizes energy consumption in the P2P Group Owner. However, as depicted in Figure 4(c), the throughput achieved by the P2P Client remains relatively stable, meaning that ASPP is able to deliver the required throughput even for high values of $U_{target}$. In the rest of the paper we will use $U_{target} = 0.8$. Regarding $K$, we can see that its effect, both in throughput and energy, is relatively small, thus in the rest of the paper we will configure $K = 0.5$.

Finally, it is worth to notice that the same algorithm dynamics hold if an uplink File Transfer is considered. For the sake of clarity though we have not included these results in the paper.

### 3.2.3. Steady State Evaluation

In this section we present the results of a steady state set of experiments where we asses the impact of the proposed algorithm on the energy consumption of a P2P Group Owner and on the performance of popular applications like File Transfers and Web browsing.

The performance of a File Transfer transmitted over TCP depends on several parameters: i) the bottleneck bandwidth, ii) the amount of buffering in the bottleneck, and iii) the path delay. In order to account for the effect of each of these parameters we perform two different experiments.

1. *NodeB Buffering Variation Experiment*: TCP file transfer of 50 MB. For each considered 3G channel, we vary the amount of buffering in the NodeB[3] while maintaining a fixed $RTT_{base}$ set to $20ms$.
2. *Path Delay Variation Experiment*: TCP file transfer of 50 MB. For each considered 3G channel, we vary the path

[3] In practice, buffering could also be limited by the mobile phone acting as P2P Group Owner instead of the NodeB. Notice though that this would not affect the validity of the presented results.

delay ($RTT_{base}$ in Figure 1), while setting the maximum buffering in the NodeB equal to 30 packets[4] per flow.

In both experiments, we compare ASPP to two competing algorithms:

- *Active* algorithm: The P2P Group Owner switches to active mode once traffic is detected in the network. This algorithm should provide an upper bound with respect to performance and a worst case bound with respect to energy consumption.

- *Static* algorithm: The P2P Group Owner advertises a fixed presence duration equal to 25ms ($\frac{Beacon\ Interval}{4}$).

Finally, in order to asses the effect that the amount of bandwidth available in the Wi-Fi network has on the performance of the algorithms under study, we consider two different Wi-Fi configurations, AC_BK and AC_VI, which are described in Table 1. The intent of these two Wi-Fi configurations is to capture the effect of the bandwidth available in the Wi-Fi network on the algorithms under study. The two configurations are named after the Wi-Fi priority[5] used to transport data traffic, and are configured following the recommendations given in the 802.11 standard [1]. Thus, the AC_BK configuration uses slower contention settings and should provide a smaller bandwidth. Instead, the AC_VI configuration uses more aggressive contention settings and allows to aggregate several packets in one Transmission Opportunity (TXOP), hence providing a higher bandwidth. Packet aggregation is also a key technology in 802.11n [19], therefore we expect the performance of the AC_VI configuration to provide hints on the performance to be expected with 802.11n.

*NodeB Buffering Variation Experiment*

Figures 5(a) and 5(b) depict the P2P Group Owner performance under our first *experiment* in terms of connection throughput and energy consumption. The upper, middle and lower plot show, respectively, the performance in the Pedestrian-A channel model (good), the Typical Urban channel model (average), and the Vehicular-A channel model (poor). In addition, the left column in Figures 5(a) and 5(b) depicts the results obtained with the Wi-Fi AC_VI configuration, and the right column the results obtained with the Wi-Fi AC_BK configuration.

The algorithms under study exhibit a similar behavior in the Pedestrian and Typical Urban channel models (upper and middle plots). Using these channel models, the Active algorithm, as expected, results in the highest throughput (specially when the buffering in the NodeB is small) and in the highest energy consumption. On the other hand, ASPP and the Static algorithm deliver similar throughput to a P2P Client when the AC_VI configuration is used, but significantly differ when the AC_BK configuration is used (with ASPP providing a much higher throughput). Regarding energy, ASPP results in the lowest energy consumption, followed by the Static algorithm. Both algorithms achieve a much lower energy consumption than the Active algorithm (> 50%).

Based on these results the following questions open up: i) Why does throughput degrade so much in ASPP and the Static algorithm when the NodeB buffer is small?, ii) Why does the AC_BK configuration, which in theory should deliver less bandwidth in the Wi-Fi network, outperform the AC_VI configuration in the case of ASPP?, and iii) Why do ASPP and the Static algorithm provide a very similar performance with AC_VI but significantly differ with AC_BK?

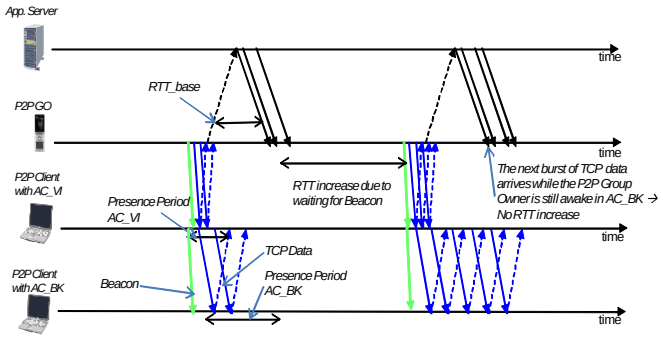
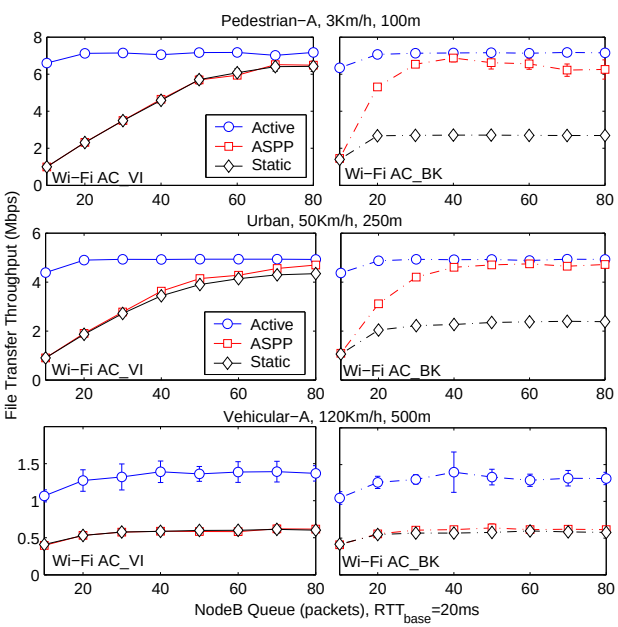



Figure 6: Dynamics of a TCP transfer when the P2P Group Owner runs a power saving protocol. Notice that the two P2P Clients illustrate a different case: the upper one a case where the file is transferred over AC_VI, and hence the P2P Group Owner can use smaller presence periods, and the lower one a case where the file is transferred over AC_BK and so the P2P Group Owner uses larger presence periods.

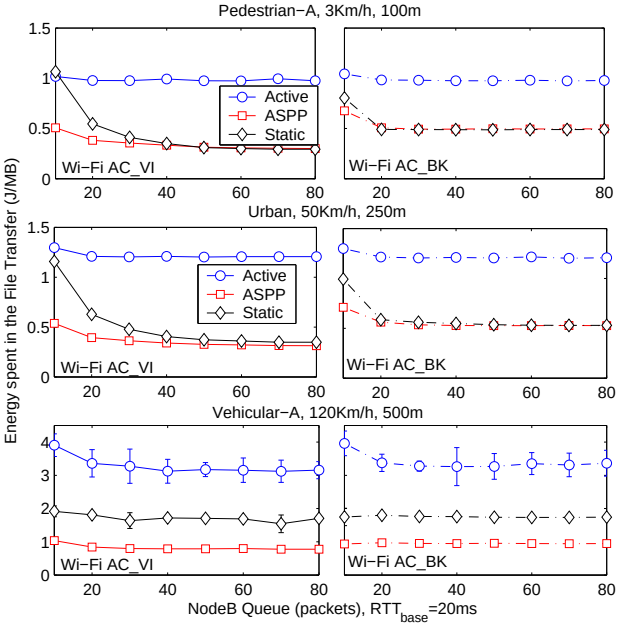
In order to answer these questions, Figure 6 depicts the dynamics of a TCP connection traversing a P2P Group Owner running a power saving algorithm like ASPP. As it can be seen in the figure, scheduling one presence period every Beacon interval might increase the RTT experienced by the TCP connection. Thus, if the TCP congestion window can not grow large enough to cover the increased bandwidth delay product, as it is likely to be the case when the amount of buffering in the NodeB is small, the 3G link becomes underutilized and the maximum achievable throughput reduces. In order to understand why ASPP performs better with AC_BK than with AC_VI, it should be noted that the higher the bandwidth in the Wi-Fi network, the smaller the presence durations that the P2P Group Owner needs to schedule in order to maintain the required level of utilization, $U_{target}$, in Equation 1. Notice that, although reducing power consumption, this behavior reduces the probability that a new window of TCP data arrives at the P2P Group Owner during an active presence period thus resulting in bursty behavior and an increased RTT, see in Figure 6 the second Beacon interval of the AC_BK case (lower P2P Client) for an explanation of why throughput increases in AC_BK. A similar interaction between TCP and the 802.11 Power Save Mode (PSM) used by Wi-Fi clients was already pointed out in [12]. Finally, in the case of AC_VI, ASPP and the Static algorithm provide a similar

[4] A MTU of 1500B is considered in our evaluation.

[5] Notice that 802.11 defines four priorities for its contention based channel access, i.e. AC_VO, AC_VI, AC_BE and AC_BK.




(a) Average File Transfer throughput




(b) Average Energy spent by the P2P Group Owner in the file transfer

Figure 5: File transfer performance when varying the buffering in the NodeB. For each figure the left column represents the results when using the Wi-Fi AC_VI configuration, and the right column the results when using the Wi-Fi AC_BK configuration.

throughput performance, and ASPP is slightly more efficient in terms of energy. The reason for the similar throughput performance is that in this case a presence period of $25ms$ turns out to be enough for the P2P Group Owner to deliver the packets that it has buffered at every Beacon frame. Indeed, when the NodeB buffer is below 20 packets even a presence period below 25ms would be sufficient, which is why the Static algorithm wastes some energy in this case. On the other hand, when AC_BK is used the P2P Group Owner needs much more than 25ms to deliver its buffered packets due to the reduced Wi-Fi bandwidth. This is the reason for the lower throughput experienced by the P2P Client when the Static algorithm is used.

Coming back to Figures 5(a) and 5(b) and focusing now on the performance achieved in the Vehicular-A (poor) channel model, we can see how ASPP and the Static algorithm deliver a similar throughput to the P2P Client, but ASPP does it in a more energy efficient manner. The reason is that presence periods of $25ms$ are larger than necessary in this case to achieve the delivered bandwidth. In addition, no difference is observed between the AC_VI and AC_BK configurations. This is due to the fact that in this case the 3G link exhibits a *good* condition during such short times that although AC_BK results in slightly higher presence periods, TCP has no time to benefit from them. Finally, it is worth to notice how much the energy consumption in the P2P Group Owner increases when the radio conditions in the 3G link degrade and the Active algorithm is used: $\sim 3.5J/MB$ in the Vehicular-A channel compared to $\sim 1J/MB$ in the Pedestrian-A chanel. The reason is that when the 3G radio conditions are poor and the Active algorithm is used the P2P Group Owner stays most of the time idle, hence wasting power, in the Wi-Fi link.

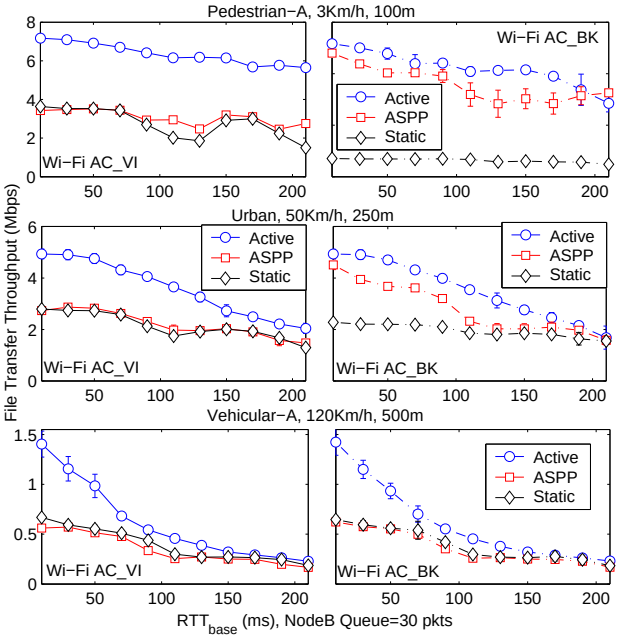
*Path Delay Variation Experiment*

To conclude our File Transfer application performance evaluation with the different algorithms under study, we analyze in this experiment the effect of varying the path delay experienced by the TCP connection ($RTT_{base}$). The corresponding results are depicted in Figure 7. It can be noticed that the lessons learned from the previous experiment where we varied the amount of buffering in the NodeB, can also be applied to explain the behavior observed in this case. A remarkable result from this experiment is the very high energy consumption incurred by the P2P Group Owner in the Vehicular-A channel when the Active algorithm is used ($\sim 7.5J/MB$). The reason is the poor radio conditions in the 3G link plus the big path delays experienced by the TCP connection which result in the P2P Group Owner being almost always idle in the Wi-Fi network.

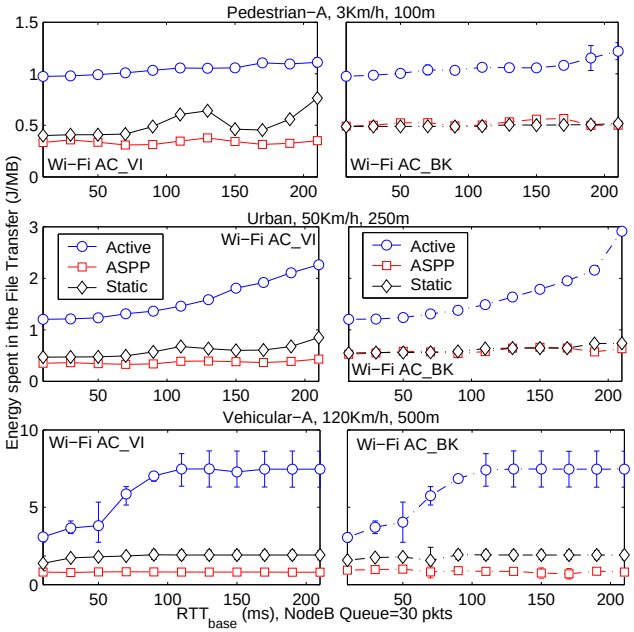
*Web Traffic Experiment*

We look now at how the different algorithms under study perform with Web traffic. In order to model Web traffic, we consider HTTP1.1 and have a P2P Client periodically[6] requesting a new web page to the Web server in the Internet, while varying $RTT_{base}$. The size and number of embedded objects of a Web page are modelled according to the statistics reported in [11]. Unlike in the case of the File Transfer, we only depict the results for the Typical Urban (average) channel in the case of

[6]More than 1000 Web pages are requested by the P2P Client in each simulation run.

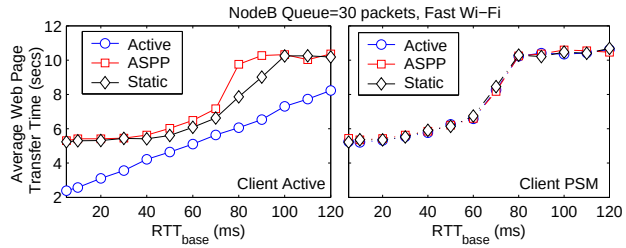



(a) Average File Transfer throughput

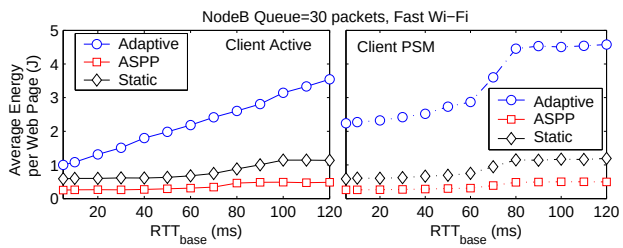



(b) Average Energy spent by the P2P Group Owner in the file transfer

Figure 7: File transfer performance when varying the path delay ($RTT_{base}$). For each figure the left column represents the results when using the Wi-Fi AC_VI configuration, and the right column the results when using the Wi-Fi AC_BK configuration.




(a) Average Web page download time




(b) Average Energy spent by the P2P Group Owner per Web page

Figure 8: Web traffic performance.

Web. The reason is that Web is mostly dominated by the path delay, i.e. $RTT_{base}$, and hence the results obtained showed the same dynamics in our different 3G channel models. For the same reason only the AC_VI configuration is considered in the case of Web. However, we introduce a variation with respect to our File Transfer experiment, we consider the P2P Client to be in active mode or in power save mode (PSM). The reason is that the performance of Web traffic already degrades when a Wi-Fi client is in power save mode [12], therefore it is important to asses whether a power saving protocol used by a P2P Group Owner degrades even further the performance experienced by a P2P Client in power save mode.

Figures 8(a) and 8(b) depict, respectively, the average web page transfer time and the average energy consumed by the P2P Group Owner per Web page. Looking at Figure 8(a), we can see that the P2P Client experiences with ASPP a Web page transfer time equivalent to the one that the P2P Client experiences when it is in power save mode. This is because in both cases the P2P Client can only transmit data during a certain period after the Beacon frame. Regarding the energy spent by the P2P Group Owner to transfer a Web page, Figure 8(b) shows that a very significant energy reduction in the P2P Group Owner can be achieved with *ASPP* with respect to the *Active* and *Static* algorithms, specially as $RTT_{base}$ grows. The reason is that the little traffic generated by the Web application together with the sequential nature of HTTP allows ASPP to operate with presence intervals as low as $L_{P_{min}}$ = 10ms, while the P2P Group Owner remains idle for long times if the *Active* or *Static* configuration are used.

## 4. AMPP: Adaptive Multiple Presence Periods

In the previous section we have presented and analyzed the ASPP algorithm which schedules a single presence period per

Beacon interval and is applicable to the two Wi-Fi Direct power management schemes described in Section 2: Opportunistic Power Save and Notice of Absence. We have seen that ASPP has a low implementation complexity and significantly reduces energy consumption in a P2P Group Owner. However, sometimes it could result in a degraded user experience with respect to an algorithm that configures the P2P Group Owner to be in active mode upon detecting traffic in the network. Our analysis has shown that the main reason why ASPP may degrade throughput is the fact that it schedules a single presence period per Beacon interval which results in an increased RTT experienced by TCP connections. Therefore, a possible way to improve upon the performance of ASPP is to design an algorithm that schedules not only one but *multiple* presence periods within a Beacon interval. In this section we present the design of such an algorithm, which is hereafter referred to as the *Adaptive Multiple Presence Periods (AMPP)* algorithm. Note that while ASPP could be implemented in both Opportunistic Power Save and Notice of Absence protocols, AMPP can only be implemented using the Notice of Absence protocol since this is the only one that allows to schedule multiple presence periods within a Beacon interval.

### *4.1. Algorithm Design*

The main goal of AMPP is to improve the performance experienced by P2P Clients compared to ASPP, whilst keeping as much as possible the power saving achieved by the P2P Group Owner with ASPP. For this purpose the design of AMPP is based on the following building blocks:

1. Adaptively dimension the size of the (multiple) presence periods advertised by AMPP. This is done based on the ASPP algorithm.
2. Design of an algorithm that estimates the raw bandwidth available in the 3G link. Note that the analysis of ASPP has shown that when the buffering in the NodeB is small a TCP connection does not always saturate the 3G link, and the amount of data received in the P2P Group significantly underestimates the bandwidth available in the 3G link.
3. Design of an algorithm that, based on the estimation of the 3G link available bandwidth, adjusts the interval between consecutive presence periods in the Notice of Absence protocol. Hence, controlling how many presence periods should be scheduled within a Beacon interval.

Next, we describe the design of the two new modules used by AMPP: i) the external bandwidth estimation algorithm, and ii) the presence interval adaptation algorithm. During our description we assume that more traffic is transmitted in downlink (network→P2P Client) than in uplink. Later in in this section we will discuss how the presented algorithm could be tailored to the uplink case.

#### *4.1.1. External network bandwidth estimation algorithm*

In this section we describe the algorithm used by AMPP designed to estimate the bandwidth available in an external network. Hereafter we will focus in the case of the external network being a 3G link. However, the same bandwidth estimation algorithm could be applied to different external networks.

According to the architectural constraint set in the previous section, the designed bandwidth estimation algorithm should not interact with the 3G interface. Therefore, we design a passive algorithm that derives the bandwidth available in the 3G link by observing, in the Wi-Fi driver of the P2P Group Owner, the interarrival times of packets arriving from the 3G link[7]. Note that such an algorithm will not interfere with the traffic flowing between the P2P Group and the 3G network.

The design of the bandwidth estimation algorithm is based on the following observations. If the buffer in the NodeB would be always full, the 3G link would modulate the arrival times of packets at the P2P Group Owner, and the 3G bandwidth could be easily estimated. In general though, the buffer in the NodeB may become empty during a TCP connection (specially if this buffer is small), and in this case the interarrival times of packets at the P2P Group Owner will not follow the 3G bandwidth. Indeed, since TCP packets are buffered by the P2P Group Owner and then transmitted during a presence period at the rate of the Wi-Fi network (usually higher than the bottleneck rate), TCP ACK compression can cause bursty increases in the bottleneck queue (NodeB) [13], see Figure 9. Therefore, when a TCP connection is not fully utilizing the bandwidth in the 3G link, a P2P Group Owner typically observes bursts of TCP packets modulated by the bandwidth in the 3G link separated by *inter-burst* times. This characteristic arrival pattern is depicted in Figure 9. Thus, an ideal bandwidth estimation algorithm should discard these inter-burst times and estimate the 3G bandwidth using only the interarrival times within a burst. In order to accomplish this, an important observation is that the inter-burst times observed at the P2P Group Owner are strongly correlated with the presence intervals being used by the P2P Group Owner. This fact is also illustrated in Figure 9 where, assuming a constant $RTT$, the time between the first packet of the two bursts of TCP data arriving at the P2P Group Owner is equal to the presence interval being used by the P2P Group Owner. As a result of the previous observations, we design our bandwidth estimation algorithm based on the following assumption : *within a time interval equal to a presence interval, the P2P Group Owner can only observe one or none inter-burst times*. In the Appendix we formalize this assumption and study under which conditions it holds.

Our bandwidth estimation algorithm is described in detail in Algorithm 1, and is run by a P2P Group Owner before transmitting the Beacon frame. Thus, before building a Beacon frame a P2P Group Owner estimates the bandwidth available in the 3G link during the last Beacon interval and uses this estimation in order to decide how many presence periods should be scheduled in the upcoming Beacon interval. Notice that, estimating the bandwidth available in the 3G link every Beacon interval (e.g. $100ms$) sets an upper limit on the 3G channel variation rate that AMPP will be able to follow. This limitation though, is not in-

[7] Note that we assume that the device's clock granularity is accurate enough to derive meaningful bandwidth estimations from the measured interarrival times.

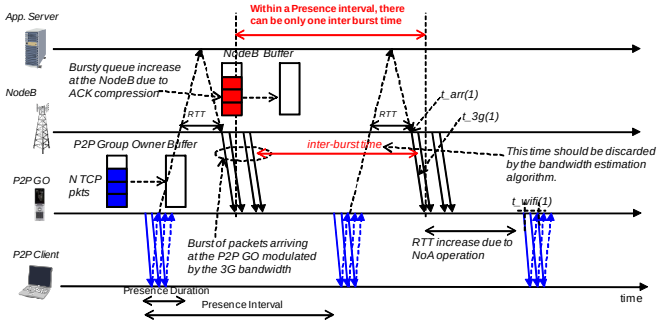



Figure 9: Insight on the bandwidth estimation algorithm: From the perspective of the P2P Group Owner, TCP data packets arrive in bursts modulated by the bandwidth of the 3G link, interleaved with inter-burst times that appear when the buffer in the NodeB becomes empty. One can notice in the figure how within a presence interval, there can only be one inter-burst time.

trinsic to AMPP but to the Notice of Absence protocol which can only update the current schedule every Beacon interval. In addition, in order to follow the variations in the 3G bandwidth, our bandwidth estimation algorithm uses an EWMA filter updated with the bandwidth estimated every Beacon interval (line 28 in Algorithm 1). Next, we describe the details of our bandwidth estimation algorithm.

A P2P Group Owner records in each Beacon interval the amount of received data, *total_bits*, the number of packets, *num_pkts*, and a list of the interarrival times between packets received from the 3G link, *interarrvs*($i$). Note that this list can be fairly small, e.g. assuming a speed of 7.2Mbps in the 3G link and 1.5KB size packets, only 60 packets could arrive within a Beacon interval of 100ms. Then, the list *interarrvs*($i$) is splitted in contiguous subintervals according to the current presence interval (*interval*) used by the P2P Group Owner (line 18).

For each of these subintervals the maximum interarrival time ($interarrv_{max}$) is tagged as a potential *inter-burst* time (line 16), assuming that within a subinterval *there can only be one inter-burst time*. Then it is decided whether $interarrv_{max}$ is indeed an inter-burst time and should therefore be discarded by comparing it against the average (*avg*) and standard deviation (*std*) of the other interarrival times contained in *subintvl*. Specifically, if $interarrv_{max} > avg+2std$ we heuristically consider $interarrv_{max}$ to be an inter-burst time and substitute it by *avg* (line 23). Notice that the challenge is to distinguish between the long tail in the distribution of interarrival times caused by the 3G link, and the inter-burst times caused by an empty queue in the NodeB. Since the Chebychev inequality [15] establishes that at maximum $\frac{1}{k^2}$ of the distribution values can be separated more than $k\sigma$ from the distribution mean, setting $k = 2$ hints that 75% of the interarrival times due to the 3G bandwidth should be correctly accounted for, while still providing a reasonable protection against inter-burst times. Notice that this heuristic allows us to follow variations in the 3G link, because our *threshold* to detect inter-burst times, i.e. $avg + 2std$, adapts to the statistics of the 3G link. Obviously the presented heuristic can fail, if the time to transmit a packet over a 3G link is higher than our threshold, although the Chebychev inequality should limit the probability of that happening, or if an inter-burst time is actually smaller than the threshold, although in that case the impact of mistaking an inter-burst time with an interarrival time caused by the 3G link should not be significant. In the next section we will evaluate how effective this heuristic is on estimating the bandwidth available in the 3G link.

**Algorithm 1:** estimate_3g_bw()

1 – *Variables definition*
2 *interarrvs* ← array containing the interarrivals of the packets received in the last Beacon interval.
3 *total_bits* ← accumulated sizes of the packets received in the last Beacon interval.
4 *num_pkts* ← Number of packets received during the last Beacon interval.
5 $t_{b2b}$ ← interarrival time below which two packets are considered back to back.
6 $M$ ← threshold on consecutive back to back packets.
7 *interval* ← Current operating interval of the algorithm.
8 *subintvl* ← {}
9 – *Routine executed when a SP completes*
10 $t_{total}, t_{subint}, m, interarrv_{max} \leftarrow 0, avg_{last} \leftarrow -1$
11 **for** $i = 0$ *to* $num\_pkts - 1$ **do**
12  **if** $interarrvs(i) > t_{b2b}$ **then**
13  $t_{total} \leftarrow t_{total} + interarrvs(i)$
14  $t_{subint} \leftarrow t_{subint} + interarrvs(i)$
15  **if** $interarrvs(i) > interarrv_{max}$ **then**
16  $interarrv_{max} \leftarrow interarrvs(i), i_{max} \leftarrow i, m_{max} \leftarrow m$
17  $subintvl \leftarrow \{subintvl, interarrvs(i)\}, m \leftarrow m + 1$
18  **if** $i = num\_pkts - 1$ *or* $m > 1, t_{subint} + interarrvs(i+1) > interval$ **then**
19  **if** $m > 1$ *and* $!interarrvs(i_{max}+1, ..., i_{max}+M) < t_{b2b}$ **then**
20  $avg \leftarrow \frac{t_{subint} - interarrv_{max}}{m-1}, avg_{last} \leftarrow avg$
21  $std \leftarrow \sqrt{\frac{1}{m-1} \sum_{j=1, j!=m_{max}}^{m} (subintvl(j) - avg)^2}$
22  **if** $interarrv_{max} > avg + 2std$ **then**
23  $t_{total} \leftarrow t_{total} - interarrv_{max} + avg$
24  **else if** $m = 1$ *and* $avg_{last} != -1$ **then**
25  $t_{total} \leftarrow t_{total} - interarrv_{max} + avg_{last}$
26  $m \leftarrow 0, t_{subint} \leftarrow 0, interarrv_{max} \leftarrow 0, subintvl \leftarrow \{\}$
27 **if** $num\_pkts > 0$ *and* $t_{total} > 0$ **then**
28  $3g\_bw \leftarrow \alpha \times 3g\_bw + (1 - \alpha) \times \frac{total\_bits}{t_{total}}$
29 $num\_pkts \leftarrow 0, total\_bits \leftarrow 0$

We set one exception to the previous rule if there are $M$ consecutive *back to back* packets after $interarrv_{max}$ (line 19), where back to back packets are defined as two consecutive packets arriving within $t_{b2b}$. The reason for this exception is that we empirically observed that a pattern of a long interarrival time followed by several consecutive back to back packets is typically caused by the Hybrid-ARQ protocol in the 3G link [14], which buffers frames to guarantee in order delivery, and not by an empty queue in the NodeB. In our implementation we set $M = 2$ and $t_{b2b} = 2ms$ which is the slot size used in HSDPA. In addition, if there is only one sample in *subintvl*, this value is not trusted, and instead the last recorded average interarrival time is considered, $avg_{last}$ (lines 24-25).

Finally, note that by adjusting the value of the variables $t_{b2b}$ and $M$ Algorithm 1 could be tailored to different cellular technologies, like LTE or WiMAX, which operate based on principles similar to the ones outlined in this section.

*4.1.2. Adapting the Number of Presence Intervals per Beacon*

In this section, we describe how AMPP uses the input provided by our bandwidth estimation algorithm in order to decide how many presence periods should be scheduled every Beacon interval. For this purpose what AMPP does is to adjust the value of the presence interval advertised in the Notice of Absence schedule being broadcasted in the Beacon, i.e. if $interval = 25ms$ then there will be four presence periods within a Beacon interval. The main ideas behind our interval adaptation algorithm are as follows:

- *Underutilization Detection*: AMPP maintains two estimates: i) an estimate of the bandwidth available in the 3G link obtained with our bandwidth estimation algorithm, $3g\_bw$, and ii) an estimate of the traffic flowing between the P2P Group and the 3G link, $thr$. In order to detect if the 3G link is being underutilized, AMPP computes the ratio between its two estimates, i.e. $ratio = \frac{thr}{3g\_bw}$.
- *Reaching Utilization Target*: AMPP takes as input parameter a desired utilization target of the 3G link, i.e. $ratio_{min}$. Based on this, AMPP checks if the desired utilization is satisfied and adjusts the selected presence interval in the following way: i) if the 3G link is underutilized, i.e. $ratio < ratio_{min}$, AMPP will decrease its operating interval (increase the number of presence periods) within a Beacon interval in an attempt of reducing the RTT experienced by a TCP connection and improve throughput, ii) if utilization is sufficient, i.e. $ratio > ratio_{min}$, AMPP will increase its operating interval to check whether a higher presence interval still delivers the desired utilization. Note that higher presence intervals are preferred since they allow for longer sleep times and thus, higher power saving.

In addition to the previous main algorithm principles more issues need to be considered in practice to achieve the desired functionality. Our first design decision is that, in order to be able to refresh the used presence interval every Beacon interval, AMPP will only select presence intervals that are submultiple of the Beacon interval, i.e. $interval = \frac{BI}{k_{curr}}$, where $k_{min} \leq k_{curr} \leq k_{max}$, $k_{curr}, k_{min}, k_{max} \in \mathbb{Z}$. Notice that if an absence period would overlap with a Beacon frame, a P2P Client might skip the Beacon, missing the updated schedule. A detailed description of our interval adaptation algorithm is provided in Algorithm 2 and summarized next.

We start discussing the case where $ratio < ratio_{min}$ (lines 9 to 29). In this case AMPP considers the link to be underutilized and thus, tries to decrease the current presence interval. However, in order to reduce spurious updates, we introduce a memory, $count_{down_{max}}$, which establishes a number of consecutive Beacon intervals before AMPP updates the current presence interval. We can see between lines 20 and 29 how AMPP decreases the presence interval by increasing the parameter $k_{curr}$, where $interval = \frac{BI}{k_{curr}}$. In addition, under special circumstances AMPP may as well decide to keep the P2P Group Owner awake during the upcoming Beacon interval. These circumstances are: i) if the current presence duration, $L_P$, is above the selected interval (line 24), or ii) if the desired utilization, $ratio_{min}$, is not achieved even when operating at the minimum allowed interval, $k_{max}$, (line 28). Notice that when staying awake the P2P Group Owner does not publish any Notice of Absence schedule in the Beacon frame.

**Algorithm 2:** adjust_interval()

1 – *Variables definition*
2 $ratio_{min} \leftarrow$ 3G link utilization threshold.
3 $count_{up_{max}}/count_{down_{max}} \leftarrow$ Variables to control the speed of interval increase/decrease.
4 $k_{min}/k_{max} \leftarrow$ Variables that control the maximum/minimum presence intervals.
5 $n\_int_{max} \leftarrow$ Threshold on the maximum number of presence periods without data before updating the used interval.
6 – *Routine executed every Beacon interval*
$3g\_bw \leftarrow estimate\_3g\_bw()$, $thr \leftarrow estimate\_thr()$
7 **if** $3g\_bw, thr > 0$, $n\_int\_no\_data < n\_int_{max}$ **then**
8 $ratio \leftarrow \frac{thr}{3g\_bw}$
9 **if** $ratio < ratio_{min}$ **then**
10 $count_{up} \leftarrow 0$, $count_{down} \leftarrow count_{down} + 1$
11 **if** $count_{down} = count_{down_{max}}$ **then**
12 $count_{down} \leftarrow 0$
13 **if** *active is* 1 **then**
14 $\Delta_{r_{act}} \leftarrow \frac{ratio - ratio_{last}}{count_{down_{max}}}$
15 **if** $\Delta_{r_{act}} < \max\{0, (1+\gamma)\Delta_{r_{slp}}\}$ **then**
16 $count_{act} \leftarrow count_{act} + 1$
17 **if** $count_{act} = n\_int_{max}$ **then**
18 $active \leftarrow 0$, $L_P \leftarrow L_{P_{last}}$, $k_{curr} \leftarrow k_{last}$, $last \leftarrow up$, $count_{act} \leftarrow 0$
19 $ratio_{last} \leftarrow ratio$
20 **else**
21 $go\_to\_active \leftarrow (last = down$ and $ratio > (1+\gamma)ratio_{last})$ or $(last = up$ and $ratio < (1-\gamma)ratio_{last})$
22 **if** $k_{curr} < k_{max}$ **then**
23 $k_{curr} \leftarrow \min\{k_{max}, k_{curr} + 1\}$, $ratio_{last} \leftarrow ratio$, $last \leftarrow down$
24 **if** $\frac{BI}{k_{curr}} < L_P$ **then**
25 $active \leftarrow 1$, $\Delta_{r_{slp}} \leftarrow \frac{ratio - ratio_{last}}{count_{down_{max}}}$, $count_{act} \leftarrow 0$
26 **else**
27 $active \leftarrow 0$, $L_{P_{last}} \leftarrow L_P$, $k_{last} \leftarrow k_{curr}$
28 **else if** *go_to_active* **then**
29 $active \leftarrow 1$, $\Delta_{r_{slp}} \leftarrow \frac{ratio - ratio_{last}}{count_{down_{max}}}$, $ratio_{last} \leftarrow ratio$, $last \leftarrow down$, $count_{act} \leftarrow 0$
30 **else**
31 $count_{down} \leftarrow 0$, $count_{up} \leftarrow count_{up} + 1$
32 **if** $count_{up} = count_{up_{max}}$ **then**
33 $count_{up} \leftarrow 0$, $last \leftarrow up$, $ratio_{last} \leftarrow ratio$
34 **if** *active is* 1 **then**
35 $active \leftarrow 0$, $L_P \leftarrow L_{P_{last}}$, $k_{curr} \leftarrow k_{last}$
36 **else**
37 $k_{curr} \leftarrow \max\{k_{min}, k_{curr} - 1\}$
38 **else if** $n\_int\_no\_data \geq n\_int_{max}$ **then**
39 $k_{curr} \leftarrow \max\{k_{min}, k_{curr} - 1\}$, $n\_int\_no\_data \leftarrow 0$, $last \leftarrow up$, $ratio_{last} \leftarrow ratio$, $count_{up} \leftarrow 0$, $count_{down} \leftarrow 0$
40 $interval \leftarrow \frac{BI}{k_{curr}}$

The P2P Group Owner should be in active mode (awake) only if the Wi-Fi interface is busy transmitting packets. Hence, before deciding to keep the P2P Group Owner awake, Algorithm 2 records in the variable $\Delta_{r_{slp}}$ the speed of increase of $ratio$. Thus, if the 3G link continues to be underutilized, i.e. $ratio < ratio_{min}$, but the P2P Group Owner is in active mode

(lines 13-19), the algorithm checks whether the speed of increase of *ratio*, $\Delta_{r_{act}}$, is above $\Delta_{r_{slp}}$. If that is not the case, keeping the P2P Group Owner awake is not being effective and the P2P Group Owner goes back to normal power saving operation. In order to effectively compare $\Delta_{r_{act}}$ and $\Delta_{r_{slp}}$, a hysteresis is introduced controlled by the $\gamma$ parameter.

Now we consider the case where the 3G link is being sufficiently utilized, i.e. $ratio \geq ratio_{min}$ (lines 30 to 37). In this case AMPP increases the operating interval by means of decreasing the variable $k_{curr}$, where $interval = \frac{BI}{k_{curr}}$. Note that, again, a memory has been introduced to reduce spurious updates, $count_{up_{max}}$. In addition, if the P2P Group Owner happened to be in active mode at that moment, the last presence interval and presence duration used before switching to active mode are restored (line 35).

One more practical aspect to be considered is the case where applications, e.g. Web, do not offer enough load to saturate the 3G link. In this case the previous logic would drive the P2P Group Owner towards using small presence intervals, which would not be energy efficient. In order to counter this effect, the variable *n_int_no_data* is defined in Algorithm 2, which accounts for the number of scheduled presence periods during the last Beacon interval where no data was transmitted. Thus, if $n\_int\_no\_data > n\_int_{max}$ the algorithm quickly increases the operating interval (lines 38-39).

Finally, the design of AMPP allows to trade-off energy and performance by configuring the parameters $ratio_{min}$, $k_{min}/k_{max}$ (maximum/minimum presence intervals), and $count_{down_{max}}/count_{up_{max}}$ (speed of decrease/increase). For instance applications with strict QoS requirements, e.g. VoIP, can be easily accomodated within the presented framework, by configuring the presence interval, i.e. $k_{min}/k_{max}$, to operate within limits that fulfill their delay constraints. These requirements can be conveyed by a P2P Client to a P2P Group Owner using the Notice of Absence protocol by sending a P2P Presence Request (see Section 2). In the next section we will study the effect of the previous parameters.

### *4.2. AMPP: Algorithm Evaluation*

In this section we will evaluate the performance of AMPP. Like in the case of ASPP, we will start analyzing the algorithm dynamics, to then present a steady state evaluation with popular applications.

#### *4.2.1. Algorithm dynamics*

We start presenting the results of an experiment where two P2P Clients connected to a mobile phone acting as P2P Group Owner download a 50MB file through the 3G network. The Typical Urban channel model is considered in this experiment. The two File Transfers from each P2P Client experience different connection delays, $RTT_{base_1} = 40ms$ and $RTT_{base_2} = 20ms$, and a 30 packets per flow buffer is considered in the NodeB.

Figure 10(a) illustrates the dynamics of the presence interval (solid line) and presence duration (dashed line) advertised by the P2P Group Owner in this experiment. As it can be observed, before 60 seconds (no traffic present) the P2P Group Owner advertises a big interval of 100ms and a small duration of 10ms. This is the default operation when there is no traffic in the network which allows a P2P Group Owner to operate in a very power efficient way. After 60 seconds, when the first File Transfer starts (signaled as con-1 in the figure), the algorithm starts adjusting the advertised presence intervals and durations. In this case AMPP needs to schedule small intervals in order to keep the throughput of the connection above the configured ratio ($ratio_{min} = 0.8$). In order to realize why this is needed recall how ASPP's throughput degraded with a buffer of 30 packets in the NodeB, shown in Figure 5(a).

At 140 seconds the second connection starts (con-2) and the aggregate load at the NodeB increases which, together with a slight improvement in the 3G channel condition (that can be observed in Figure 10(b)), allows the P2P Group Owner to achieve the required throughput using bigger intervals and presence durations. After 200 seconds the first file transfer completes and the P2P Group Owner is forced to use again smaller intervals to maintain throughput. However, it is interesting to notice in this case that since $RTT_{base_2} < RTT_{base_1}$, the P2P Group Owner can now operate with bigger intervals than with the first File Transfer.

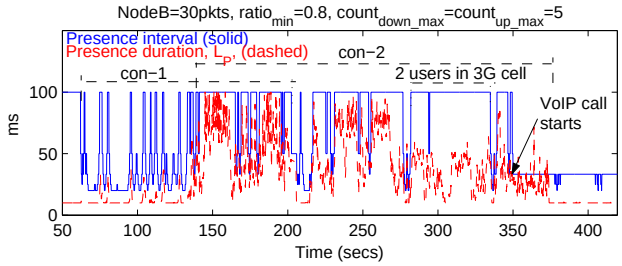
Between 280 and 330 seconds, a second user enters the 3G cell that was serving the P2P Group Owner halving the bandwidth available to the mobile phone (P2P Group Owner) in the 3G link [8]. AMPP's bandwidth estimation algorithm detects this bandwidth reduction (see Figure 10(b)) and, in order to save energy, the P2P Group Owner starts operating with a bigger interval and a smaller presence duration.

Finally, at 350 seconds a VoIP call enters the system. The VoIP client sends a P2P Presence Request frame to the P2P Group Owner requesting the P2P Group Owner to be present at least every 40ms in order to maintain the QoS needed in the Voice call. AMPP can easily accomodate the incoming call by configuring $k_{min} = 3$ upon receiving a P2P Presence Request from the P2P Client, which establishes a maximum presence interval of $33ms$.

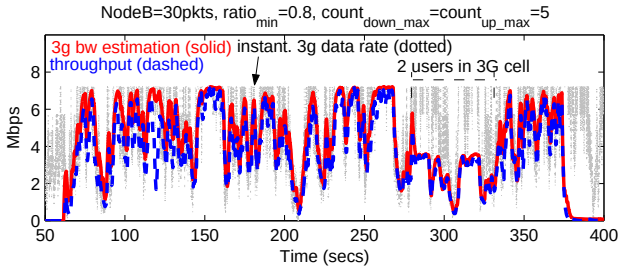
Figure 10(b) depicts for the same experiment the instantaneous rate in the 3G link, the $3g\_bw$ estimation and the throughput delivered to the P2P Client. Notice how by means of adapting interval and duration, AMPP is able to deliver to the P2P Client all the bandwidth available in the 3G channel.

We study now how the AMPP algorithm behaves when not one but several TCP connections are established concurrently through the P2P Group Owner. Figure 10(c) depicts the result of an experiment where we analyze the average presence interval and duration selected by AMPP, together with the ratio between the throughput delivered to the P2P Clients and the bandwidth available in the 3G link. In the experiment we incrementally increase the number of P2P Clients connected to the P2P Group Owner from 1 to 8. Each of the P2P Clients in our experiment retrieves a 50MB file using a TCP connection that experiences a different path delay ($RTT_{base}$) between $20ms$ and

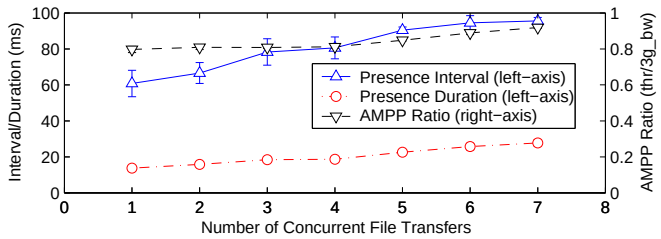
[8] In our simulation the 3G cell operates using a Round Robin scheduler



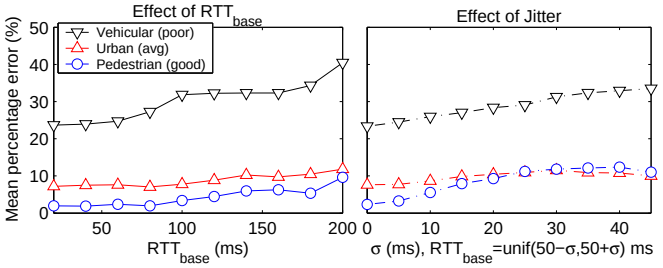

(a) Duration and Interval dynamics. Solid/Dashed lines depict the variation of the presence interval/duration.



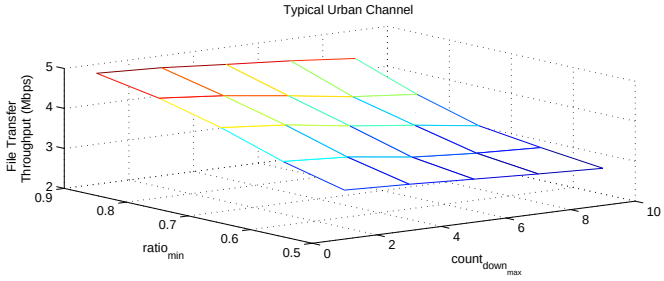

(b) $3g\_bw$ and achieved throughput. Notice how the $3g\_bw$ estimation and the delivered throughput follow the 3G available bandwidth.



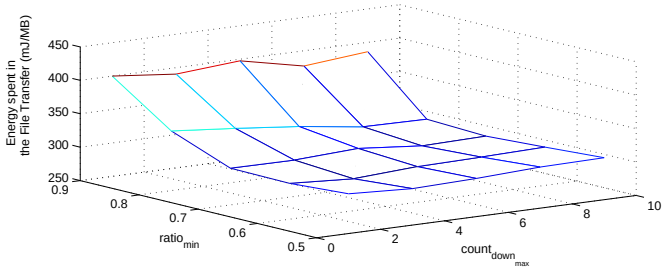

(c) Dynamics when increasing the number of concurrent file transfers. Notice that *AMPP Ratio* is plotted against the right axis.




(d) 3G Bandwidth Estimation Accuracy. The solid lines are plotted against the lower x-axis, and the dashed ones against the upper x-axis.




(e) AMPP Throughput as a function of $ratio_{min}$, $count_{down_{max}}$ and $count_{up_{max}}$




(f) AMPP Energy as a function of $ratio_{min}$, $count_{down_{max}}$ and $count_{up_{max}}$

Figure 10: Dynamics of the *AMPP* algorithm.

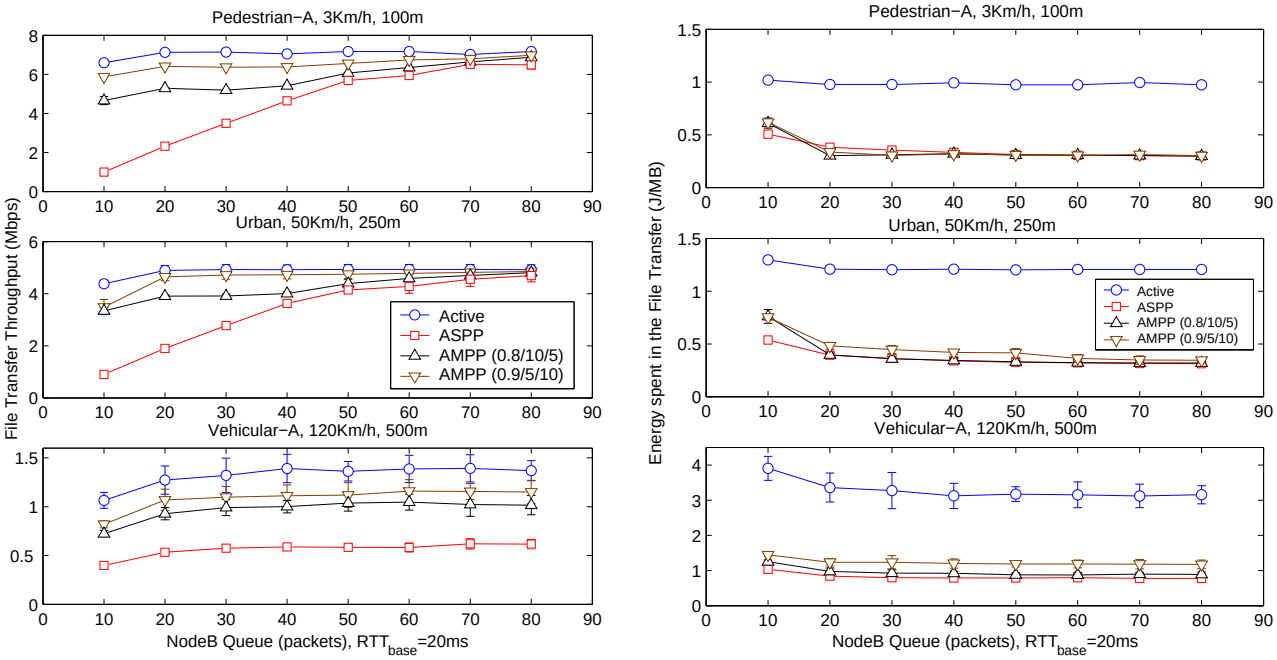
$80ms$. In the figure the average presence interval and duration are plotted against the left y-axis while the AMPP ratio is plotted against the right y-axis. Notice that from the perspective of AMPP it does not matter if packets come from the same or different connections.

From the results in Figure 10(c) we can observe that in all cases AMPP is able to deliver the required ratio of the 3G bandwidth ($ratio_{min} = 0.8$). However, if only one client is present, AMPP needs to use smaller presence intervals (schedule more presence periods within a Beacon frame). The reason is that, as the number of concurrent TCP connections increases, the buffer in the NodeB becomes empty less often. Thus, AMPP benefits from this phenomenon by operating with higher presence intervals which is more energy efficient.

In order to evaluate the accuracy of our bandwidth estimation algorithm, Figure 10(d) presents the mean percentage error achieved by our bandwidth estimation algorithm in the following experiments. A first experiment (left sub-graph) where we increase the value of $RTT_{base}$ while keeping the same buffer size in the NodeB. Notice that a higher path delay increases the bandwidth delay product and hence the likelihood of having an empty buffer in the NodeB, which is the case that should be detected by our bandwidth estimation algorithm. A second experiment (right sub-graph), where we introduce an increasing jitter around an average path delay of 50ms.

Notice that, as shown in the Appendix, jitter is one of the conditions that challenge the fundamental assumption of our bandwidth estimation algorithm, i.e. the existence of a single inter-burst time within a presence interval. We repeated the previous two experiments for our different 3G channel models. As seen in Figure 10(d) the error of our 3G bandwidth estimation algorithm is kept below 10% for the whole parameter range in the Pedestrian-A and Typical Urban channels. Only in the fastly varying Vehicular-A channel, the estimation error reaches values around 30% because the TCP connection often stalls in this case not providing enough packets for AMPP to have a reliable estimation.

In order to gain a deeper understanding on the influence




(a) Average File transfer throughput.

(b) Average Energy spent by the P2P Group Owner in the file transfer

Figure 11: File transfer performance when varying the buffering in the NodeB.

of the $ratio_{min}$, $count_{down_{max}}$ and $count_{up_{max}}$ AMPP parameters, Figures 10(e) and 10(f) depict respectively the throughput and energy spent in the P2P Group Owner in the Typical Urban channel, when a P2P Client downloads a 50MB File and a 30 packets buffer is considered in the P2P Group Owner. In this experiment, $ratio_{min}$ is varied between 0.5 and 0.9 and $count_{down_{max}}$ is varied between 1 and 9, having $count_{up_{max}} = 10 - count_{down_{max}}$. We can see looking at the previous figures how increasing $ratio_{min}$ or decreasing $count_{down_{max}}$ and increasing $count_{up_{max}}$ increases both throughput and energy consumption, being $ratio_{min}$ the strongest parameter, since it results in the P2P Group Owner offering more presence intervals within a Beacon interval[9].

Finally, note that AMPP could be applied also to uplink File Transfers. In this case though, AMPP should estimate the 3G uplink available bandwidth from the returning TCP ACKs. We have not further studied this approach because our experiments reveal that given the usually limited uplink 3G bandwidth, ASPP (no interval adaptation) suffices to provide an adequate performance in this case.

### 4.2.2. Steady State Evaluation

In this section we evaluate the performance of our *AMPP* algorithm with popular applications like File Transfers and Web traffic. The same experiments defined in section 3.2.3 for the ASPP evaluation are used here to study AMPP's steady state performance. In this case we will focus in the results obtained with the AC_VI settings since this was the Wi-Fi configuration providing the best energy efficiency in ASPP but exhibiting the highest performance degradation. It is therefore the goal of AMPP to overcome the performance problems of ASPP while achieving similar energy efficiencies.

Two different configurations for AMPP that illustrate the effect of its configurations parameters will be considered:

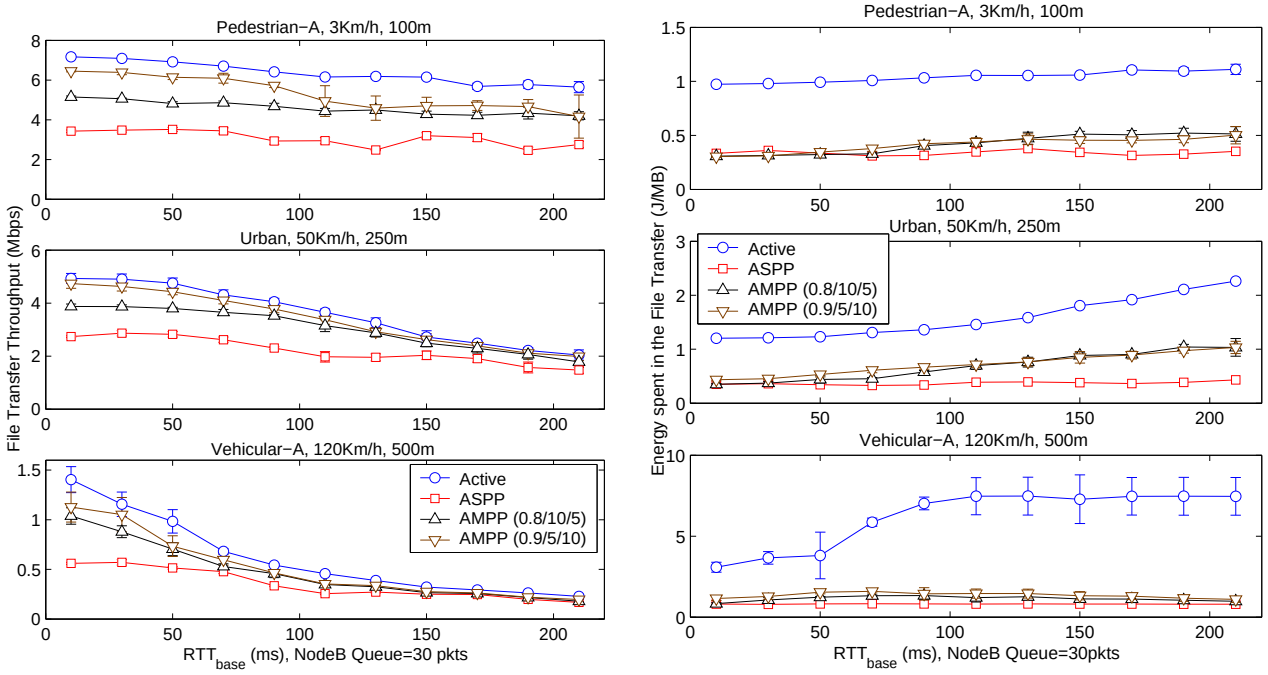
- *QoS configuration*: $ratio_{min}$, $count_{down_{max}}$ and $count_{up_{max}}$ set to 0.9, 5 and 10 respectively.
- *Energy configuration*: $ratio_{min}$, $count_{down_{max}}$ and $count_{up_{max}}$ set to 0.8, 10 and 5 respectively.

The maximum and minimum allowed presence intervals $k_{min}/k_{max}$ are set to $20ms$ and $100ms$ respectively. Finally, the parameters $n\_int_{max}$ and $\gamma$ are empirically set to 3 and 0.1 respectively.

*NodeB Buffering Variation Experiment*

Figures 11(a) and 11(b) depict the performance of the considered algorithms during a File Transfer when varying the buffering available in the NodeB and setting the path delay equal to $RTT_{base} = 20ms$. In the figure it can be clearly observed how for all of our considered 3G channel models, i.e. Pedestrian-A, Typical Urban and Vehicular-A, the AMPP algorithm significantly outperforms ASPP introducing only a marginal power consumption increase. In particular, we note that the AMPP QoS configuration delivers in all cases a File Transfer throughput very close to the one delivered when the P2P Group Owner is always active and at the same time an energy consumption

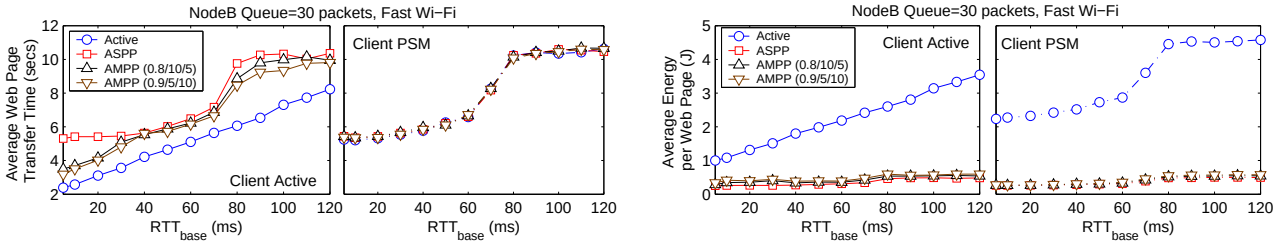
[9]Notice though that the energy profile depicted in Figure 10(f) may slightly vary depending on the power characteristics of the considered Wi-Fi chipset.




(a) Average File transfer throughput.

(b) Average Energy spent by the P2P Group Owner in the file transfer

Figure 12: File transfer performance when varying the path delay ($RTT_{base}$).




(a) Average Web page download time

(b) Average Energy spent by the P2P Group Owner per Web page

Figure 13: Web performance.

very close to the ASPP one. The reason why AMPP is able to provide this improved performance is that it successfully identifies when the 3G link is being underutilized and in such cases schedules additional presence periods within Beacon intervals.

*Path Delay Variation Experiment*

Figures 12(a) and 12(b) depict the performance of the algorithms under study when varying the path delay, $RTT_{base}$, and fixing the buffer size in the NodeB to 30 packets per flow. Like in the previous experiment, AMPP significantly outperforms ASPP for all the considered 3G chanel models operating close to the performance of an always Active solution while providing energy efficiencies similar to those of ASPP. The only exception is the case when $RTT_{base}$ is large for the Pedestrian-A (good) and Typical Urban (average) channels. In these cases AMPP results in a higher energy consumption than ASPP, although still much lower than the one of the Active algorithm. The reason for a higher energy consumption in these cases is that TCP can not fill the path's bandwidth delay product and AMPP ends up operating with reduced intervals, i.e. $20ms$ in our experiments, with a small portion of the TCP congestion window being transmitted in each presence period.

*Web Traffic Experiment*

Finally, we complete our evaluation by studying in Figure 13 how AMPP performs with Web traffic. In the figure it can be observed how AMPP reduces the time to transfer a Web page compared to ASPP, specially when $RTT_{base} < 50ms$. The reason is that since Web traffic usually can not fill up the 3G link, *ratio* falls below $ratio_{min}$ and AMPP decreases the used presence interval. However, when an interval smaller than $RTT_{base}$, i.e, the delay between the NodeB and the Web server, is selected, the P2P Group Owner schedules some presence periods where no data is transmitted and hence, AMPP increases

the used presence interval because $n\_int\_no\_data > n\_int_{max}$. Therefore, when $RTT_{base} < 50ms$, the P2P Group Owner ends up scheduling a presence interval that ocillates around $RTT_{base}$, which is obviously a desirable behavior in the case of Web. If $RTT_{base} > 50ms$ AMPP often schedules its immediately higher interval which is $100ms$ resulting in a behaviour similar to ASPP. Regarding energy, we can see in Figure 13(b) how adapting the presence interval does not significantly increase the energy consumption of AMPP with respect to the one of ASPP.

## 5. Summary and Conclusions

Bringing device to device connectivity to a mass market is a key milestone in Wi-Fi's evolution roadmap. For this purpose, the Wi-Fi Alliance has recently developed the Wi-Fi Direct technology which should become the key device to device communication enabler. Among the different requirements to be fulfilled by this technology, battery usage efficiency is a central one due to the high penetration of Wi-Fi in mobile devices.

In this paper we have analysed the two power saving protocols defined in Wi-Fi Direct allowing APs to save power, Opportunistic Power Save (OPS) and Notice of Absence (NoA), and designed two algorithms to efficiently use them: Adaptive Single Presence Period (ASPP) and Adaptive Multiple Presence Periods (AMPP). These algorithms allow a portable device implementing Wi-Fi Direct (e.g. a mobile phone) to offer access to an external network (e.g. a cellular network) while addressing the trade-off between performance and energy consumption in a configurable manner. ASPP and AMPP performance has been thoroughly analyzed by OPNET simulations considering both their dynamic and steady state behaviour.

From our results the following conclusions can be drawn: i) ASPP and AMPP successfully manage to significantly reduce the power consumption of Wi-Fi Direct devices acting as Access Points (50-90%) without introducing a major user experience degradation, ii) the NoA protocol in combination with the AMPP algorithm delivers a close to optimal user experience and energy efficiency, and iii) AMPP's tuning parameters can be configured to prioritize either energy saving or user experience according to the device manufaturer preferences.

As future work, we consider two main directions to improve the presented algorithms. First, AMPP could be extended to adaptively tune its multiple parameters based on remaining battery capacity. Second, P2P Group Owner algorithms could be enhanced to control the fraction of bottleneck bandwidth obtained by its own applications and the one delivered to the associated P2P Clients.

## 6. Acknowledgments

The research leading to these results has been partially funded by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement N° 257263 [21].

## References

[1] IEEE 802.11-2007 Standard, *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, 2007.
[2] Wi-Fi Alliance, Quality of Service (QoS) Task Group, *Wi-Fi Multimedia (including WMM PowerSave) Specification v1.1*, 2005.
[3] Wi-Fi Alliance,*Wi-Fi Protected Setup Specification v1.0h*, Dec. 2006.
[4] Wi-Fi Alliance, http://www.wi-fi.org.
[5] Wi-Fi Alliance, P2P Technical Group, *Wi-Fi Peer-to-Peer (P2P) Technical Specification v1.0*, December 2009.
[6] Yangyang Li, Terence D. Todd and Dongmei Zhao, *Access Point Power Saving in Solar/Battery Powered IEEE 802.11 ESS Mesh Networks*, Second International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks (QShine), 2005.
[7] Todd, T.D., Sayegh, A.A., Smadi, M.N. and Dongmei Zhao, *The need for access point power saving in solar powered WLAN mesh networks*, IEEE Network, May-June 2008.
[8] http://www.opnet.com.
[9] *Enhanced UMTS Radio Access Network Extensions for NS2 (EURANE)*, http://eurane.ti-wmc.nl/eurane.
[10] Atheros AR6001, *http://www.atheros.com*.
[11] *Average Web Page Size*, http://www.websiteoptimization.com/.
[12] R.Krashinsky and H.Balakrishnan, *Minimizing energy for wireless web access with bounded slowdown*, Proceedings of the eighth Annual International Conference on Mobile Computing and Networking (MOBICOM), September 2002.
[13] J.C. Mogul, *Observing TCP Dynamics in Real Networks*, ACM Sigcomm Computer Communication Review, Volume 22 Issue 4, Oct. 1992.
[14] *3GPP TS 25.321 V6.5.0*, Technical Specification Group Radio Access Network; Medium Access Control (MAC) protocol specification (Release 6), 2006.
[15] Grimmett and Stirzaker, *Probability and Random Processes*, Oxford Science Publications, ISBN 0 19 853665 8, Problem 7.11, proof available at: http://www.mcdowella.demon.co.uk/Chebyshev.html.
[16] S. Shenker, L. Zhang and D. Clark, *Some Observations on the Dynamics of a Congestion Control Algorithm*, Computer Communication Review 20(5):30-39, October,1990.
[17] Third Generation Partnership Project (3GPP) High Speed Packet Access (HSPA) Specifications, *http://www.3gpp.org/HSPA*.
[18] List of deployed HSDPA networks worldwide, *http://en.wikipedia.org/wiki/List_of_HSDPA_networks*.
[19] IEEE 802.11n Standard, *IEEE Standard for Information technology–Telecommunications and information exchange between systems–Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput*, 2009.
[20] K. Miyuki, *Radio Communication Systems*, Patent from Sony Corp., JP 2004336401.
[21] EU FP7 Project on FLexible Architecture for Virtualizable future wireless Internet Access (FLAVIA), N° 257263, http://www.ict-flavia.eu.

## Appendix

Consider in Figure 9 a presence period at time $t_P$ with $N$ packets in the P2P GO belonging to $M \geq 1$ TCP connections. Each packet, $i = 1...N$, will result in one or two (if this packet is the last of the current congestion window) new packets arriving at the NodeB at time $t_{arr}(i) = t_P + t_{wifi}(i) + RTT(i)$, where $RTT(i)$ is the RTT experienced by packet $i$, excluding the time spent in the Wi-Fi network, and $t_{wifi}(i)$ is the time to transmit the TCP Data and TCP Ack in the Wi-Fi network. Consider $\{t_{arr}(k)\}$ to be the ordered set of arrivals at the NodeB. Algorithm 1 assumes that the buffer at the NodeB never gets empty while processing these packets, which is true unless $\exists k$ such that $t_{arr}(k) - t_{arr}(1) = \Delta_{wifi}(k) + \Delta_{RTT}(k) > \sum_{j=1}^{k-1} t_{3g}(j)$, where $t_{3g}(j)$ is the time of transmitting packet $j$ over the 3G link, $\Delta_{wifi}(k) = t_{wifi}(k) - t_{wifi}(1)$ and $\Delta_{RTT}(k) = RTT(k) - RTT(1)$.

Given the fact that typically the Wi-Fi bandwidth is much higher than the 3G bandwidth, in general $\Delta_{wifi}(k)$ is significantly smaller than $\sum_{j=1}^{k-1} t_{3g}(j)$. In addition, $\Delta_{RTT}(k)$ contains the RTT difference between two packets separated $k$ positions in the set of arrivals at the NodeB. This difference can be assumed to be small if these packets belong to the same TCP connection and potentially high otherwise. However, it is interesting to notice that even packets belonging to different connections, as shown in [16], tend to get clustered in a bottleneck queue, which would favor the bursty behavior assumed in Figure 6. We have extensively evaluated whether this condition holds in practice in Figure 10(d), where a random $RTT(i)$ was introduced for each packet. Finally, notice in Figure 6 that the next burst of packets at the P2P GO will not arrive before $t_P$ + *interval*, therefore in general only one or no inter-burst times will be present within a presence interval, which is the basic design assumption of Algorithm 1.