WiFi-Opp: Ad-Hoc-less Opportunistic Networking

Sacha Trifunovic, Bernhard Distl, Dominik Schatzmann, Franck Legendre
Communication Systems Group
ETH Zurich, Switzerland
lastname@tik.ee.ethz.ch

ABSTRACT

Opportunistic networking offers many appealing application perspectives from local social-networking applications to supporting communications in remote areas or in disaster and emergency situations. Yet, despite the increasing penetration of smartphones, opportunistic networking is not feasible with most popular mobile devices. There is still no support for WiFi Ad-Hoc and protocols such as Bluetooth have severe limitations (short range, pairing). We believe that WiFi Ad-Hoc communication will not be supported by most popular mobile OSes (i.e., iOS and Android) and that WiFi Direct will not bring the desired features.
Instead, we propose WiFi-Opp, a realistic opportunistic setup relying on (i) open stationary APs and (ii) spontaneous mobile APs (i.e., smartphones in AP or tethering mode), a feature used to share Internet access, which we use to enable opportunistic communications. We compare WiFi-Opp to WiFi Ad-Hoc by replaying real-world contact traces and evaluate their performance in terms of capacity for content dissemination as well as energy consumption. While achieving comparable throughput, WiFi-Opp is up to 10 times more energy efficient than its Ad-Hoc counterpart. Eventually, a proof of concept demonstrates the feasibility of WiFi-Opp, which opens new perspectives for opportunistic networking.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless Communications

General Terms

Design, Performance

1. INTRODUCTION

The increasing penetration of smartphones with extended communication capabilities (3G/LTE, WiFi, Bluetooth) opens new networking possibilities. Opportunistic networking is one promising dimension which would bring networks back to the users and may instill the excitement of the Internet debut again.

Opportunistic networks bring many benefits to wireless infrastructure-based networks (e.g., 3G). They can increase capacity [1], efficiently use the available radio spectrum [2] and offload the infrastructure [3]. Moreover, they are a valuable asset to cope with partial or complete communication infrastructure outages caused by natural disasters or government censorship. Communication would be uphold in those situations at the cost of some delay.

There is objective evidence that opportunistic networking can bring many advantages. Nevertheless, no opportunistic application has found its way into an application store yet. The concept of opportunistic networking and smartphone applications leveraging these concepts are since years confined to research labs and small-scale testbeds [4, 5]. There are several reasons for this: Firstly, there is still no support for seamless and high throughput short to mid-range Ad-Hoc communications in stock phones. WiFi Ad-Hoc is still not supported on most popular phones[1] and will probably not be in the near future, if ever [6]. Secondly, Bluetooth suffers from its limited bandwidth and the cumbersome peering/service detection procedure, not to mention implementation incompatibilities. Thirdly, even if some platforms such as Windows Mobile or Symbian support WiFi Ad-Hoc, these suffer from the lack of a popular application store and low or decreasing market share. Eventually, WiFi Ad-Hoc is very power inefficient, draining the battery in only a few hours. One might claim that WiFi Direct [7] will improve this issue, though nothing guards us from mobile OS restrictions (see iPhone WiFi tethering mode) or similar Bluetooth-like cumbersome pairing procedures [8].

We believe that applications leveraging opportunistic principles currently face a chicken and egg problem: *As long as opportunistic applications are not popular, smartphone manufactures will not implement the APIs required to setup WiFi Ad-Hoc connections. But as long as application developers do not have APIs to exploit the advantages of opportunistic networks, how can they create such popular applications?* Instead of relying on wishful thinking [6], hoping for the coming support of WiFi Ad-Hoc or WiFi Direct, we reconsider the problem from a more Cartesian approach. By looking at what is available now (and not what we wish to be feasible in the future), we ask ourselves how we can overcome this chicken and egg problem using the current smartphone generation.

In this paper, we propose WiFi-Opp, an alternative way of enabling opportunistic networking based on current smartphones' communication features and APIs. WiFi-Opp leverages the mobile WiFi AP feature (also known as tethering) as well as stationary open APs to support opportunistic communications between classical WiFi client.

To summarize, the main contributions of this paper are:

- We propose WiFi-Opp, a simple and energy efficient way of enabling opportunistic networks using current smartphones' technology (Section 2).

[1]Unless they are rooted or jailbreaked.

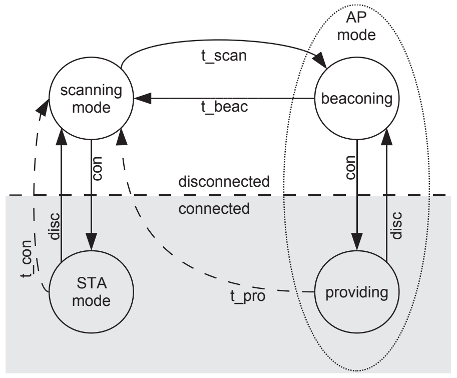

Figure 1: State transition diagram of WiFi-Opp.

- We evaluate the performance and energy efficiency or WiFi-Opp for content dissemination. We show that our approach is up to 10 times more energy efficient than WiFi Ad-Hoc for comparable dissemination performance (Section 3).
- We present a proof of concept demonstrating WiFi-Opp's feasibility (Section 4).
- We discuss the implications of becoming a spontaneous mobile AP, the current device heterogeneity in the market and the benefits of WiFi-Opp over WiFi Direct (Section 5).

2. WIFI-OPP: AD-HOC-LESS OPPORTUNISTIC NETWORKING

Wifi Ad-Hoc is absent from most popular smartphones (Android, iPhone) which requires us to find an alternative technology to enable opportunistic communications between co-located nodes. Devices should hence use the well established 802.11 infrastructure mode and leverage the stationary access points (APs) within range by associating to them and using them as relay to exchange data. But what if an AP prevents client-to-client communications [9], if it is protected by a key (WEP, WPA), or if none is within range in the first place? To overcome these limitations, we propose that some mobile devices themselves switch spontaneously to mobile APs. This can be achieved by putting the mobile device in the so called tethering or mobile AP mode. While this mode is initially meant to share one's 3G Internet access to clients, it can be used for client-to-client communications and even mobile AP-to-client communications thus enabling opportunistic communications. This is the core concept of WiFi-Opp.

Yet, one main drawback of 802.11 is that co-located APs cannot discover each other as well as co-located clients or stations (STA) connected to different APs (either stationary or spontaneous) will not be able to communicate with each other hence resulting in isolated clusters centered around APs. If we aim at content dissemination, we need to enforce some topological dynamics for content to spread. This can come for free thanks to the mobility of stations, which might roam between different APs hence carrying content further (this also applies to mobile APs). However, the scenarios we are targeting are mostly static (e.g., public transports, campus, bars, concert). Besides, most traces have highlighted that nodes are stationary for a large portion of time interrupted by mobility events. We hence introduce randomized switching between the spontaneous AP and STA modes. This will allow breaking clusters resulting in dynamic topological reconfigurations. In addition, this switching strategy will even the energy consumption between nodes, for a STA does not consume as much as an AP which relays traffic for others.

In WiFi-Opp, nodes can operate in 3 different modes as shown in Fig. 1: scanning, station (STA), and access-point (AP). For all scenarios we use the following parameter definitions:

- t_scan is the time a mobile device scans for APs.
- t_beac the time it advertises its presence in AP mode by sending SSID beacons.
- t_con the time a STA stays connected with a specific AP.
- t_pro the time a mobile device is providing AP functionality for connected STAs.

In the rest of this section, we present five different WiFi-Opp based communication scenarios, all using the aforementioned principles. The first one, called *Static*, is the simplest base scenario. We extend it to enforce topological reconfigurations with the *Flexible STA* and *Flexible AP* scenarios. The *Manual* case considers the manual switching from STA to AP by users. Those scenarios still ignore the presence of stationary open APs. Therefore, in a last step, the *Fixed APs* scenario considers the usage of existent infrastructure APs.

2.1 WiFi-Opp – Static

In the *Static* approach (see Fig. 1), each device periodically scans the spectrum[2] for $t\_scan \sim U[t\_scan_{min}, t\_scan_{max}]$ seconds to discover any APs in range. If an AP is found it will connect to it, otherwise, a node becomes an AP itself after its scanning time has expired. This strategy allows the randomized election of an AP for co-located nodes (within range). Devices becoming AP will stay up for $t\_beac \sim U[t\_beac_{min}, t\_beac_{max}]$ seconds awaiting for STA associations. If the AP has clients (STA) it will stay up as long as at least one STA is associated. Otherwise, after t_beac is over, it will fall back to scanning mode again for t_scan seconds. This way, an isolated node (without any neighbor) will not uselessly stay in AP mode (and hence save energy). A STA that gets disconnected from an AP due to a contact loss or any other reason, will fall back to scanning mode. In this case nevertheless, the node will not scan for t_scan seconds (which might be very long) but immediately try to find a new AP or becomes one within a few seconds (fast reconnect). The fast reconnect feature allows for a quicker response to topological changes and is also used in the next two scenarios.

2.2 WiFi-Opp – Flexible STA

The *Static* approach might lead to multiple independent clusters that cannot communicate although they are physically co-located. The reason is that stations associated to one AP cannot communicate to stations associated to another. Although mobility of the nodes might aid in this situation, for a temporally static scenario (e.g., people sitting in offices), we extend the *Static* approach and introduce the *Flexible STA* scenario, allowing for stations to disconnect from an AP after the time $t\_con \sim U[t\_con_{min}, t\_con_{max}]$ has elapsed (see Fig. 1) and scan for other APs or become one itself (fast reconnect explained above).

2.3 WiFi-Opp – Flexible AP

Although the *Flexible STA* approach might improve connectivity by dynamically reconfiguring the clusters, the AP cannot fall back to scanning mode as long as it has associated stations. This might lead to uncontrolled battery drainage. Hence the *Flexible AP* scenario provides APs the possibility to disconnect stations and become scanning again. In this case these stations will then do a fast reconnect (described above) in order not to loose valuable contacts.

In this scenario, APs switch back to scanning mode after t_pro seconds (see Fig. 1), despite associated stations. However, we make t_pro depend on the AP's importance i.e., the number of connected

[2]We assume that scans are triggered every 5s.

	H06	MIT	ETH
# Nodes	78	96	280
Time Period	93 hours	14.9 weeks	14.6 weeks
Type	Bluetooth	Bluetooth	AP Association
# Contacts Total	128'979	75'432	99'024
# Contacts / Node	1654	786	354

Table 1: Properties of contact traces.

stations ($\#_{STA}$) as follows: $t\_pro \sim U[\#_{STA} \cdot t\_con_{min}, \#_{STA} \cdot t\_con_{max}]$, where $[t\_con_{min}, t\_con_{max}]$ is the range of connection time per station (also used for the *Flexible STA* scenario).

2.4 WiFi-Opp – Manual

Some of the current smartphones do not yet provide the possibility to automatically switch the AP mode on/off[3]. In this case, this procedure has to be performed manually by the user. We hence consider the case where a certain percentage of users perform this action in order to exploit the opportunistic contacts while on the move. In such a case, spontaneous APs might be fewer but available for longer time periods (e.g., 30 minutes commuting time). In a more extreme settings such as disaster scenarios or situations where the network infrastructure is teared down, the incentive for manual configuration will probably be quite high.

2.5 WiFi-Opp – Fixed APs

In major urban environments, there exist many spots with open APs (e.g., stores, public WiFi hotspots). Although these open APs usually redirect users willing to access the Internet to a web portal to authenticate, they can still be leveraged for opportunistic communications. Two users connected to the same AP can still communicate without necessarily being authenticated. We consider this scenario assuming a certain density of stationary open APs.

3. PERFORMANCE EVALUATION

We compare the performance of WiFi Ad-Hoc and WiFi-Opp by replaying real-world contact traces and simulating content spreading. We assess the impact of our main parameters on the dissemination performance as well as the energy efficiency of each approach.

3.1 Mobility Traces

We evaluate the WiFi-Opp algorithm by replaying three real-world contact traces presented below. These traces differ in terms of size, duration, and contact frequency. Their main characteristics are summarized in Table 1.

Haggle Infocom 2006 (H06): During Infocom 2006 contact traces of 78 conference attendees wearing Bluetooth devices were collected for the Haggle project [10]. The granularity of the direct contacts collected on 4 continuous days is given by the 2 minutes scanning interval of the devices.

MIT Reality Mining (MIT): The Reality Mining project [11] collected Bluetooth contacts with a 5 minute scanning interval of 96 students and staff members on the MIT campus during several month. We extract the 15 weeks of the trace with the highest contact density.

ETH Campus (ETH): On the ETH campus access point (AP) associations where logged during 15 weeks. We only consider 280 users that have an AP association for at least 5 days a week. The trace is preprocessed in order to remove short disconnections due to interference and the well known ping-pong effect where devices jump back and forth between different APs. Two nodes are considered in contact while associated to the same AP.

[3]This is the case with iOS while it is feasible with Android (see Section 4).

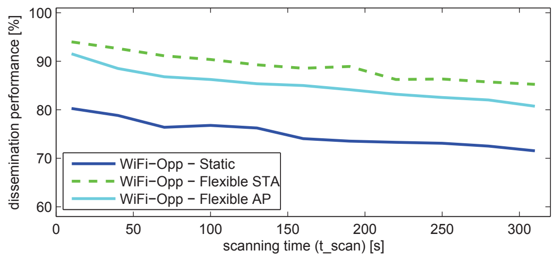

Figure 2: Dissemination performance depending on scanning time (ETH trace). 100% corresponds to the WiFi Ad-Hoc performance.

3.2 WiFi Ad-Hoc vs. WiFi-Opp: Dissemination Performance

We compare the amount of data any node can disseminate to any other node using either WiFi Ad-Hoc or WiFi-Opp. We use a simple epidemic spreading model, thus information is flooded to all nodes without any sophisticated distribution scheme. Nodes getting content from the source can then relay it to others. In each simulation run, we consider a different node as the source of content and average the results over all runs. We assume infinite buffer space so that the performance outcome only depends on the parameters of our approach.

Regarding the link capacity model, we constantly track the number of k nodes that are in contact in the traces by computing the cliques composing these nodes. In a clique of size k there are $l = k(k-1)$ directed links. In the WiFi Ad-Hoc case, assuming nodes composing a clique are in contact for duration t, we calculate the individual link capacity as $d = \frac{t}{l} \cdot throughput$. For the WiFi-Opp approach, there are also l directed links. However, only the $2(k-1)$ AP-to-STA links have capacity d. The STA-to-STA links have only half the capacity ($\frac{d}{2}$), since the traffic has to be relayed through the AP. Note that in both cases, we did not consider any interferences nor possible overheads.

In the sequel, the dissemination performance of the WiFi-Opp variants is always represented as a percentage of the dissemination performance based on WiFi Ad-Hoc communication that we use as a baseline. Although WiFi-Opp takes probabilistic decisions, the variance from one simulation run to the next is negligible since we are replaying traces. Two exceptions though are the *Manual* and *Fixed APs* approaches where we choose a random subset of participating nodes hence generating more variance. In these cases, we average over 10 simulation rounds. If not specified otherwise, the times t_scan_{min} and t_beac_{min} are set to 10 seconds and t_con_{min} is set to 120 seconds. The respective maximum times are always 3 times the min times, e.g $t\_scan_{max} = 3 \cdot t\_scan_{min}$. Note that the connection time t_con is only relevant for the *Flexible STA* and *Flexible AP* variants.

We analyze the impact of the three main WiFi-Opp parameters in isolation, namely the scanning time t_scan, the beaconing time t_beac, and the connection time t_con (and implicitly the AP providing time t_pro). The following results are all shown for the ETH trace. Nevertheless, simulations based on the Haggle or the MIT trace show similar results.

3.2.1 *The impact of the scanning time*

Intuitively, if two scanning nodes meet, the average time until one of them switches into AP mode, thus making a connection possible, is longer if the scanning time t_scan is longer. This can impact the performance, since more connection opportunities are missed. This impact is visible in Fig. 2. Depending on the scanning time (x-axis) the plot shows the average dissemination performance

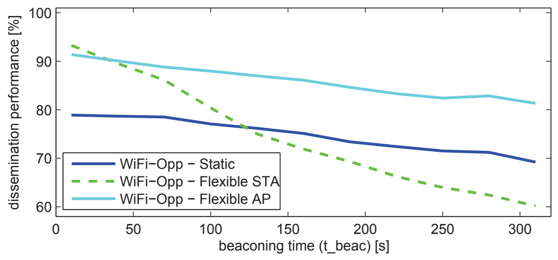

Figure 3: Dissemination performance depending on beaconing time (ETH trace). 100% corresponds to the WiFi Ad-Hoc performance.

(y-axis). The *Flexible STA* variant (green dashed line) generally performs best with 85-95% of the classical WiFi Ad-Hoc performance. The *Flexible AP* variant (light blue solid line) performs slightly worse since it disconnects stations simultaneously causing a short connection break. The *Static* variant (dark blue solid line) has significantly lower performance than the *Flexible STA* and *Flexible AP* variants due to a lack of dynamics in the network. For all the variants, the performance only decreases around 10% when increasing the scanning time from 10-30 seconds to 5-15 minutes.

Conclusion: The WiFi-Opp performance is almost independent of the scanning time, thus permitting long scanning times of 5-15 minutes (and saving energy as we will see later) with only a slight performance decrease compared to WiFi Ad-Hoc. Besides, enforcing topological reconfigurations with WiFi-Opp Flexible STA/AP improve by at least 10% the performance compared to the Static variant, which relies only on mobility to generate dynamics.

3.2.2 *The impact of the beaconing time*

The beaconing time t_beac defines how long a node advertises its presence in AP mode. Fig. 3 shows that increasing the beaconing time (x-axis) actually lowers the performance (y-axis) for all WiFi-Opp variants, namely the *Static* (dark blue solid line), the *Flexible STA* (green dashed line), and the *Flexible AP* variant (light blue solid line). The reason for this performance loss is that two beaconing nodes (in AP mode) can become co-located due to mobility i.e., two nodes switch to AP mode and then come into proximity. They thus cannot discover each other and the connection opportunities are missed until one node falls back to scanning mode. The *Flexible STA* variant is more affected since disconnected stations will beacon for a long time, even if there are no scanning nodes around (the AP they where connected to is also still beaconing as seen in Fig. 1).

Conclusion: Increasing the beaconing time in AP mode has a negative impact on the performance since mobility might lead to several co-located APs which cannot communicate between them.

3.2.3 *The impact of flexible connection times*

Flexible connection times between STAs and APs, t_con and $t\_pro = \#_{STA} \cdot t\_con$, allow an increase of topological dynamics. Nevertheless, if the connections are very short, the performance decreases (see Fig. 4). The *Flexible STA* (green dashed line) and the *Flexible AP* variant (light blue solid line) have the best dissemination performance (y-axis) if the connection time (x-axis) is above 40 seconds in the ETH trace. For longer connection times the performance stays constant.

Conclusion: The performance of the Flexible STA and Flexible AP variants of WiFi-Opp is largely independent of the connection times (if $\geq$ 40s), reaching 90% of the WiFi Ad-Hoc performance.

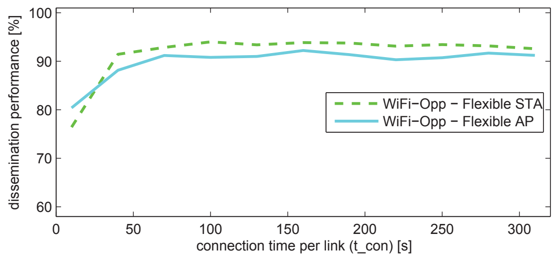

Figure 4: Dissemination performance depending on associated time (ETH trace). 100% corresponds to the WiFi Ad-Hoc performance.

Operation	**Battery consumption**
802.11 scanning	4.80% per 24h
UDP beaconing (station)	28.66% per 24h
802.11 SSID beaconing (AP/Ad-Hoc)	124.55% per 24h

Table 2: Energy consumption for different operations expressed as the percentage of a Nexus One battery they consume in 24h.

3.3 WiFi Ad-Hoc vs. WiFi-Opp: Energy Consumption

In order to perform an energy consumption analysis we measured the required energy for different WiFi-Opp operations on real phones. We used 3 Nexus Ones running Android 2.3.4. As a baseline, we verified how much energy they consume with WiFi turned off. We measured then the energy to scan for WiFi APs (scanning mode), to maintain tethering mode (AP mode), and to send UDP beacons in 2 second intervals (STA mode). The UDP beacons are necessary for station associated to the same AP to discover each other and are not to be confused with the SSID beacons an AP sends. The AP also relays all these UDP beacons. WiFi Ad-Hoc consumes the same amount of energy as the AP mode, assuming no devices are connected to it. Connected devices distribute the SSID beaconing process among them.

The energy measurements can be seen in Table 2. Energy consumption of an operation will always be represented as a percentage of a Nexus One battery it consumes in 24h. This means that if an operation consumes 100% energy it would drain a Nexus One in 24h (assuming the phone does nothing else). More than 100% means that the battery is drained in less than a day, thus the smaller the values, the longer is the phone's lifetime.

In Table 2 we can see that most energy is required for sending SSID beacons. WiFi-Opp should thus reduce the sending of useless SSID beacons, i.e. being an unused AP (time t_beac), as much as possible. We thus analyze the impact of the scanning and the beaconing time on the energy consumption next.

3.3.1 *The impact of the scanning time*

The energy consumption is mainly impacted by the scanning time t_scan and the beaconing time t_beac. Since scanning is the cheapest operation in terms of energy, the longer we scan and do not beacon, the less energy we consume. This is confirmed by Fig. 5 for the MIT trace. By increasing the scanning period (x-axis) the energy consumption (y-axis) of WiFi-Opp (blue solid line) drops considerably. For long scanning periods (5-15 minutes), energy consumption drops to 12% of what WiFi Ad-Hoc (red dashed line) consumes (36% for H06 and 10% for ETH). Recall from Fig. 2 that performance is still good for long scanning periods.

Conclusion: WiFi-Opp can consume as low as 10% of its WiFi Ad-Hoc counterpart while still achieving almost 90% of the WiFi Ad-Hoc dissemination performance (see Fig. 2).

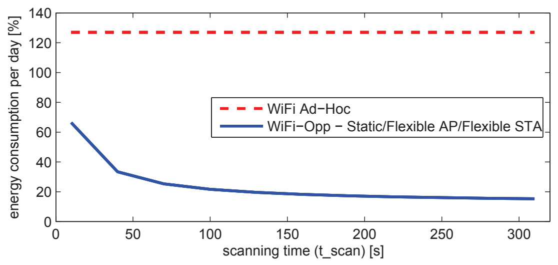

Figure 5: Energy consumption depending on scanning time (MIT trace).

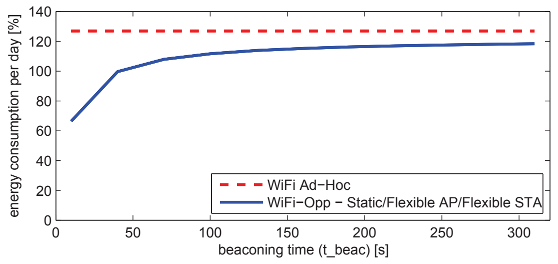

Figure 6: Energy consumption depending on beaconing time (MIT trace).

3.3.2 *The impact of the beaconing time*

While increasing the scanning time reduces the energy consumption, increasing the beaconing time t_beac does the opposite. This can be seen in Fig. 6 for the MIT trace. By increasing the beaconing time (x-axis) the energy consumption (y-axis) of WiFi-Opp (blue solid line) increase drastically, close to the consumption of a classical WiFi Ad-Hoc network (red dashed line). As we saw in Fig. 3 there is actually no performance gain from increasing the beaconing period.

Conclusion: The beaconing time should be kept to a minimum. Increasing it has a negative impact, both on the energy consumption (Fig. 6) and dissemination performance (Fig. 3).

3.4 Manual Scenario

In the *Manual* scenario, we assume that a certain percentage of users participate in the creation of the network by manually setting their device into AP mode. They do so twice a day, (e.g. in the morning and evening while commuting from/to work), for a time between 30 minutes to 1.5 hours. The rest of the time all the nodes are in scanning mode. The resulting dissemination performance for the Haggle trace is shown in Fig. 7. As expected with an increasing percentage of participating users (x-axis) the dissemination performance (y-axis) increases. The performance between 2% and 17% might seem low in relative terms but looking at the actual amount of data that can be transferred, this shows a different picture. In the Haggle trace, 17% of the capacity means that a node can disseminate around 1400 seconds worth of data to each other node on average during the 3 days of the trace. Assuming a transfer rate of $500kB \cdot s^{-1}$ this corresponds to nearly $700MB$ of data in 3 days.

Conclusion: In a manual scenario the capacity naturally shrinks drastically compared to the automatic scenarios but in terms of amount of data that can be transferred it is still significant.

3.5 Fixed APs Scenario

Open AP infrastructure has a lot of potential in aiding opportunistic communication. We simulated its effect by choosing a fraction of random nodes in the trace to be fixed infrastructure nodes. For the Haggle trace, Fig. 8 shows the performance (y-axis) that

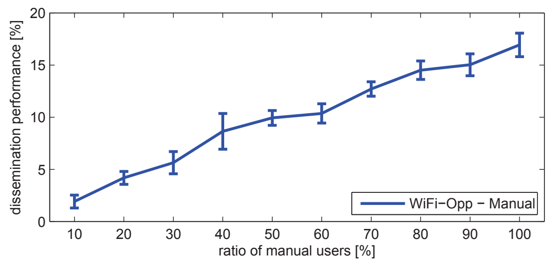

Figure 7: Dissemination performance depending on manual users (H06 trace). 100% corresponds to the WiFi Ad-Hoc performance.

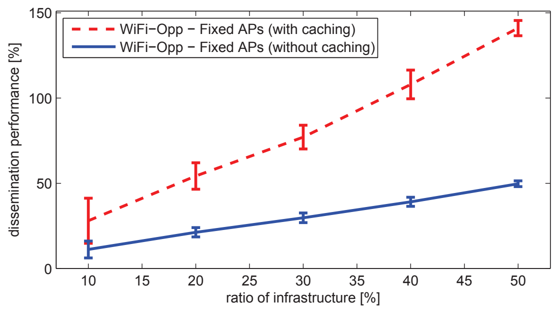

Figure 8: Dissemination performance depending on available APs (H06 trace). 100% corresponds to the WiFi Ad-Hoc performance.

can be gained by just exploiting open APs for different amounts of infrastructure nodes (represented on the x-axis as a percentage of total nodes). By just relaying data the infrastructure can provide considerable performance (blue solid line). This will be the general case. Nevertheless, if some APs, e.g. belonging to a campus network, want to foster opportunistic communication they can more than double the performance by caching data (red dashed line). By caching data the dissemination performance of the network can even surpass the one of classical WiFi Ad-Hoc networks. This can be explained by the higher bandwidth of AP-to-STA communications (twice the STA-to-STA bandwidth) and the increased availability of content at one-hop i.e., AP with cache instead of multi-hop dissemination.

Conclusion: Open infrastructure APs hold a lot of potential for opportunistic communications and even more if adjuncted with caches.

4. PROOF OF CONCEPT

We implemented a WiFi-Opp proof of concept application on Android. The implementation was tested on stock (unrooted) Google Nexus One phones running Android 2.3.4. The application switches to tethering mode (AP), disabling 3G connection while in it, and then goes back to scanning once the beaconing time is over. To automatically switch to the tethering mode, we used the Java Reflection API since there is no Android API call for this purpose. Disabling the 3G connection to protect the Internet access can be done by renaming the APN (Access Point Name) in the 3G configuration. For the rest of the functionalities, normal API calls were used. To continuously send UDP broadcast packets (beacons) while being associated to an AP, even while the screen is off and the device is running on battery, we needed to enable the WiFi Lock feature of Android. In order to omit the UDP beacons from the AP, which serve the sole purpose of communicating the current IP address, a station can find the IP address of the AP in the DHCP information. To the best of our knowledge, WiFi-Opp features are currently fully supported by Android and with some limitations (only for the AP mode) on Symbian, Windows Mobile and the iPhone.

5. DISCUSSION AND RELATED WORK

In the following, we discuss (i) the issue of establishing opportunistic communications and having access to the Internet simultaneously, (ii) the current device heterogeneity in the market, (iii) the relation of WiFi-Opp with WiFi Direct, and (iv) how WiFi-Opp could be further enhanced to be more energy efficient.

Internet access: The tethering (AP) mode is explicitly designed to share a smartphone's 3G Internet access with other devices. Nevertheless, users running WiFi-Opp in AP mode (tethering) might not be willing to share their 3G access with others. Currently we just disable 3G while using WiFi-Opp. The only way we found to protect 3G access from station access is by setting up a L2TP VPN connection which has to be launched manually. Further, users might not have a 3G subscription and exclusively use WiFi in order to connect to the Internet. WiFi-Opp might prevent these users from adopting opportunistic networking since they will lose their ability to access the Internet while WiFi-Opp is running. This could be solved by virtualizing the WiFi cards but this is not yet supported by mobile OSes. The *Fixed AP* approach, assuming the AP in question provides Internet access, is currently the ideal solution. For the remaining automatic WiFi-Opp scenarios the phone would need to switch between the opportunistic and the infrastructure realm. A simple tradeoff would be to enable opportunistic networking while the phone is in standby (screen off) and connect to the WiFi Internet AP once the phone wakes up (screen on) and the user needs it. This will be integrated into WiFi-Opp.

Device Heterogeneity: Currently we only have a proof of concept for the Android OS (version 2.2 and higher). Nevertheless, it is currently the most popular OS with the highest and still increasing market share. A WiFi-Opp application should also work for operator branded Android phones which only allow enabling the tethering mode with an additional subscription. Since only the AP functionality but not the Internet sharing feature of the tethering mode is used, operators have no reason to block a WiFi-Opp application from the market. Other OSes like the iOS only allow for a manual change into the tethering mode. Nevertheless, all devices with WiFi capabilities are able to participate as a station. We believe with the current mix of available devices and taking into account the fixed open APs, opportunistic networking is feasible.

WiFi Direct: The recently released WiFi Direct standard [7] enables two or more peers to discover each other and communicate over WiFi. Nevertheless, it was not designed with opportunistic networking in mind, but aims at connecting groups of WiFi enabled devices. Pairing such as entering a PIN is compulsory in the WiFi Direct standard and thus group formation can take up to 2 minutes and requires user interaction. Since we want to enable communication in a dynamic environment this is not too limiting. Security should be addressed at a higher layer. Nevertheless, security-less WiFi Direct might be used to enable opportunistic networking but since the AP mode is much simpler and works just as well, why not use WiFi-Opp. Furthermore, the WiFi Direct standard was just released and it is yet unclear how it will be supported in mobile OSes.

Context-aware WiFi-Opp: Although WiFi-Opp is energy efficient, there is still a lot of room for improvement by making it context-aware. The most promising extension is to use the embedded low-power sensors (e.g., accelerometer) to infer the motion context and adapt the WiFi-Opp strategy [12]. One could also go one step further and adapt the discovery strategy depending on the battery level. This is done by Google Latitude in order to provide an energy efficient position tracking service [13]. Such additional measures could also be exploited by WiFi-Opp and would render the energy consumption argument against opportunistic networks not holding anymore.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed WiFi-Opp, a viable and energy efficient alternative to WiFi Ad-Hoc to support opportunistic communications by exploiting the mobile AP mode of today's smartphones. A proof of concept of WiFi-Opp has been implemented on stock smartphones and does not require root privileges. Future work will investigate incentives for users to become APs and share their Internet access. We will also consider more realistic traffic patterns for our evaluation. Eventually, we will further enhance the energy efficiency of WiFi-Opp. We are currently implementing WiFi-Opp as a fully-fledged opportunistic framework for Android and the iPhone which can be used by application developers to design opportunistic (and hybrid) applications. Once implemented, it will be deployed in our testbed for an in situ performance evaluation and be released to the public.

Acknowledgments

We want to thank the anonymous reviewers for their advice in improving the paper and Cristian Tuduce for the ETH access point trace. We also thank the Google4EDU project for providing us with Nexus Ones. This work was partially funded by the European Commission under the SCAMPI (258414) and the OpenLab (287581) FIRE projects.

7. REFERENCES

[1] M. Grossglauser and D. N. C. Tse, "Mobility increases the capacity of ad hoc wireless networks," *IEEE/ACM ToN*, vol. 10, 2002.

[2] V. Vukadinovic and G. Karlsson, "Spectral efficiency of mobility-assisted podcasting in cellular networks," in *ACM MobiOpp*, 2010.

[3] B. Han, P. Hui, M. Marathe, G. Pei, A. Srinivasan, and A. Vullikanti, "Cellular traffic offloading through opportunistic communications: A case study," in *ACM Chants*, 2010.

[4] "PhotoShare Haggle Demo," http://code.google.com/p/haggle/wiki/PhotoShare.

[5] A.-K. Pietiläinen, E. Oliver, J. LeBrun, G. Varghese, and C. Diot, "Mobiclique: middleware for mobile social networking," in *WOSN*, 2009.

[6] "Android issue 82: Support ad hoc networking," http://code.google.com/p/android/issues/detail?id=82.

[7] WiFi Alliance, "Wi-Fi Peer-to-Peer (P2P) Technical 7 Specification," www.wi-fi.org/Wi-Fi_Direct.php.

[8] "WiFi Direct Demonstration at CES 2011," http://www.youtube.com/watch?v=_mv-XFZmwNA.

[9] Wikipedia, "Wireless lan security," http://en.wikipedia.org/w/index.php?title=Wireless_LAN_security&oldid=415749161.

[10] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Impact of Human Mobility on the Design of Opportunistic Forwarding Algorithms," in *IEEE INFOCOM*, 2006.

[11] N. Eagle and A. Pentland, "Reality mining: sensing complex social systems," *Personal Ubiquitous Comput.*, vol. 10, 2006.

[12] K.-H. Kim, A. W. Min, D. Gupta, P. Mohapatra, and J. Singh, "Improving energy efficiency of wi-fi sensing on smartphones," in *IEEE INFOCOM*, 2011.

[13] "Google Latitude - Location Update Frequency," www.google.com/support/mobile/bin/answer.py?hl=en&answer=136647.