Slicing the Battery Pie: Fair and Efficient Energy Usage in Device-to-Device Communication via Role Switching

Sascha Trifunovic, Andreea Picu, Theus Hossmann, Karin Anna Hummel
Communication Systems Group
ETH Zurich, Switzerland
lastname@tik.ee.ethz.ch

ABSTRACT

By using device-to-device communication, opportunistic networks promise to fill the gap left by infrastructure-based networks in remote areas, to support communication in disaster and emergency situations, as well as to enable new local social networking applications. Yet, to become practically feasible and accepted by the users, it is crucial that opportunistic communication is energy-efficient.

In this paper, we measure and analyze the energy consumption of today's device-to-device communication technologies: WiFi Direct, Bluetooth and WLAN-Opp (a solution based on the WLAN access point mode). We compare the energy consumption of individual operations like neighbor discovery and connection establishment/maintenance across the different standards. We find that all of these technologies suffer from two problems. First, neighbor discovery is expensive and can quickly drain the battery if implemented carelessly. We analyze this by measuring the impact of scanning frequency on battery lifetime for the different technologies. Second, all technologies suffer from unfairness issues once a connection is established. The "host" of a connection uses more than three times the energy of a "client". We propose strategies to increase fairness by alternating the hosting role among the peers. We compute the frequency of switching roles based on the distribution of the residual connection time, to achieve a good trade-off between fairness and switching cost.

1. INTRODUCTION

Today, most people carry mobile phones featuring technologies like Bluetooth or WiFi Direct, that allow device-to-device communication. Using these technologies, devices can exchange data whenever they are within mutual transmission range, thereby forming an opportunistic network [1, 2]. Such opportunistic networks were proposed as a solution to fill the gap of the existing networking infrastructure in remote and rural areas [3], to enable communication when infrastructure breaks down during natural disasters [4], or to circumvent censorship. Furthermore, opportunistic networks can mitigate the pressure on infrastructure caused by exponentially growing traffic demands, by offloading certain traffic [5]. Even novel applications and services, which cannot easily be implemented relying on fixed network infrastructure, such as local social networking [6], are imaginable.

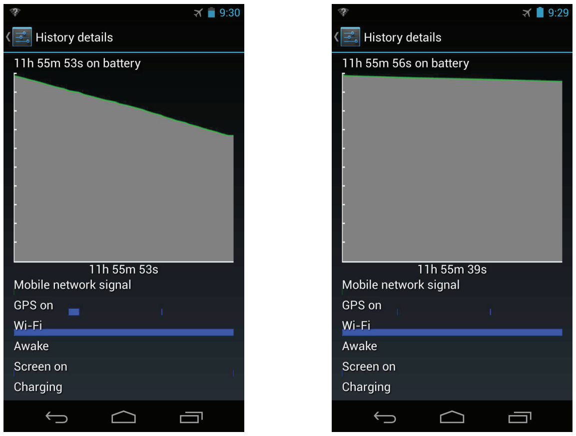
However, to make opportunistic networking practically feasible, users must accept to contribute, despite their resource constrained phones. One critical challenge to achieve this is minimizing the impact of device-to-device networking on battery lifetime. Thus, studying energy consumption is crucial for the success of an opportunistic network. In particular, the energy spent in the background on discovering opportunistic peers has been well investigated in the past [7, 8]. That research has mainly focused on the theoretical analysis and derivation of adaptive scanning strategies: minimizing the number of required scanning operations, while detecting a maximum number of communication opportunities by adapting the scanning interval. Most approaches either try to learn the optimal scanning rate for the current arrival rate of new peers [7], or they adapt the scanning rate based on the current context of the user [8].

In this paper, we go one step further and break down the opportunistic networking functions (neighbor discovery, connection establishment/maintenance) to single operations (e.g., waking up from sleep mode, performing a scan) and states (e.g., being discoverable). For these simple building blocks, we carry out *extensive measurements* of the respective energy consumption. We compare in detail the energy consumed by three technologies, commonly used for device-to-device communication: the widespread Bluetooth and WiFi Direct[1], as well as WLAN-Opp [10], a method based on traditional WLAN, which puts a subset of devices in access point (AP) mode so that any other device can connect. Since none of these technologies were explicitly designed for creating opportunistic networks, it is essential to understand and compare their usage for this purpose.

In particular, we show that, for *peer discovery*, Bluetooth uses less than half the energy needed by WLAN-Opp, which in turn only uses half the energy needed by WiFi Direct. This is true across a wide range of scanning intervals, and it follows from each technology's characteristics and design choices. Further, we find that, once a peer is found, all three technologies suffer from fairness issues in the establishment and maintenance of a connection to this peer. More precisely, of two devices, the one that "hosts" the connection incurs a much higher energy cost than the "client" device. An example of this effect is shown in Figure 1. The figure shows the battery usage of two Samsung Galaxy Nexus phones running Android 4.2 (both in sleep state and airplane mode), connected to each other for almost 12 hours without transmitting data. We can see that one of the battery drains much faster than the other, which is in line with the measurements we show in this paper.

Based on the above insight, we propose the concept of *fair role switching*, which increases fairness and can also substantially extend the battery lifetime of mobile phones. Fair role switching consists of periodically alternating the hosting role between the devices, so as to assure equal depletion of their batteries. The challenge is to estimate the best switching frequency, which achieves a good trade-off between fairness and overall energy consumption for maintaining the connection. To address this challenge, we use the fact that the connection duration of two devices in opportunistic networks was found to follow a power law distribution [1]. We analytically derive the distribution of the remaining contact duration, in function of the elapsed contact duration, and use this to estimate a fair role switching interval. Applying this to real connection traces, shows that it reduces the energy required for role switching by up to 98% (for long contacts) compared to static role switching interval while maintaining a good level of fairness.

[1]The ad-hoc part of the IEEE 802.11 standard [9] is unfortunately not supported on major smartphone operating systems such as iOS and Android (unless they are rooted or jailbroken).



(a) WiFi-Direct AP. (b) WiFi-Direct Station.

Figure 1: Measured battery usage of a WiFi-Direct connection.

Summarizing, our work makes the following main contributions:

- We describe an accurate measurement method, using the Monsoon Power Monitor, and present energy consumption readings for basic (networking) operations on a Galaxy Nexus phone (Section 2).
- We dissect the energy consumption of the discovery process for Bluetooth, WiFi Direct, and WLAN-Opp, and identify the most promising energy saving methods (Section 3).
- Finally, we devise a scheme which determines the best role switching interval between a pair of equal devices, so as to achieve fair depletion, minimize the number of role switches, and extend the battery lifetime of the two devices (Section 4).

2. ENERGY MEASUREMENT

Due to limited battery capacity, energy-efficient operation is one of the most critical issues for smartphones. Hence, measuring and modeling energy consumption has attracted considerable attention in the research community, both from the mobile phone sensing perspective [11] and the networking domain [12, 13, 14, 15]. In this section, we introduce our methodology for measuring the energy consumption of individual events (e.g. scanning) and states (e.g. being discoverable), that are relevant for opportunistic networking.

2.1 Measurement Setup

Measuring the energy consumed by a single operation is a challenging problem. The operating system typically provides estimates of the current state of the battery (in percentage of the full capacity). However, deducing the consumed energy from this information is inaccurate, as it is heavily influenced by background processes running on the phone, as well as the state of the battery, which typically loses capacity during its lifetime. To obtain a clearer picture of energy consumption, the state-of-the-art approach [12, 13, 14] is to circumvent the battery and directly record the power consumed by the phone. For this purpose, we use the Monsoon Power Monitor [16]. The Power Monitor replaces the battery and shows (and records), in real-time, the power that is consumed by the device at a resolution of 500 Hz (or 2 ms). For our measurements, we use a Samsung Galaxy Nexus smartphone, running Android 4.2. While the power consumption differs among device models, experiments with a few other phones show qualitatively similar results.

A sample output of a recorded change of state is shown in Figure 2: the phone is initially sleeping, then it wakes up and enters the idle state, and finally it switches back to sleep mode. The different levels of energy consumed during the sleep and awake states are clearly visible. The plot also shows that the device requires around one second to wake up, which makes the actual time awake slightly longer than the desired ten seconds.

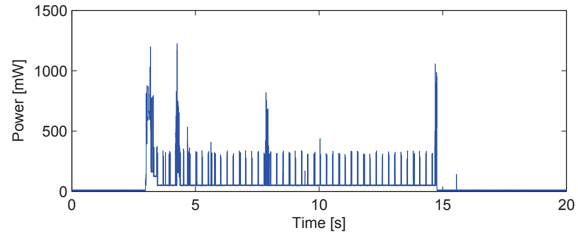


Figure 2: A Monsoon Power Monitor measurement.

Abbr.	*Operation*	*Power/Energy*
P^{CPU}_{sleep}	CPU sleep	11.60 mW
P^{CPU}_{awake}	CPU awake	52.43 mW
E^{CPU}_{wake}	CPU wake up	277.23 mJ

Table 1: Power and energy consumption of the basic operations.

2.2 Measurement Method

The Monsoon Power Monitor allows us to read the power consumed for different patterns of operations, such as waking up the CPU every 20 seconds for a duration of ten seconds. To derive the consumed energy, we integrate over time the power used for each pattern (note that one pattern consists of several operations). In order to extract the energy consumption of individual operations, we perform independent experiments, running different patterns of operations. By formulating the patterns of operations as linear equations, we obtain a system of linear equations, which we solve for the energy consumption of a single operation.

We distinguish between *states* (e.g., awake or asleep), which continuously use power and are hence measured in terms of power (Watt), and *events* (e.g., discovery), which are limited in duration and measured in terms of energy (Joule). Any event will happen either while the device is asleep or while it is awake. The energy consumed by an event does not include the energy required for maintaining basic states, i.e., being asleep or awake. Measurement is particularly tricky for events that involve waking up and going back to sleep. The measurement for the wake up event includes the energy consumed for being awake but not for sleeping. Further, it is difficult to exclude the energy consumed by the awake state itself, because the duration of the waking up operation is not necessarily precisely measurable.

2.3 Basic Energy Consumption

The most effective way to save energy in smartphones is to keep a device in the sleep state for as long as possible. This is typically the case when the screen is off. However, applications may request to keep the CPU awake or an alarm might trigger it to wake up. As seen in Table 1, the phone consumes nearly five times more power if the CPU is awake (but idle). To put these number into the perspective of the battery life of a smartphone, the Galaxy Nexus has a battery with an energy capacity of 6.48 Wh. It could thus theoretically last for 558 hours (over 23 days) in the sleeping state, or for 123 hours (over five days) with only the CPU running idle.

In case background processes are running and waking up the CPU frequently, we need to take into account the one time energy cost of waking up the CPU, which is also shown in Table 1. Considering the amount of energy it takes to wake up the CPU, the benefit of putting the phone into sleep mode depends on the time the phone will stay in sleep mode. With the tested Galaxy Nexus, the device must sleep for at least seven seconds before waking the CPU up again, in order to achieve a gain in energy.

3. ENERGY EFFICIENT DISCOVERY

A key operation in opportunistic networking is neighbor discovery. As neighbor discovery is a process continuously running in the

Abbr.	State	Power [mW]
P_{disc}^{BT}	Bluetooth discoverable	14.73
P_{disc}^{D}	WiFi-Direct discovery	352.27
P_{on}^{AP}	WiFi AP on	205.43

Table 2: Power consumption of discoverable states.

Abbr.	Event	Energy [mJ]
E_{disc}^{BT}	Bluetooth discovery (no peer)	1741.49
	Bluetooth discovery (1 peer)	2167.41
E_{on}^{D}	Wifi-Direct turn on	665.36
E_{scan}^{WIFI}	WiFi scan	695.49
E_{on}^{AP}	WiFi AP turn on	1501.53

Table 3: Energy consumption of discovery operations.

background, energy efficiency is particularly important. By measuring the energy consumption of the discovery process, we will answer two questions in this section: (i) *How do the energy consumptions of Bluetooth, WiFi-Direct and WLAN-Opp compare to each other?* (ii) *How much energy is necessary for each of these technologies to guarantee finding a peer within a given time T?* Note that, as mentioned in the introduction, related work focuses mostly on adapting the scanning interval to the current context (e.g., [8]). Here, our goals are different. First, we want to provide a comparison between the available technologies, to allow an informed decision when designing an opportunistic network. Second, we believe it is important to provide a guarantee of discovering a peer within a certain time interval. Thus, we need to limit the maximum time between consecutive discovery operations.

3.1 Bluetooth

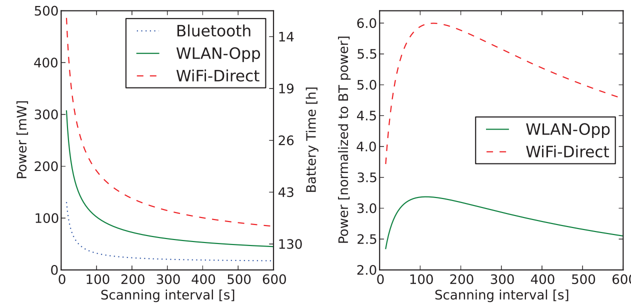
Bluetooth was designed to set up a personal area network, to easily connect several user devices and appliances with one another (e.g., a headset to a mobile phone). However, it also allows to establish a local network, consisting of a master serving several slaves (seven slaves in a local piconet). Its range and throughput are small compared to the IEEE 802.11a/g/n technology.[2]

Despite its drawbacks, Bluetooth has the big advantage of being a very energy efficient ad-hoc communication protocol, and our measurements confirm this. This is partly due to low transmission power (leading to smaller transmission range), but also to other reasons described in the following. First, our experiments show that the CPU can stay asleep, while being detectable to other devices' scans (discoverable, in Bluetooth terms). And second, as shown in Table 2, being discoverable is extremely energy efficient. From Table 1, the power consumption of the sleep state is 11.6 mW, so being Bluetooth discoverable consumes only about 3 mW. Thus, for automatic discovery in opportunistic networks, a device can continuously be discoverable without wasting much energy.

The process of actively scanning for peers naturally consumes much more energy. A Bluetooth scan takes about 13 s and consumes 1741.49 mJ during this time, if no peer is present. If a peer is found, the scan consumes even more energy, as shown in Table 3. Thus, discovery operations need to be scheduled carefully. If we schedule discovery at regular intervals of time t_{scan} (i.e., two devices are guaranteed to discover each other after t_{scan}, and will find each other after $\frac{t_{scan}}{2}$ on average), the power a device consumes for the discovery process is given by:

$$P^{BT}(t_{scan}) = P_{disc}^{BT} + \frac{E_{disc}^{BT}}{t_{scan}} \tag{1}$$

[2]While the range of Bluetooth in smartphones is acceptable, note that especially the range of Bluetooth LE (low energy) is too limited to be usable for opportunistic networking.



(a) Energy vs. scanning interval. (b) Energy vs. Bluetooth energy.

Figure 3: Energy consumed by the discovery process.

However, this whole process will prove to be very energy efficient, in comparison to other technologies in Section 3.4.

3.2 WLAN-Opp

WLAN-Opp [10] provides a flexible way to setup and maintain ad-hoc wireless connectivity, by leveraging existing 802.11 networking technology. To communicate using regular WiFi technology, one device must be in access point (AP) mode, so the peer can connect to it. In a standard setting (e.g., a public hotspot), the role of the access point is assigned to a specific device. In the case of opportunistic networking, however, any device can be either access point or client. Therefore, the WLAN-Opp approach needs a discovery function.

In order to become discoverable, a device must become an AP for a certain time t_{disc}, while regularly performing WiFi scans (in client mode) the rest of the time. The scans must be frequent enough not to miss another device in AP mode, i.e. at least every t_{disc}. The time interval until becoming an AP (i.e., t_{scan}) defines how fast a device will be discovered (guaranteed and on average)[3].

The energy consumptions for both the AP mode (switching on and being on) and the scanning are shown in Tables 2 and 3. In contrast to Bluetooth, the CPU must be awake to perform a WiFi scan or to switch to AP mode. Both being discoverable and performing a discovery are active processes that consume considerable energy. The energy of the whole discovery function is given by:

$$P^{W}(t_{scan}, t_{disc}) = P_{sleep}^{CPU} + \frac{E_{scan}^{WIFI}}{t_{disc}} + \frac{E_{on}^{AP} + P_{on}^{AP} \cdot t_{disc}}{t_{scan}}. \tag{2}$$

While the scanning interval t_{scan} depends on how fast devices must be able to discover each other (it is thus a design decision), the duration of the AP mode t_{disc} (which also defines the time interval until the next WiFi scan) can be optimized to minimize the power consumption P^{W}. By setting the derivative of $P^{W}(t_{scan}, t_{disc})$ with respect to t_{disc} equals zero we get the optimal scan duration t_{disc}^{opt}, depending on the scanning interval as follows:

$$t_{disc}^{opt}(t_{scan}) = \sqrt{\frac{E_{scan}^{WIFI} \cdot t_{scan}}{P_{on}^{AP}}} \tag{3}$$

We can now plug this optimal scan duration into Eq. (2) and we will use this for our power consumption comparison, in Section 3.4.

3.3 WiFi-Direct

WiFi-Direct is WiFi Alliance's new ad-hoc communication protocol for interconnecting smart devices[4], by allowing to setup a "soft access point" for high bandwidth WiFi communication. WiFi-Direct requires a secure pairing procedure which adds delay (up to

[3]For the sake of simplicity, we assume t_{scan} is fixed; in a real implementation, this interval should have a random component.

[4]WiFi-Direct today has largely replaced WiFi Ad-Hoc, which was never adopted widely.

two minutes). While WiFi-Direct could be used for opportunistic networking, it was designed for different purposes and without energy efficiency in mind. The discovery process is very costly, as seen in Table 2, and requires both devices to actively scan at the same time for a successful discovery. By design, WiFi-Direct is, therefore, mostly suited to consciously connect two or more devices at a specific point in time, and is not optimized for a continuous discovery process in the background.

Since the WiFi-Direct specifications do not include a discovery process with duty cycling, we must design this function ourselves. Ensuring that both devices are scanning at the same time (and thus discover each other) without requiring synchronization between the devices is not straight forward. An efficient approach is to mimic the scheme described above for WLAN-Opp. Translated to WiFi-Direct, this means to scan for a long duration t_{disc} every t_{scan} (this corresponds to being in AP in WLAN-Opp). In between such long scans, we perform short scans every t_{disc} (this corresponds to scanning for an AP in WLAN-Opp). The duration of this short scan can be set to minimum time that allows a discovery (t_{disc}^{min}). Since there is no specified minimal scan time t_{disc}^{min} to guarantee discovery in WiFi-Direct, we take the value of $t_{disc}^{min} = 5.5\,\mathrm{s}$, the median of successful discovery times reported in [17].

The whole power consumption of the discovery process thus comprises the sleeping CPU power P_{sleep}^{CPU}, the frequent short scans and the less frequent long scans:

$$P^D(t_{scan}, t_{disc}) = P_{sleep}^{CPU} + \frac{E_{on}^D + P_{disc}^D \cdot t_{disc}^{min}}{t_{disc}} + \frac{E_{on}^D + P_{disc}^D \cdot t_{disc}}{t_{scan}} \tag{4}$$

Again, the time a device performs a long scan (t_{disc}) defines the interval between the short scans and again this time can be optimized depending on the interval of the long scan (t_{scan}). The optimal duration for a long scan is thus:

$$t_{disc}^{opt}(t_{scan}) = \sqrt{\frac{(E_{on}^D + P_{disc}^D \cdot t_{disc}^{min})\, t_{scan}}{P_{disc}^D}} \tag{5}$$

3.4 Discoverability vs. Energy Consumption

Given the expressions for average power consumption and the power measurements, we have now all the ingredients to compare the three technologies. Naturally, the energy consumption of all discovery mechanisms depends on the duty cycle interval t_{scan}. A smaller time results in faster discovery but also requires more energy. The interval t_{scan} is the time of guaranteed discovery, while two devices will discover each other on average after $\frac{t_{scan}}{2}$.

The average power consumption depending on the duty cycle interval is shown in Figure 3a. As expected, Bluetooth performs best in terms of preserving battery life. For example, it is able to guarantee discovery within 2 minutes for 9 days and 7 hours, while WLAN-Opp reaches 2 days and 21 hours and Wifi-Direct only 1 day and 13 hours. All in all, we can see in Figure 3b, showing the power consumption normalized to Bluetooth, that Bluetooth is 2.5-3 times more efficient than WLAN-Opp, which is in turn twice as efficient as Wifi-Direct.

One of the main benefits of Bluetooth, and a reason for its efficiency, is that it is able to operate while the phone is asleep. Thus, if battery lifetime is the only concern, Bluetooth is a good choice for opportunistic networking. However, if the additional transmission range and throughput calls for a WiFi based approach, an access point based scheme like WLAN-Opp has clear benefits over WiFi-Direct in terms of energy consumption.

4. FAIR CONNECTION MAINTENANCE

Once a peer is discovered, the pair[5] needs to be able to communicate. All three technologies (Bluetooth, WLAN-Opp, and Wifi-Direct) establish some type of server-client connection. There is always one device in master or AP mode, while the other is connected to it as a slave or station. As can be expected, the energy required to be a master/AP or a slave/station is not equal, resulting in an unfair battery drain for some devices. While this is already true for the connection establishment phase, it is even worse when also accounting for connection maintenance and actual traffic.

State	*Power in mW*
Bluetooth connected (slave)	73.09
Bluetooth connected (master)	240.21
WiFi-Direct connected (station)	77.04
WiFi-Direct connected (AP)	286.25
WLAN-Opp associated (station)	72.42
WLAN-Opp associated (AP)	205.43

Table 4: Power consumption of connection states.

Event	*Energy in mJ*
Bluetooth connect (slave)	811.69
Bluetooth connect (master)	183.41
WiFi-Direct connect (station)	4263.57
WiFi-Direct connect (AP)	312.14
WLAN-Opp associate (station)	2554.50
WLAN-Opp associate (AP)	4746.42

Table 5: Energy consumption of connection events.

The energy cost of maintaining a connection between a pair of devices for the different technologies (Bluetooth, WLAN-Opp, and WiFi Direct) is shown in Table 4. The role of the device impacts energy consumption by a factor of about 3 for all technologies. The overall energy cost of communication includes the additional one-time contribution of the connection establishment, which also differs with the device's role, as shown in Table 5.

4.1 The Fairness–Efficiency Trade-off

In order to render fair the energy consumption of each device, one option is to alternate the master/slave roles in a round robin fashion, at the cost of short disconnections. This scheme involves an obvious trade-off between fairness and overall energy efficiency. On the one hand, switching should be kept at a minimum, to avoid both disconnections and the one-time cost of setting up the connection. On the other hand, the duration of the physical proximity of two devices is uncertain; therefore, to maintain fairness, switching must be done as often as possible.

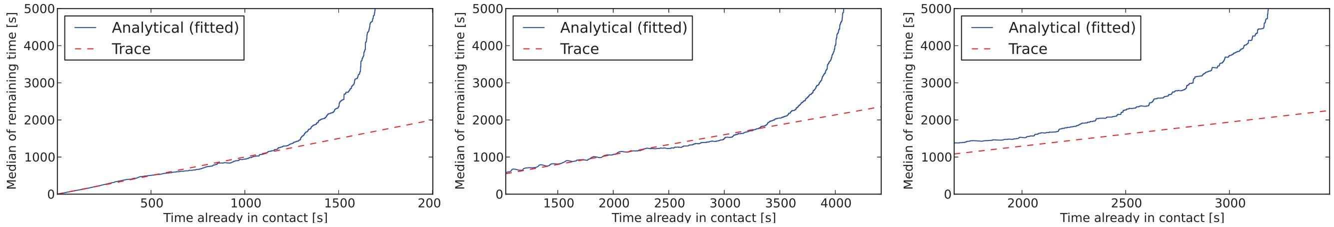
While, for the sake of network usability, we must ensure that disconnections are infrequent, there is also a minimal time t_{min} below which switching simply does not pay off from an energy perspective. As seen in Tables 4 and 5, the one-time cost of connection establishment is higher for the slave device, while the continuous cost of connection maintenance is higher for the master device. Therefore, the slave device will start out having consumed more energy, but the master device will "catch up" after a short time. This time is our desired t_{min}. In other words, the minimal time t_{min} is the time it takes for the energy consumed by the master device to become equal to the energy consumed by the slave device, including the energy required for switching:

$$t_{min} \cdot P^M = E^S + t_{min} \cdot P^S. \tag{6}$$

From this, the minimal time t_{min} is given by:

$$t_{min} = \frac{E^S}{P^M - P^S}. \tag{7}$$

[5]Though we focus here on pairwise communication, we believe the results are easily extensible to group communication and plan to demonstrate this in future work.



(a) PL trace. (b) H06 trace. (c) ETH trace.

Figure 4: Median of the remaining contact time distribution depending on the time already in contact.

	PL	H06	ETH
# contacts	1 000 000	227 657	22 968
α	1	1.62	1.39
x_m	1	1 032	1 673
# nodes		98	20
# scanning interval		2 min	2 s

Table 6: Contact traces.

Having established a basic condition for energy-efficient switching, in the following we focus on devising a switching heuristic aimed at achieving a good trade-off between the total energy consumption of the pair of devices and the fairness of how the two devices share this consumption.

4.2 Role Switching Scheme

A role switching scheme that is both fair and efficient should minimize the number of switches, while ensuring that the overall energy cost is split equally between the two communicating devices. If the duration of the communication opportunity (or *contact*, in opportunistic networking terms) were known in advance, it would be easy to calculate exactly the optimal number of switches, based on the reference energy consumption readings from Tables 4 and 5. However, in opportunistic networks, contacts are typically caused by node mobility, which is non-deterministic. Therefore, rather than calculating a fixed number of switches per contact, a good role switching heuristic should continuously re-evaluate the *remaining lifetime* of the contact, and produce a switching decision based on this and the above reference consumption readings.

Analyses of real world experiments with opportunistic communication have shown that contact duration is distributed as a power law [1, 7]. Using this finding and reliability theory, in the following we derive in closed-form the distribution of the residual contact duration and then use this, to propose a role switching heuristic.

Let the contact duration X be distributed as Type I Pareto distribution (a power law), which has a cumulative distribution function:

$$F_X(x) = P(X \leqslant x) = \begin{cases} 1 - \left(\frac{x_m}{x}\right)^\alpha, & \text{if } x \geqslant x_m \\ 0, & \text{if } 0 < x < x_m, \end{cases} \tag{8}$$

where $\alpha > 0$ is the power law exponent and $x_m > 0$ is the absolute minimum duration of a contact. Then, the distribution $F_T(t)$ of the remaining contact duration T is given by the probability that the contact finishes at or before time $t_{elapsed} + t$, given that it already lasted for $t_{elapsed}$ time units. Using the definition of conditional probability and basic algebra, and noting that $t_{elapsed} > x_m$, we find:

$$\begin{aligned} F_T(t) &= P(T \leqslant t) = P(X \leqslant t_{elapsed} + t \,|\, X > t_{elapsed}) \\ &= 1 - \left(\frac{t_{elapsed}}{t_{elapsed} + t}\right)^\alpha . \end{aligned} \tag{9}$$

Based on this, we can use, for example, the median[6] remaining contact duration to decide when to switch roles. The median remaining contact duration can be easily derived from Eq. (9) as:

$$t_{med} = t_{elapsed} \cdot \left(2^{\frac{1}{\alpha}} - 1\right). \tag{10}$$

As the median remaining contact duration increases linearly with the time already spent in contact, we can dynamically re-evaluate the switching time t_{switch} at each role change as follows:

$$t_{switch} = \begin{cases} \frac{t_{med}}{2}, & \text{if } \frac{t_{med}}{2} > t_{min} \\ t_{min}, & \text{else,} \end{cases} \tag{11}$$

where t_{min} is the minimum switching time for an efficient operation, determined in Eq. (7).

This strategy should provide a good trade-off between the energy consumption of role switching, while maintaining fairness by switching frequently enough during shorter contacts.

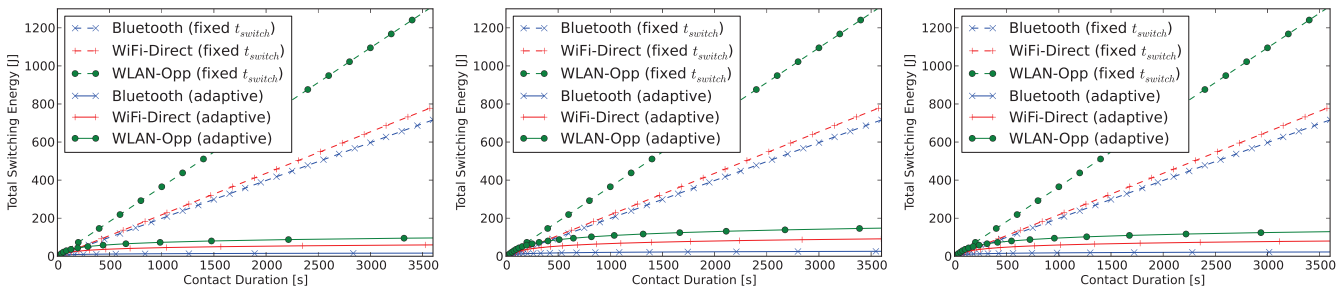
4.3 Evaluation of Our Role Switching Scheme

To analyze and evaluate the trade-off, we simulate the energy consumption and fairness of our adaptive strategy in comparison to using a constant switching time t_{switch}. For this, we first use contact durations generated by the Pareto distribution in Eq. (8). Then, for a more realistic evaluation, we also use the contacts from two different real world traces: the Haggle 2006 trace, collected during the three days of the Infocom conference in 2006, and the ETH trace collected on the ETH campus in 2005. The characteristics of the used datasets of contact durations are summarized in Table 6. For both real world traces, a Pareto distribution was fitted to the set of all contact durations, using the maximum likelihood method described in [18]. This provides the likeliest set of parameters: minimal time x_m and exponent α.

First, we check how good the power law fit is in the traces, by confirming that the median of the remaining time is actually also increasing in the traces. As seen in Figure 4, the median indeed increases with the time already spent in contact. The red dashed line shows the theoretical value of the median depending on the corresponding α and the blue line is the measured value. For longer times, the measured values drift away from the theoretical value because there are not enough samples of long contacts.

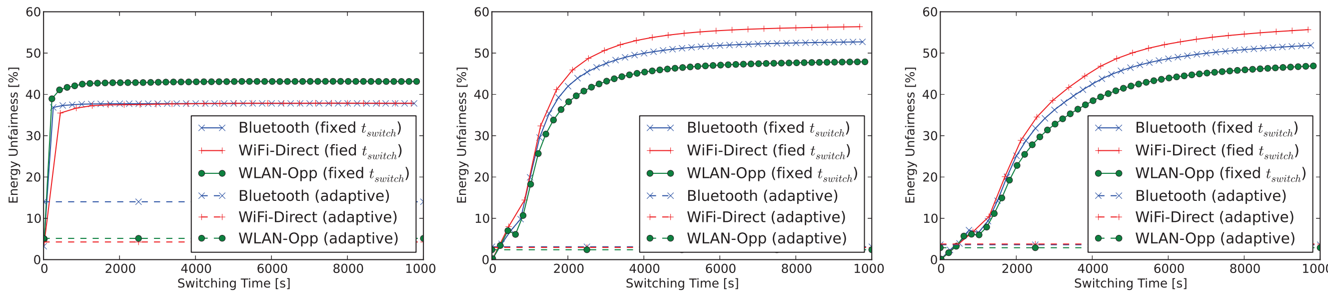
As the basis for our adaptive role switching heuristic holds, we can calculate the energy required for the role switching operation depending on contact duration, as well as how much energy we would save using an adaptive switching time instead of a fixed switching time. The results can be seen in Figure 5 for all three traces and for all three technologies. This adaptation of the switching time allows us to save a considerable amount of energy, especially for longer contacts. For an hour long contact we can save up to 98% energy for the switching process (for Bluetooth, a bit less for other technologies). To put this in perspective, this corresponds to a saving of at least 700 J, which is 3% of the battery. In this case, the energy was compared to the smallest constant switching time, that is still energy efficient: $t_{switch} = t_{min}$.

[6] Since the distribution of the remaining contact duration is also power law, the median is better than the average, as a representation of the "typical" contact duration. The average may even be infinite, depending on the exponent α.



(a) PL trace. (b) H06 trace. (c) ETH trace.

Figure 5: Energy consumed to switch roles depending on the contact duration.



(a) PL trace. (b) H06 trace. (c) ETH trace.

Figure 6: Energy fairness depending on the switching time. Smaller values are fairer.

The advantage of a small switching time is that it is fair. To measure the fairness of our heuristic, we calculate the imbalance of energy, i.e., the difference in energy consumption of the two devices during every contact and divide it by the total energy this contact requires. This fairness ratio is shown in Figure 6 as a function of the switching time. The fairness decreases with an increasing switching time, as it allows for more and longer unbalanced connections. We can also see that our adaptive algorithm is a good trade-off, resulting in good fairness values (usually below 5%), while saving a lot of battery power.

5. CONCLUSIONS AND FUTURE WORK

Energy-efficient operation is a key prerequisite for user acceptance of opportunistic device-to-device communication. We presented extensive energy measurements for the states and operations of major ad-hoc wireless communication technologies available for recent mobile phones, i.e., Bluetooth, WiFi-Direct, and WLAN-Opp based on traditional wireless LAN.

For peer discovery, we found that Bluetooth consumes less than half the energy of WLAN-Opp, which in turn consumes only half of WiFi-Direct. Further, we showed that each technology is potentially unfair as the different roles of the devices required to maintain a connection, such as being a master versus being a slave, show different energy consumption footprints. Using our concept of fair role switching based on estimating the remaining contact duration as a function of the elapsed contact duration, we could assure fair depletion of batteries while reducing the energy cost of role switching up to 98% in several real connection traces.

6. REFERENCES

[1] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Pocket switched networks: real-world mobility and its consequences for opportunistic forwarding," University of Cambridge Computer Laboratory, Tech. Rep., 2005.

[2] V. Lenders, M. May, G. Karlsson, and C. Wacha, "Wireless ad hoc podcasting," *ACM SIGMOBILE Mobile Computing and Communications Review*, 2008.

[3] S. Guo, M. H. Falaki, E. A. Oliver, S. Ur Rahman, A. Seth, M. A. Zaharia, and S. Keshav, "Very low-cost internet access using KioskNet," *ACM SIGCOMM Computer Communication Review*, 2007.

[4] T. Hossmann, P. Carta, D. Schatzmann, F. Legendre, P. Gunningberg, and C. Rohner, "Twitter in disaster mode: security architecture," in *SWID*, 2011.

[5] B. Han, P. Hui, V. A. Kumar, M. V. Marathe, G. Pei, and A. Srinivasan, "Cellular traffic offloading through opportunistic communications: a case study," in *CHANTS*, 2010.

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

[7] W. Wang, V. Srinivasan, and M. Motani, "Adaptive contact probing mechanisms for delay tolerant applications," in *MobiCom*, 2007.

[8] Y. Wang, B. Krishnamachari, Q. Zhao, and M. Annavaram, "Markov-optimal sensing policy for user state estimation in mobile devices," in *IPSN*, 2010.

[9] IEEE-SA, "IEEE 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications," 2007.

[10] S. Trifunovic, B. Distl, D. Schatzmann, and F. Legendre, "WiFi-Opp: ad-hoc-less opportunistic networking," in *CHANTS*, 2011.

[11] K. K. Rachuri, C. Mascolo, M. Musolesi, and P. J. Rentfrow, "Sociablesense: exploring the trade-offs of adaptive sampling and computation offloading for social sensing," in *MobiCom*, 2011.

[12] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani, "Energy consumption in mobile phones: a measurement study and implications for network applications," in *IMC*, 2009.

[13] A. Carroll and G. Heiser, "An analysis of power consumption in a smartphone," in *USENIX Annual Technical Conference*, 2010.

[14] R. Friedman, A. Kogan, and Y. Krivolapov, "On power and throughput tradeoffs of WiFi and Bluetooth in smartphones," in *INFOCOM*, 2011.

[15] M. Dong and L. Zhong, "Self-constructive high-rate system energy modeling for battery-powered mobile systems," in *MobiSys*, 2011.

[16] "Monsoon Power Monitor." [Online]. Available: http://www.msoon.com/LabEquipment/PowerMonitor/

[17] D. Camps-Mur, A. Garcia-Saavedra, and P. Serrano, "Device to device communications with WiFi Direct: overview and experimentation," *IEEE Wireless Communications Magazine*, 2013.

[18] A. Clauset, C. Shalizi, and M. Newman, "Power-law distributions in empirical data," *SIAM Review 51*, 2009.