Adapting network interactions of a rescue service mobile application for improved battery life

Anpassning av en mobilapplikations nätverksinteraktioner för ökad batterilivslängd

Dennis Dufbäck
Fredrik Håkansson

Supervisor: Simin Nadjm-Tehrani
Examiner: Nahid Shahmehri
Upphovsrätt


Copyright

The publishers will keep this document online on the Internet – or its possible replacement – for a period of 25 years starting from the date of publication barring exceptional circumstances. The online availability of the document implies permanent permission for anyone to read, to download, or to print out single copies for his/hers own use and to use it unchanged for non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional upon the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility. According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement. For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its www home page [http://www.ep.liu.se/](http://www.ep.liu.se/).

© Dennis Dufbäck
Fredrik Häkansson
Students in the 5 year Information Technology program complete a semester-long software
development project during their sixth semester (third year). The project is completed in mid-
sized groups, and the students implement a mobile application intended to be used in a multi-
actor setting, currently a search and rescue scenario. In parallel they study several topics
relevant to the technical and ethical considerations in the project. The project culminates
by demonstrating a working product and a written report documenting the results of the
practical development process including requirements elicitation. During the final stage of
the semester, students create small groups and specialise in one topic, resulting in a bachelor
thesis. The current report represents the results obtained during this specialisation work.
Hence, the thesis should be viewed as part of a larger body of work required to pass the
semester, including the conditions and requirements for a bachelor thesis.
Abstract

Today, it is not unusual that smartphone devices can’t survive even one day of regular use until the battery needs to be recharged. The batteries are drained while using power hungry applications made by developers who haven’t taken their application’s energy impact into consideration. In this thesis we study network transmissions as made by a mobile application, and the impact these have on the battery life. The application was developed with the local rescue and emergency service as a hypothetical target group. We test how the mobile network technologies 3G and WiFi together with the device’s current signal strength and battery level affect the energy usage of the battery when uploading data to a server. We develop an adaptation mechanism on application level which uses a mathematical model for calculating a suitable adaptation of scheduling of network interactions. The adaptation mechanism makes use of burst buffering of packets, and adjusts for 3G tail times as well as for different priorities of incoming requests. Custom packet scheduling profiles are made to make consistent measurements, and with this implementation we are able to reduce the amount of energy consumed using 3G and WiFi with 67 % and 39 % respectively during tests.
Acknowledgments

We would like to thank our group members Adrian Sonnert, Villiam Rydfalk, Johan Lindstöm, Pontus Hero Ek, Alexander Ulander and Karl Fredrik Gudjonsson for a good work in the parallel project during the semester this thesis were written, where much of the underlying coding for this project were done. We would like thank our supervisors Simin Nadjm-Tehrani and Mikael Asplund for all the feedback we got from them during the writing of the thesis.
## Contents

**Abstract** iv  
**Acknowledgments** v  
**Contents** vi  
**List of Figures** viii  
**List of Tables** ix  

1. **Introduction** 1  
   1.1 Motivation 1  
   1.2 Aim 1  
   1.3 Problem statement 2  
   1.4 Approach 2  
   1.5 Report structure 2  
   1.6 Limitations 2  

2. **Background** 4  
   2.1 Mobile rescue service application 4  
   2.2 Application context 4  
   2.3 Quality of Service 5  

3. **Theory and related work** 6  
   3.1 Lithium-ion battery 6  
   3.2 Energy impact of different Radio Access Technologies 7  
   3.3 Energy impact of Signal Strength 8  
   3.4 Energy impact of transfer size 9  
   3.5 Related work 9  

4. **Method** 11  
   4.1 Adaptation mechanism 11  
   4.2 Packet scheduling profiles 13  
   4.3 Tools used 13  
   4.4 Measurement environment 14  

5. **Deriving an adaptation logic** 15  
   5.1 Scheduling mechanism 15  
   5.2 Mathematical model 16  
   5.3 Implementing the adaptation mechanism 17  

6. **Results** 19  
   6.1 Battery impact 19  
   6.2 RAT impact 19
List of Figures

3.1 Example of a discharge curve for a Lithium-ion battery ........................................ 6
3.2 Radio Resource Control (RRC) states of UMTS/WCDMA in a 3G network ............ 7
3.3 Energy Breakdown for 3G and WiFi for downloading 10kB ................................. 8

4.1 Adaptation mechanism .................................................................................................. 12

5.1 Flow chart of the adaptation mechanism ....................................................................... 18

6.1 Battery level impact on energy usage with and without adaptation ......................... 20
6.2 Adaptation impact on 3G ............................................................................................ 20
6.3 Adaptation impact on WiFi ........................................................................................ 20
6.4 Signal level impact on energy usage .......................................................................... 21
6.5 No adaptation with high priority requests .................................................................. 21
6.6 Adaptation with high priority requests ...................................................................... 22
6.7 Adaptation without high priority requests .................................................................. 22
List of Tables

5.1 Battery level thresholds ............................................. 17
1 Introduction

1.1 Motivation

Today, smartphones are ubiquitous and there are more than 7 billion mobile subscriptions worldwide [7]. Hardware specifications and computation capability is continuously increasing which in turn requires battery manufacturers to develop sufficient high-end batteries as well. The general opinion from the average smartphone user is however that the battery time is insufficient for daily use, and recharging the device twice a day is more the rule than the exception. Since the main use of smartphones includes running different applications, these will have a substantial impact on the device’s energy usage. A big challenge for app developers, and something that is rarely taken into consideration is accounting for how the app they are developing is actually affecting the battery life of the device, and in some sense the overall user experience. The bulk of the energy consumption in smartphone apps comes from CPU processes, the display and network I/O [17]. The energy usage is further affected by varying conditions, such as what radio access technology (3G, WiFi etc.) is being used, the current condition of the network or what the user’s set brightness level is, etc. Some of these factors can’t be controlled by the app developer, but if she can adapt the behavior of the app in a calculated manner based on varying conditions, the application would contribute to retaining the battery life longer in an energy-efficient manner. By adapting the behavior while maintaining a high Quality of Service (QoS), she can ensure that the application still runs reliably with a high performance.

1.2 Aim

In this thesis, our aim is to develop and evaluate a software mechanism that will increase the battery life of a device running an Android application whilst still maintaining a good QoS. We will use a preexisting application intended for use by search and rescue personnel and develop an adaptation mechanism in order to decrease the energy usage caused by made network transmissions. To achieve this, we will take the current network connection into consideration and transmit data based on the characteristics and conditions of the connection. The mobile device’s current battery level will function as a threshold for the adaptation, regulating its behavior. Using our adaptation logic, we aim to increase the total battery life time of the device by decreasing the energy used by the application.
1.3 **Problem statement**

In order to develop an adaptation customized for conditions of the network we need to get a deeper understanding of how various aspects of a network connection affect the battery and what the impacts are. To address this, we define the following questions which we will answer and discuss throughout this thesis.

1. How do different conditions of a mobile network connection affect the energy usage of a mobile smartphone?
2. How can adaptation of network interactions occur in order to reduce the energy used by a mobile application?
3. How should the adaptation occur based on current network conditions as implemented into our application whilst still maintaining a good QoS?

1.4 **Approach**

The approach for answering our problems involves studying properties of different Radio Access Technologies (RAT) and the ones considered in this thesis are 3G and WiFi technologies since these are the two most commonly used ones [8]. These differ in terms of functionality as well as energy usage, and we want to make sure we can derive a logic applicable for both technologies. On deciding on a scheduling logic, we identify the interactions that are made by the application to the server. Interactions are given different priorities in accordance with a considered hypothetical usage of the application by a rescue organisation. When deriving the adaptation logic we consider the delay tolerance of our application whilst taking the QoS into consideration.

The extended functionality that was implemented in this thesis is made on application level. It could be possible to implement an adaptation mechanism on kernel level on the smartphone or having some middleware between the software and the hardware regulating the network traffic. Then the functionality would be adaptable to all network traffic performed on the device. But in this thesis, we will implement the functionality in a specific application due to the limited knowledge in kernel programming of the authors.

In order to maintain a good QoS in the application, we consider the application within its context which also helps us to define what boundaries we are able to stretch the application’s performance to.

1.5 **Report structure**

Following this introduction, Chapter 2 will explain the requisites for this thesis and some fundamental concepts used throughout the thesis. In Chapter 3 we go deeper into what impact several variable factors have on the battery life, and in Chapter 4 we present how we proceeded to develop an environment for performing measurements of energy usage and what tools were used to accomplish this. Then follows Chapter 5 in which we present the adaptation logic derived during the process. Finally, we present our results in Chapter 6 and discuss these in Chapter 7.

1.6 **Limitations**

Our work has a few limitations which need to be kept in mind when reading our results. We are only measuring and comparing WiFi and 3G connections and the result does not cover other technologies like for example Bluetooth or 4G.
Our measurements only consider data being transferred from the device, not the data received. This is because we are adapting the mobile client and have no control over the servers sending protocol. Our adaptation also relies on the built-in methods of the Android SDK for retrieving information about the connectivity status of our device. Since the reliability of these tools aren’t verified and may be lacking in comparison with other advanced measuring tools there is a margin for error in our measurements.
In this chapter, we will give a more detailed description of the application in which we implemented the adaptation mechanism. We also try to provide more context and discuss the significance of implementing Quality of Service.

2.1 Mobile rescue service application

The application we will utilize during this project was developed during a parallel project in this course together with six other students. The complete platform used in this thesis is an Android application which communicates with a web server running Java and storing data in a MySQL database. The application’s intended area of use is as a communication platform to be used by the local rescue and emergency service.

The application utilizes a set of interactions with the server, more precisely, video call, retrieving other employees’ contact information and show ongoing emergencies on a map. The application also performs a task in the background, consisting of updating the device’s GPS position to the server every 8 seconds. This GPS data is intended to be used by some incident commander or other administrative personnel and can be used for tactical positioning of the operative personnel in the field or documenting user behaviour and movement patterns. These interactions with the server can be interpreted as either requested or scheduled. A requested interaction is triggered by the smartphone user and relies on an immediate answer from the server. For example, when requesting information about an ongoing emergency (stored on a server), we can’t allow the request to the server to be postponed. All interactions except for the background task can be interpreted as requested interactions in our project application. The background task consists of scheduled interactions. The data transferred through requested and scheduled interactions is referred to as foreground data and background data, respectively.

2.2 Application context

Given the behaviour of the application, we can ask ourselves why it would be relevant to implement a battery saving mechanism into this specific application. The application is, as mentioned, developed as a mock up application for the local emergency and rescue services.
As their responsibilities include addressing different kinds of emergencies, it is to be assumed that having access to a battery charger is not always a given. Also, considering an organization that utilizes advanced mobile technology and is dependent on it operating reliably, not having sufficient battery to function as desired could have negative consequences. As mentioned, one of the main functionalities of our application is sending GPS data as background data to a server. These connections to the server are established regularly, and during an ‘ordinary’ day of use, these background connections make up the majority of connections made to the server. Since transferring data to a server directly impacts the battery life of the mobile phone (more on this in Chapter 3), and since the users will not be able to monitor the battery life at all times, we consider it of great importance to ensure resource-efficiency of the service.

2.3 Quality of Service

As mentioned, our application is to be used by a local emergency service and is one that is dependent on information being handled in a reliable fashion. The concept of implementing QoS to a service is to provide for a better network performance and better reliability for the user. Several aspects, or characteristics of the network service are usually considered and managed when treating QoS. The aspects that are managed are related to the specific service, and could include bandwidth, latency and availability. Prioritizing network traffic can be a way to achieve QoS. Giving higher priority to more important packets, as specified by developers based on requirements elicited from the users, is a way of obtaining reliable and predictable results and guaranteeing a desired level of performance [10].
In this chapter, we present some earlier work about the energy consumption of a mobile smartphone and what aspects of the network communication that affect the battery the most. We also compare our work with some previous work similar to the one we present in this thesis.

3.1 Lithium-ion battery

The most common battery type used in smartphones today is Lithium-ion batteries and the discharge curve of these batteries is monotonically decreasing, as shown in Figure 3.1. The state of discharge represents how much of the battery has depleted. The y-axis shows the battery voltage output and we can see that the cutoff voltage (voltage when battery is no longer usable/fully discharged) occurs at around 3500 mV. Notable for a Lithium-ion battery is that the discharge curve can change with the discharge current, temperature and age of the battery [26].

These battery types are widely used because of their high capacity and small size. Lithium batteries are however sensitive to temperature changes, and a low temperature can severely attenuate the capacity [19].

Figure 3.1: Example of a discharge curve for a Lithium-ion battery [26].
3.2 Energy impact of different Radio Access Technologies

Since the main purpose of this project is to reduce the energy consumed by a smartphone by adjusting made network interactions, we first need to examine the factors that affect the battery in a smartphone during a network interaction. In order to figure out the impact on the battery of the different network properties we do a literature study on the subject to get a full understanding on how we should develop our implementation.

We have chosen to examine the RATs 3G and WiFi, as they are dominant wireless technology choices among smartphone users. They also apply well within the context of our application prototype as it intended to be used. The techniques in these connections differ a lot as pointed out by Huang et al. and Mittal et al., and so does the corresponding energy that is being used by the device.

3G Energy use

In 3G we will focus on the technique Universal Mobile Telecommunications System (UMTS) since it is one of the most common techniques today and will be in the future. It is also the technique used by the mobile operators in Sweden.

A wireless smartphone device connected to 3G can be in several different modes of operation, called power states. A Radio Network Controller (RNC) determines the device’s power state using a Radio Resource Control (RRC) protocol. Depending on what state the device is in, different amount of power is being consumed. The 3G network has four power states: Dedicated Channel (CELL_DCH), Forward Access Channel (CELL_FACH), Paging Channel (CELL_PCH) and IDLE. The device exists in the CELL_DCH state when transmission occurs and this state utilizes a dedicated channel to ensure low delay and high throughput for the transmission. This is the power state that uses the highest amount of energy, see Figure 3.2. The CELL_FACH state is used for low transfer rates and does not have a dedicated channel. In this state, some energy is still being used by the device, but only about half as much as in the CELL_DCH state.

![Figure 3.2: Radio Resource Control (RRC) states of UMTS/WCDMA in a 3G network](image)

The RRC inactivity timers determine when the device should transition to a lower performance power state (T1 - T3). As these timers trigger, the device remains inactive in its higher power state before the transition happens. This is to alleviate the delay that would incur, were the device to move up to a high power state from IDLE, but this also means that the device is spending excess energy, known as tail energy. The duration of the inactivity timers, or, the tail time, can differ in duration depending on configurations made by the mobile network operator, and this in turn can lead to differences in energy consumption during transmissions.
Measurements made by Vergara and Nadjm-Tehrani [23] show the energy consumption values of the different power states for a device using the mobile operator TeliaSonera. They get around 1400 mW for CELL_DCH, 600 mW for CELL_FACH and 200 mW for CELL_PCH. Similar values can also be found in [5]. These values differ in different scenarios depending on factors such as mobile operator and signal strength, these are good benchmarks when comparing the different RAIs[17].

WiFi Energy use

WiFi operates similarly to 3G by having power states and inactivity timers. Almost all the 802.11 protocols are using a Power Save Mode (PSM)[3] which has the different states Deep Sleep, Light Sleep, Idle, and High. The PSM induces excess tail energy, as the device remains in High power state after transmitting/receiving packets due to a pre-defined inactivity timer called PSM timeout. Tests made by Mittal et al. [17] show measured energy consumption values of 10 mW, 120 mW, 400 mW and 600 mW for Deep Sleep, Light Sleep, Idle and High respectively. A more inclusive comparison between WiFi, 3G as well as GSM is made by Balasubramanian et al. [5]. They conclude that of the three, WiFi consumes the least amount of energy with regards to transfer energy and tail energy.

3.3 Energy impact of Signal Strength

Regardless of whether the device is using 3G or WiFi, the signal strength can vary due to variation in cellular network coverage, and as it does, so does the energy consumption [8][17][14]. The Received Signal Strength Indicator (RSSI) is measured in dBm (decibel meters) and ranges from about -100 dBm to 0 dBm for WiFi and from about -115 dBm to -45 dBm for 3G. An ‘average’ user connected to 3G receives a signal strength that is considered poor (< -91.7 dBm) 47 % of the time, and WiFi users experience poor signal strength (< -80 dBm) 25 % of the time, concluded by Ding et al. [8]. They also conclude that the average user transfers 43 % and 21 % of foreground data, as well as 19 % and 4 % of background data while experiencing poor signal strength using 3G and WiFi respectively. Transmitting data using poor signal strength can result in a higher energy consumption compared to when having good signal strength, for both 3G and WiFi. In short, low signal strength causes a reduced rate of data during the transmission, prolonging the time it takes to transmit packets. It can also lead to packets being dropped and having to be retransmitted.

Figure 3.3 shows the electric charge used under 3G and WiFi when downloading 10kB of data for different RSSI values. Ding et al. [8] chose to present their y-values in µ Ah (micro ampere hours) but use the denotation Energy which is incorrect but was made for simplicity.
in their report. The values come from dividing the energy used in Wh (watt hours) with the standard voltage supply of a Lithium battery which is 3.7 V \( \left( \frac{\text{Wh}}{3.7 \text{ V}} = \mu \text{Ah} \right) \). The actual energy used is then proportional to the presented \( \mu \text{Ah} \)-value with a factor of 3.7 which makes the graphs relevant to our purpose when comparing energy usage for different RSSI values.

The graphs show us that for 3G, the dominating energy used is made up of tail energy (Tail1 and Tail2) for all measured RSSI values. Energy spent in promotion phases, (as the device transitions to a higher power state), is denoted Prom1 and Prom2. Under WiFi, PSM tail energy and idle energy makes up most amount of energy cost for all RSSI values. We can also see however that when the signal strength gets low enough, the idle energy consumes the most amount of energy and this is due to reduced data rate. More data frames are retransmitted as well, causing the energy spent in receiving retransmitted data frames (ReRx) to go up notably. Energy spent in receiving unique data frames (Unique Rx), transmitting unique TCP frames (Unique Tx) and retransmitting TCP frames (ReTx) makes up a relatively small portion of the energy usage.

Both graphs have a similar appearance in that a lower signal strength increases the energy used quite drastically. It is also apparent that sending files using 3G uses up much more of the battery than sending data via WiFi.

### 3.4 Energy impact of transfer size

The amount of data being transmitted affects the energy consumption in both 3G and WiFi. Transmission energy for a single data packet is higher for large packets according to [5] and especially when the size exceeds 100kB. However, it is not always true that smaller packet equals lower energy consumption. As pointed out in [21] the energy consumed by small packet transmissions can be of equal size, or even greater, as for larger transmissions. One example is when the payload of the TCP/IP packet is smaller than the header. Then the overhead of the packet would stand for the majority of the size instead of the data itself. True for both RATs however is that for small amounts of data (1 - 10kB) the same amount of energy is being consumed, regardless of the packet size [5, 8].

### 3.5 Related work

A lot of previous work has been done within the area of preserving battery life of a mobile device. One interesting solution to reduce energy usage in a mobile application was made by Vergara and Nadjm-Tehrani [23]. They took advantage of sharing one cellular connection between multiple cellphones by creating a small LAN between the phones and sending all data through the same cellular connection. Their results show they could save up to 59 % of the energy consumed when 8 phones are collaborating.

We are going to use a tool named EnergyBox [24, 9] to estimate energy consumption caused by the network transmissions. Since we will be using it in a lot of our work we have studied the work by Andersson [4] and Green Olander [11] in order to get a deeper understanding in how we can use EnergyBox. Their work gave examples of usage and what they have achieved with the help of this tool.

Balasubramanian et al. [5] have developed TailEnder that reduces energy consumption by taking advantage of tail times that occur under 3G, GSM and WiFi. During our study we use their TailEnder algorithm for saving energy while connected to 3G.

On top of the research in [5] the work by Tang et al. [20] extend their solution by implementing machine learning on the server side in a client-server architecture. The client sends recorded information about the packets being sent from the client to the server where it calculates future scheduling for the client depending on historical data collected. The server learns to predict this schedule over time and needs a big set of data to be able to calculate a good result. They characterises this method as “online scheduling”. But they also make an “offline
scheduling” algorithm which is similar to TailEnder developed in [5]. By comparing these two different approaches they were able to show that the offline scheduling is more efficient in terms of reducing energy consumption.

Finally, a thesis with a similar subject has been written by Hall and Luckey [12]. Here the authors bundle data packets to save energy. Their approach is very similar to ours but they do not cover connection over WiFi or varying RSSI values.
In this section, we explain our method of deriving a mechanism for adapting server interactions in order to save energy. We also describe the tools used to collect and analyze data from made measurements. In short terms, the method consists of the following:

1. Deriving a mechanism for adapting interactions made with the server. The mechanism gives different outcomes depending on the device’s battery level and on varying network conditions.
2. Creating packet scheduling profiles that correspond to different usages of the application and that can be used during performed measurements.
3. Running experiments to measure the impact of different network conditions as well as the impact of the adaptation mechanism.
4. Comparing and evaluating the energy usage for the different experiments.

4.1 Adaptation mechanism

From Chapter 3, we know that there are several factors that affect the battery when a device interacts with a server. These factors have different impacts on the battery life, and are factors that cannot necessarily be controlled by the application, but perhaps adjusted for. We can measure the separate values associated with these factors (for instance the signal strength’s RSSI value) and adjust how the device interacts with the server according to the measured values in order to save energy.

Figure 4.1 shows an overview over our adaptation mechanism, which takes readings from three factors as input variables. The three factors are the device’s battery level, its current RAT and its signal strength and the associated measured values are obtained in real time from the device itself. These factors were chosen not only since their impact on the battery life is predominant but also since variations of their values apply well within the context of our application as a mobile rescue service application. The device’s battery level is used to set thresholds for when the adaptation mechanism should change its behaviour. The adaptation mechanism itself is developed to include a mathematical model functioning in conjunction with mechanisms for scheduling of packets. Their underlying adaptation
logic is derived in Chapter 5. The adaptation mechanism allows our application to alter its behaviour based on varying network conditions. However, the data packet type that is to be sent impacts the adaptation as well.

**Packet priorities**

In Section 2.1, we make the distinction between *requested* interactions (for transferring foreground data), and *scheduled* interactions (for transferring background data) made with the server. This distinction is important to retain when deriving an adaptation logic for the scheduling mechanism. Since requested interactions expect an immediate response in order to retain an acceptable QoS, foreground data packets are therefore marked as *high priority* packets. As they are received, the transmission is immediately initiated. Packets that constitute the background data are marked as *low priority* packets, and these packets always go through the adaptation mechanism.

**Adaptation policy**

In Chapter 2, we gave a description of the application and its interactions with the server and put it into the context of its intended use. Since we are interested in providing a good QoS for the users of the application, we feel it is necessary to introduce a clear specification of the adaptation mechanism. The adaptation policy describes what is required during the adaptation and what is and isn’t allowed. It is also used as a framework when deriving the adaptation logic to ensure that we achieve desired goals. The general functionality of the adaptation is that it allows for delaying of requests sent by the application depending on several condition variables. The adaptation policy is set as follows:

- Packets marked with high priority are to be sent immediately with minimal delay.
- Transmissions of packets are allowed to be delayed for a set maximum amount of time as well as a set minimum amount of time. These values should apply well within context of the application and support a good QoS.
- A low value of the device’s current battery level should allow a longer delay of packets than a higher battery level in order to save energy.
- A low value of the device’s current signal strength should allow a longer delay of packets than a higher signal strength in order to save energy.
- When using 3G, a request arriving while the device remains in the CELL_DCH state should be transmitted immediately, in order to make use of tail time.
By setting these requirements, we can ensure that we get a delay-tolerant application that still performs in a reliable and predictable manner. It is also worth mentioning that the background data sent during the scheduled interactions is considered significant enough to not allow for too long of a delay.

4.2 Packet scheduling profiles

In order to perform multiple measurements of energy usage with various conditions on battery level, RAT and signal strength we define packet scheduling profiles that can be reapplied for several experiments. These scheduling profiles allow us to repeat experiments with consistency and to compare results from measurements made with and without adaptation of the server interactions.

We utilize two distinctive packet scheduling profiles in our experiments. The first profile contains only scheduled network interactions from the application which consist of regularly upload of GPS coordinates. These interactions have a frequency of 8 seconds, derived from the specification of the application in Section 2.1. The second packet scheduling profile consist of a combination of scheduled background traffic and multiple requested interactions that will occur at different times throughout the duration of the scheduling profile. These infrequent requests can be interpreted as a user interacting with the application.

The packet scheduling profiles have a total duration of 3 minutes and will be repeated several times for each test. During these 3 minutes the conditions (input variables) will be kept as static as possible in a psychical environment.

All requests sent to the server are approximately 300 bytes in packet size.

4.3 Tools used

Obtaining necessary network information from the device can be quite challenging without proper equipment. The tools we used to perform measurements, and why we used them, is described in this section.

Tcpdump

To create necessary trace files of the network activity, we use the command-line packet analyzer Tcpdump\footnote{Tcpdump - \url{http://www.tcpdump.org}} running on our server. Tcpdump captures the packets that are sent from the device over a certain network interface and outputs information about the content of the packets to a .pcap trace file. The packets that are sent follow the packet scheduling profile, and by running the analyzer as the interactions with the server are made we can generate the required trace file to be used with EnergyBox.

EnergyBox

EnergyBox is a software used to estimate energy consumption of a smartphone application’s network activity\footnote{EnergyBox - \url{http://www.energybox.org}}. EnergyBox needs several parameters in order to calculate the theoretical values which are shown as a graph of energy consumed over time. It requires a trace file of the network activity from the mobile application. The trace files we use are generated with Tcpdump using our packet scheduling profile. By specifying the source IP address that the device used during the trace, we get only the packets needed for the calculation of the energy used. EnergyBox also requires two configuration files in order to make its calculations. The two configuration files contain information about the RRC inactivity timers, as seen in Figure 3.2, and device specific data of the amount of energy the mobile phone consumes in the different RRC power states. Information about the RRC inactivity timers (T1 - T3) is not
publicly available for our mobile operator, therefore we use the default configuration settings provided with EnergyBox from its GitHub repository [9]. We also used the data of the energy consumed during the different RRC power states, available at the GitHub repository. Even if the configuration values do not exactly match the correct values of the device or of the operator, the comparative results would still be relevant when using our method.

4.4 Measurement environment

As mentioned, our application is made for a device running Android and during the measurements it runs on a Google Nexus 5X. All measurements are made in a physical environment from the same geographical location, using the Swedish mobile operator Tele2. Variation in temperature is small, and the device is kept as static as possible to prevent unpredictable variations caused by different conditions in the network.
5 Deriving an adaptation logic

In this chapter, we describe how we developed a scheduling mechanism and a mathematical model for adapting network interactions in order to conserve energy. We also go through decisions we made based on our theoretical study and on measured data received from testing our application solution. We will go through the logic of the adaptation mechanism and explain why this is a good method of solving the problem of unnecessary energy consumption.

Decisions made in this chapter are mainly based on the theory we presented in Chapter 3 in combination with the adaptation policy that takes QoS requirements into consideration.

First, we discuss how the scheduling of network packets is handled by our adaptation mechanism. Later, we present a mathematical model that impacts the delay of transmissions based on the mobile devices battery level, RAT and signal strength.

5.1 Scheduling mechanism

In this section, we explain how scheduling of network packets are implemented into our adaptation mechanism and how it works in conjunction with the mathematical model.

Burst buffering

In order to allow for delaying of network interactions, and by doing so, reduce the amount of energy used by network transmission, we use burst buffering of packets. This implies buffering multiple packets for some duration of time before sending them all in a burst. This reduces the numbers of transmissions made, and alleviates unnecessary tail energy by increasing time spent in IDLE for both 3G and WiFi, as mentioned in Section 3.2 [22].

Keeping our adaptation policy in mind, we set suitable maximum and minimum amounts of time that we allow buffering of packets for. These times vary based on the device’s current battery level, RAT and signal strength. Measurements in Vergara et al. [25] show that a deadline of 90 seconds in comparison to 30 seconds has a great energy reduction on frequent background data transmissions.
High priority packets

As mentioned in Section 4.1 and in compliance with our adaptation policy we also have some packets that we don’t tolerate to be delayed and these packets will be sent immediately. The mathematical model will be overridden as a high priority packet is added to the queue and all packets that are currently buffered will be sent together with the high priority packet.

Use of tail time

Another feature we implemented in our adaptation mechanism is inspired by the works of Balasubramanian et al. [5] and their TailEnder. The feature lets our adaptation utilize the tail times in 3G networks by not buffering a packet if it arrives when the device still exists in a high power state and instead transmitting it immediately. As mentioned in 4.3, we use the information about the RRC inactivity timers that was available through the EnergyBox GitHub repository which specifies a tail time of 4100 ms.

5.2 Mathematical model

Our adaptation mechanism consists of a mathematical model that affects how the scheduling of network packets is to be performed. The model calculates a deadline timer for the burst buffering based on the current battery level, RAT and signal strength. The model consists of the following parameters:

- \( t_d \): Burst buffer deadline [s]
- \( t_{\text{min}} \): Minimum possible buffer deadline for a specific battery level [s]
- \( t_{\text{max}} \): Maximum possible buffer deadline for a specific battery level [s]
- \( \alpha \): RAT type factor, \( 0 \leq \alpha \leq 1 \)
- \( \beta \): Signal strength factor, \( 0 \leq \beta \leq 1 \)

Here, \( t_d \) is calculated with \( t_{\text{min}}, t_{\text{max}}, \alpha \) and \( \beta \) as input parameters according to the following equation:

\[
 t_d = t_{\text{min}} + (t_{\text{max}} - t_{\text{min}}) \cdot \alpha \cdot \beta \tag{5.1}
\]

This gives us our required output \( t_d, t_{\text{min}} \leq t_d \leq t_{\text{max}} \). In the following part of this section we go through how we define the input parameters.

Battery level

In section 4.1 we specified that the adaptation solution is to have the device’s current battery level to set thresholds for the adaptation that modify the adaptation behavior. The theory presented in Section 3.1 show that the battery discharge is high at low battery level. In order to smooth out the discharge curve and extend the battery life we choose the battery level threshold values based on the characteristics of the battery discharge curve in Figure 3.1.

Battery level thresholds

By extracting the current battery level of the device from the `BatteryManager`\footnote{BatteryManager - \url{http://developer.android.com/training/monitoring-device-state/battery-monitoring.html}} class, built into the Android SDK, we are able to categorize the battery level into three discrete levels, as presented in Table 5.1. The threshold values are selected based on the theory about battery properties in Section 3.1 where the discharge rate is higher when we reach a lower battery level. These levels can be modified based on the type of battery used in the smartphone.
For the different battery thresholds, we set a maximum and minimum value of the buffer deadline \( t_{\text{min}} \) and \( t_{\text{max}} \). These values are also shown in Table 5.1. These values are derived based on the frequency of the background requests and in appliance with the adaptation policy. They are also motivated by results presented in [25].

<table>
<thead>
<tr>
<th>Battery level (%)</th>
<th>Value</th>
<th>( t_{\text{min}} ) [s]</th>
<th>( t_{\text{max}} ) [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>40 - 100</td>
<td>High</td>
<td>8</td>
<td>30</td>
</tr>
<tr>
<td>20 - 39</td>
<td>Medium</td>
<td>16</td>
<td>60</td>
</tr>
<tr>
<td>0 - 19</td>
<td>Low</td>
<td>32</td>
<td>90</td>
</tr>
</tbody>
</table>

Table 5.1: Battery level thresholds

**RAT**

Since energy usage differs quite a lot depending on what RAT is being used we want to customize the adaptation as well as possible for each network type. 3G and WiFi are treated separately and the RAT type factor \( \alpha \) is adjusted according to these technologies’ characteristics.

Since transmitting data over WiFi is less energy consuming than 3G it is reasonable that \( t_d \) should be lower for connections via WiFi compared to 3G connections given the same current battery level. We let the parameter \( \alpha \) represent the difference in consumption between WiFi and 3G. From our example values in Section 3.2, we can see energy consumption values of 1400 mW for 3G and 600 mW for WiFi in their respective transmission state. We set \( \alpha_{3G} = 1 \) for 3G and \( \alpha_{WiFi} = 1 - \frac{600}{1400} = 0.57 \) for WiFi and this gives us a RAT type factor based on the corresponding energy consumption.

**Signal Strength**

Similar to \( \alpha \), we want to have the factor \( \beta \) representing the difference in energy consumption for different signal strengths. We choose to discretize the signal strength into four different values, based on their energy consumption presented in Figure 3.3, for 3G and WiFi separately.

In order to get a normalized value of \( \beta \) we use feature scaling according to

\[
\beta = \frac{x - \min(x)}{\max(x) - \min(x)}
\]

(5.2)

where \( x \) is mapped to the energy usage for the current RSSI value, according to our discretization. A high RSSI reading will output a value of \( \beta = 0 \), a low reading will output \( \beta = 1 \) and the values in between will have a value \( 0 < \beta < 1 \).

### 5.3 Implementing the adaptation mechanism

The adaptation mechanism is implemented at application level as an Android Service\(^2\). The implementation consists of a single Java class of approximately 400 lines of code. The class also handles the network connections, over which the data is transferred via the TCP/IP protocols.

Since the adaptation utilizes burst buffering, when the adaptation is active, incoming packets are placed in a send queue to be sent later. A flow chart of how the scheduling of packets occurs can be seen in Figure 5.1.

As a user interacts with the application, or as scheduled background interactions occur, custom request objects are created and placed in the send queue. Upon receiving a packet

\(^2\)Android Service - [https://developer.android.com/guide/components/services.html](https://developer.android.com/guide/components/services.html)
when the queue is empty, the burst buffer deadline $t_d$ is calculated using Equation 5.1 from the mathematical model. Until $t_d$, incoming request objects are placed in the queue according to the First-In-First-Out (FIFO) principle. The queue is emptied and the requests are sent to the server when the deadline timer is reached or if a high priority request arrives. Over 3G, tail time is utilized by transmitting the request immediately if it arrives as the device is in a high power state.

Note that without the adaptation active, all incoming requests are sent immediately.
6 Results

This section shows the results from measurements using the adaptation as well as from measurements with no adaptation at all. All the graphs are derived using the estimations gathered from EnergyBox and will therefore contain theoretical values as mentioned earlier in this thesis.

To start with, we choose to fix two of the three input variables to present how one variable affects the output energy used. All experiments have been executed multiple times with the same packet scheduling profile and an average value of the energy usage is calculated to avoid temporary variations.

6.1 Battery impact

The first measurements were of the battery level and of how our adaptation impacts the energy used. As stated earlier in this thesis, our goal is to use less energy at lower battery levels in order to extend the battery life time. Therefore, our algorithm will be more strict in its energy usage at low battery levels according to Chapter 5, more precisely, as in Table 5.1.

These measurements were made with the RAT type fixed to WiFi and with a signal strength level of 2 which has a dBm interval between $-85$ and $-95$.

The orange bar represents values with adaptation active at different battery levels. The blue bar represents a reference value when we utilize the packet scheduling profile with no adaptation with the same conditions.

The results represented in Figure 6.1 shows significantly reduced energy used the lower the battery levels get. Even with high battery ($>40\%$) we use $24\%$ less energy and at low battery ($<20\%$) we use $39\%$ less energy with the earlier described conditions.

6.2 RAT impact

As discussed throughout this thesis, the energy consumption in 3G and WiFi is significantly diverse. From the theory, we made the conclusion that 3G is much more energy consuming when transmitting than WiFi and therefore we felt it was crucial to reduce the energy significantly in 3G. The result in Figure 6.2 shows that we reduced the energy consumption when connected to 3G with $67\%$ at $10\%$ battery level and high signal strength.
In Figure 6.3 we have corresponding values for WiFi. With 10% battery and high signal strength we reduced the energy with 39%, using our adaptation. These measurements demonstrate that the adaptation reduces the battery usage more for transmissions over 3G than over WiFi.

We only show values with low battery level since showing results for all possible cases would yield an unnecessarily big result section. At 10% we have the strictest adaptation and the biggest impact of the adaptation is therefore visible at this battery level for different RATs. We are only interested in the difference in adaptation between 3G and WiFi, and other fixed values for the battery level or signal strength would give the same general appearance of the power used.

6.3 Signal strength impact

We now present how our adaptation mechanism reduces energy usage depending on the signal level of the current connection. We performed measurements at the low battery level and with WiFi. We see in Figure 6.4 that without adaptation we have a higher energy consumption at lower signal strength which confirms the theory in Chapter 3. Because of this,
we wanted our adaptation to do the opposite and use less energy at low signal strengths to compensate, which is achieved according to the result.

![Signal level impact](image)

**Figure 6.4: Signal level impact on energy usage**

### 6.4 Mixed interactions

Finally we tested the packet scheduling profile containing high priority requests with a 3G connection in order to display how our adaptation mechanism behaves during different interactions with the server.

Starting in Figure 6.5, we have no adaptation active and all the packets are sent immediately as they are scheduled. Due to the high frequency of the background traffic the antenna almost never goes down to IDLE, but continuously remains in the CELL_DCH or CELL_FACH state. The total energy consumed by this experiment is 178 J.

![No adaptation with high priority requests](image)

**Figure 6.5: No adaptation with high priority requests**

In Figure 6.6, we have the same packet scheduling profile but with adaptation active. We can clearly see the scheduling mechanism bundling the packets into a few bursts and reducing the time spent in high power state. Then total energy consumed here is only 93 J, which corresponds to a reduction with 47 %.

As mentioned earlier in this thesis, the high priority packets will be sent immediately and in order to distinguish the influence of the high priority packets we repeat a similar packet scheduling profile, but without the high priority requests. The result from this trace is displayed in Figure 6.7. The total energy consumed in this trace is 72 J.

![Packet scheduling profile](image)

**Figure 6.7: Packet scheduling profile**

During one experiment with 3G and low signal strength, where the adaptation enables the longest $t_d$, we bundled a total of 13 packets in one burst for our application. The total packet size of this transmission is approximately 4 kB.
If we now compare Figure 6.6 & 6.7 more closely, it is possible to do an approximate identification of where the high priority packets were sent. The high priority requests were programmed to be sent at $t = 20, 48, 120$ and $157$ [s], which can be identified by looking at Figure 6.6.

Figure 6.6: Adaptation with high priority requests

Figure 6.7: Adaptation without high priority requests
In this chapter, we are going to discuss our results from experiments made with and without our adaptation mechanism to show and pinpoint the progress we made and approximate how much energy our implementation actually saved.

7.1 Results

Starting with the result from before the implementation we confirmed our predictions based on the theory and could easily see that the 3G connection used notably more energy than the WiFi connection. This wasn’t exactly a new discovery, but what the result shows us is a more exact difference in how much more energy the cellular connection uses.

By this, we were able to answer the first research question that read “Understand how different radio access technologies (RATs) affect the amount of energy that is being used up by the battery?” with help from the theory presented in [5][18].

We chose not to test and display all possible input combinations with the motivation that we didn’t have the means of performing these tests with the consistency we desired. We instead show how the adaptation changes based on one input at the time in order to derive their particular impact.

The measurements on the RAT impact, shown in Figure 6.2 and 6.3 show a higher percentage of reduction in energy consumption. This means that our adaptation functions as desired. Worth mentioning here is that the reduction could have been higher or lower, depending on how we choose our fixed values. But as mentioned earlier, the results are used to show the variations in adaptation based on the current RAT.

From the measurements on the energy impact of signal strength, we can see that the energy used at medium signal level in Figure 6.4 is consuming more energy than both low and high signal level when adapting. This is due to a combination of the facts that the scheduled interactions have a frequency of 8 seconds, and of our choice of $t_{min}$. During our measurements of high and medium signal strength, the difference in $t_d$ is small enough that the same amount of packets are bundled for the two measurements. The energy usage is still reduced compared to when not adapting, but a higher set value of $t_{min}$ for medium signal strength would have given a more desirable outcome of the results. If we had known this before the experiment, we would have either chosen a higher $t_{min}$ or set it dynamically based on the frequency of the packets.
It is important to remember that the numeric values presented in the results are estimated values obtained from EnergyBox. These values are calculated using the configuration files containing data of RCC state consumption and timers. These values vary across smartphones, tablets and other devices and the values used in this thesis corresponds with a generic device for 3G and a Samsung Galaxy S2 in WiFi. Using other configurations would generate other values. The estimations in EnergyBox have been subjective to extensive evaluations and compared to physical measurements. The estimations on 3G and WiFi have been proven accurate to a degree of 94–99 % and 95–99 % respectively [11].

7.2 Method

Our approach to the problem to reduce energy usage by using a dynamic algorithm for adapting network transmissions was shown to be very successful. We were able to save more energy with our adaptation policy compared to the results in [12]. By combining different techniques obtained from previous works we managed to create a adaptation mechanism with good characteristics.

A question that may arise while reading the thesis is why we don’t choose to have the strictest adaptation at all times. As our result show, we have the largest reduction of energy consumption when the strictest adaptation is active. So, if we wanted to save as much energy as possible we should always run the adaptation with the strictest settings. But the goal is to adapt dynamically with the change of battery level and network conditions and not affect the QoS when the conditions are in our favour. We also, throughout the process of deriving our adaptation solution consider the application as intended to be used by a rescue service organisation. The collection of background data is valued highly enough that we didn’t want to allow a maximum amount of delay of the information at all times.

While deriving the results there are some parts that we think we could have improved. In our mathematical model, we derived $\alpha$ and $\beta$ from theoretical studies. During further investigation and comparison with our results we sometimes found the factors inadequate. For example, in $\alpha$ we used values of the power a RAT consumes while in a specific state. It would have been relevant to include the duration spent in this state for each transmission since this adds up to the total consumption of performing a transmission. Then the difference of $\alpha_{\text{wifi}}$ and $\alpha_{\text{3G}}$ would have been remarkably increased compared to the ones used in our mathematical model.

Another approach to having threshold values for the battery level in the adaptation mechanism would be having the battery level as a variable in the mathematical model instead of using set values for $t_{\text{min}}$ and $t_{\text{max}}$ as performed in this thesis. Then the values of $t_{\text{min}}$ and $t_{\text{max}}$ would change dynamically in relation to the battery level.

When reading values of the device’s current battery level, we are provided with a battery percentage by the Android API. We have no means to verify the accuracy of these values, and made the experiments under the assumption that the provided values were accurate. The fact that we introduce threshold values is however favorable, since the adaptation remains consistent for a range of battery level values.

7.3 Wider context

As mentioned in Section [1.1] the batteries in mobile smartphones are struggling to keeping up with the increased energy used by network transmissions in mobile applications. But as shown in this thesis, great reduction of energy usage can be achieved with quite simple implementations. With the rapid increase of mobile devices in the world, we insist on making awareness of energy consumption a fundamental knowledge in software development for mobile devices.
7.4 Future work

Our work has mainly covered the wireless radio access technologies 3G and WiFi. Today, the 4G network is well established in several countries and its usage is heavily increasing \cite{6}. It would have been interesting to investigate how this technique would compare to the ones used in this thesis. At the start of this project, EnergyBox didn’t support calculations for 4G but have recently been extended with support for this technique. Therefore, it would be both possible and interesting to redo our work on the 4G network and compare to the results in this thesis.

During our experiments we ran tests from a static geographical position with the same condition during the complete experiment. Since our adaptation mechanism is adaptive to changing conditions it would be interesting to run extensive experiments with movement and varying conditions to measure the real impact of the implementation during a long time with realistic variation of the conditions.


