Contents

List of Corrections

1 Introduction

When located in an unknown area, it can be useful to have information about the area around you. Examples of this could be map data for navigation, plans for public transport in the area or locations of nearby restaurants. It is usually possible to retrieve this kind of data using a smart phone with a mobile Internet connection. Data connections can however not be trusted to be available everywhere. This could be a concern for someone traveling in an area they haven't visited before.

An example of potential connection problems can be seen in figure 1.1. The map displays a screenshot of a coverage map, showing a few problematic areas, especially north of Aars. The coverage is that of the telephone company TDC hosted on their website[1]. Speeds of "Up to 0,2 mbit" are not shown as it can not be considered a reliable data connection (up to 0,2 can also be 0). There are however further issues than shown directly on the map.

By TDC's own description (translated): "The speeds displayed on the coverage map is derived from a calculation of plausible speed, not a measure of actual speeds. The measure takes into account the altitude of the masts and altitude differences in the terrain." While this makes sense, it also means that obstructions (other than terrain) has not been considered for the map. Adding these in could make the map even worse, as mentioned on the TDC webpage: "(...) If one is outside the building, trees and surrounding buildings can also attenuate the speed, but typically not in the same amount". To make it even worse the displayed speeds are only with 70% certainty: "The displayed speeds shows the expected capacity for the majority of use (∼70%)."

Adding all these considerations into the map, it is obvious that a real-time coverage map would not match the theoretical model.

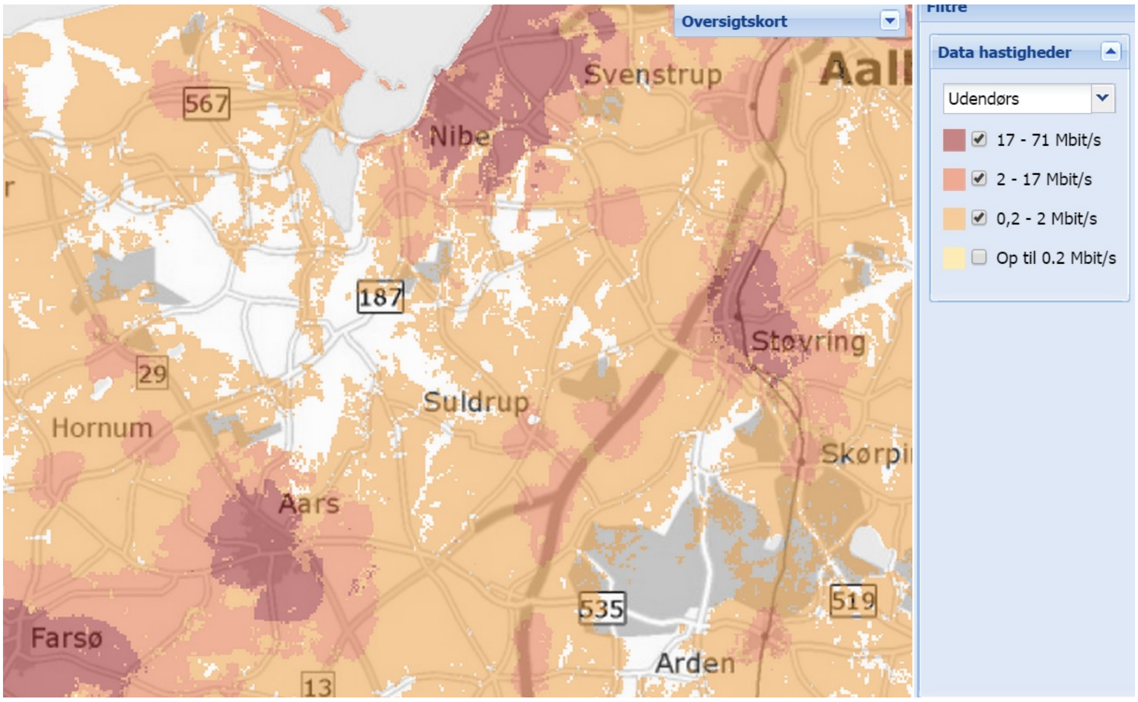


Figure 1.1: Example of missing coverage

Regardless the state of available connections, it would be ideal to always have relevant data on the phone. This would however require that one can predict when to get the data, before the connection drops out. As a user it is possible to download these data manually, but also impractical as it would require manual monitoring of the signal strength. Furthermore it would require some effort to continuously update the phone with all the relevant data. It would be ideal if the smart phone could handle it automatically.

1.1 Roadmap

The following is a brief description of each chapter in the report

1.1.1 Design and Implementation

1.1.2 Test

1.1.3 Future Work

1.1.4 Conclusion

2 Problemanalysis

When developing an application for android devices with the purpose to ensure availability of web resources with varying connection to the Internet, it is important to be able to predict when a connection will be available and when it is about to disappear. In order to do so several approaches are available.

A simple approach is to measure the signal strength of the device and use that as the base for predicting when connection will be unavailable. Using this method a stable but slowly dropping signal strength might indicate a future with no connection available, in this event the required web resources should be cached so they are available doing an offline period. Another scenario could when the signal is unstable but the average remains high, this could indicate interference in the area but no risk of a significant offline period, in this case caching is not necessary.

Another approach is to base the prediction on historical data of the near area. This can be done by gathering data of where Internet connections usually drops and build a map for the available providers. Then the caching will be done when a device is predicted to move into such an area.

3 Prototypes

The following sections are solutions towards detecting an incoming connection problem - only a part of the solution to the problem. Later we will be writing how to maintain a service once one of these trigger reports about endangered connection.

3.1 Downloading Based on Signal Strength

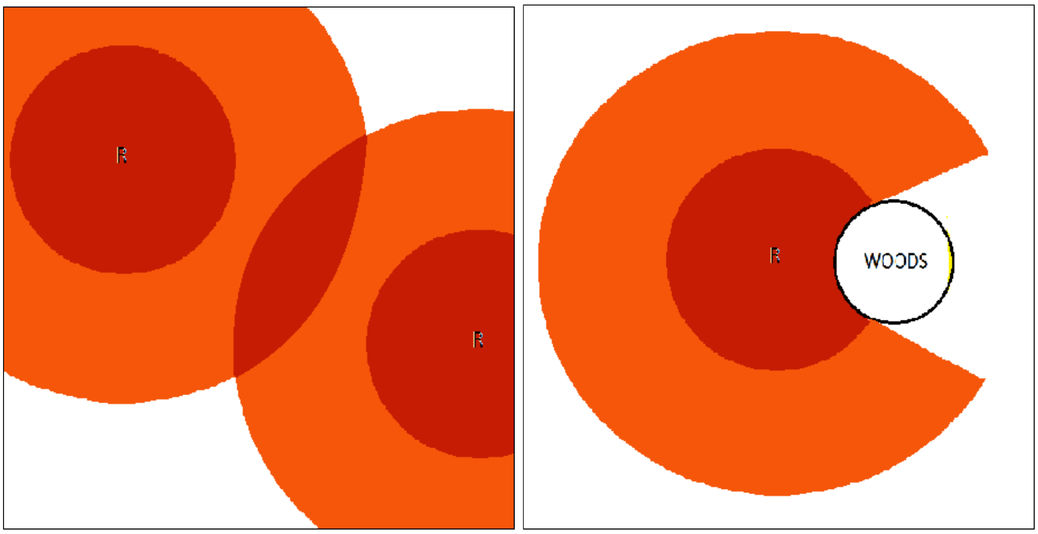
This idea is simple application where the device constantly checks its signal strength. When the application detects a drop low enough in the signal strength, it asks the server for a map according to coordinates. The idea is simple to implement but runs into some restrictions about how much data can be sent in the given time available. If the trigger is set too low there wont be enough time or data to send a big map from the server, and if the trigger is set too high it will trigger from small swings in the signal strength and send too much data all the time.

Advantages	Disadvantages
• Flexible in a sense that it can be used anywhere • Very thin client side	• Very hard to predict sudden swings in signal strength

Table 3.1: this is a dummy

3.2 Dangerzone Map Based on Analysis

The thought behind this idea is simple and easy to implement given we have the information needed to do the analysis. The idea is to design a heat map based on some sort of information - in this example we will be using location of radio towers and the location of forests. The closer to a radio tower, the more heat on the map and as we traverse away from the tower the signal would grow colder. Whenever two signals overlap the heat would rise in that section as illustrated in figure 3.1a. In our example forests obstruct the signal, therefore creating less signal on the opposing side of the forest, this is illustrated in 3.1b. With these implications we can build the entire map based on information gathered before the application is launched.



(a) 2 Radiotower signals (b) Signal being obstructed by a forest

Figure 3.1: Illustrations of the idea

When the heat-map is complete and ready to use, a specific heat point would have to be a trigger point. Solutions that could be implemented when reaching said trigger point is described in . Different Internet Service Providers have already created some heat maps based on theoretical calculations, this is very similar to what we would strive to create. An example of said map is the coverage map created by TDC - this is based on users on the tower, distance to the tower and the surroundings. Another thing these maps consider is differentiating contours. One could imagine that towers on higher points provide signal further than a tower in a slump.

HUSK EN REFERENCE HER - til after triggers afsnit

TDC map kilde (http://daekning.tdc.dk/privat.html)

Advantages	Disadvantages
• No data besides the map needs to be saved • The map is built before application launch	• It assumes there's no unexpected interference • Requires a lot of information beforehand • Requires updates to the map every time new information is out

Table 3.2: this is a dummy

3.3 Coverage Maps Based on Crowd Sourced Data

The idea is based on the same premises' as the map based on analysis from section 3.2 except it is based on connectivity information gathered from different users. The idea is to have many different devices constantly sending information like coordinates and signal strength. This would create a map which would be based on actual data sent from the devices, and could be updated often based on new datasets. This is a more of an hands on approach to creating a coverage map, as signal strength might vary at unexpected events.

In addition to holding the coordinates up against the created map, as the dangerzone map, this solution needs to send some data which can be saved on the database. It is necessary to send coordinates and signal strength, as they are needed to create the map. Dependant on what information you find important, you can save additional data in the database such as provider or device type.

Advantages	Disadvantages
• highly interchangeable map • Requires no data prior to implementation • It is possible to add additional data to the database	• More data needs to be sent from client side • Requires a lot of data before the map is usable

Table 3.3: this is a dummy

3.4 Crowdsourced Analysis

Based on nearby devices network-connection - this trigger solution will track where devices previously lost connection and where they regain connection. Use this information to make a live registration of nearby connection dead zones, with a limited lifespan. Since new connectivity-data will be available more often than planned dead zones scans, the lifespan can be kept low and the available data may be kept dynamic for highly populated areas. The lifespan of connectivity-data may then be increased in areas less populated.

According to [link], crowdsourcing can be used to outsource routine tasks to a large unknown crowd of participants. The participants require no or limited skills to participate in the dead zone analysis, they will most likely not even notice, since their participation will be automated after installation of a mobile-application. This mobile application will automatically log data and send it to a server. Uniting this data, from a large number of participants, will give a live map of connection dead zones.

(temp link until JabRef is set up: http://raptor1.bizlab.mtsu.edu/s-drive/DMORRELL/Mgmt

Advantages	Disadvantages
• Connectivity-data will have short lifespan and refreshed often • Does not require field analysis by designer	• Requires a large number of participants to effectively work • Requires participants to resident over a wide area

Table 3.4: Crowsourced analysis advantages and disadvantages

3.5 Handlers

3.5.1 Packaging

A solution to losing network services and therefore the possibility to download maps, are to detect when the device is losing network-connection in appropriate time. Hereby the device may start downloading maps over the nearby area and store for use once the network connection is lost. These maps should then be compartmentalized into groups with different prioritzation, based on a criteria of nearest-package-first.

Downloading the by nearest-package-first requires an indexation of map-packages for a given location of a device. Since our service are to provide maps even when network connection is lost, we intend to use a map-service like Google Maps and not develop our own. Google Maps already divided a map into small packages, which we could use for downloading and displaying upon request depending on simple GPS-coordinates. New packages will be downloaded and displayed when the user moves the map on a mobile-device. The problem that has to be dealt with, are which packages to download without knowing more than GPS-coordinates.

We put forward three solutions to this problem:

- Define a circular perimeter around the GPS-coordinates. Start downloading map-packages on the device GPS-coordinates, then prioritize after nearest-package-first until the perimeter is covered with downloaded packages.
- Define a rectangular perimeter around the GPS-coordinates and proceed as above.
- Process the GPS-coordinates over time, which will provide direction and velocity of the device. This should be used to download packages in the direction that the device is moving, and disregard the package behind the device. Considering the velocity, the width of required packages can be defined and as velocity increase packages further away can be downloaded. This solution does not follow nearest-package-first defined on first set of GPS-coordinates but rather try to dynamically follow nearest-package-first based on time and movement.

Advantages	Disadvantages
• adv • adv	• disadv • disadv

Table 3.5: this is a dummy

3.6 Supplements

3.6.1 Crowdsourced sharing

Advantages	Disadvantages
• adv • adv	• disadv • disadv

Table 3.6: this is a dummy

3.6.2 Event based handlers

Advantages	Disadvantages
• adv • adv	• disadv • disadv

Table 3.7: this is a dummy

4 Architechture

5 Transmission Control Protocol

The Transmission Control Protocol (TCP) is a part of the Internet protocol suite. Together with the Internet Protocol (IP), TCP is so widely used that the entire Internet protocol suite, containing many protocols, is often just called TCP/IP.

In the following, the parts of TCP that are particularly important for this project are described. This chapter is based on [3]. [1]

5.1 Data Transfer

Because we are working with transfer of data to a device with an uncertain connection, it is relevant to consider how TCP handles transferring data. An important aspect of this is to ensure reliable transmission, i.e. to make sure all the data is transferred correctly. To ensure this, each byte of data gets a sequence number, allowing the destination host to reconstruct the data in case of for example packet loss. Additionally, when a packet is received, the receiver sends back an acknowledgement, and if such an acknowledgement is not received the packet will be sent again. To ensure correctness of the packet content, each packet has a checksum included which the packet's content can be compared to upon arrival at its destination.

This possibility of error checking comes in handy in our project. When a map is transmitted to a client, it ensures that the map data is not corrupted. This is very desirable since corrupted map data could make the map unusable in the best case, and show wrong map data in the worst case.

5.1.1 Flow Control

It is possible for the server to sent more data than the receiver can process, and to accommodate this a flow control protocol is used. When data is sent, the receiver

[1] FiXme Fatal: Find better source.

answers back with a *window size*, telling the sender how much more data it is able to process. If the window size reaches 0, a persist timer will be set to account for the possibility that the updated window size was simply lost. When the timer runs out the server will send a small packet, probing the receiver for an updated window size.

5.2 Predicting Signal Loss

To predict when a device might suffer a signal loss and become disconnected from the internet, two different scenarios must be considered. The scenarios are as follows

- The device is moving into the country side
- The device is in the city[2]

In order to predict when the signal may be at risk of being lost in the first scenario, it makes sense to look at measurements from the antenna of the device. A number of measurements, five for example, can then be stored on the device. These values should be error corrected (e.g. using standard deviation) to make sure sudden fluctuations will not trigger an alarm, telling the device that it has lost its signal. Since coverage is usually worse outside of big cities compared to in the cities, the signal will be lost gradually and not suddenly. Therefore it makes sense to measure a number of antenna measurements, and decide whether they are falling at a rate which would indicate heading towards the country side.

This functionality uses the moving average, which has the purpose of smoothing out temporary fluctuations in measurements. [2]

In the city, it is very hard to accurately predict signal loss, even with precise heat maps. This is partly because phones and their antennas vary in quality and partly because signal loss in cities is sporadic and the signal usually returns quickly. Therefore signal loss is not predicted in the city, but only outside.

Determining Location

When deciding what web-content is required available it is important to know the locational context of the user, specifically whether the user is in a urban area or not. Different methods can be used to determine this, such as basing it on map data or connectivity.

When basing it on map data the idea is to compare the current GPS-coordinates to the map data, in order to determine

[2] FiXme Warning: should this be removed? is the scenario not needed anymore?

6 Implementation

7 Conclusion

8 Scenarios

8.1 Scenario 1: Maps & Navigation - City Areas

In a large city, for example one of the 5 largest cities in Denmark, data connectivity can be obscured by many big buildings, concrete parking garages or when descenting into the subway. This type of data connection loss can be unpredictable and happen in an instant.

A scenario can be a travelling salesman arriving at an unknown large city. The salesman is trying to find a customer and have to navigate through public transportation. The customer is awaiting the salesman for a business meeting, and therefore the salesman is limited by time. The salesman is relying on information about public transport timetables and a journey planner to direct him to the customer.

A number of challenges may arise for the salesman regarding connectivity issues. The salesman may have plans to ride one bus to the customer, but upon arriving at the bus stop he realises the bus is delayed. He now has to consult his device for an alternative bus to ride. Since the salesman is under time pressure, he is relying heavily on having data-connection. Another challenge may arise if the salesman descents into the metro-station. The salesman may have become unwary about which train to enter, and is now relying on still having data-connection. The subway is underground and build from concrete which may severely reduce the data-connection.

The challenges arise when the salesman is under time pressure and has to deal with unpredictable traffic and public transportation. The salesman would be more secure if a service would be able to provide the desired information even if the device has lost data-connection.

8.2 Scenario 2: Maps & Navigation - Countryside Areas

When travelling out of cities, data connectivity can be unavailable for a longer period of time. This type of data connection loss is slowly fading away as you move further away from the connection source.

A scenario can be a couple going for a walk in the woods on the countryside. They are visiting an unknown area and walks into the woods. They happen to get lost and wants to find their way back to the car, but in the woods no data connection is available and they are lost.

A modification of this scenario could be a couple that wants to get additional knowledge of the area they are walking around in, e.g. visiting Rold skov in Denmark, they might want to read the stories about the robbers from Rold. It is also possible that they want to know about the attractions in the area, and directions to them.

The challenges are predicting that the data connection will be lost in the near future, and what services should be attempted to download to a mobile device.

8.3 Scenario 3: Bus and Train

In this scenario the problem be solved is to minimize Internet drop-outs when travelling on frequently used bus or train lines. The idea is to collect data on a user base and use that data to identify areas where the user might lose connection.

The solution should be able to detect travelling by bus or train and predict the route and destination. It should reduce the user experience of no Internet during the travel and make relevant information available for the destination, should it be in a poorly connected area.

9 Transmission Control Protocol

The Transmission Control Protocol (TCP) is a part of the Internet protocol suite. Together with the Internet Protocol (IP), TCP is so widely used that the entire Internet protocol suite, containing many protocols, is often just called TCP/IP.

In the following, the parts of TCP that are particularly important for this project are described. This chapter is based on [3]. [1]

9.1 Data Transfer

Because we are working with transfer of data to a device with an uncertain connection, it is relevant to consider how TCP handles transferring data. An important aspect of this is to ensure reliable transmission, i.e. to make sure all the data is transferred correctly. To ensure this, each byte of data gets a sequence number, allowing the destination host to reconstruct the data in case of for example packet loss. Additionally, when a packet is received, the receiver sends back an acknowledgement, and if such an acknowledgement is not received the packet will be sent again. To ensure correctness of the packet content, each packet has a checksum included which the packet's content can be compared to upon arrival at its destination.

This possibility of error checking comes in handy in our project. When a map is transmitted to a client, it ensures that the map data is not corrupted. This is very desirable since corrupted map data could make the map unusable in the best case, and show wrong map data in the worst case.

9.1.1 Flow Control

It is possible for the server to sent more data than the receiver can process, and to accommodate this a flow control protocol is used. When data is sent, the receiver

[1] FiXme Fatal: Find better source.

answers back with a *window size*, telling the sender how much more data it is able to process. If the window size reaches 0, a persist timer will be set to account for the possibility that the updated window size was simply lost. When the timer runs out the server will send a small packet, probing the receiver for an updated window size.

Bibliography

[1] TDC (2014, October 6). *Dækning.*

[2] Wikipedia (2014a, October 20). *Moving average.* http://en.wikipedia.org/wiki/Moving_average.

[3] Wikipedia (2014b, October 12). *Transmission Control Protocol.* http://en.wikipedia.org/wiki/Transmission_Control_Protocol.