

# One week in the life of the iPhone's Internet Traffic

Proseminar Technische Informatik SoSe 11

**Raphael Urban**

Freie Universität Berlin  
raphael.urban@fu-berlin.de

## **ABSTRACT**

This paper is the result of a study which analyzes the characteristics of mobile data with regard to both amount and content of the data collected. It will be shown what information is revealed through a man-in-the-middle attack on an unjailbroken iPhone 3GS within one week of internet usage. The results will reveal that the average transfer size of a datastream is very small in comparison to the headersize of a datapacket as well as a strong varying transfer rate and response time, highly dependent on signal strength. Furtheron, the distribution of traffic in terms of port usage and data traffic per host will be shown. The paper will conclude with a final analysis of the data collected and summarize the information revealed as well as pointing out the dangers associated with it.

## **1. INTRODUCTION**

Since the introduction of the iPhone and other alike smartphones, the demand for mobile internet has raised increasingly fast. This development is projected to increase even more over the next years. A projection by Nokia Siemens Networks [2] says that mobile data traffic will increase by 300 times and outrun hard-line traffic by 10 times in the next five years. The possibility to gain full access to the internet without sitting in front of a computer has become an important criterion for the development of new smartphones. The applications range from receiving and answering one's emails everywhere and at any time to real-time video chat in HD with people around the world. In combination with a GPS receiver, the possibility of locating oneself increases the range of possible applications even more. One example is an application showing restaurants nearby, ordered by distance or quality which is determined by customer ratings published on different homepages (e.g.: Qype.com [1]). Another example of the use of mobile internet in combination with built-in tools of smartphones can be seen in the magnetometer of the iPhone 3GS (which is later replaced by the Gyroscope in the iPhone 4 to increase precision). The magnetometer enables the mobile device to determine in which direction it is pointed so that it can download information from the internet to present what can be seen in this direction. (e.g.: AroundMe [3])

In this paper, the results of a self-study are presented, which analyze the data collected from

an iPhone 3GS within one week of use. Its primary aim is to point out and to characterize the characteristics of internet traffic on smartphones in general as well as the search for data that could become a threat to personal privacy or security. It will start with the presentation of the technique used to collect data and of how much was collected. Afterwards, it will show which technical characteristics can be seen, which application was used to what extent and how the data differs from data collected on a computer at home. It will end with a summary of what information can be drawn from the data about the user of the iPhone.

## **2. MOTIVATION / RELATED WORK**

Triggered by the tremendous possibilities and the fast expansion of smartphones into more and more areas of life, risks have emerged and intensified which did not exist before. Personal data that can be found on smartphones usually include direct personal data like the full name, birthday or the like, interpersonal data like emails or the address book and location based data like places visited or the current location of the smartphone. As most of this data is very sensitive information, the user wants to ensure that this data can not be read from anyone except him/herself. Because of the rapid development, new information or even new types of personal data can be collected and transmitted over the internet which causes new possible dangers. Ensuring the security of this data is of growing public interest as more and more data is stored on smartphones as well as the number of smartphones rising constantly. In order to guarantee the security of personal data, it is crucial to perform recurrent inspections of traffic produced and searches for possible threats to personal privacy. Since this study has primarily been performed to get a rough picture of the characteristics of produced traffic by a smartphone and secondarily to seek for any possible threat to the personal data of the user, the latter aspect is only touched on in this paper.

The study is inspired by the work of a study [8] about the traffic on smartphones in general. It presents a wide range of data, collected from 2 datasets of 10 and 22 users. As it concentrates on Windows Mobile and Android devices, this study focuses on the iPhone and the iOS 4 and takes only one dataset as source.

Another study [11] deals with the iPhone and its traffic but rather intends to show the differences in user acceptance and internet usage in comparison to equivalent devices of Nokia. The study explains the success of the iPhone and roughly characterizes the distribution among users in terms of internet usage.

As most thematically related studies work with multiple and very large datasets in order to present representative data, this study is more focused on the procedure and demonstrates the results of a random sample with data that may not be representative for larger datasets.

### 3. CAPTURING DATA

To capture data from an iPhone for later analysis, there are several possible methods of monitoring the traffic produced:

- The first method is to use the tcpdump function of the Terminal app which is included in the iPhone OS. This would be the most precise way of capturing every byte of traffic produced and being the most elegant way at the same time because it does not require any setup outside the iPhone. The disadvantage with this method is that the Terminal app is not accessible on an unjailbroken iPhone. So in order to access and use the tcpdump function, the iPhone has to be jailbroken, which is undesirable because it is an illegal act of work.
- The second method is to provide a wifi network and send all traffic via wifi, which is automatically preferred by the iPhone as mode of transmission, because it ensures a higher digit rate than the UMTS or EDGE network and the iPhone automaticly choses the available connection type with the highest data rate. The traffic of the iPhone in the wifi network is monitored by a computer or laptop with a wireless network card and a network sniffer. This methods has its disadvantages in several aspects. One is that it cannot be ensured that all traffic is transferred via wifi because the wifi connection to the network does not stay alive once the iPhone enters stand-by mode. Any traffic produced between waking up from stand-by and the establishment of the connection will be sent via UMTS or EDGE and will not be captured by the computer. The second and stronger counter-argument against this method is that it is not possible to use the device anywhere else but in the range of the wifi network which makes some applications useless.
- The third and in this study preferred method is a man-in-the-middle attack on the iPhone using a VPN server as proxy to route all queries from connected devices (the iPhone will be the only one) to the requested location. The requested data will also be returned via the VPN server. In doing so, both the outgoing and incoming data will be transmitted using the VPN server, making it possible to create a full log of all traffic produced. The advantages of this method are that the VPN server is accessible from everywhere because the device can connect to the server using both wifi and UMTS network. Again the major disadvantage is that the VPN server disconnects due to a timeout and the build-up of the connection requires some time in which data can be transmitted without being monitored by the server.

Taking all aspects into account, the setting up of the VPN server is the preferred method because mobility is not restricted as with the wifi setup method but no jailbreaking of the device is necessary. The only negative aspect of this method is the delay when connecting to the VPN server, in which some data may be transmitted.

#### **4. ANALYSIS**

This section describes the general setting and environmental factors of the used device as well as the following analysis of the data using different functions and methods of a packet analyzer tool.

The used iPhone 3GS, operating with iOS 4.3.3 has an unlimited data plan (o2-de) and an activated account of MobileMe [6]. MobileMe is a service provided by Apple, offering the possibility to have the calendar, mails and notes as well as the contact list synchronized on many different devices. This is a very important aspect for the study as it changes the functionality of the Mail, Calendar and Notes applications (although the Notes app and the address book was not used in this study) in the way that it automatically uploads any changes to all of these apps to the MobileMe server and thereby causes data traffic. At every call up of one of these apps, the current server status is also checked for changes on other devices.

In the analysis, use is made of the packet analyzer tool Wireshark [4], formerly named Ethereal. It provides an insight into the traffic collected by the logger of the VPN server, breaking it down into single data packets, providing all necessary information about headersize, content and affiliation to datastreams. Wireshark also offers a wide range of statistical information and graphical representation of datastreams and packets. To analyze different datastreams, it also offers very detailed and precise filter options. With these filter options and the statistics functionality, many different sights can be drawn from the data to analyze it. The following sections will present the found characteristics and later draw the line when reconsidering the differences to hard line traffic and possible threats to the data collected.

#### **5. ANALYZED DATA / EVALUATION**

This section will present the findings of the analysis by showing the most significant and interesting aspects of the data collected.

In the week of use, altogether 14.32 MB of traffic has been produced and monitored, transmitted in 20315 data packets. At first, it is important to visualize the distribution of traffic among the used servers to procure an understanding of the ratio of traffic produced per host. During the monitoring, the device produced traffic using applications and services that were transmitted between the device and 50 different IP-addresses. Figure 1 shows the top seven

applications as to usage of traffic with regard to the number of packets as well as to the amount of data. The pie chart on the left represents the amount of packets sent and received by these applications whereas the chart on the right distinguishes the portion of bytes produced by this application. The seven shown apps account for more than 86% of traffic produced.

The first aspect of interest is the fact that the streaming of a video on Youtube covers more than 66% of all traffic. This aspect shows that in contrast to other mobile traffic producers, video streaming requires a very large amount of data. In the figure, the video data is encapsulated from the rest of the Youtube

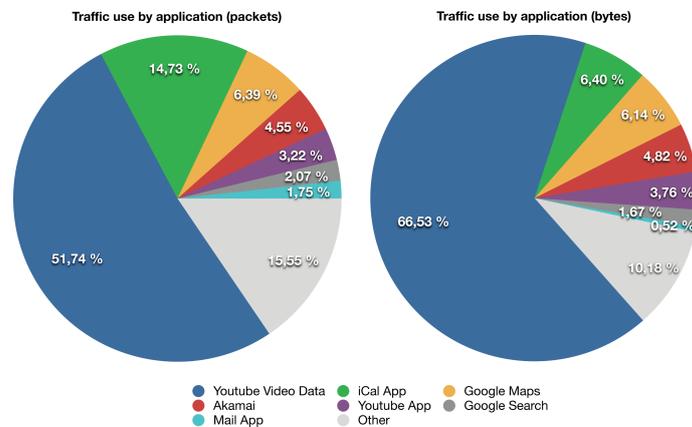


Figure 1: Traffic use by application

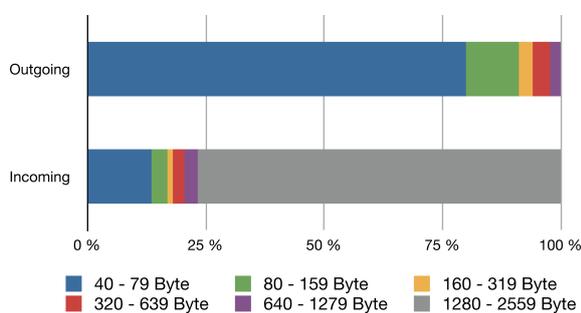
app to point out that the video data transfer alone exceeds all other data transfers. The functionalities of searching for videos and previewing them which are offered by the Youtube app are shown separately, but even they produce more traffic than most apps so that both video data and preview and search data are placed under the top seven of traffic producers.

The next interesting aspect to be seen in this figure is that the Calendar app is the 2nd biggest traffic producer, directly followed by the Google Maps app. To understand the fact that the Calendar app is placed as 2nd highest producer of traffic, it is critical to point out that the device possesses an activated MobileMe account because otherwise the Calendar app does not produce any traffic at all. Taking a closer look at the Google Maps app, it can clearly be seen that the downstream is much higher than the upstream. The reason for this is that the main functionality of Google Maps is to view the surroundings of the current location of the iPhone or any location chosen by a search function in map or satellite view. Therefore it needs to download image data from its server which is the main producer of traffic in this app. The upload of data is relatively small because only the current location or the search input needs to be uploaded. The fourth largest traffic part belongs to an internet service hoster named Akamai. It offers a worldwide server network which is used by Apple as primary content network [7] for applications such as the iTunes store and as provider for their website.

Since the monitoring of the traffic was done by using a VPN server, the representation of the Mail app is misleading because most of the traffic is produced by the push technology of Apple. This means that a new email arriving on the MobileMe server is instantly forwarded to the iPhone. Since the connection to the VPN server timeouts if not used for a certain time, it has

to be manually reactivated once leaving stand-by mode. Thus, most of the traffic produced by the Mail app occurs when the device is in stand-by mode, which causes that most of the traffic of this application is not monitored and therefore renders the captured data of the Mail app unusable.

**Transfer sizes** Now, a closer look at the composition and characteristics of the collected data packages is taken. The first interesting aspect in this matter is that although around 42% (8545) of the packets logged were produced by outgoing traffic, the data volume of the outgoing traffic accounts for only 6.2% ( $\approx$  0.89 MB). Therefore, the average packet size diverges widely between outgoing and incoming traffic.



**Figure 2:** Distribution of packet size

Figure 2 shows the distribution of packet size as produced by outgoing and incoming traffic in relation to all send/received packets. The packet size ascends from left to right, with every color change representing packets being twice as large as the packets before. As outgoing traffic mostly consists of queries for data which are normally very small, the number of small packages is almost 80% of all outgoing packages.

The incoming traffic shows a very different result, that is to say a very small number of small packages and a large number of big packets in which much content is transmitted. As already shown, the most used app with regard to the amount of data transmitted is the Youtube app which is also the app with the largest upload volume. The reason for this is that for every incoming data packet, the TCP connection requires an ACK packet to confirm the receiving data on the server. Thus the size of a video stream, the iPhone has to upload many ACK packages when streaming a movie. The measured packet loss ratio of 8.7% for the video increases the traffic even more.

**Traffic composition** The next aspect is the technical analysis of data transfers and their properties, which will be followed by the analysis of the packets themselves and their characteristics with regard to their content.

The first differentiation of the traffic monitored is to divide it into outgoing and incoming traffic. As the outgoing traffic has an average throughput of 1.43 KB/s, incoming traffic is received with an average of 21,6 KB/s. While the ratio of the amount of incoming packages to outgoing packages is 42% to 58%, the ratio of the amount of data sent and received is 6%

to 94% which implies a strong diverging build-up of packet size as already shown. This is also reflected in the average packet size which is about 104 KB in outgoing packets compared to 1141 KB in incoming packets. The retransmission rate gives a first hint about how stable the connections are and how much data has to be resent because of packet loss. It takes up about 2.86% which is very high compared to wired internet at a home computer. According to the SLAC laboratory [9], the average median loss rate is less than 0.1% in Germany.

Now, a closer look is taken on how the data is transmitted. The vast majority of data is transmitted via unencrypted TCP and takes up more than 95.1% of data. The remaining 4.9% traffic are encrypted SSL traffic via TCP (4.6%) and UDP (0.3%) traffic, only used with the DNS service to translate a given hostname into an IP and the DHCP service for automatic network configuration. The only two applications used in this study that utilize SSL over TCP are the Calendar app and the Mail app. The SSL encryption is offered by the American company Verisign [10] which can be taken from the contents of the TCP stream.

**Usage of common ports** To precise the found aspects of traffic types, it may be helpful to analyze the usage of common ports as they break down connection types to standardized services.

As it can be seen in Figure 3, the all-dominant port used is port 80, that is to say http. It covers almost 77% of the number of packets sent as well as more than 88% of bytes transmitted. The following port HTTPS (port 443) is used for 10.5% of traffic. HTTP is used for browsing websites and especially for downloading data as HTTPS, being the secured version of HTTP, is used for secure websites and by the Mail and Calendar app. These apps are the only ones used in this study that transmit data via secured conversations. Taking both of these ports together as secured or unsecured HTTP traffic, they cover almost all traffic, meaning 99.22% of bytes and 97.04% of packets transmitted. The third port used by TCP traffic in this study is the port IMAPS (port 993), which is used by the secured version of the IMAP protocol, that is to say for secured email transfer. As it is also transmitted via TCP, it increases the percentage of TCP traffic even more to 99.73% of all transmitted bytes. It also belongs to the Mail app.

Ports used		
Port	Packets (%)	Bytes (%)
<b>HTTP (80)</b>	76,81 %	88,67 %
<b>HTTPS (443)</b>	20,22 %	10,54 %
<b>IMAPS (993)</b>	1,75 %	0,51 %
<b>DNS (53)</b>	0,90 %	0,14 %
<b>DHCP (67)</b>	0,31 %	0,14 %
<b>other</b>	<0,01%	<0,01%

**Figure 3:** Traffic usage by port

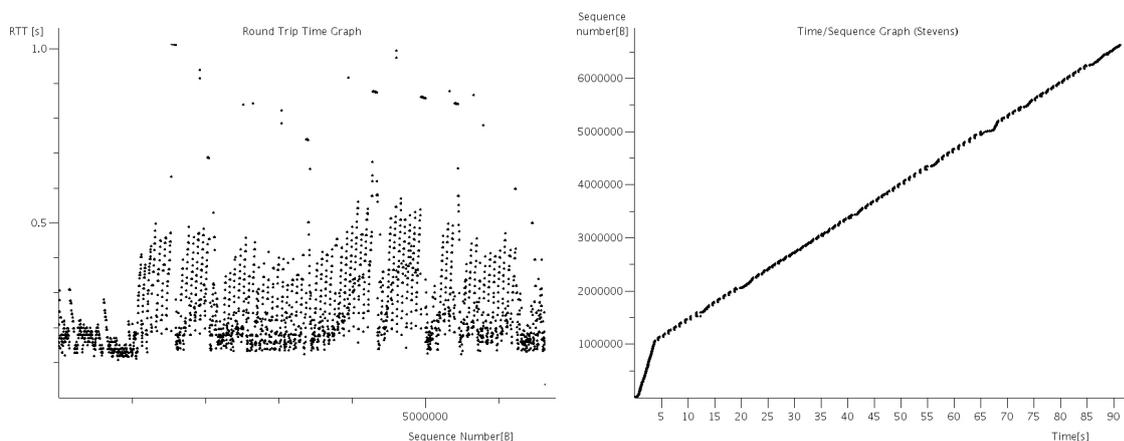
All the rest of the monitored traffic ( $\approx 0.27\%$ ) belongs to protocols using UDP for transfer, almost evenly distributed among DNS (port 53, 0.14%) and DHCP (port 67, 0.13%). The usage of these two services is explained at the beginning of this part of the analysis. To sum this

part up, the observation of common ports used refine the view on the utilization of methods of traffic by demonstrating again the magnitude of TCP usage. In comparison to the study on traffic on smartphones [8], it can be seen that the amount of tcp traffic is analogical high. A difference is that in the given study, HTTPS is used more than HTTP (31.66% to 22.16%). This cannot be seen by the results of this study.

**Performance** In this part, the performance will be investigated by analyzing an exemplary TCP stream in order to gain information about retransmission rate and round trip time of packets as throughput has already been shown at the beginning of this paragraph.

To analyze the performance of connections, a common way is to consider the round-trip-time necessary to receive the data inquired as well as the time/sequence graph, showing the development of the sequence numbers of packets over time. Thus most conversations do not have the granularity because of the little data transfer, the representative TCP connection to analyze is the video stream mentioned at the beginning because it provides enough data.

Figure 4 shows the round-trip-time on the left hand side and the time-sequence graph by W. Richard Stevens, both printed by the tool Wireshark. Thus one graph per tcp stream is created, the figures shown are taken from the largest tcp stream which is, as already pointed out the streaming of the youtube video.



**Figure 4:** Round trip time (rtt) graph and time/sequence graph for the Youtube app video data

The figure on the left is a scatter plot, showing the round-trip-time for any data package transmitted in this tcp stream. A dot represents the time (y-axis) between the sending of a packet, specified by its sequence number (x-axis) and the response packet from the server. The scatter plot shows that most packets need between 100ms and 500ms with a concentration at 100ms to 300ms to return to the device. It can also be seen that a few packets need over 0.8 seconds to return to the device. This results in an estimated median rtt of 200-300ms.

The results of the study [8], mentioned in the paragraph related work, shows a result in which the rtt is notably larger but as the results of this analysis are only based on one tcp stream, there is a high probability that other analyses show the same picture.

The figure on the right presents the development of the sequence numbers (y-axis) over time (x-axis). It can be seen that there is a linear increase in the sequence numbers with a strong change in speed at around 5 seconds. As the increase in the sequence numbers starts with a rate of approx. 300,000 numbers per second, it is reduced to approx. 70,000 numbers per second. After this break, the rate is not modified during the rest of the conversation except for two small irregularities, approximately at sequence number 2,000,000 and 6,500,000. At these points, the sequence number per time rate drops at first and then rises again to return to the overall rate.

When taking both figures into account, some parallels can be seen and explained. In the rtt graph, it can be seen that the round trip time for the first  $\approx 1,000,000$  sequence numbers is very good compared to the rest of the graph. Therefore the sequence numbers can increase faster. At the time of the break at the time/sequence graph, it can easily be seen that the round trip time increases abruptly. Thereby the sequence numbers increase more slowly which leads to the slab of the graph. The small irregularities described above can also be seen in the rtt graph because at the given points of the time/sequence graphs, the number of packets with a high round trip time is significantly lower which leads to the stronger increase in sequence numbers.

To sum up the latest observations, it can be seen that there is a relatively great variation in the round trip times which is probably triggered by different signal strength.

**Qualitative analysis** This paragraph deals with the characteristics and contents of the transmitted packets itself, starting by describing technical specifications and ending with an insight into the personal data legible from the collected packages.

As the general composition of the traffic concerning amount and size of packages was already shown in the beginning of the analysis, the amount of overhead per package due to the addition of headers of lower layer protocols is also an interesting fact concerning the structure of data traffic. In TCP traffic, the headerlength for data transfer packets is 20 Byte long with the possibility of adding bytes as additional options. To this value, the IP header of 20 Bytes is added to every packet. The found lengths of TCP headers added to data transfers are 20 and 32 Byte. The 32 Byte headers contain an additional option, presenting information about the round trip time as it contains the current timestamp of the sender of the packet. This timestamp is returned with the ACK segment, answering the given TCP packet. With

this information, the server can make assessments of the round trip time more precise. As this option is not set in every data packet monitored, this functionality is not relevant for the study.

The calculated headersize adds up to  $\approx 944$  KB. Compared to 13.4 MB, it represents an amount of 6.65% which is a relatively high value because it 6.7 MB out of 100 MB of traffic is overhead. Separating outgoing and incoming traffic concerning overhead, it is found that 46.32% of outgoing traffic is overhead, due to very small package size (as already shown above) whereas incoming traffic produces only 4.27% of overhead in transfer. This huge discrepancy supports the fact that outgoing traffic mostly consists of queries for data and very few uploads of data whereas incoming packets are mostly full data packets where the overhead has the same absolute size but differs strongly when it comes to the relation of overhead and the amount of data.

The following part will now deal with the handling and dangers of personal data on smartphones, especially the iPhone as it is used in this study. Therefore, the contents of the monitored traffic is searched for any readable personal data.

To search for personal data efficiently, some presorting has to be done to avoid searching in over 270 tcp streams, that is to say, ruling out applications that use 100% encrypted traffic or are less likely to transmit personal data because their functionality does not require them. As the Calender and Mail app contain the most interesting data for its purely personal character, they both

```
<query id="30" timestamp="0"
  type="getforecastbylocationid">
  <list>
    <id>GMXX0007|638242</id>
    <id>USNY0996|2459115</id>
  </list>
  [...]
</query>
```

**Figure 5:** Content of a weather forecast query

transmit every datapacket via secured tcp streams. Another important aspect of personal data is the creation of an exact or diffuse movement profile. For information concerning this profile, the most interesting apps that have already been mentioned are the Maps app and the AroundMe app as well as the Weather app, although the last one does not determine the current location. As AroundMe is not used in this study, this app is not observed any further. The readable content from the Maps app transmissions is the current time of the location request as well as the version number of the Maps app. All content with regard to locations parameters or coordinates are encrypted by software and sent unencrypted via TCP streams. The most interesting fact concerning location is sent by the weather app. While the con-

tent of the Weather app, meaning the weather forecast itself, is transmitted encrypted, the locations saved in the app are transmitted in clear text. Figure 5 shows an excerpt from the query for the weather forecast for the given list of locations. The locations are given in the format `< id >xxxxyyyy| [...] < /id >` of which only the combination of 4 characters (x) and 4 numbers (y) is relevant for determining the location. The fact that the stored locations for weather forecasts are transmitted in plain text is not directly a threat to personal security as it does not contain the current or past locations because the places are entered manually. On the other hand, it can easily be argued that users of this app want to gain information about the weather in their city or area as well as places of interest. In the case of this study, the weather app transmits the locations Berlin (DE) and New York (USA), which represent the hometown and another place of interest of the author of this study. Concluding, it can be said that this information can be used to create a diffuse movement profile about the user which can definitely be classified as personal information.

## 6. CONCLUSION

Based on the data of one device, monitored in one week of usage, this paper presented a rough overview on traffic on smartphones, in this case the iPhone 3GS. It was shown that mobile traffic has a relatively high overhead, a significant loss rate and a long round trip time. It also showed the proportions of usage per application and service, finding that there are high discrepancies in the data volume consumed. Other technical findings are the high download in contrast to a very small upload and the very huge amount of TCP based communication. The second aspect of interest is that the content of the monitored data concerning personal data should not be human readable. To this point, it is revealed that although the most sensitive data is securely transmitted, some personal information is revealed in plain text which may contribute to a diffuse movement profile.

**Significance of this study** As this study bases its findings on only one dataset, constructed in only a week of usage, it is not very likely that the results found would match the results if another dataset was collected from multiple sources and over a longer period of time. This study is to give an idea about the procedure for finding the information presented but not present data for comparison.

## 7. REFERENCES

- [1] Qype reviews: <http://www.qype.com/>
- [2] N. Wood.: *"Mobile data traffic growth 10 times faster than fixed over next five years."*  
<http://www.totaltele.com/view.aspx?ID=448681>
- [3] AroundMe: <http://www.tweakersoft.com//aroundme.html>
- [4] Wireshark, original name Ethereal, Version 1.4.7, August 2010:  
<http://www.wireshark.org/>
- [5] Akamai: <http://www.akamai.com/>
- [6] MobileMe, Apple Inc.: <http://www.apple.com/mobileme/>
- [7] [http://www.akamai.com/html/about/press/releases/2003/press\\_042903.html](http://www.akamai.com/html/about/press/releases/2003/press_042903.html)
- [8] H. Falaki, D. LyMBERopoulos, R. Mahajan, S. Kandula and D. Estrin:  
*"A first look at Traffic on Smartphones"*, IMC November 2010
- [9] L. Cottrell and S. Khan. ICFA SCIC network monitoring report:  
<http://www.slac.stanford.edu/xorg/icfa/icfa-net-paper-jan09/report-jan09.doc>, 2009.
- [10] Verisign: <http://www.verisign.com/>
- [11] Rich Ling and Pål Roe Sundsøy, *"The iPhone and mobile access to the internet"*,  
Chicago, Illinois, USA, May 2009