Location aware performance measurements

G.Inberg

Thesis for a Master of Science degree in Telematics from the University of Twente, Enschede, the Netherlands.

November 17, 2006

Graduation committee:

ir. R.G.A. Bults

Dr. ir. B.J.F. van Beijnum

Dr. ir. I.A. Widya

Abstract

Nowadays, a lot of mobile devices exist that are characterized by their high processing power and low costs. Next to this, they have the ability to connect to the Internet using wireless communication technologies like GPRS, UMTS or WLAN. These advances in mobile device capabilities and communication technologies have enabled new mobile applications by which services are deployed on the mobile device which are available to users on the Internet. This is called nomadic mobile service provisioning and has been successfully used in the remote healthcare monitoring domain. Nomadic mobile services are characterized by the limitations of their mobile devices and their wireless communication technology. The ASNA chair developed a Mobile Service Platform (MSP) to deal with these issues.

This thesis focuses on performance measurements of MSP, using a WLAN as MSP transport system, and how the location of a mobile device can affect this performance. Integrating these performance measurements within applications enables pro-active adoptation of an application.

To accomplish this goal a software framework has been developed to retrieve time and location information via GPS and an evaluation system is build to evaluate the MSP performance. The evaluation system is using MSP as a service infrastructure with example workload parameters from healthcare applications which have, in other circumstances, already used MSP.

My research has shown that the bottleneck in MSP is not the underlying communication technology (WLAN) but the protocol entities which are used in MSP. The maximum goodput that is reached in the experiments is less than 5% of the (theoretical) WLAN maximum throughput. Measurements performed at multiple locations pointed out that the MSP goodput can differ more than 5%.

Contents

List of Figures

List of Tables

Preface

This thesis is the result of my Master of Science assignment that I have performed for my study Telematics at the University of Twente. This assignment has been carried out from September 2005 to November 2006 as part of the AWARENESS project at the Architecture and Services of Network Applications chair. I have done research in the performance evaluation area, which is a very interesting and challenging area according to me.

I would like to thank a number of people who have given me the opportunity to complete this thesis. First, I would like to thank my supervisors Richard Bults, Ing Widya and Bert-Jan van Beijnum for their feedback, suggestions and continued support during my assignment. I would also like to thank Aart van Halteren, Pravin Pawar and Kate Wac for their support, and the students at the Lab for the pleasant working environment. Furthermore, I would like to thank my friends and student colleagues who have given me their view on my work. And last but not least, I would like to thank my mother who has made it possible for me to complete my study.

Enschede, November 17, 2006

Ger Inberg

Chapter 1

Introduction

This chapter presents an overall introduction to the research in this Master thesis. First, it will present the motivation behind the research which is followed by the objectives to be achieved. Next, the approach how to achieve these objectives is given. The chapter ends with an overview of the structure of this report.

The work in this thesis has been carried out as a part of the Freeband Awareness[1] project. This project aims at researching and designing an advanced context-aware network and service infrastructure. The AWARENESS project focuses on a service and network infrastructure that enables rapid and easy development of context-aware and pro-active applications in a secure and privacy-conscious manner.This network and service infrastructure is validated through prototyping intelligent (context-aware) applications.

1.1 Motivation

The ubiquitous availability of communication infrastructures (e.g. WiFi, GPRS, UMTS) is one of the main driving forces behind the popularity of mobile applications. Traditional mobile applications are un-aware of their context; i.e. are not able to automatically adjust their functionality and/or performance based on the location of the end-user, but next generation mobile applications will be context aware.

Advances in mobile device capabilities and communication technologies have enabled new applications by which services are deployed on the mobile device which are available to users on the Internet. Nomadic mobile services are characterized by the lim-

[1](A Dutch collaborative project on context AWARE mobile NEtworks and ServiceS

itations of their mobile devices and their wireless communication technology. The Mobile Service Platform (MSP) has been developed to deal with these issues. MSP uses the Jini Surrogate architecture to support the nomadic mobile service provisioning paradigm by hiding the limitations of the mobile devices and their wireless communication technology. In this thesis the focus is on the performance between the mobile service provider (at a mobile host) and it's Surrogate (at a fixed computer called the surrogate host)

Applications need a certain performance of their underlying transport system to function correctly. For applications like Mobihealth [22], which takes care of remote health-care monitoring of patients, one can imagine it is of vital importance that the data from the patient has to be available on time for e.g. the hospital. The (data) transport system (i.e. communication infrastructure) used can be a bottleneck in the performance of such a distributed application. To narrow the performance measurements, I decided only to focus on streaming applications. An example of such an application is a mobile health application, running on a mobile device worn by patient, that transmits patient data to a healthcare center. Because for streaming applications only the uplink[2] behaviour is of importance, this thesis will only focus on the uplink behaviour.

The MSP transport system is a so called hybrid transport system, because it consists of a mobile part and a fixed part. This hybrid system can be a major factor in the MSP performance due to the incompatible characteristics of the wired and wireless parts. For example the TCP sliding window mechanism is based on a more or less constant RTT[3] which is mostly available in fixed networks but not in mobile networks and thus limiting the optimal functioning of this mechanism for the wireless network.

The performance a mobile device can obtain depends on a lot of factors, for example network capacity, number of users, signal strength and buildings in the environment. Therefore the location of a mobile host in a certain network can be an important factor in the performance of a transport system.

1.2 Objectives

The goal of this assignment is to perform location specific performance measurements for a Wireless Local Area Network (WLAN) as the mobile part of the MSP transport

[2]from mobile device to a host on the Internet

[3]Round Trip Time; the time it takes to send a packet to a remote host and receive a response

system. To analyze the WLAN performance, measurements are performed between a mobile service provider in the WLAN and the surrogate host in the fixed network. The measurements should give a good indication of specific Quality of Service (QoS) aspects of the transport system.

To measure certain performance metrics like one-way delay the clocks of the mobile service provider and the host in the fixed network should be synchronized, as accurate as possible. Besides, the mobile host should provide it's (geographic) location to the fixed host to relate it's location to the network performance. The assignment should result in an analysis of how the network location influences the WLAN performance and how this relates to the mobile service provider. To visualize this performance a map will be created which shows the relation between the mobile service provider it's location and it's network performance at that location in a WLAN environment. This map should provide valuable information for applications that make use of WLAN as a transport system.

1.3 Approach

In the last section, the objectives of this thesis are stated. In this section the tasks that have to be performed to reach these objectives are explained. These tasks can be summarized in the following phases:

- Study the architecture and design of the AWARENESS service and network infrastructure in order to get a good understanding in which environment the performance measurements will be performed.
- Study time synchronization and location determination protocols and services and conclude which of these are suited for this project.
- Research which performance methodology can be used for a good evaluation of the performance.
- Design and implement a software framework which is able to perform the location specific performance measurements for the WLAN transport system. The framework should also include the time synchronization and location determination module.
- Execute the measurements, analyze the results, formulate general conclusions and suggest recommendations for future work on this subject.

1.4 Thesis structure

In chapter two the state of the art of current technologies in time synchronization and location determination is discussed. This chapter ends with a conclusions of. Chapter three explains the AWARENESS service and network infrastructure which is used as the framework to execute the performance measurements. Chapter four deals with the actual performance methodology while chapter five explains the design and implementation of the software framework. The evaluation from the measurements are discussed in the chapter six, which is followed by the conclusions and recommendations for future work in chapter seven. A list of acronyms, a glossary, a reference list and the appendices follow chapter seven.

Chapter 2

State of the art

In this chapter technologies in the area of time synchronization and location determination are discussed. As explained in chapter 1, time synchronization is needed for accurate performance measurements between the mobile host and the fixed host and location determination to relate these measurements to a certain location.

In section 2.1 different methods are discussed to retrieve the location of a device, this can be done using hardware, software or a combination of both. Then, in section 2.2 time synchronization protocols are discussed which is followed by the conclusion7.1 of which technologies of the discussed ones will be used for this project.

2.1 Positioning Service

In order to perform (location aware) performance measurements, the fixed host should be able to retrieve the (location) information from the mobile host. The positioning service is a service that makes the location, of the mobile host, available to other interested parties.

2.1.1 Positioning hardware resources

Several technologies exists that can be used to determine the location of a mobile host, such as GSM, GPS, WIFI and Bluetooth. Although only GPS was designed to determine one's location, the other hardware technologies can also be used for this, next to their original purpose. Each of these technologies has its advantages and disadvantages, with respect to accuracy, coverage, cost, reliability, etc. Next to these hardware technologies, software, that uses one or more of those (hardware) technologies, is needed to calculate the location of the mobile host. Place Lab[25] is a java library that can calculate the current location based on GSM, GPS, WiFi or Bluetooth beacons. Place Lab can be used as a standalone application as well as a building block for other applications, embedding the location information in higher level applications.

GPS

The Global Positioning Service (GPS) uses 24 satellites that are in 6 orbital planes above the earth. The system makes it possible to determine an entities geographic location on the earth with a GPS receiver. GPS determines this (current) location based on the differences in time signals that the various satellites send to the receiver.

The satellites are placed in such a way that from any point on the earth there will be at least four satellites above the horizon. Each satellite contains a computer, an atomic clock and a radio. The satellites continually broadcast their location and time. On the earth surface a GPS receiver receives this information and calculates its location (longitude and latitude) based on the information of at least three satellites. The third component of a location, the altitude, can be determined based on the information from four of more satellites.

GPS can achieve a base accuracy between 4 and 40 meters. The Wide Area Augmentation System (WAAS)[35], available since August 2000, increases the accuracy of GPS signals to within 2 meters for compatible receivers. GPS accuracy can be improved further, to about 1 cm over short distances, using techniques such as Differential GPS (DGPS)[4].

GPS coverage is in principle worldwide and continuous because the satellites can be reached from all over the world. However the GPS receiver needs to have a clear view of the sky to be able to reach the satellites, so unfortunately it cannot be used indoors and may suffer from distortion in a city with high buildings. The GPS service is owned and controlled by the US Department of Defense, but is available for general use around the world. [38]

GSM

Another positioning method uses the GSM network, the widely used mobile phone network. The GSM network consists of a large number of cells. Each cell has a unique ID, which can be used to identify the Base Transceiver Station (BTS) that the host is communicating with. The coordinates of each BTS are known in advance. When the user knows the ID of the BTS that supports GSM communication, the location of the user is also available. The accuracy of this method depends on the size of the cell. A GSM cell can measure between some 100m to 35km depending on the environment (e.g. buildings, open space, mountains) but also expected traffic. [28]

Another positioning method based on GSM cells uses the Timing Advance (TA) measure to estimate the Time of Arrival (TOA = propagation delay). The TA is a quantized measure of the transit time between the mobile host and the base transceiver station. When TA's from at least three base transceiver stations have been acquired the location of the mobile host can be determined (triangulated). The accuracy of this method also depends on the size of the cells. Techniques based on this method can achieve an accuracy of 150 meters.

The big advantage of using GSM networks for positioning is that it can be used both indoors and outdoors. Coverage is relatively good, but not as good as GPS. In densely populated areas the coverage is very good, but less populated areas may not even be covered completely. [9]

WiFi

Next to GPS (accurate but only outdoors) and GSM (less accurate but indoor and outdoor) other technologies exist that can be used for location determination, such as WIFI access points (AP's). This technology can be used in a relatively small area, such as within buildings or on a campus.

IEEE 802.11 is the international standard for the Wireless Fidelity (WiFi or WLAN) technology. WiFi can be used both indoors and outdoors. WiFi works also with the cell-based idea of GSM, but the user is now communicating with access points. Mobile hosts can determine their current location based on the ID of the access point they are communicating with. Accuracy depends on the size of the WiFi cells, which can have a diameter up to 100 meters. When a user can detect more than one access point, it can calculate its location more accurate by comparing the signal strengths of the different access points. [16]

Bluetooth

Bluetooth is a technology that enables (short range) wireless communication between hosts in so-called ad hoc piconets. It may be used as a personal area networks (PAN) with very limited coverage and without the need for a pre-existing infrastructure. Although Bluetooth is designed for wireless communication between hosts, it can also be used for location determination in small areas. Bluetooth hosts can communicate with other Bluetooth hosts that are in a range of 10 meters, with special transceivers even up to 100 meters.

Positioning using Bluetooth uses the knowledge of the locations of some fixed Bluetooth hosts or beacons. When the mobile host detects the fixed hosts the location of the mobile host can be determined. Because Bluetooth cells are relatively small, the accuracy is rather high. [28]

Sensors

Several technologies have been especially developed for positioning, mostly using ultrasonic, infrared or ultra-wideband radio. Most of these technologies are based on the location of fixed sensors that are located in a building or on a campus. Based on the location of the sensor, the signal-strength or the round trip times of signals of these sensors the location of a user can be calculated. The primary focus of these systems is to optimize accuracy, which can be in the 5-50 centimeter range. However the coverage of these systems is very limited. Another disadvantage of the infrared sensor systems is that there must be a direct line of sight between the sender and receiver. [16]

2.1.2 Place Lab application

Place Lab is a Java environment that provides libraries for location determination. One of the Place Lab design criteria is that it should work in all places where people spend their time (e.g. indoor as well as outdoor); this means Place Lab should have a good coverage of radio beacons. Next to this, it shouldn't be expensive for users and application developers to use the system. In Place Lab, the precision of the location estimates, though important, is secondary to coverage and privacy. However experiments have been done to characterize the accuracy of Place Lab as it relates to the types and densities of beacons in the environment.[25, 16]

The Place Lab Architecture

The whole Place Lab system consists of three key elements: Radio beacons in the environment, databases that hold information about beacon locations and the Place Lab clients that use this data to estimate their own location.

Place Lab works by listening for the transmission of beacons such as WiFi access points, fixed Bluetooth devices and GSM cell towers. The protocols of these beacons assign unique or semi-unique ID's to these beacons, which simplifies the process of calculating the client's location. The amount and type of beacons in the client environment determines the coverage and accuracy, see section 2.1.1 for an overview of this.

In order to estimate a client's location, the beacon locations should be available for Place Lab, otherwise the location cannot be determined. In Place Lab, the beacon database stores the beacon location information relative to client devices. There can be multiple beacon databases (either public or private) and it's not specified how clients authenticate with a database and how many databases a client should use data from. Many of these databases come from institutions, universities and companies who own a large number of these beacons. Mostly these databases are quite accurate.

Another source for Place Lab databases are produced by the war-driving community. War-driving is simply driving around with a mobile computer equipped with a GPS device and a radio signal receiver in order to collect a trace of network availability. Each trace is a time-coded sequence of the latitude and longitude of the record that was taken, added with the radio signal receiver sources and their signal strengths that could be measured at that time. A disadvantage of war-driving is that these databases contain only estimates of the beacons' locations. This comes from the fact that GPS is used to obtain the current location, and GPS has a limited accuracy.

Place Lab clients use live radio signal observations in combination with stored beacon locations to calculate their location. The client software is separated in three parts to make the software both portable and extensible, namely spotters, mappers and trackers.

Spotters are responsible for the observation of radio beacons. Clients typically use one spotter per radio protocol that is supported by the device. For example, a laptop probably has a WiFi spotter while a mobile phone has a GSM spotter, but both can have a Bluetooth spotter. Spotters share the information of nearby beacons (the ID's) with the other components in the system.

The mapper's job is to provide the location of the known beacons. The information provided always includes the latitude and the longitude of the beacon but can also contain other useful information such as the antenna altitude, the 'freshness' of the data or the power of the transmitter. The mapper can obtain his data directly from a database, or from a cached portion of a database. The size of the beacon-area, that is stored in the cache, depends on the capacity of the device.

The tracker is the component that uses the streams of the spotter and the mapper to produce estimates of the client location. The tracker includes the knowledge of how various signals propagate and how propagation relates to distance, the physical environment and location. Trackers can also use extra data, next to the data from the spotter and mapper, like road maps to produce more accurate estimates.

Place Lab Integration

Place Lab supports six ways of communicating location information to an application:

- Direct Linking: Applications may link against the Place Lab Java library and invoke a single method to start the location tracking service.
- Embedding Place Lab: Applications may embed a Spotter, Mapper and Tracker to make estimates of the device's location.
- Daemon: For lighter-weight interactions, Place Lab can be run in daemon mode and applications can query Place Lab via loopback HTTP. This HTTP interface allows programs written in most languages and styles to use Place Lab.
- Web Proxy: Place Lab supports location-enhanced web services by augmenting outgoing HTTP requests with extension headers that denote the user's location. By setting their web browser to use the Place Lab daemon's web proxy (in the same way one uses a corporate firewall's proxy), web services that understand HTTP headers can provide location-based service to the user.

- JSR 0179: To support existing Java location-based applications Place Lab supports the JSR 0179 Java location API.
- NMEA 0183: Place Lab provides a virtual serial-port interface that can mimic an external GPS unit by emitting NMEA 0183 navigation sentences in the same format generated by real GPS hardware.

For the development of services for mobile devices (e.g. mobile phones, PDAs) the first two ways look most suitable, because the overhead is the least. Testing (on a PDA) proved that embedding Place Lab in the application is the fastest and most light-weight solution. The results are shown below in table 2.1 it clearly shows that embedding Place Lab in the application's code is faster.[32]

	First estimate[1]	Subsequent estimates
Direct Linking	320ms	115 ms
Embedding Place Lab	60 ms	6ms

Table 2.1: Place Lab testing results

Place Lab accuracy

To investigate the relationship between beacon density and accuracy, research has been done to compute the median accuracy achievable by Place Lab using 802.11. Figure 2.1 shows a graph comparing the accuracy of Place Lab as it relates to the number of unique beacons the client saw during the previous 10 second window. From this figure we can conclude that if 802.11 density is high enough for clients to see at least 3 distinct beacons during a 10 seconds window, the density is sufficient for Place Lab to achieve its "peak" median accuracy of 15-20 meters. [16]

So, including Place Lab into our architecture means that the accuracy of our map is also limited to 15-20 meters.

[1]This includes the loading of drivers

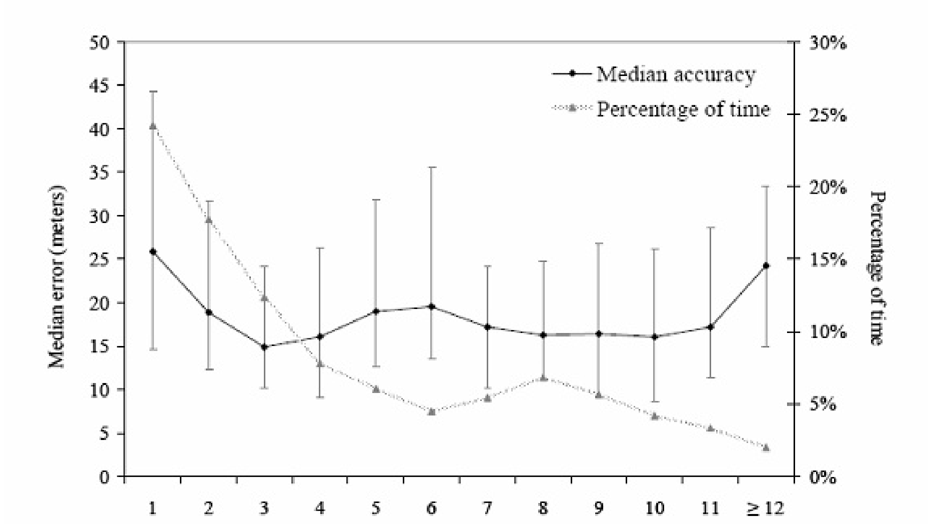

Figure 2.1: This graph shows the effect AP density on Place Lab's accuracy. The accuracy line is drawn through the medians, while the error bars represent the 1st and 3rd quartile readings (50% of the readings fall between the error bars) The second line shows how often we saw the number of distinct AP's. [16]

2.2 Time Synchronization protocols

Time synchronization protocols providing a mechanism to synchronize the local clocks of computers in a network have been studied for a couple of decades.

The main protocol to synchronize (Internet) computers clocks is the Network Time Protocol (NTP)[21]. This protocol is a client-server protocol, as most are, were clients can synchronize their clocks with a time server. These protocols all have some basic common features: a simple connectionless messaging protocol, exchange of clock information between clients and one or more servers, methods for mitigating the effects of non-determinism in message delivery and processing and an algorithm on the client for updating the local clock based on information received from a server. There are however also protocols that differ from this traditional approach of senders synchronizing with receivers; those protocols use broadcasting to synchronize a set of receivers with one another, for example GPS uses this technique.

In some systems, causality is more important than absolute time. Lamport's system focused on giving events a total order rather than measuring the time differences between them. [17] This system has emerged as a major influence in mobile sensor networks, because many of those applications only require relative time. For the performance measurements the clocks should be synchronized to each other (e.g.to measure one-way delay) but absolute time is not needed. In those cases synchronization to an absolute time source, like in NTP, is not needed. However those networks mostly need a better accuracy than in the traditional fixed networks because of their close coupling with the physical world and their energy constraints. For example, when sensors are used to estimate the velocity of an object, based on proximity detections, one can imagine that time synchronization in the order of milliseconds can already lead to an imprecise estimate of the object's velocity.

2.2.1 DCF-77

DCF-77 is a example of a radio-clock system. A radio clock is a clock that is synchronized by a time code bit stream transmitted by a radio transmitter connected to a time standard such as an atomic clock. A radio (controlled) clock consists of an antenna for intercepting the RF time code signal, a receiving circuit to convert the time code RF signal into a digital time code and a controller circuit to decode the time code bit streams and to drive and output circuit which can be LCD in case of digital clocks or stepping motors in case of analog clocks. [38] DCF-77 is a longwave time signal, which is transmitted from Mainflingen, close to Frankfurt in Germany. The 77,5kHz carrier signal is generated from local atomic clocks that are linked with the German master clocks in Braunschweig. The transmitter can be received in large parts of Europe, as far as 2000km from Frankfurt. Compared to satellites, the radio clocks have an advantage that the signal can penetrate into buildings, whereas satellites require a clear sky view. A disadvantage of DCF-77 is its accuracy, which is in the range of 5-25msec. [26]

2.2.2 Network Time Protocol

The Network Time Protocol is an Internet standard that enables clients to synchronize their computer's clock with time servers. NTP has many advantages compared to other protocols including its scalability, robustness to failures and sabotage and ubiquitous deployment. The main disadvantage is that the NTP algorithm requires a symmetrical link, which is not always available in (especially mobile) networks; this is discussed below after an explanation of the synchronization mechanism. The time servers are synchronized by external time sources mostly using the GPS, see section 2.1.1, a constellation of satellites operated by the U.S. Department of Defense. [15] Commercial GPS receivers can achieve an accuracy of better than 200nsec relative to Universal Time Coordinated (UTC). [11] However GPS requires a clear sky view, which is unavailable in many application areas for example inside buildings. In addition, receivers can require several minutes of settling time and may need too much power for small nodes in a network.

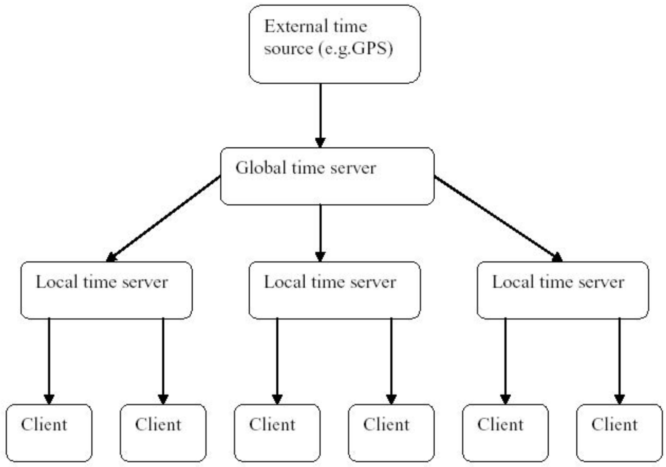

Figure 2.2: The NTP system, the arrows indicate the synchronization direction.

In figure 2.2 the structure of the NTP system is given. Clients can synchronize their clock to a local time server (e.g.ntp.utwente.nl), that gets its time from a global time server. The global time server gets their time via a very accurate external clock, for example via GPS. In figure 2.2, when moving down in the graph, accuracy will degrade because mostly the connections are getting less stable (because of wireless links). The NTP clients can synchronize their clocks to NTP time servers with accuracy in the order of milliseconds. The protocol is based on statistical analysis of the round trip time between the client and the server.
It works in the following way: Node A (client) sends a synchronization pulse packet (including T1) at time T1 according to its local clock. Node B (server)receives the packet at time T2 according to its local clock. Node B transmits an acknowledgement at time T3 including time values T1, T2 and T3. Node A calculates the clock drift and propagation delay defined below, and synchronizes itself to node B.

In the formulas below delta is the relative clock offset and D is the propagation delay.

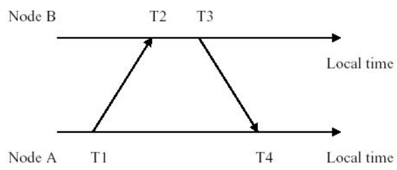

Figure 2.3: Synchronization mechanism in NTP

$$\Delta = \frac{(T2 - T1) - (T4 - T3)}{2}, D = \frac{(T2 - T1) + (T4 - T3)}{2} \tag{2.1}$$

As mentioned, this method requires symmetric round trip times for a correct working of the protocol. However non-determinism of random events can occur and this leads to asymmetric delays, which contribute directly to synchronization errors. To better understand the source of these errors, it is useful to decompose the source of a message's latency. The latency can be characterized by having four distinct components:

- Send Time: the time spent at the sender to construct the message. This includes kernel protocol processing and variable delays introduced by the operating system, e.g. context switches and system call overhead incurred by the synchronization application. Send time also accounts for the time required to transfer the message from the host to its network interface.

- Access Time: delay incurred waiting for access to the transmit channel. This is specific to the MAC[2] in use. Contention-based MAC's (e.g., Ethernet) must wait for the channel to be clear before transmitting, and retransmit in the case of a collision. Wireless RTS/CTS[3] schemes such as those in 802.11 networks require an exchange of control packets before data can be transmitted. TDMA[4] channels require the sender to wait for its slot before transmitting.

- Propagation Time: the time needed for the message to transit from sender to receiver once it has left the sender. When the sender and receiver share access to the same physical media (e.g., neighbors in an ad-hoc wireless network, or a LAN), this time is very small as it is simply the physical propagation time of the message traversing

[2]Medium Access Protocol
[3]RTS=Request To Send, CTS=Clear To Send
[4]Time Division Multiple Access

the media. In contrast, propagation time dominates the delay in wide-area networks, where it includes the queuing and switching delay at each router as the message transits through the network.

- Receive Time: processing required for the receiver's network interface to receive the message from the channel and notify the host of its arrival. This is typically the time required for the network interface to generate a message reception signal. If the arrival time is time stamped at a low enough level in the host's operating system (e.g., inside of the network driver's interrupt handler), the receive time does not include the overhead of system calls, context switches, or even the transfer of the message from the network interface to the host. Existing time synchronization algorithms vary primarily in their methods for estimating and correcting for these sources of error. When performing many request/response experiments it is likely that at least one experiment will not encounter random delays, this experiment is then used for calculating the correct time.

A different approach to reduce the clock synchronization error is to exploit the broadcast channel. The idea behind this approach is that a message that is broadcast via the physical channel will arrive at a set of receivers with very little variability in its delay. Although the send and access time may be unknown, the property of broadcast is that for a particular message these quantities are the same for all receivers. [12] For a discussion of protocols that use this property see the next section.

2.2.3 Reference-Broadcast Synchronization

Reference Broadcast Synchronization (RBS) uses broadcast communication to allow the receivers of a synchronization message to synchronize their clocks with each other. By removing several components of non-determinism (jitter) from traditional time synchronization (where receivers try to synchronize with the sender), Elson et al. achieve better accuracy than previous methods, e.g., NTP. [12] RBS requires a physical broadcast channel and cannot be used in networks that employ point-to-point. Nodes can synchronize time (1) relative to each other or (2) relative to an external timescale, e.g., UTC. Many sensor applications require only relative time.

The most significant limitation of RBS is that it requires a network with a physical broadcast channel. It can not be used, for example, in networks that employ point-to-point links. However, it is applicable for a wide range of applications in both wired and wireless networks where a broadcast domain exists and higher-precision or lower energy synchronization is required then what NTP can typically provide.

By synchronizing a set of receivers, the send and access time are removed from the critical path because only the receive time is important. The simplest form of RBS is broadcast of a single pulse to 2 receivers, allowing them to estimate their relative phase offsets. That is:

1. A transmitter broadcasts a reference packet to two receivers (i and j).
2. Each receiver records the time that the reference was received, according to its local clock.
3. The receivers exchange their observations, calculate the offset of their recorded times and change their clock to half of this offset (2 receivers).

Based on this single broadcast, the receivers have sufficient information to form a local timescale. [12]

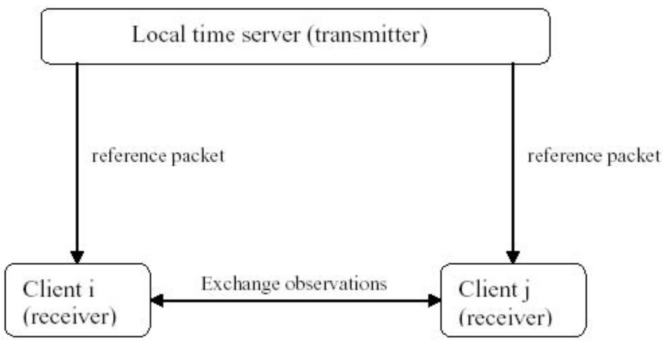

Figure 2.4: The RBS system with 2 receivers

2.2.4 GPS synchronization over Bluetooth

GPS can achieve an accuracy of better than 200nsec to UTC. The main limitation of GPS is that it requires a clear sky view. However there are devices on the market in which the GPS time can be communicated via some technology (e.g. Bluetooth) to other devices. When synchronizing the mobile device to the GPS receiver via bluetooth the accuracy degrades. Only in the case that the time needed to send a packet from the mobile device to the GPS receiver takes the same time as vice versa the accuracy doesn't degrade. Unfortunately this is normally not the case. The accuracy is in the worst case one half of the Round Trip Time (RTT); this is the time between sending a packet and receiving a packet on the mobile device side. In figure 1.13 an overview is given of this method. The GPS receiver gets his (precise) time from GPS satellites and via Bluetooth this is passed on to the client.

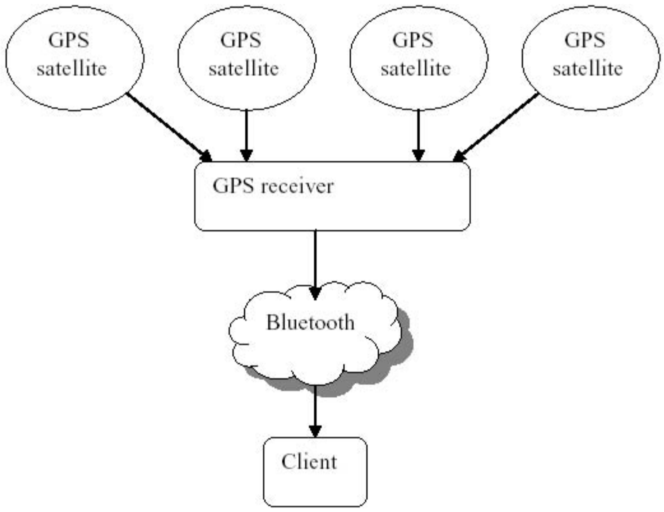

Figure 2.5: GPS synchronization over Bluetooth

2.2.5 In-band vs. Out-of-band synchronization

In [36], a performance evaluation study over a wireless network is performed. In this evaluation study 2 methods for time synchronization are distinguished, which are defined below.

Definition In-band time synchronization: This is the exchange of time synchronization information on the same channel than that of the user information.

Definition Out-of-band time synchronization: This is the exchange of time synchronization information on a channel that is dedicated for that purpose and separate from the user information channels.

Performing measurements in a network while using the channel at the same time for time synchronization can lead to corrupted measurements results in case of an asymmetrical and heavily loaded communication channel. [36] This is due to the fact that most time protocols (such as NTP) take into account the one-way delay between the server and the client (based on the RTT) when generating a reference time stamp for a client.

2.2.6 Clock drift

Two physical clocks tend to drift away from each other; this means that the difference between their clock values increases unbounded as time increases. System clock drift occurs due to external factors (e.g. temperature) on the clock behaviour. The clock drift has a negative impact on the quality of the obtained measurement results; the accuracy of these results will decrease. Therefore it is important to discover this clock drift. To keep the clock values of two devices between at certain value, time synchronization has to be performed in certain intervals. When the clock drift is known, these synchronization intervals can be calculated.

2.2.7 System clock resolution

Different operating systems have different clock resolutions; this means that these clocks take different times in response to an input event. These values influence the measurement data, for example when the delay of a message is measured, the time must be recorded (clock request) when a packet is send and received.

Time differences measured must be "scaled" according to these resolutions. The values for different operating systems are given in the table below.

Operating system	System clock resolution [ms]
Linux(2.2,x86)	1
Mac OS X	1
Windows 2000, XP	10
Windows 98	60
Solaris (2.8, i386)	1
Solaris (2.7, sun4u)	1

Table 2.2: System clock resolution for different Operating systems [18]

2.3 Conclusion

In order to realize the objective of location aware performance measurements, the clocks of the mobile host and the fixed host should be synchronized to each other (absolute time synchronization not needed) and the mobile host should provide it's location to the fixed host.

For time synchronization, the out-of-band method is preferred, because with this method the time synchronization data will not interfere with the measurement data. The Reference Broadcast Synchronization protocol cannot be used for this assignment, because the fixed host can be anywhere on the Internet and thus broadcast is not available. NTP is not suited because it requires a symmetrical round trip time which is mostly not available in wireless networks.

Because a GPS receiver can provide a very accurate time and also provides the location this method is the preferred option for time and location synchronization is is therefore used in this project.

Chapter 3

Awareness system

As briefly stated in chapter 1 this thesis is performed as part of the Freeband AWARENESS project, a Dutch collaborative project on context AWARE mobile NEtworks and ServiceS. Eight partners from industry and academia work together in this project: Lucent Technologies, University of Twente, Roessingh R&D, Telematica Instituut, Twente Institute for Wireless and Mobile Communications, Ericsson, Yucat and Twente Medical Systems International.

The Mobile Service Platform (MSP), which supports nomadic mobile services (see section 1.1) is part of the AWARENESS system. MSP uses the Jini Surrogate architecture to make these services available for other users. This architecture specifies an interconnect protocol that takes care of the communication between the mobile service and its surrogate object in the fixed network.

In this chapter the AWARENESS system is explained and how MSP is related to this system. Section 3.1 explains the AWARENESS goal and discusses it's system architecture and functional overview, whereas section 3.2.3 zooms into the different MSP components. The chapter ends with a discussion of the interconnect protocol in section 3.3

3.1 System overview

This section discusses the AWARENESS architecture and a function overview after an introduction of the goal of the AWARENESS project.

3.1.1 AWARENESS goal

In the AWARENESS project vision a human (user) is always and everywhere surrounded by a networking environment (ubiquitous). This environment is able to detect, or sometimes even determine the identity of, the user and the upcoming context information that is (or might become) relevant to provide a service (service provisioning). This enables applications and services to become attentive and make appropriate selections in resource-constrained environments, which means that they will be able to react on changes in the end-users context, such as available resources (e.g. quality of network) or a user environment (e.g.location). The AWARENESS project focuses on an infrastructure for context-awareness that enables (pro-active) responsiveness of applications, and trialled this through prototyping with mobile health applications. Pro-active responsiveness means that applications can timely react on (predicted) changes in the context of the end-user. Mobile health applications make it possible to monitor patients, who might get in a dangerous situation, and even to treat patients at a distance.

The AWARENESS project aims to support the nomadic mobile service provisioning paradigm. In this paradigm, services are offered from mobile devices (e.g. PDA's and (smart) mobile phones) to other mobile users, but also to users connected to the Internet.

Nomadic mobile service provisioning is a new way to offer services. Traditionally, most of the functionality of a mobile service is deployed on a computing system that is connected to a fixed network. This computing system has sufficient processing and communication capacity and a light-weight client on the mobile device gives the end-user the ability to invoke the service. However, in nomadic mobile service provisioning the functionality of a mobile service is deployed on a mobile device and may be invoked from a fixed network. This deployment of services on a mobile device introduces some challenges: the processing and storage capacity of the mobile device is limited, how and when the device connects to a network is unpredictable and the quality of the network connection may fluctuate a lot.

3.1.2 AWARENESS architecture

The AWARENESS network and service architecture consists of three layers which are shown in 3.1: the network infrastructure layer, service infrastructure layer and mobile applications layer. The network infrastructure layer and the service infrastructure layer together form the AWARENESS infrastructure.

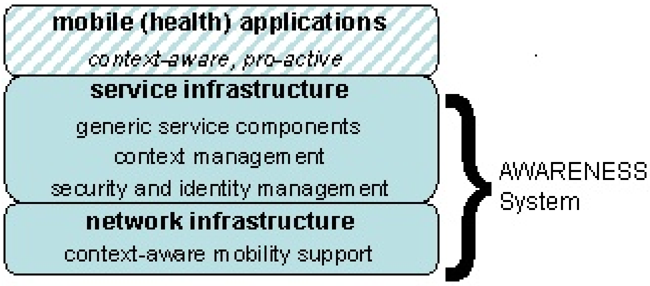

Figure 3.1: Awareness system architecture [2]

The network infrastructure

The network infrastructure provides context-aware mobility support in an environment where different networks co-exist. Context-aware mobility is supported in two ways: 1) The network infrastructure takes the context of the user into account when controlling the (network) connectivity that is provided to this user, for instance security settings and network selection 2) The network infrastructure itself is a source of context information, for instance location information and available bandwidth (i.e.transport capacity)

The service infrastructure

The service infrastructure consists of generic service components that support development of pro-active applications. The support functionalities include context management, intelligent context information processing, federated identity management, 3rd party access control, mobility management, service discovery, privacy enforcement and security mechanisms.

Mobile (health) applications

The added value of the AWARENESS system will be trialled in the healthcare domain. The AWARENESS project develops a mobile health services platform and mobile health applications. The health applications will support tele-treatment of patients with chronic pain, tele-monitoring of epileptic seizures and uncontrolled movements in spasticity.

Part of the mobile health services platform is a health Body Area Network (BAN) that collects sensors data and sends this data to health care centers and/or healthcare professionals. [2]

3.1.3 Functional overview

A functional overview of the AWARENESS system is presented in figure 3.2.

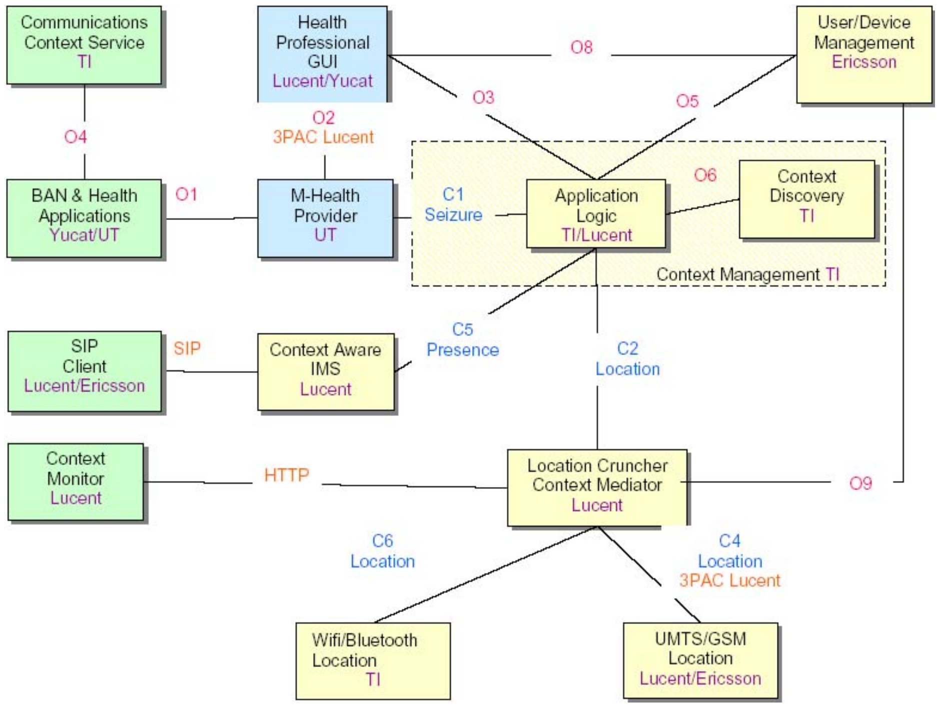

Figure 3.2: Awareness functional overview

The lines represent the interactions between the functional blocks, the name and or number specify the type of interaction. An explanation of these interactions (including parameters) supporting a health application can be found in [10].

The mobile health applications are located at the left side of the picture and the network and service infrastructure are located at the right side of the picture. The location determination, which is a part of this assignment, can also be seen in this figure. The functional block "Location Cruncher, Context Mediator" gets the location from terminals in a UMTS/GSM or WiFi/Bluetooth network and stores this information. On request, it transfers terminal location information, or the location of terminals that are nearby a given location to the "Context Management" block for further processing or to the "Context Monitor" to represent this information on a mobile application.

The blocks BAN & Health applications and M-health Provider with the interaction O1 are of importance to my assignment, because MSP is located in these blocks. BAN & Health applications are applications running on mobile hosts. An example application is an Health BAN that can predict an upcoming epileptic seizure for epilepsy patients [10]. The BAN collects data about some of the patient's vital signs and transmits this over the (partly mobile) network to a M-health center (M-health Provider), where the data is further processed. The interaction O1 implements the transport of BAN measurement signals from the terminal to the M-Health Provider domain. In the next section the blocks BAN & Health applications and M-health Provider are decomposed to retrieve how MSP is embedded in these blocks.

3.2 Mobile Service Platform and Jini service

As briefly mentioned in chapter 1 MSP uses the Jini surrogate architecture. This architecture is discussed in the next section, while MSP is explained in section 3.2.3

3.2.1 Jini architecture

Jini[30] is a de facto standard based on the idea of federating groups of users and the resources required by those users. The focus of Jini is to make the network a more dynamic entity, that better reflects the (dynamic) nature of service users, by enabling the ability to add and delete services flexibly. Jini uses the Service Oriented Architecture (SOA), which is a design for linking computational resources (principally, applications and data) on demand to achieve the desired results for service consumers (end users or other services). Jini tries to address the problems in current SOA implementations, which include assumptions about the network quality and reliability and assumptions about the static nature of a network.

Jini services do not need the network to be reliable; Jini services are able to repair themselves without the need for an administrator to fix things. Jini uses the concept of leasing. Leasing means that a service is registered for a certain time, after this time the lease has to be renewed or the service is removed from the registry. A Jini service can handle changes in the network, for example there is no need for an administrator to change

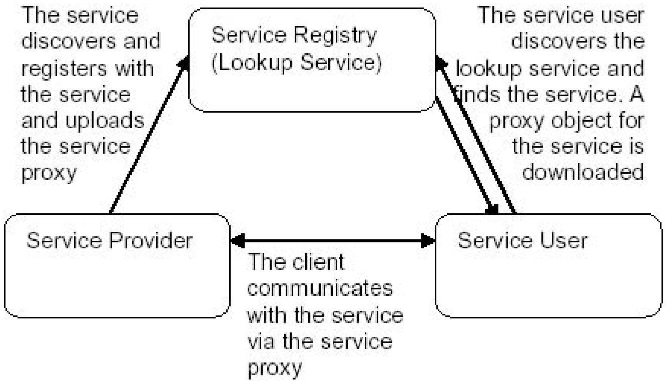

Figure 3.3: Jini architecture

the URL in software or properties file, the Jini service itself responds to such changes. Communication with Jini services occurs through service proxies that are downloaded when the service is discovered and invoked. Due to this feature the administrator does not need to install code, which is necessary in traditional architectures.

A Jini architecture consists, just like other SOA implementations, of a service provider, a service user and a service registry (which is called the lookup service). The Jini service provider discovers the lookup service and registers its service with this lookup service. A service proxy is uploaded to the lookup service. The lookup service keeps track of all Jini Services in the Jini network. On request, the service user discovers a lookup service and requests a certain type of service. If this type of service is registered with the lookup service, a service proxy will be returned to the service user. The service user can use this service proxy to invoke the service. The underlying system that is used most of the time to allow service users to communicate with a Jini service provider is RMI (Remote Method Invocation). Though other underlying systems may also be used. [5, 13, 23]

3.2.2 Jini Surrogate Architecture

For a service to join a Jini network (a network of Jini technology-enabled services), it must satisfy several critical requirements: first it must be able to participate in the Jini discovery and join protocols[1]. Furthermore, it must be able to export classes written in Java so that the service is available for downloading by a service user. These requirements

[1] A sequence of steps to register themselves with the lookup service

are easy to meet for most services, but it may be a problem for some devices with limited computational resources or limited network connectivity.

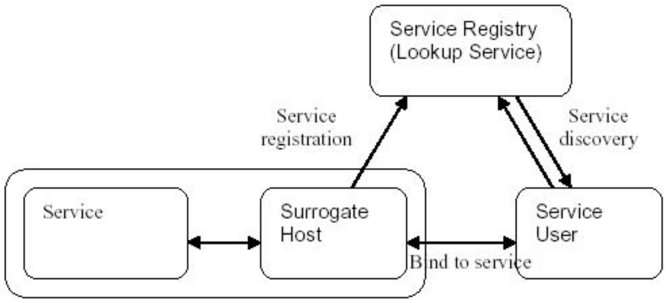

Figure 3.4: Jini Surrogate Architecture

The Jini Surrogate Architecture addresses these problems by defining a means by which these devices, with the aid of a third party, can participate in a Jini network while still maintaining the plug-and-work model of Jini network technology. [30]

The Jini Surrogate Architecture provides a place (i.e. surrogate host) where processing power can be placed for a surrogate object (surrogate) that acts on behalf of an attached mobile service. Furthermore the Jini Surrogate Architecture specifies an interconnect protocol, a communication gateway for the physical and logical connection between the mobile service and the surrogate. The Jini Surrogate Architecture allows devices that are not directly connected to a Jini network or are not able to join a Jini network directly, to join a Jini network. A surrogate (written in Java) that is able to access the Jini network and has access to the Jini technology infrastructure, is provided for a nomadic mobile service. This surrogate represents a service on a mobile device that is not Jini technology-enabled in the Jini network. A surrogate host is an environment for the hosting of surrogates. Surrogates can be loaded (e.g. from a web server) into a surrogate host and executed. A surrogate host provides a place where surrogates can participate fully in a Jini network, thereby enabling the service on the mobile device (represented by the Surrogate) to participate fully in a Jini network. Surrogate hosts manage the life cycle of surrogates by loading, starting, stopping, and disposing of surrogates when necessary.

The connection between the nomadic mobile service and the surrogate is implemented by the interconnect protocol. This protocol is explained separately in section 3.3.

Figure 3.5: Interconnect protocol as as intermediate between mobile service and surrogate

3.2.3 Mobile Service Platform

Before explaining MSP, first the embedding of MSP in the AWARENESS system is described. In figure 3.6 this embedding is depicted. As can be seen in this figure, the surrogate host (including Surrogate) is located in between the sub-systems BAN & Health applications and the M-Health Provider. The MSP system is part of the sub-system BAN & Health applications and attaches to the Surrogate. The MSP system uses the MSP transport system to exchange Interconnect messages between the BAN & Health applications sub-system (i.e. the mobile service) and the Surrogate. The Jini system is part of the sub-system M-Health Provider and attaches at the other side of the Surrogate than MSP does. The MSP system is not only focussing on health applications but rather on generic mobile services. The goal of MSP is to support mobile services in resource constrained environments [1].

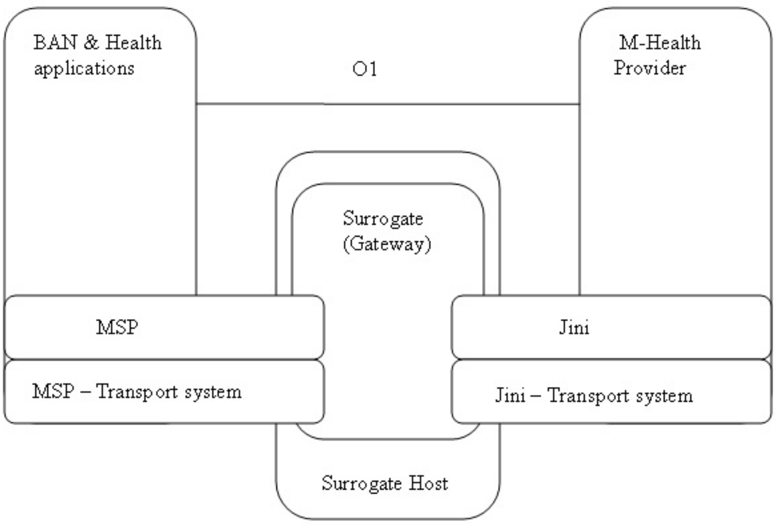

Figure 3.6: MSP located in AWARENESS system

In figure 3.7, the MSP & Jini service is given. The MSP domain is considered to be the mobile service, the Interconnect Protocol and part of the Surrogate. As can also be seen in this figure, the Surrogate is the gateway between the MSP domain and the Jini service domain. The Jini service domain consist of service users, the lookup service and part of the Surrogate. The function of Jini, amongst others, is to dynamically discover the mobile service delivered by an application and make this service available to service users.

The distinction between the MSP domain and the Jini Service domain is made in order to be more flexible; in the future other technologies can be used to make mobile services available to service users.

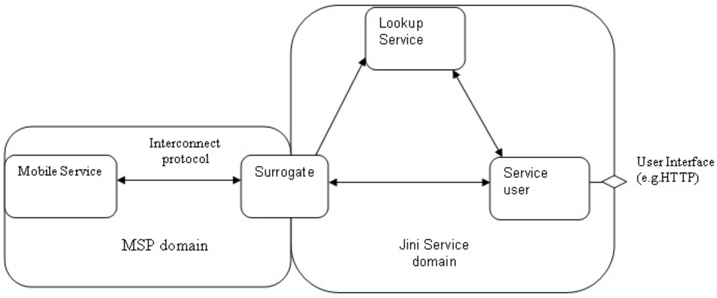

Figure 3.7: MSP I & Jini service

Components

The MSP and Jini service consists of the following parts:

1. Mobile Service
2. Surrogate
3. Jini Lookup Service
4. Service User
5. Transport system

1.Mobile service

The mobile service is offered by an application running on a mobile host. A mobile host can have multiple mobile services. The mobile service communicates via the Interconnect Protocol over respectively a mobile and fixed network with the Surrogate.

2.Surrogate

The Surrogate is located on the surrogate host and is a representation of a mobile service in the fixed network. The surrogate host is responsible for instantiation of surrogates representing a mobile service to the Jini service users. The mobile service registers itself to the surrogate host when starting up.

3.Lookup service

The lookup service is responsible for the registration, discovery and advertisement of the available services of a particular mobile host in the Jini service domain.

4.Jini service user

The Jini service user can search for a certain service at the lookup service. When this service is available, a service proxy is returned to the service user. The service user can use this proxy to invoke the mobile service. It offers this service to fixed network users (e.g. Internet users) by providing an application interface (e.g. HTTP)

5.Transport systems

The MSP transport system is the transport system between the mobile service and the Surrogate and consists of a mobile and a fixed part. The mobile service and the Surrogate communicate via the Interconnect Protocol which is discussed in section 3.3. The Jini transport system takes care of the communication between the Surrogate, the lookup service and the service user. It doesn't have an hybrid character, as in the MSP transport system, but is a fixed transport system.

3.3 Interconnect Protocol

The interface of the Interconnect Protocol is specified by Sun Microsystems. The protocol functions as a type of middleware layer which can be implemented on top of several forms of communication.

The features that should be supported by the Interconnect Protocol, according to are:[29]

- Device discovery

- Surrogate upload
- Keep-Alive

Device discovery

The purpose of the device discovery is to make the surrogate host aware of the device (mobile host) existence (including its services) and vice versa. The discovery mechanism must define a way in which a device, when attached to an interconnect, can either: announce its presence such that a surrogate host can detect it, or detect a surrogate host on that interconnect. When both hosts have discovered each other, the mobile host its services are represented in the the fixed network.

Surrogate upload

The surrogate host stores a Surrogate as a representation of the mobile service in the fixed network The mobile host can upload the Surrogate directly to the surrogate host or it can send the location from where the Surrogate can be downloaded.

In the University of Twente implementation [2], this works a bit different; the surrogate host returns a URL to the mobile host pointing to a specific surrogate object (Surrogate).

Keep-Alive

The keep-alive mechanism is used to inform the surrogate host if the mobile host is still active and connected. If the mobile host cannot confirm its online status for a certain time period (i.e. keep alive messages are not received by the Surrogate) the surrogate host will terminate the service and its corresponding surrogate.

In the University of Twente implementation, this works a bit different; the mobile service is directly communicating with the Surrogate, so the surrogate host is not an intermediate node in the communication process.

[2] The source code of this implementation can be found at http://janus.cs.utwente.nl:8000/twiki/bin/view/MSP/CVS

Protocol working

In the Interconnect Protocol messages are sent between the mobile service and the Surrogate in the body of HTTP messages[3] (in the HTTP request as well as in the HTTP response). The mobile service sends HTTP requests on a regular basis and it puts the messages for the Surrogate in the HTTP request body.

The Surrogate reads the messages from the HTTP request, processes this message and sends a response back, in the body of the HTTP response, to the mobile service. The mobile service sends HTTP requests (keep-alive request or other messages) on a regular basis, responses to these requests can be used to send messages back to the mobile service.

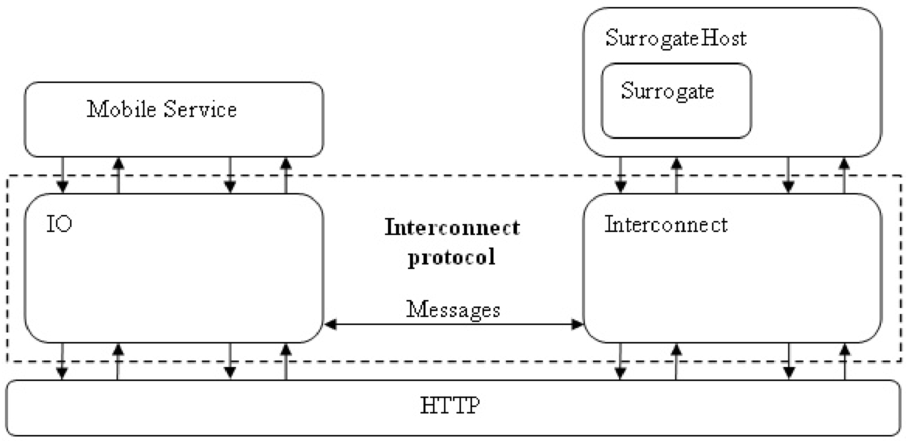

Figure 3.8: Interconnect Protocol [32]

Messages

The MSP messages package (nl.utwente.msp.messages) implements the communication part from the interconnect protocol specification by Sun Microsystems. [29] It contains the messages that can be sent between the mobile service and the Surrogate. The other modules of the MSP, IO and Interconnect, are described in [32]. The message package also contains functionality for encoding and decoding of these messages.

Three types of messages are available in the Interconnect Protocol: RequestMessage, ReplyMessage and OnewayMessage. The RequestMessage contains a request and can

[3] For an explanation why HTTP is used as underlying layer for the Interconnect Protocol see [32]

be initiated by the mobile service as well as by the Surrogate. The ReplyMessage contains the response to the RequestMessage and can also be send by the mobile service as well as the Surrogate.

Before the ReplyMessage can be send, the RequestMessage should be processed in order to send some relevant data back. Processing of such a message takes time, which may change from several milliseconds up to seconds depending on the method that has to be executed. Also, the communication takes time and especially the communication in the wireless part of the MSP transport system can introduce a significant delay.

The OnewayMessage does, as its name indicates, not require a response. This message may also be send by both the mobile service and the Surrogate. OnewayMessages were added to the protocol for efficiency reasons. In some cases it is simply not necessary to send data back to the sender. This is for example the case when the mobile service sends location updates to the surrogate. In these cases a OnewayMessage is useful.

Mobile Service request

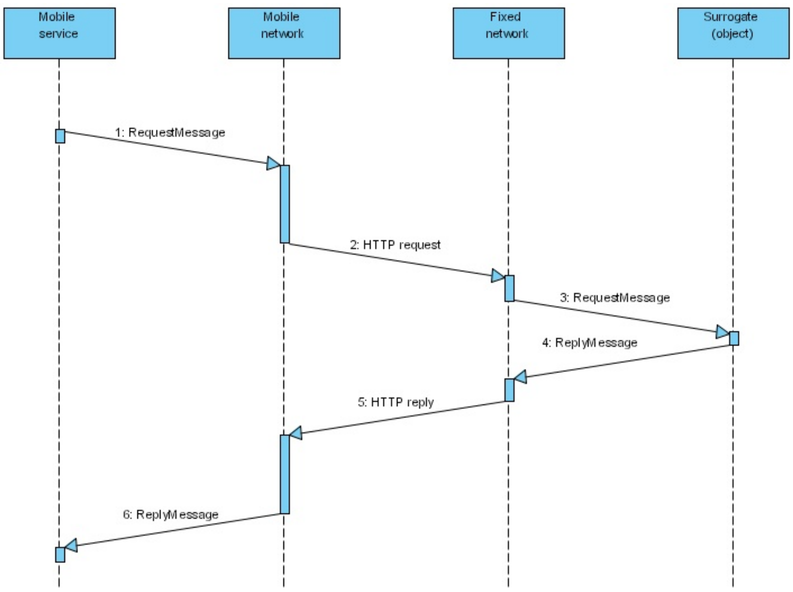

Figure 3.9: Mobile service request

In figure 3.9, the mobile service does a request to the Surrogate. The Interconnect protocol transfer is split into two parts to visualize that the message traverses through a mobile network a fixed network, which can make clearer the delays involved in the communication. The biggest part of the delay is expected to be located in the mobile network, therefore the blue thick line in the Interconnect Mobile network is drawn longer then for example the interconnect protocol fixed network. The RequestMessage is sent via a HTTP request to the Surrogate (which is located at the Surrogate Host). After processing this request, the Surrogate sends a ReplyMessage to the mobile service, using a HTTP reply.

Surrogate request

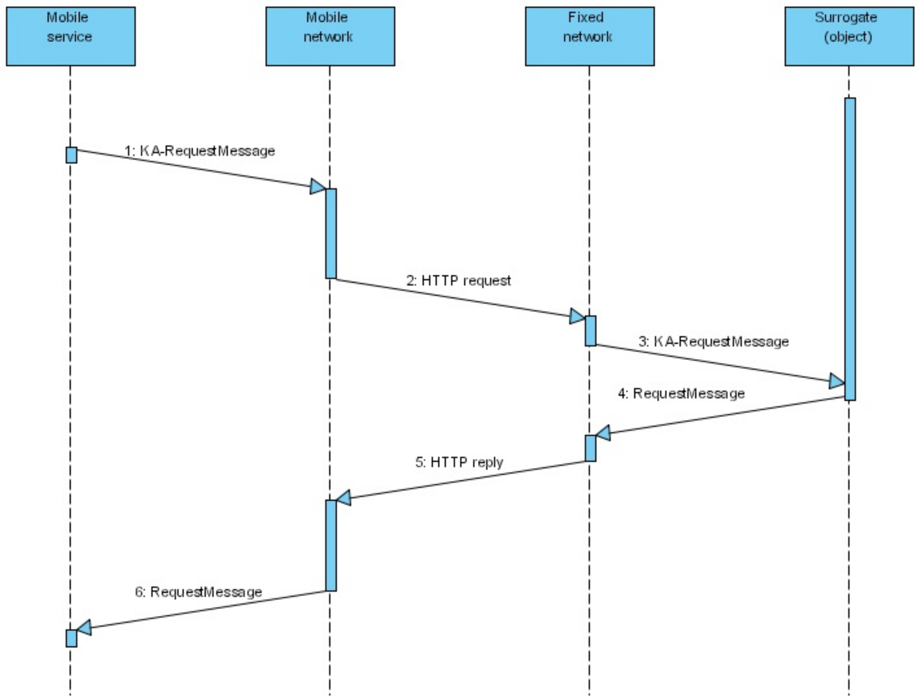

Figure 3.10: Surrogate does a request for a Mobile Service

In figure 3.10, the Surrogate does a request to the mobile service. For example this request can come from a service user (not shown) Before the request can be send all the way to the mobile service, first a keep-alive message (dotted line) should be received from the mobile host (blue thick line indicates the Surrogate is waiting for keep-alive message). The keep-alive mechanism is part of the Interconnect protocol specification and is used to indicate there is still a connection available to the mobile service. Due to the mobile part of the MSP transport system, which is mostly not as reliable as the fixed part, this cannot be taken for granted.

When the keep alive message is received, the waiting request can be forwarded (i.e. piggy-backed at the Keep Alive message) to the mobile service. The mobile service then sends a ReplyMessage after processing of the RequestMessage. Because the keep alive message is also a RequestMessage, the mobile service receives a ReplyMessage for confirmation.

OnewayMessage

In figure 3.10, the mobile service sends an OnewayMessage to the Surrogate. This can be useful when the mobile service wants to send an update of its location to its Surrogate; in this case no reply is needed from the Surrogate.

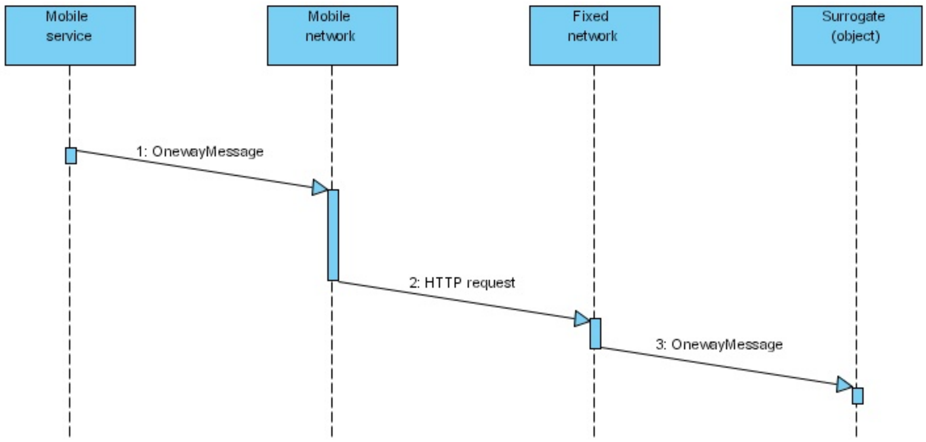

Figure 3.11: Mobile Service sends a OnewayMessage to the Surrogate

Message structure

The messages used in the Interconnect Protocol all extend from the class Message, resulting in a common structure for these messages. The message starts with a ServiceId, followed by an operationId, a sequenceId and ends with the body. The ServiceId is the Id of the mobile service that sends the message, or the Id of the mobile service the message is destined for when sending from the Surrogate to the mobile service. The operationId is a unique Id that is given to the operation of the mobile service or the Surrogate. Each operation (e.g.location request) at the mobile service or the Surrogate needs to have a unique Id, so one side can trigger a certain operation at the other side. The SequenceId identifies the order of the messages. The body of a message contains operation specific data. For example, if the mobile service sends a message with location updates, the body of this message will contain the location updates. [32] In figure 3.12 the structure of the message is visualized.

Figure 3.12: The structure of a message

Chapter 4

Performance methodology

4.1 Performance evaluation

Performance of a system can be measured in many different ways. Depending on the (user of an) application that needs a certain Quality of Service (QoS) of a system the right performance metrics[1] have to be chosen. For example, a streaming video service has other performance requirements from a system than a simple text messenger. A performance evaluation methodology is needed for a systematic approach to evaluate the performance of a system. The performance of a system depends, among other factors, on the underlying transport system, which is in the scope of this research a Wireless Local Area Network (WLAN). Therefore, in section 4.1.1 an overview is given of factors that influence the performance of a WLAN. The performance metrics are discussed in section 4.1.2 The methodology discussed in section 4.1.3 will be used to structure the process of evaluating the performance of the WLAN. The reason this methodology is chosen is, because it is used in a similar project before ([36]) and is proven to be useful. In this methodology, also the performance metrics that are relevant to my system will be chosen.

4.1.1 WLAN performance factors

[2] The performance in a WLAN can vary a lot, depending on many factors, which are described below.

1. WLAN (802.11) protocol: The IEEE 802.11 standard defines various physical-layer rates

[1]The criteria used for performance evaluation of a system[27]

[2]The material in this section is derived from [20]

for different types of WLANs, such as 1,2,5.5 and 11Mbps for 802.11b. Rates for 802.11a and 802.11g include 6,9,12,18,24,36,48 and 54Mbps. The user goodput is less than these link rates for the following reasons:

- Each packet includes additional data, such as preambles, headers (MAC, IP, and TCP) and checksums.
- When every directed (unicast) message is received, the receiver transmits a short acknowledgement packet back to the sender.
- Transmitters wait for short random times between packets to allow other users to contend for and share the channel.

Given these reasons, the theoretical maximum user-level performance for the various 802.11 systems is:

	Number of channels	Modulation	Maximum link rate (Mbps)	Maximum TCP rate (Mbps)	Maximum UDP rate (Mbps)
802.11b	3	CCK	11	5.9	7.1
802.11g with 802.11b	3	OFDM/CCK	54	14.4	19.5
802.11g	3	OFDM/CCK	54	24.4	30.5
802.11a	19	OFDM	54	24.4	30.5
802.11a turbo	6	OFDM	108	42.9	54.8

Table 4.1: assumes 1500-byte packets, encryption enabled, default 802.11 MAC configurations, zero packet errors and maximum available channel bandwidth (operating at close range).

2. Radio environment (medium): Several issues affect the way the radio signal propagates from sender to receiver.

- Radio energy attenuates when it propagates. As radio waves propagate outwards spherically, the energy spreads out over an ever increasing area. In free space, doubling the distance decreases the received power by a factor 4. Radio signals also attenuate when they pass near or through objects such as floors, walls, furniture and people. The attenuation increases with the object's conductivity (due to metal or water content, for example). The combination of these attenuation effects reduces radio signal strength by $\frac{1}{R^3}$ to $\frac{1}{R^4}$, or even $\frac{1}{R^5}$ (R=Radius from source to signal). In other words, each time you double the distance, the received power might decrease by 8, 16 or even 32 times.

- Antenna design affects how much radio-frequency (RF) energy is transmitted or received and where it is directed.

- Scattering and multi-path (signals can take different paths to receiver) cause fading effects. Signal strength can change rapidly as a function of location because the received signal is the sum of potentially numerous signals scattered from nearby objects. As the transmitter or other objects in the environment move, the scattered signals sometimes add together and sometimes cancel each other. Fading occurs over time as well as over location due to small changes in the environment. As a result the received signal strength can change over time, even when the location of the receiver and transmitter is fixed.

- Scattering and multi-path result in delay spread. The received signal might contain several slightly delayed copies of the transmitted signal, as the scattered signals travel via different physical paths of different lengths.

- Other devices occupying the same or nearby channels cause interference. For example, the 2.4 GHz spectrum might be occupied by Bluetooth devices, microwave ovens, and cordless telephones.

3. Frequency: A common misconception is that free-space propagation depends upon frequency, so higher frequencies are assumed to propagate less well than lower frequencies. However, effects such as antenna efficiency, RF component performance, and absorption through and scattering around objects do depend upon frequency. Here are some of the frequency-dependent effects:

- Generally, antennae of the same physical size tend to become more directional (have higher gain in some directions and less in others) as the frequency increases.

- Absorption due to propagation through objects tends to increase with frequency.

- Scattering around objects might have a positive or negative effect on signal strength as a function of frequency, depending upon the relative sizes and locations of the objects.
- Noise and spurs generated by nearby electronics (for example, inside the Access Point (AP) or PC laptop) in addition to co-channel interference, such as Bluetooth devices, cordless phones and microwave ovens, will degrade lower frequencies sensitivity more than high frequencies.
- Attenuation increases with frequency, so antenna cables (if present) in the AP or laptop will have more attenuation at high frequency, unless more expensive cables are used.

4.Vendor equipment: Equipment from different vendors exhibit significantly different performance due to architecture, design, manufacturing and software variations, as well as proprietary features and enhancements.

5.Vendor interoperability: Products that undergo WiFi-certification are certified to interoperate with a wide variety of vendor products. However, these tests mainly verify basic connectivity and do not enforce stringent throughput requirements. You might be able to connect a client device to a different vendor's access point, but you might not get a very high throughput.

6.Security: No matter what security standard (e.g. WEP, WPA, and WAE) is involved, the way the standard is implemented can affect the WLAN's performance. Specifically, some vendors implement encryption in software, which can dramatically reduce throughput compared to advertised rates. Depending upon the application of interest, the measurements are performed with encryption enabled or disabled.

4.1.2 Performance metrics

The term "performance metric" refers to the criterion used to evaluate the performance of a system[27]. For example, response time (the time to service a request) could be used as a metric to compare two systems (or sub-systems) The ITU-T provides a systematic method of identifying system performance metrics and their related performance parameters for digital networks. This method is known as the 3 x 3 matrix approach [8]

In [36], the authors conclude that although the focus of this approach is aimed at circuit switched networks, it can be used to select the performance metrics of interest for a packet switched network. In the 3 x 3 matrix approach, the following definitions are used:

Primary performance parameter

> A parameter or a measure of a parameter determined on the basis of direct observations of events at service access points or connection element boundaries (e.g. delay).

Derived performance parameter

> A parameter or a measure of a parameter determined on the basis of observed values of one or more relevant primary performance parameters and decision thresholds for each relevant primary performance parameter (e.g. goodput).

The 3 x 3 matrix[3] approach is presented in table 4.2.

Function \ Metric	Speed	Accuracy	Dependability
Access			
User information transfer			
Disengagement			

Table 4.2: ITU-T 3 x 3 matrix: relation between communication functions and performance metrics

The main features are:

- Each row represents one of the three basic and distinct communication functions. Note: The access function represents the connectionless as well as connection-oriented services that are possible with transport systems. Note: I will focus only on the connection-oriented service because the MSP system, which is used for the performance measurements, uses TCP (i.e. connection-oriented)
- Each column represents one of the three mutually exclusive[4] outcomes possible when a function is attempted.

[3]The description of the matrix approach is copied from [8] document and modified for my purposes
[4]Outcomes being related such that each excludes or precludes the other[37]

- The 3 x 3 matrix parameters are defined on the basis of events at the system boundaries and are termed "primary performance parameters". "Derived performance parameters" are defined on the basis of a functional relationship of primary performance parameters.

- Transport system primary performance parameters should be defined so as to be measurable at the system boundaries of the system to which they apply.

Description of the basic communication functions

- Access: The access function begins upon issuing an access request or its implied equivalent at the interface between a user and the communication network. It ends when either: 1) The access request is accepted and a response is issued to the calling users; or 2) At least one packet of user information is input to the network (after session establishment).

- User information transfer: The user information transfer begins on completion of the access function, and ends when the "disengagement request" (terminating a communication session) is issued. It includes all formatting, transmission, storage, and error control and media conversion operations performed on the user information during this period, including necessary retransmission within the network.

- Disengagement: There is a disengagement function associated with each user in a communication session: each disengagement function begins on issuing a disengagement request signal. The disengagement function ends, for each user, when the network resources dedicated to that user's participation in the communication session have been released. Disengagement includes both physical disconnection (when required) and higher level protocol termination activities.

Description of the performance metrics

- Speed: Speed is the performance metric that describes the time interval used to perform the function and the rate at which the function is performed. Note: The function may or may not be performed with the desired accuracy.

- Accuracy: Accuracy is the performance metric that describes the degree of correctness (e.g. error probability) with which the function is performed. Note: The function may or may not be performed with the desired speed.

- Dependability: Dependability is the performance metric that describes the degree of certainty (e.g. blocking probability) with which the function is performed regardless of speed or accuracy, but within a given observation interval.

4.1.3 Methodology

A widely used performance evaluation methodology of computer systems is based on the methodology of R.Jain [27], who identified ten phases in this process. According to [36], the methodology can be adapted and/or refined for the use in telecommunication systems.

Three techniques exist for performance evaluation of computer/telecommunication systems, namely: analytical modelling, simulation and measuring a real system. In this assignment the measuring technique is chosen, because the UT-WLAN (a WLAN available at the University of Twente), which is the transport system to be tested, is already fully operational and the accuracy of this method is better then the others when executed correctly. [27] This leads to the following nine phases, which will be explained in the remaining parts of this chapter:

1. State the Goals and System Definition
2. List Services and their Outcomes
3. Select Performance Metrics
4. List System and Workload Parameters
5. Select Factors and their Levels
6. Select System and Workload Parameters
7. Design and Execute the Experiments
8. Analyze, Evaluate and Interpret the Data
9. Present the Results

1.State the Goals and System Definition[5]

[5] The material in the rest of this section is derived from [36]

The first phase of the measurement activity is answering the question what system performance characteristics are of importance and why (i.e. what is the objective of the study) and what is the system as an object of the study (i.e. what are the system boundaries). A description of the system should be provided as far as it is known and relevant to the stated objectives. For complex systems, that consists of many subsystems, this activity can imply a system decomposition process, such that the system of interest is identified, or even the particular component of the system is highlighted as of interest. The objective of the study is the key consideration while decomposing a system. Moreover, the goal set strongly affects the level of accuracy required for a performance study. For example, when considering the goal of speed determination of a PC, the goals and system definitions stated by the end-user and the system architect will be different. Namely, the end-user would usually state the interest in the speed of particular application running on this PC, while the system architect will scale the goal down to the speed determination of the (particular) processor, as a component of this complex system.

2.List services and their outcomes

The next phase is to answer the question what services the system offers and what their respective outcomes are. In the particular case where a system offers many services a selection of services of interest and their respective outcomes is necessary. Generally, these outcomes are divided into three categories: the system may perform the service correctly, incorrectly or refuse to perform the service. Some of these outcomes are desirable and some are not. If the system performs the service incorrectly, an error has occurred. For example, the database system provides searching and sorting services. The service of interest may be that database responds to a searching query. It may answer to the query correctly, incorrectly (e.g. due to its inconsistency) or not at all (e.g. due to deadlocks). The outcome of interest may be the correct one.

3.Select Performance Metrics

The next phase is to identify, select and define the performance metrics of the transport system. Phase 1 defines the system and phase 2 the services and their respective outcomes. Performance metrics are associated with the three possible service outcomes (i.e. successful service delivery, error, and service unavailability) and relate to speed, accuracy and availability of the services offered by the system. For example, the performance metrics of the transport system can be availability, speed (e.g. delay, goodput) and reliability (e.g. error rate of the transported data). For each service offered by the system, there are a number of speed metrics, a number of reliability metrics and a number of availability metrics. For the system that offers more than one service, the number of metrics grows proportionately. Let us consider the example of the database system from the previous phase. If the system answers to the query correctly, its performance is measured by: the elapse time to perform this service (responsiveness), the rate at which this service is performed (productivity) and the resources consumed by this service (utilization).

The desired accuracy of the evaluated performance parameters is of vital importance. The question: "What level of detail is required for the estimation of particular parameter?" is raised. According to the desired accuracy level, the mean values, variances or complete distribution of measures of interest need to be obtained.

4.List System and Workload Parameters

In this phase, the main concern is the set of parameters that affect the system performance. There are system-specific parameters and workload parameters. System parameters include both the hardware and software parameters, which generally do not vary among various installations of the system. Workload parameters are characteristics of user's requests, which vary from one installation to the next. For example, the buffer sizes (system parameter) and volume of data being transported (workload parameter) affect the transport system performance.

5.Select Factors and their Levels

The list of identified parameters can be divided into these that will be constant during the measurements and these that will be varied. The latter group of parameters are called factors and their values are called levels. The selection criteria of parameters as system factors should be: a) the influence level of particular parameters on the system performance, and b) the feasibility of changing these parameters. Regarding the transport system, the important factors can be size and arrival rate of the transported packets.

6.Select System and Workload Parameters

The workload consists of the list of service requests and their arrival intensities. For a particular service request, the values of workload and system parameters are determined.

The list of service requests is a representative of the system usage in real life or the planned system usage. For measurements, the workload consists of user executable scripts on the system. The measurements done on the transport system need to explore the variety of possibilities of transport service requests.

7.Design and Execute the Experiments

The decision on what experiments and in which order to perform these experiments to obtain the maximum information with minimal effort is taken. The equipment and/or additional code needed for experiments and their instrumentation needs to be listed. The report on when, where, how and by whom the experiments should be executed follows.

8.Analyze, Evaluate and Interpret the Data

At this phase, the (raw) measurement results need to be collected and (roughly) validated. In addition, the generation of the statistical data based on the (raw) data obtained from each individual experiment is assumed. Data interpretation phase can follow only if the obtained results are explainable and accepted, otherwise the performance evaluation measurement process needs to be revised and, if necessary, repeated (i.e. start from phase 1 again).

9.Present the Results

At this point, the obtained data is analyzed, presented and discussed. It is important that the results are presented in an easy understandable manner (e.g. in graphic form). Often at this point, the knowledge gained by the study may require going back and reconsideration of some of the decisions made in previous phases. For example, redefining the system boundaries or inclusion of other factors and performance metrics that were not considered before. In this case, another cycle through all the steps is required.

4.2 Objectives

Measuring performance can be time-consuming and difficult because systems may be complex and there are many performance metrics. Therefore, it is important to state the objectives clearly; why is this performance study actually needed?

Context

In this section, the context in which the measurements will be executed are discussed. As explained in chapter 2, the measurements are executed within the SoD. The positioning service (see [32]) makes use of the MSP by offering a mobile host location to other service users. With this service, it is possible to relate the (mobile service perceived) network performance and location, which is the ultimate goal of these performance measurements.

Transport system of interest

The MSP transport system consists of different systems because it has a mobile and a fixed part. However, for this assignment the measurement activity is limited to one mobile transport system, the transport system of interest:

The WLAN of the University of Twente (UT-WLAN).

MSP - MSP TS interaction

The MSP system uses the MSP Transport System (TS) as a lower level service(See figure 3.6). The Interconnect Protocol offers a means of transportation for the application messages from the mobile service to the Surrogate and vice versa. . The exchange of Interconnect messages (RequestMessage, ReplyMessage and OnewayMessage) is visualized in section 2.3. Because the Interconnect Protocol specification supports multiple features (i.e. device discovery, Surrogate upload and Keep-Alive mechanism) the MSP transport system should be reliable. In section 4.2, the service of the MSP transport system is precisely defined.

Goal

The UT-WLAN is the mobile transport system part, of the MSP transport system, between the mobile host and the surrogate host. The Interconnect Protocol supports communication between a mobile host (offering mobile services) and the surrogate host (offering surrogate objects). Objective:

- Measure the quantitative behaviour of the UT-WLAN as part of the MSP transport system for the Interconnect Protocol.

4.2.1 System description and boundaries

In chapter 3, the MSP transport system is presented, which consists of several sub-systems. The mobile host and the surrogate host are geographically dispersed; the mobile host is located in the UT-WLAN while the surrogate host can be anywhere on the Internet. However, in this assignment the surrogate host is on the (fixed) campus network of the University of Twente (UT-NET). The mobile service is located at the mobile host and the surrogate object (Surrogate) is located at the surrogate host.

System decomposition

As mentioned in the last section, the MSP transport system consists of the UT-WLAN and the UT-NET, which are both controlled by the University of Twente.

The transport system of interest is the UT-WLAN. Because the surrogate host is located on the fixed network (i.e. UT-NET), a gateway[6] is needed to make a connection between the mobile network (location of the mobile host) and the fixed network. Figure 4.1 presents the decomposed abstract model of the MSP transport system.

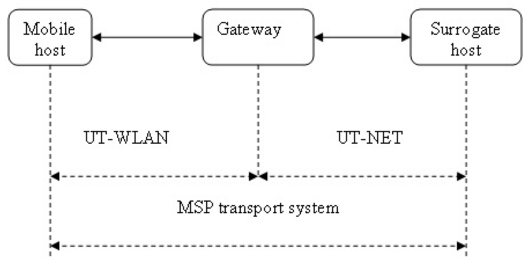

Figure 4.1: Decomposed abstract model of the MSP transport system

System under Test

In this section a conceptual framework is discussed that will be used to describe the MSP transport system in an abstract manner. The conceptual framework consists of a System of Discourse (SoD) that contains a System Under Test (SUT) [6], and is based on the "black-box - white box" principle [36]. The framework also consists of an evaluation system, which is discussed in 4.2.

[6]The gateway device is considered to be for a part in the UT-NET and for a part in the UT-WLAN

Black-box white box principle

The black box - white box principle is applied to the decomposed abstract model of the MSP transport system (see fig 4.1. Figure 4.2 depicts the MSP system, including the black box - white box model of the MSP transport system; MSP transport system is represented as a white box and UT-WLAN and UT-net as black boxes. The MSP protocol service uses the MSP transport system to support the MSP service.

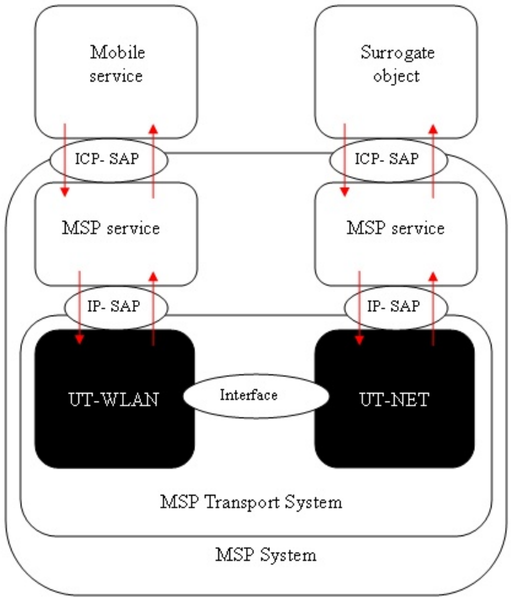

Figure 4.2: Black-box - white box model of the MSP transport system

A "black box" represents each transport system's sub-system. The MSP interactions with sub-systems take place at the boundary of each system by means of an Internet Protocol (IP) interface.

The mobile service and the surrogate object both access the MSP through an InterConnect Protocol Service Access Point (ICP-SAP). Service data can only be exchanged in Service Data Units (SDUs) between adjacent service layers by means of a Service Primitive[7] (arrows in SAP) exchange at their individual SAPs. The MSP service accesses the MSP transport system through an Internet Protocol SAP (IP-SAP). The MSP transport system as a whole can be represented as a black box because for the MSP service it is of no

[7]Interaction that can occur at a SAP, defined by stating its purpose and parameters [34]

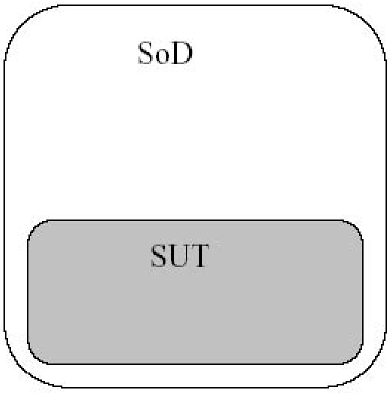

Figure 4.3: SoD including SUT

interest how this system delivers its service.

System of Discourse and System Under Test

The System of Discourse (SoD) is defined as:

> The total system that is of importance to the measurement activity and about which sufficient information is available. [6]

Because the goal of this assignment is to study the effect of the UT-WLAN on the MSP system (see fig 4.2), the MSP system is the SoD. The subject of the performance measurements is located inside the SoD and is called the System Under Test (SUT). In figure 4.3, an abstract representation is given of the SoD and the SUT.

The SUT is defined as:

> That part of the system of discourse of which the quantitative aspects of behaviour are under study (keeping the qualitative aspects of the behaviour the same). [6]

The objective is to measure the performance of the UT-WLAN, thus:

> The UT-WLAN system is defined as the SUT.

The non-SUT part of the SoD may influence the performance measurements of the SUT, so a description of the non-SUT part is necessary. The black box - white box model of the

MSP transport system is now integrated with the SoD conceptual framework and presented in figure 4.4. The mobile host and the surrogate host are respectively replaced by an abstract "service user 1" and "service user 2" because in the next section, the services of the sub-systems which are delivered to its users, are specified.

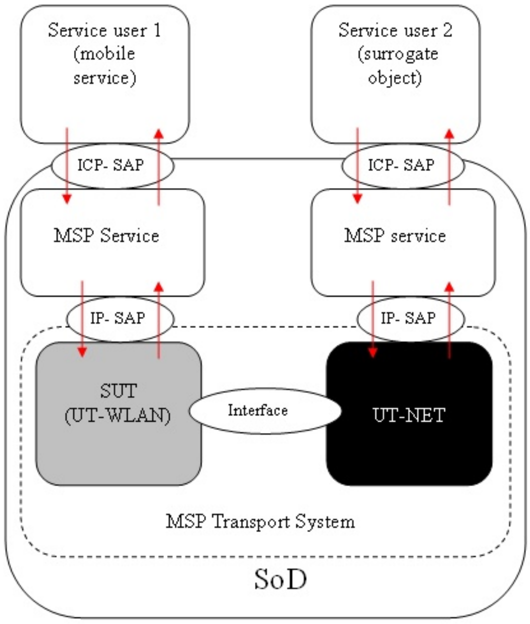

Figure 4.4: SoD including SUT as white box

4.3 Services and outcomes

For performance measurements, it is interesting to know the services provided by the SUT. The external observable behaviour of the SUT defines those services. The SUT communicates via an IP interface[8] to the UT-NET and delivers an IP service to the MSP protocol entity. Figure 4.4 abstracts from the service interaction between the SUT and the SoD. However, since the non-SUT part of the SoD may influence the performance measures of the SUT, a description of the services delivered by this part is relevant and therefore a SoD service decomposition is required.

[8]Implemented by a router or gateway

4.3.1 Service decomposition

In this section, the service (i.e. function) of the SoD is described. A service normally consists of sub-services which cooperate to deliver the service. Decomposition is an iterative process to split-up the service in those sub-services until you have reached "a satisfactorily level". The SoD service decomposition is given in figure 4.5.

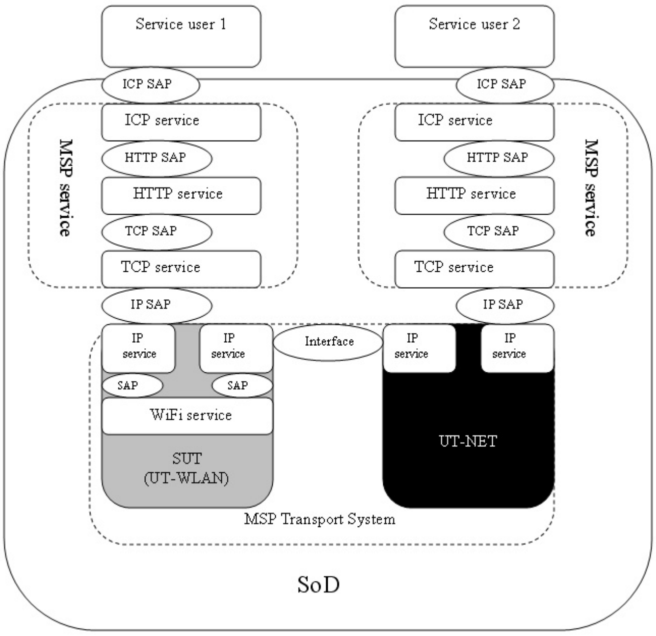

Figure 4.5: Sod decomposition

Service user 1 (mobile service) communicates with service user 2 (Surrogate) via the Interconnect Protocol (ICP), which is discussed in section 3.3. This protocol uses the HyperText Transfer Protocol (HTTP) which is an application protocol to transfer information over the Internet. HTTP in turn uses the Transmission Control Protocol (TCP). TCP is a reliable (i.e. no loss), connection oriented (i.e. before transmitting data a connection is created) in-sequence (i.e. the sequence of messages remains the same) delivery service for transmitting data segments (portions of the byte stream transmitted via TCP).

The SUT delivers an IP service, which is a connectionless and unreliable service that transports datagrams between hosts in a network. Actually, the IP service is implemented on the hosts but [36] conclude that this is of no influence on the SoD behaviour and that this service can be modelled in the SUT. This choice makes the big picture clearer, because the SUT(i.e. UT-WLAN) now delivers the well defined IP service. Concluding, the protocol stack between the SUT and the SoD consist of the following protocols; ICP, HTTP and TCP.

4.3.2 Service characteristic analysis

The service decomposition described in the previous section abstracts from the detailed services characteristics. These characteristics are relevant for the performance metrics of interest selection (section 4.4).

SUT

The UT-WLAN (i.e.SUT) delivers an IP datagram service. WLAN has different standards as is introduced in section 4.1.1, which have their own throughput rate. The UT-WLAN uses the 802.11b standard, which implements a symmetric communication link; the downlink and uplink have equal transport capacity with a maximum rate of 11Mbps[9].

The UT-WLAN architecture consists of eight VLAN's (Virtual Local Area Network), which are logical subgroups within a local area network (created via software) that allows traffic to flow more efficiently within populations of mutual interest. [31] In table 4.3 the use of these VLANs is explained.

SoD

The SoD delivers the Interconnect Procotocol service at the SoD boundaries (SAPs). This is a service that enables the exchange of Interconnect messages (i.e.RequestMessage, ReplyMessage and OnewayMessage) between Interconnect SAPs in a reliable and in-sequence way (achieved by using TCP as a transport service). Service execution is correct because the probability of data corruption in the sub-systems of the SoD, SUT and UT-NET, is assumed to be zero.

4.4 Evaluation System

An evaluation system is used to study the quantitative aspects of (parts of) the SoD behaviour. The evaluation system in figure 4.6 consists of a workload generator and a

[9] http://www.utwente.nl/wireless-campus/en/technology/WebWC-technology1.doc/

VLAN number	Name	Information
1	WLAN	In this VLAN the AP's reside and this VLAN is used for the Radius communication - native VLAN
2	WLAN @ UT	VLAN for employees and students
3	GUEST	VLAN for guest-usage
4	COM1	VLAN for commercial and test usage
5	COM2	VLAN for commercial and test usage
6	COM3	VLAN for commercial and test usage
7	COM4	VLAN for commercial and test usage
8	COM5	VLAN for commercial and test usage

Table 4.3: UT-WLAN VLANs

measurement function.

As can be seen in this figure, the workload generator has a direct relation with the SoD. In fact, the workload generator should have a direct relation with the SUT (system to measure) but because of instrumentation issues (i.e. how to provide workload at IP level?) this is not possible. This indirect way of measuring the reaction of the SUT to workload generator stimuli is valid under the assumption that the SoD does not influence the measurements. The workload generator influences the SoD's behaviour by generation of stimuli and is able to react to stimuli from the SoD. The measurement function measures the re-

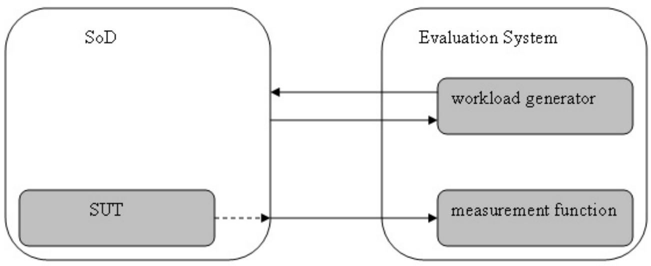

Figure 4.6: Functional overview of the SoD and an Evaluation System [36]

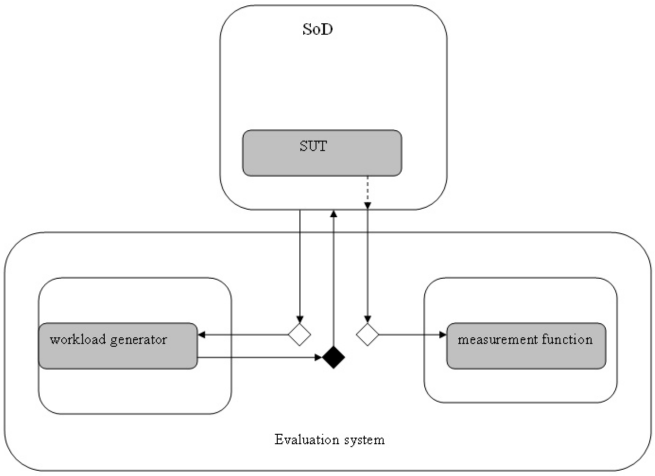

Figure 4.7: Functional overview of a distributed Evaluation System

action to generated stimuli (from the workload generator) at the SoD boundary (i.e. due to instrumentation issues). There is no direct relation between the measurement function and the SUT, because the measurement function shouldn't influence the SUT's behaviour. The fact that the workload generator and the measurement function have different relations to the SoD implies that they cannot co-operate with each other. The measurement function should not be able to interact with the workload generator because otherwise the measurement function is able to influence (indirectly) the SUT's behaviour.

Distributed structure

Performance measurements of a SoD which has a distributed structure (i.e. consists of geographically dispersed components) require a distributed evaluation system.

As can be seen in figure 4.7 the evaluation system consist of 2 distributed parts. One part consists of the workload generator that generates stimuli and can react to stimuli from the SoD. The measurement function only measures the reaction to generated stimuli. Because the measurement function depends on information from the workload generator when measuring one-way delay (i.e. the start time of the stream) the distributed parts should be synchronized in time. This is achieved through the use of GPS receivers for both parts.

Workload generator

The workload generator (WG) creates a workload that is offered to the SoD SAP. To measure the SUT performance this workload should be offered at the SUT SAP. Because the application runs on top of ICP and the SUT at the IP service it has to pass the ICP, HTTP and TCP service first. It can take the initiative to generate a stimulus, or it can respond to a stimulus from the SUT SAP (e.g. trigger event). The offered stimuli at the SoD SAP should be specific to reveal the SUT's behaviour. The stimuli consist of one or more Service Primitives (SP) exchanges between the workload generator and the SoD. The interaction can be seen as service elements (SE) exchange; SP's which are logically grouped, with a prescribed temporal ordering. The workload provided to the SoD should al least consist of multiple SE's which are specific to the (performance) measurement. Thus, the workload can be specified as multiple instances of the SE's offered at the SoD SAP.

The SAP at the SoD boundary is called an ICP-SAP. Because the workload generator and the measurement function both communicate, separately, via this SAP the SAP contains two connection endpoint identifiers. Thus the workload generator interacts with the SUT via the ICP-SAP and a its connection endpoint identifier. The behaviour of the SoD in response to the generated workload may influence the SUT's behaviour, which may result in a decrease of the overall performance measurement accuracy.

The workload generator can be parameterized, which means that values can be assigned to certain parameters of the workload. These parameters are called workload parameters. They can be seen as characteristics of user requests, which can vary depending on the SoD and SUT instance.

SoD and SUT instantiations can also be varied by system parameters, these include both hardware and software parameters that generally do not vary among various instantiations of the SoD and SUT. Examples of these parameters are the Network Interface Card (hardware) and an operating system (software). A particular workload choice is called a workload instance. Both system and workload parameters should be specified in order for the performance measurements to take place. This is needed because in a well performed research, the results should be reproducible and when these parameters are not specified this is not possible.

Measurement function

The measurement function only operates at the SoD or SUT SAP level. It is only interacting with the SoD through the ICP SAP (with its connection endpoint identifier), see figure 5.1

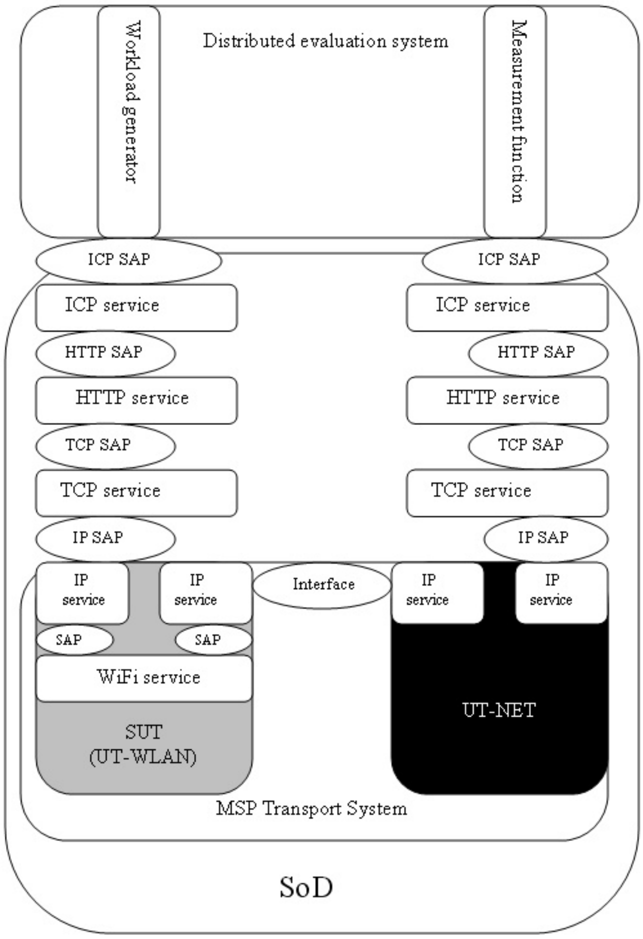

Figure 4.8: Evaluation system parts as service users of SoD

The measurement function should have a passive role; it should only listen to events (i.e. the arrival of an SP or SE) at the ICP-SAP. However, the main goal of the measurement function is not to record the events at this SAP but to relate them to their original events from the workload generator. In this way the effect of the generated stimuli can be analyzed. [36]

SoD and evaluation system

The evaluation system can be seen as a service user, because the stimuli of the workload generator should be offered/received at the service user level. The ICP-SAP is the only interaction point between the service user and the SoD.

Parameter / Metric	Delay	Jitter	Goodput
Speed	Primary	Derived	Derived

Table 4.4: Performance metric and performance parameters of interest

4.5 Select performance metrics and parameters

Recall the ITU 3x3 matrix in section 4.2, where the communication functions and performance metrics are specified. Based on the objectives defined in section 4.2, the focus is on the communication function user information transfer and its performance metric speed. The [8] description of performance metric speed is:

Speed is the performance metric that describes the delivery time that is used to successfully perform a transfer function and the rate at which this transfer is performed. The speed metric is linked to the primary performance parameter[10] delay and derived performance parameters[11] jitter and goodput see table 4.4.

delay

Primary performance parameter that is a measure for the time interval used to transfer a SE from a mobile host to a surrogate host. In figure 4.9 this is the time needed to transfer a SE from the ICP-SAP of service user 1 to the ICP-SAP of service user 2 (and vice versa).

jitter

Derived performance parameter, which is a measure of the variation in delay

goodput

Derived performance parameter, determined as the size of the data-carrying component (payload) of a particular SE, divided by the observed time at which this SE is being transferred.

[10] A parameter or a measure of a parameter determined on the basis of direct observations of events at service access points or connection element boundaries [8]

[11] A parameter or a measure of a parameter determined on the basis of observed values of one or more relevant primary performance parameters and decision thresholds for each relevant primary performance parameter.[8]

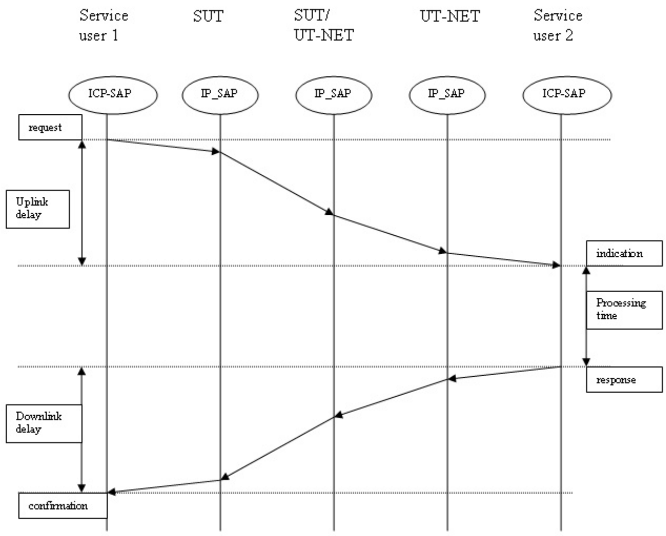

Figure 4.9: Time sequence diagram for the transfer of a SE

Figure 4.9 illustrates a time sequence diagram, which is used to calculate the transport delays in the system. When service user 1 does a request, this message is transferred via the SUT and the UT-NET to service user 2. The time needed to transfer this request is the uplink delay. The time it takes to transfer the response message from service user 2 to service user 1 is called the downlink delay.

Note: Recall, that I focus on the streaming part of the mobile applications. Hence, the performance evaluation of the uplink (communication link from mobile host to fixed host) is the focus of this research

The total delay (from request to confirmation) is given in the following formula: Delay = uplink delay + processing time + downlink delay. In general, the delay (also called latency) is the sum of the propagation -, transmit - and queue time. [19]. The formulas are:

Delay = Propagation + Transmit + Queue, where Propagation = Distance / Signal propagation speed and Transmit = Size / Bandwidth

In the whole transmission path from service user 1 to service user 2, four delays can be recognized. The steepness of the lines differs due to their (expected) differences in the transmission delay. The difference in transmission time is, for example, caused by the lower bandwidth in a mobile network (SUT).

4.6 System Parameters of influence

System parameters are being related to instantiations of a SoD and SUT. This section discusses the system parameters that may influence the outcome of the SUT performance measurements; these are called the system parameters of influence. The workload parameters, which are related to user requests, will be discussed in section 4.7.

System parameters of influence

A precise definition of a system parameter of influence is needed to clearly understand what these parameters are influencing. [6] uses the probability theory to answer this question and concludes with a clear definition of system parameters of influence. The following quotations are crucial to understand the meaning of the definition:

1. The performance measure is either a random variable, i.e. a variable that has some probability distribution function (PDF) or a probability. In case of a random variable, e.g. delay, one may think of the performance measure as described by the moments of the PDF of the random variable, such as mean delay and variance of the delay.

2. A value of the measure being the outcome of a single probabilistic experiment is taken as an observation for statistical inference.

3. A sequence of observations is called a 'sample'. It results from repeating the probabilistic experiment.

A measure is defined as:

> A random variable or a probability that quantitatively characterizes the behaviour of the SUT [6] Note: I consider the performance parameters of interest (section 4.3) as performance measures and conform to the "measure" definition from this point onwards in my thesis.

The system parameter of influence is defined as:

> A parameter whose value is a priori considered to affect the probability distribution function (PDF) of the measure (significantly).

This definition implies an as good as possible understanding of the SoD and SUT. It also limits the set of parameters to those that significantly affect the probability distribution function. Identification of candidate system parameters of influence is based on table 10

Parameter class	Parameter sub-class	Candidate system parameters of influence
System-description-related parameters	SUT structure	Mobile host IP interface (WiFi card), Access Point (AP)
	SUT internal behaviour	WiFi Service (transmission of data between clients)
	SUT external behaviour	IP service (parameters): - SP_send (header, payload) - SP_rcv (header, payload) - TOS field = not used - TTL field = assumed of no influence (i.e. does never reach the value of 0) - MTU size = 2304 bytes[33]
	Quantitative aspects of the SUT	Uplink transmission capacity Downlink transmission capacity Number of concurrent users
System-usage-related parameters	Workload	See section 4.6 for workload parameters
	Background load	SUT and SoD

Table 4.5: Identification of candidate system parameters of influence

in [6] and is presented in table . As can be seen in this table, these parameters are either related to system-description or to system-usage.

SoD candidate system parameters of influence that are located in the non-SUT part of the SoD are also identified. This is not according to the approach of [6], but these are selected because the SoD may influence the performance measurements of the SUT.

Thus, the candidate SUT system parameters of influence are combined with the SoD system parameters of influence. After analyzing these parameters, a selection is made based on their effect on the performance measurements or for which the quantitative aspects are known. The selected system parameters of influence are given in table 4.6. The levels of these selected parameters (factors) are given in section 4.7

SoD and SUT instance

A service delivered by a system discloses the external behaviour of this system. The internal working of the system is not visible to the system user and considered to be irrelevant to the users. However, for a performance evaluation perspective the implementation may play a significant role in the quality (e.g. speed) in which the service is delivered.

One implementation of the system may perform better than the other. The selected system parameters of influence are considered parameters that modify the implementation of a particular system (or system component). Many combinations, of system parameters of influence, are possible that deliver the same type of service but result in a different perceived Quality of Service (QoS). Each implementation of a SoD and SUT is considered to represent a particular instance of the system.

Performance load

The generated workload for the performance measurements should be realistic for mobile health applications in the AWARENESS project. The AWARENESS architecture describes that the mobile network can be a GPRS, UMTS or WLAN network. [10] The bandwidth of these networks differs and thus the application's throughput will be limited to bandwith of the network used. The bandwidths of the networks are presented in table 4.7.

[12] I assume the UT-NET is not influencing the measurements, see section 4.6 for an explanation

Parameter class	Parameter sub-class	Selected system parameters of influence
System-description-related parameters	SUT structure	Mobile host, Access Point (AP)
	SoD structure[12]	Mobile host IP interface (WiFi card), SUT, UT-NET, Surrogate host
	SUT internal behaviour	Assumed of no influence (generated workload is low compared to SUT's transport capacity, see section "performance load")
	SoD internal behaviour	Assumed of no influence (generated workload is low compared to SoD's transport capacity, see section "performance load")
	SUT external behaviour	IP service (parameters): - SP_send (header, payload) - SP_rcv (header, payload) - TOS field = not used - TTL field = assumed of no influence (i.e. does never reach the value of 0) - MTU size = 2304 bytes
	SoD external behaviour	Interconnect service: - SP_send (header, payload) - SP_rcv (header, payload) Interconnect specific options: - Keep Alive time interval
	Quantitative aspects of the SUT	Uplink transmission capacity Downlink transmission capacity Number of concurrent users Location of the Mobile host
	Quantitative aspects of the SoD	Keep Alive time interval
System-usage-related parameters	Workload	See section 4.6 for workload parameters
	Background load	SUT (UT-WLAN): Can be assumed of no influence. SoD (UT-NET): Can be assumed of no influence.

Table 4.6: Selected SoD and SUT parameters of influence

Network	Bandwidth (theoretical)
GPRS	171.2 Kbit/s [13]
UMTS	2 Mbit/s
WLAN: IEEE 802.11b	11 Mbit/s

Table 4.7: Mobile networks and their bandwidth

Because the GPRS network has the least bandwith, the mobile heath applications are tailored (i.e. tuned) to this network. When comparing the theoretical bandwidth of GPRS to that of WLAN (which is used in the experiments) it is about 1,5%. Thus, the generated workload for the performance measurements is at most 1,5% of the WLAN capacity and can thus be neglected. The fixed UT-NET has an even higher bandwidth than the WLAN (100 Mbps Ethernet), thus the load on that network is smaller than 1,5% and thus the load of the performance measurements on this network can also be neglected.

4.7 Select Factors and their Levels

In this section the workload and (selected) system parameters (factors) with their values (levels) are discussed.

Workload parameters

Workload parameters are parameters for which the effect on the performance measurements can be investigated. In [27] the workload parameters are defined as:

Measured quantities, service requests, or resource demands, which are used to model or characterize the workload.

If workload parameters characterize the workload, then they specify the parameters and rate of SE's generated by the workload generator. These parameterized SE's are typically generated to obtain a statistical correct measure of the primary performance parameter (delay). The performance objective (section 4.2) is to measure the delay between the mobile service and the Surrogate for a streaming application. This is the delay for one service user, thus the number of concurrent users, which is used in performance measurements with a scalability objective, is not a workload parameter. With the parameters size,

[13] Using 8 time slots and CS-4 coding

rate[14] and stream time, the uplink behaviour of the SUT can be found for different stream lengths, sizes and rates of SE's, and thus the primary performance parameter (delay) can be measured.

Thus the workload parameters are: Size, rate and stream time

Workload parameters and values

Workload parameters should have values in order for the performance measurements to be meaningful. In [27], this is called assigning levels (i.e. values) to factors (i.e. parameters to be varied). Thus, the factors are size,rate and stream time. The workload will be generated by the workload generator, which is part of the evaluation system. Note: I will look at the evaluation systems that are currently available to see if they are suited for my performance measurements. If this is not the case, I have to implement the workload generator and the measurement function

Size

The packet size is the amount of bytes of a SE that is transmitted between the mobile host and the surrogate host. Because the delay partly depends upon the SE size (see section 4.4), different SE sizes are chosen to calculate the relation between them. Before the "real" measurements will take place, I will do some experiments to see what the data size is like in the current MSP system. Based on this information, I can design the experiments for the actual measurements.

> Note: based on data from mobile health applications, 1267 Byte is an average SE size. The SE size is calculated from the factors sample rate, sample size and a compression factor from the software. Common values for the sample rate, sample size and compression were 128,18 and 45% which leads to a SE of 1267 Bytes.

Rate

The packet rate of the SE offered to the SUT's SAP is chosen to be a percentage of the (theoretical) maximum link capacity of GPRS, because mobile health applications are tuned to this capacity (see previous section).

[14] SE's per second

The increment is chosen so that the workload can be classified in terms of normal, medium to high and overloaded traffic. Based on this, the packet rate are chosen to be 1,10,20,30,40,50,60,70,80,90,100.

Stream time

The stream time specifies the length of the workload stream. Because a streaming application its performance can degrade because of the Keep Alive mechanism, this mechanism should be taken into account and thus the stream should be at least last 3 seconds (Keep Alive interval). Because of limited time available, and enough observations should be done for a reliable result, the stream length is chosen to be 10 sec.

Note:A value of 100 observations per measurement is chosen, because the variation in delay didn't change when more observations than 100 were performed.

In the rest of this chapter a Service Primitive (subscript) [15] will be used to indicate the number of packets per message (e.g.SP_10 means 10 packets per message)

The packet size is fixed to 1267 Bytes. Also the stream time and the message rate[16] are fixed in my experiments (because of limited time). The stream time of 10sec is chosen to take also in account the influence of the keep alive mechanism[17] The only variable left to increase or decrease the workload is the number of packets per message. Because the message rate is fixed at one per second the number of packets per message is equal to the packet rate (number of packets per second). Furthermore, a value of 100 observations per measurement is chosen, because the variation in delay didn't became smaller when more observations than 100 were performed.

Workload parameter	**Value**
Packet size	1267 Bytes
Packet rate	1,10,20,30,40,50,60,70,80,90,100
Stream time	10 sec

Table 4.8: Workload parameters

[15] Interaction that can occur at a SAP, defined by stating its purpose and parameters [34]
[16] number of messages per second
[17] This value is chosen to be 3 seconds in chapter 4

System parameters

Systems parameters are system "parameters of influence" which are classified as system description related parameters. To compare performance measurements between different SoD/SUT instantiations, these parameters must not change during one sample. In this way the effect of a system parameter can be studied and using this information the performance of the SUT can be optimized. Four system parameter classes are identified :

- SUT structure

  The SUT is considered to be the UT-WLAN. The UT-WLAN consists of a part of the mobile host (IP service of UT-WLAN is implemented on the mobile host) and the Access Points (AP's). The AP's are connected to the UT-NET via switches, routers and gateways.

- SoD structure

  The SoD consists of the mobile service (at the mobile host), the UT-WLAN, the UT-NET[18](including gateway) and the Surrogate (at the surrogate host).

- Quantitative aspects of the SUT

  The quantitative aspects of the SUT consists of the uplink transmission capacity. The uplink transmission capacity is the maximum amount of data that can be transported in the uplink direction.

- Quantitative aspects of the SoD

  The quantitative aspect of the SoD has only one factor: the Keep Alive interval. The Keep Alive mechanism takes system resources as well as network resources. The Keep Alive interval specifies the time between Keep Alive messages from the mobile service to the Surrogate and thus relates to the amount of resources that is necessary to support this mechanism.

[18]Although the impact of the UT-net to the SUT performance measurements is considered not measurable, it remains part of the SoD for reasons of completeness.

System parameters and values

In table 4.9, the values for the different parameters of influence are given.

Parameter class	Parameter sub-class	Examples of parameters of influence
System-description-related parameters	SUT structure	Mobile host: PDA Access Point: "Available" and "not available"
	SoD structure	Mobile host: PDA UT-WLAN: "Available" and "not available" UT-NET: "Available" and "not available" Surrogate: "Available" and "not available"
	Quantitative aspects of the SUT	Uplink transmission capacity: 11 Mbps[19] Downlink transmission capacity: 11 Mbps Number of concurrent users Location of the Mobile host
	Quantitative aspects of the SoD	Keep Alive interval:3000ms

Table 4.9: System parameters and values

[19] This capacity can only be reached if the signal strength of the mobile device to the AP is at least -85dbM. Otherwise the capacity drops to 5.5, 2 or even 1 Mbps[3]

Chapter 5

Implementation

This chapter describes the software framework that is developed in this project. This framework consists of the time synchronization module, location determination module and the evaluation system needed to execute and evaluate the performance measurements.

As concluded in chapter 2, time synchronization and location determination are both performed via GPS. How this is achieved is described in section 5.1. In section 5.2 and 5.3 the implementation of the mobile host and the surrogate host are discussed respectively. The chapter concludes with section 5.4, in which some technical challenges are mentioned that have been encountered during the project.

5.1 GPS Synchronization

For this project, there were a couple of GPS receivers that could be accessed via different technologies (e.g.Bluetooth,serial port,usb). For the mobile device, a Bluetooth GPS receiver is used because the GPS receiver should be at the same location as the mobile device and thus a wireless connection between the receiver and device is needed. At the surrogate host, I first used a GPS receiver via serial port communication but it gave varying results in the time it took to receive the GPS data. I couldn't find out what the problem was but because there was a Bluetooth GPS receiver that did not have the same problem, I didn't spend too much time to find out.

5.1.1 Surrogate host - TCP socket connection

For the surrogate host, a Bluetooth GPS receiver is used to synchronize the system clock with the GPS satellite network. The surrogate host is running on top of an Ubuntu Operating System (OS), as a VMware image, on a Windows XP host. A (usb) Bluetooth dongle on the host OS connected to the GPS receiver. Because VMmware player had problems to access the high (usb-to-serial) port number of the Bluetooth dongle, I decided to use the tool TCP-comm that can route the data from a serial port to a TCP port. The surrogate host thus accessed the GPS data via a TCP socket. This didn't significantly influence the time synchronization accuracy because the synchronization time was only a couple of milliseconds.

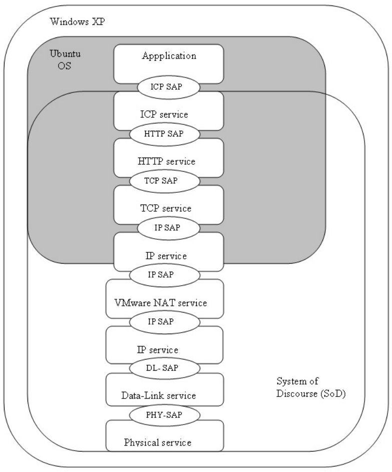

Figure 5.1: Ubuntu running as a guest OS in Windows XP

5.1.2 Mobile host - Bluetooth connection

For the mobile device a (direct) Bluetooth connection is used to retrieve the data from the the GPS receiver. Unfortunately I had some problems to setup a connection to the GPS receiver because my device (acer n50 premium[1]) has a Widcomm Bluetooth stack and the implementation of the Bluetooth stack in MSP is based on the Microsoft Bluetooth stack. On the Internet I found avetanaBluetooth, a JSR-82[2] implementation that pro-

[1]The specification of this device can be found at http://handheld.softpedia.com/devices/Acer/Acer-n50-Premium-71.shtml

[2]The official Java Bluetooth API

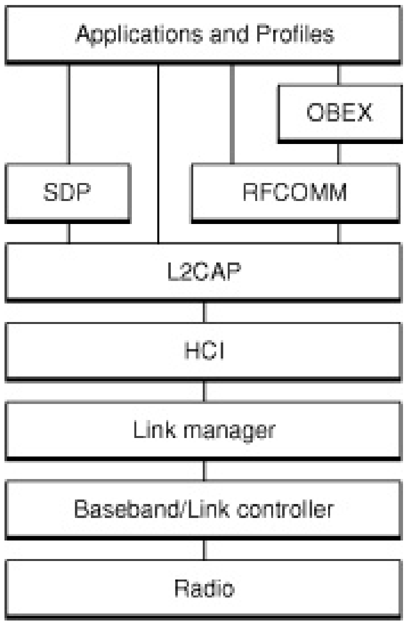

Figure 5.2: The Bluetooth protocol stack

vides Bluetooth functionalities to Java software in a standardized way. AvetanaBluetooth supports the Serial Port Profile (SPP). The Serial Port Profile defines the requirements for Bluetooth devices necessary for setting up emulated serial cable connections using RFCOMM between two devices. Using this implementation a connection to the Bluetooth GPS receiver could be made.

Bluetooth and WLAN interference

After some experimenting I found out that Bluetooth and WLAN didn't cooperate, the WLAN connection became unstable if I established a Bluetooth connection. Then I realized that Bluetooth and WLAN operate in the same frequency band, the unprotected ISM-band (Industrial, Scientific, Medical) at 2.4 GHz.

Thus the time synchronization could not be performed at the same time as the measurements were performed. The original plan was to use a daemon to synchronize the clock to GPS once in a while but because of the above described limitation this was not an option. Instead the time synchronization is performed before and after the measurements.

5.1.3 NMEA parser

Most GPS receivers use the NMEA 0183 standard to output data. This standard defines electrical signal requirements, data transmission protocol and time, and specific sentence formats. NMEA 0183 sentences are all ASCII. The Emtac GPS receivers used for this project support the following sentences of the NMEA 0183 specification:

- GPGSA = GPS DOP and Active Satellites
- GPGGA = Global Positioning System Fix Data
- GPGPS = Satellites in View
- GPRMC = Recommended Minimum Specific GPS/TRANSIT Data
- GPGSV = Track Made Good and Ground Speed

Only the GPRMC sentence is suited to retrieve both the current location and time from the GPS receiver. A specification of this sentence can be found in Appendix A

Because the data received from the GPS device is just plain bytes, a parser is needed to translate these bytes into sensible user data (e.g. time and location). The protocol for parsing the data from the GPS receiver works in the following way:

1. Read a NMEA sentence from the Bluetooth connection
2. Check if the checksum of the sentence is correct, if not go to step 1 otherwise proceed
3. Check if the message type is correct (GPRMC),if not go to step 1 otherwise proceed
4. Parse the data from the NMEA sentence to obtain the current time and location
5. Set the system clock and save location information to a local file.

5.1.4 Java Native Interface

To set the system clock, an a system call to the underlying operating system has to be used. Because Java is operating system independent, it cannot directly set the system clock. However it can call native methods of the underlying operating system via Java Native Interface (JNI) [14].

For the mobile host the underlying operating system is Windows Mobile 2003 2nd edition and for the surrogate host it is Ubuntu. The system call for Ubuntu (linux) is settimeofday which has a struct timevalue and a struct timezone as arguments. On Windows Mobile 2003 I used the nativeSetClock method which has as arguments the values for year,month,day,hour,minute,second and milliseconds. Because GPS outputs the time in the format hh:min:ss and day:month:year for the settimeofday this format had to be translated to UTC time.[3] because this functions requires UTC time as an argument.

5.2 Mobile Host

The software programmed for the mobile host consists of the workload generator module and the module for time synchronization and location determination. The software is run in the OSCAR environment, which is explained in the next section.

5.2.1 OSCAR

Oscar is an open source implementation of the Open Services Gateway Initiative (OSGi) framework specification.

OSGI

The OSGi technology is designed to ease the development of new and exciting services and applications for the latest generation of networked devices. Adding an OSGi Service Platform to a device, enables to manage the life cycle of the software components in the device from anywhere in the network. Software components can be installed, updated, or removed on the fly without having to disrupt the operation of the device. By exploiting these unique after-market sales possibilities, device manufacturers, service providers and software developers are able to improve time-to-market.[24]

The OSGi framework is small and reasonable lightweight, which makes it well suited for small devices, like PDAs.

[3] Universal Time Coordinated

Bundles

In OSGi, components are called bundles. A bundle is simply a JAR[4] file containing a manifest file and some combination of Java classes, native code, embedded JAR files, and resources.

In my implementation I have used the following bundles, which will be discussed shortly:

- TimeLocation bundle
- WorkloadGenerator bundle

5.2.2 TimeLocation bundle

Because the time synchronization module and location determination module retrieve their information from the same source, the GPS receiver, I decided to combine them in one bundle. In appendix C the structure of the timelocationbundle is given in a class diagram. According to the OSGI specification each bundle should have an Activator class which starts the bundle. The Activator calls the TimeLocationService which is the main class. This class calls the GPS module to retrieve the time and location information, calls the SystemClock class to set the system clock and has a method to write the latitude and longitude coordinates of the current location to a file.

The time synchronization module of this bundle for the mobile host is equally as for the surrogate host.

5.2.3 Workload Generator bundle

In appendix C the structure of the workload generator is given in a class diagram. The workload generator generates workload based on statistics from a Mobihealth application. The workload generator is immediately started after the time synchronization to keep possible errors (e.g. due to clock drift) as small as possible.

The workload generator class first creates a packet of a certain type and size. The packet is filled with random data. The total workload has properties such as the number of packets per message, the message rate, the time of the stream and the number of observations per measurement. These properties are specified in the msp.properties file,

[4]Java ARchive; an archive format for the programming language Java

which is read when OSCAR is started, so these properties can be changed without adapting the code. In appendix B an example is shown on the msp.properties file. These properties result in a certain bitrate, to achieve this bitrate, a Thread.sleep() is called after transmitting a packets. The packets are written to the outputstream of the surrogate connection.

5.3 Surrogate Host

5.3.1 Measurement Function

As explained in chapter 4, the measurement function should measure the primary performance parameter delay. The performance parameters goodput and jitter are derived from this parameter. The algorithm to calculate the delay is as follows:

- Listen for incoming connections
- Receive request to create surrogate of workload generator, which acts as the measurement function.
- Receive the workload stream, and count the number of packets and bytes received.
- Retrieve the send time of the stream at the mobile device
- Calculate the total delay of the stream and the average packet delay.
- Store information in database

In appendix C the class diagram of the measurement function is given.

5.3.2 Database connection

For storing the performance measurements data I have chosen to use a MYSQL database. The stored data contains the device number, the location coordinates of the Mobile Host(latitude and longitude) and the performance parameter delay. The database is located at the Surrogate Host. With specific MYSQL commands tuples (value-pairs) can be added and deleted to a table of a database. MYSQL also offers functions to calculate the minimum-,maximum and average value of a column this is used for the delay calculation.

The database contains 2 tables, one that contains basic information about the workload like number of packets received, packet size. The other table contains meta information about the workload like the performance parameters average delay of a packet plus variation and the jitter and goodput.

5.3.3 Performance map retrieval and processing

To create a performance map a map of the environment should be obtained. After searching for webservices that can be used for this purpose I found Google Maps , the Yahoo Maps Web Service and the Microsoft MapPoint Webservice.

The retrieve a map of a location the coordinates of that location have to be known. Currently there are webservices that have a geocoding function which means that they support the mapping of location to the coordinates. However, since GPS outputs the coordinates of the location, such functions are not needed. The location coordinates I stored in the database are the input for the map request. The scale should be chosen such that the map covers all locations where performance measurements have been done. The

purpose of the map processing part is to visualize the performance data. The challenge is to represent the performance measurements in such a way that a user can immediately see in which area a good performance can be expected and in which are a worse performance can be expected. Because I had only a couple of locations where I did the measurements the map is not complete, however it should give an impression of how it can be used in the future. Another option is to include multiple transport technologies so an overview is available which technology is suited in a particular area.

I choose to represent the secondary performance parameter goodput in colors. This means that for different goodput levels a different color is used, white for a good (e.g.high) goodput and darker when the goodput decreases. For a human colors are better interpreted that numbers, so it should be immediately clear in what location a good performance can be expected.

5.4 Technical problems encountered

- Bluetooth stack.I had some problems to setup a connection to the GPS receiver because the device used (acer n50 premium) has a Widcomm Bluetooth stack and the implementation of the bluetooth stack in MSP is based on the Microsoft Bluetooth stack. On the Internet I found a company called Avetana, which has made a java implementation of the RFCOMM protocol[5] This implementation can setup a connection to a bluetooth device via a serial port connection, and supports the Widcomm stack.

- Haicom GPS receiver unreliable synchronization time. First, I used a Haicom GPS receiver to synchronize the Surrogate Host but this gave some unpredictable results. The synchronization times varied from about 1 ms to more than 500 ms sometimes. One could compensate for this to measure this time and add this value to gps time. But because we had some other GPS device available I decided to test it with that device and this device didn't had the varying synchronization times.

- Mobile host clock setting. Setting the system clock of the mobile host seemed to go well, but after a while I fount out that the clock could not be set more accurate than 1 second on my PDA. Setting the year,month,day,hour,minute and seconds was no problem but the operation system just ignored the millisecond field in the system call to set the clock. Because the system call didn't return an error value I didn't notice this error immediately only after I did some measurements and saw some strange results. Fortunately the operating system of the surrogate host, Ubuntu , is able to set the clock to millisecond accuracy. The solution I used is to synchronize the clock of the surrogate host to the clock of the mobile host (and indirectly to GPS). First, at the mobile host I retrieved the GPS time and stored the offset of the local clock to GPS time. Then a message is sent to the surrogate host that request for synchronization of the surrogate host. The surrogate host retrieves the GPS time from the GPS receiver and sets it's clock to the offset of the mobile host which is included in the message. By calculating it's own clock offset to GPS time, the offset between the clocks of the mobile host and the surrogate host is known.

[5] The RFCOMM connection is simulating a serial port connection.

Chapter 6

Analysis and Evaluation

This chapter covers phase 8 and 9 of the performance methodology presented in chapter 4. These phases consist of analyzing, evaluating and interpreting the data that is obtained through the measurements. In section 6.2, this data will be discussed, while in section 6.1 the performance characteristics for a service user are evaluated. The bottleneck analysis in the SUT/SoD performance will be explained in section 6.3, while section 6.4 evaluates the influence of the mobile host its location on the SUT performance.

6.1 Data analysis

In chapter 4, the workload parameters have been specified. Recall, the workload consists of a stream of 10 seconds with a fixed packet size of 1267 Bytes. The only variable is the packet rate, which is the number of packets per second that is included in the stream. The SUT delay is measured as the time between sending the first bit of data and receiving the last bit of data minus the stream time.

In figure 6.1, the raw (unprocessed) data from an experiment with workload parameter SP_1(i.e. 1 packet/sec) is drawn. The value on the vertical axe indicates the SUT delay[1].

As can be seen, the SUT delay is increasing with the number of observations[2]. The reason for this is the difference in clock drift between the clocks of the mobile host and the surrogate host. Clock drift, which is measured over time, influences the clock

[1] Recall fig 4.4, SUT is part of the SoD

[2] For the experiments with a higher packet rate, the relative influence of the clock drift on the observed delay is less because of the larger delay.

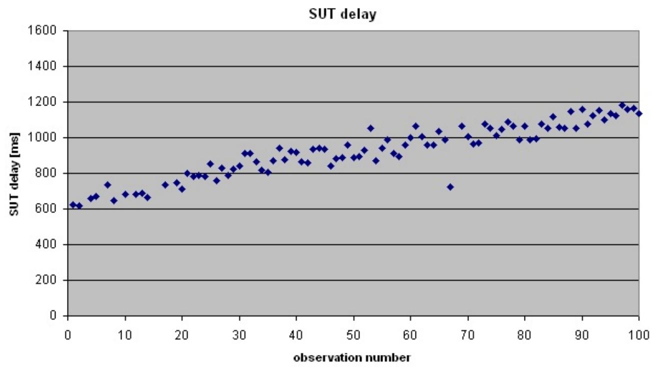

Figure 6.1: SUT delay for a stream with workload parameter SP_1 and 100 observations

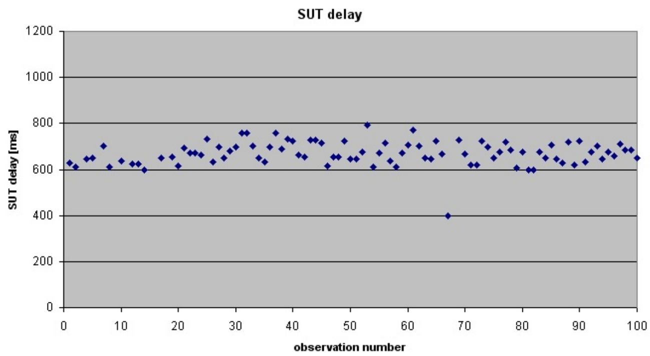

Figure 6.2: SUT delay corrected for clock drift

offset which is measured at a certain instant in time. As explained in chapter 5, the time synchronization process and the workload generator process could not run at the same time because of Bluetooth/WLAN interference. To deal with the clock offset and clock drift, the clocks have been synchronized to each other before and after the experiments[3] and the offset between the 2 clocks has been stored. When assuming that clock drift is a linear process, the delays can be recalculated according to the clock drift per time unit. The measured clock offset after the experiments is -571ms (clock of the fixed host runs faster than clock on the mobile host). Assuming that after the synchronization just before the clocks are equal, the formula for recalculating the SUT delays is: $f(x) = g(x) - (571/100) * x$ with f(x) the

[3]An experiment consists of 100 observations

corrected SUT delay, g(x) the uncorrected SUT delay and x the observation number. In figure 6.2 the result of this operation can be seen when applied to the data in figure 6.1. After this correction for clock drift, the average and standard deviation for the SUT delay can be calculated. The SUT delay is assumed to have a normal distribution, which is shown in figure 6.3.

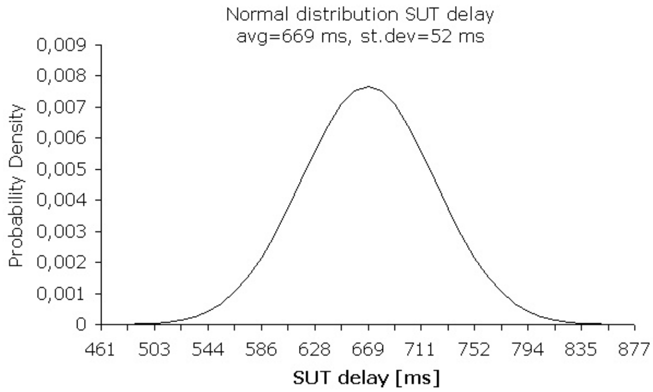

Figure 6.3: Normal distribution of the SUT delay for SP_1

6.2 Meta data analysis

In this section the delay characteristics of the SUT are discussed as observed by a service user (see fig 4.4)

6.2.1 Packet rate vs SUT delay

In figure 6.4, the SUT delay for different packet rates is shown. Recall, the SUT delay is the delay of a stream with a length of 10 seconds, where every second packets are inserted according to the packet rate. The vertical bars in the graph indicate the standard deviation. The first thing to notice when looking at the figure are the extreme high delays. This will be discussed in section 6.3. A logical result is that the SUT delay increases when the packet rate increases, because more data has to be send which leads to a greater delay. However, the slope of the graph is not constant, which means that a certain change in packet rate not always translates to a certain change in delay. The angle of the slope tells something about the goodput of the system. When the packet rate is doubled and the delay increased less than two times this means that the goodput has increased. As can be seen in the graph the slope is first getting smaller (goodput is increasing) and after the point of 60 packets per second the slope is getting larger which means that the goodput is decreasing.

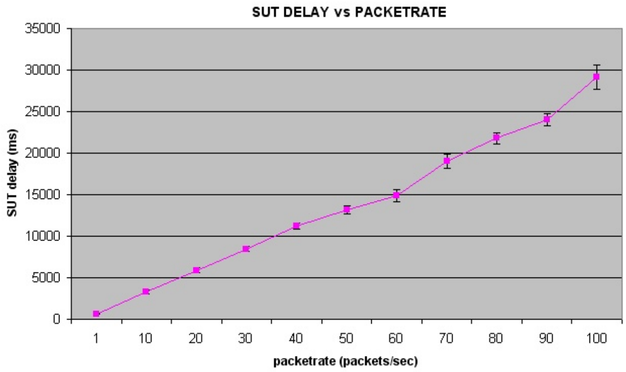

Figure 6.4: SUT delay versus packet rate

6.2.2 Packet rate vs goodput

The goodput is one of the secondary performance parameters that can be derived from the primary performance parameter delay. Goodput is the amount of data that is received at the receiver side of the application. As can be seen in figure 6.5, the goodput is first increasing and after a certain point it is decreasing.

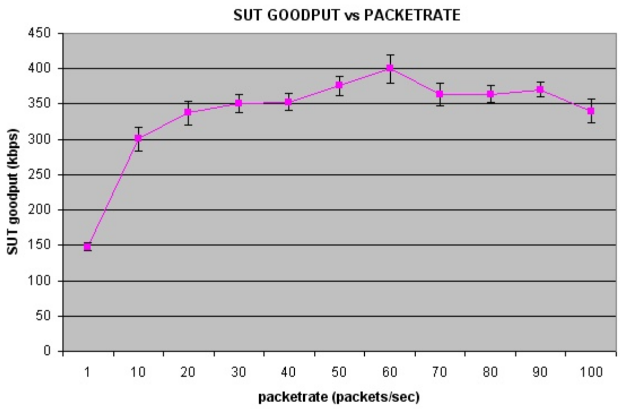

Figure 6.5: SUT goodput versus packet rate

The goodput is much lower than expected. According to 4.1 the maximum practical goodput of a 802.11b WLAN is 5.9 Mbps while in these measurements not even 10% of this limit is reached.

6.2.3 Packet rate vs average calculated packet delay in the stream

Because the packet rate is the variable in the experiments it is interesting to look at the relation between the packet rate and the average packet delay. This average packet delay is calculated by dividing the SUT delay by the number of packets received per stream. As can be seen in figure 6.6 the average packet delay at first decreases when increasing the packet rate but when the packet rate is higher than 60, it increases again.

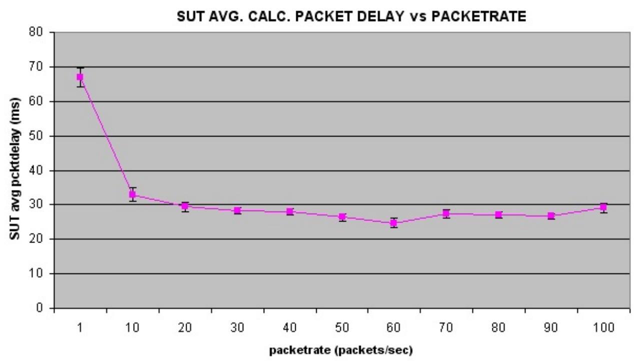

Figure 6.6: Average calculated packet delay of the SUT versus packet rate

As could also be concluded from the SUT goodput graph, there is an optimum in the range of about 60 packets per second. The reason why the average packet delay drops so much after a packet rate of 1, is that buffering takes place at the sender. This will be explained in section 6.3

6.3 SUT performance bottleneck

Recall chapter 4 in which the SoD is identified as the MSP system and the SUT as the WLAN. The MSP service decomposition done in 4.4 gives us a better view of where the performance bottleneck can be.

From the results in the last section, one can conclude that the performance of the SUT is much lower than the nominal capacity of the SUT. One possible explanation is that

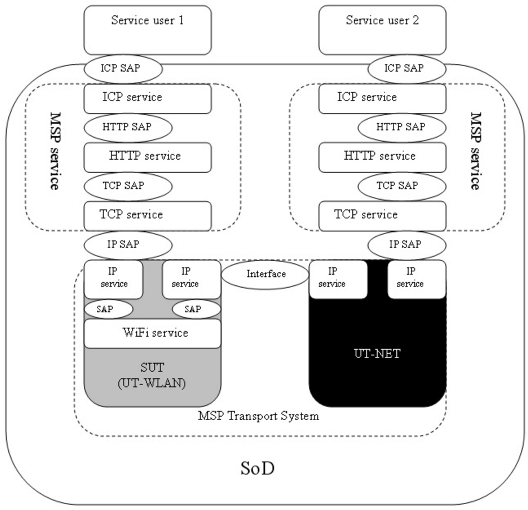

Figure 6.7: SoD decomposition

(some of) the protocol layers TCP,HTTP and ICP degrade the performance a lot. To find out if this is the case and which layer degrades the performance most, the influence of the SoD has to be reconsidered according to step 8 in Jain's methodology (section 4.1.3):

> Data interpretation phase can follow only if the obtained results are explainable and accepted, otherwise the performance evaluation measurement process needs to be revised and, if necessary, repeated (i.e. start from phase 1 again).

The performance bottleneck encountered can be partly due to resource starvation at the mobile host (e.g. CPU load). However, because of time constraints this influence could not be analyzed precisely and is therefore mentioned as a recommendation for future work in chapter 7.

6.3.1 MSP protocol layers

To research in which protocol layer a possible bottleneck resides, measurements have been executed using only TCP (on top of IP) and using ICP over TCP. The measurements have been carried out at the packet rate which has the highest goodput using the Interconnect Protocol (see fig 6.5, which is 60 packets/sec. The other system and workload parameters also remained the same. In figure 6.8 the goodput of TCP, ICP over TCP and ICP over HTTP and over TCP is compared. As can clearly be seen, using HTTP degrades the performance most.

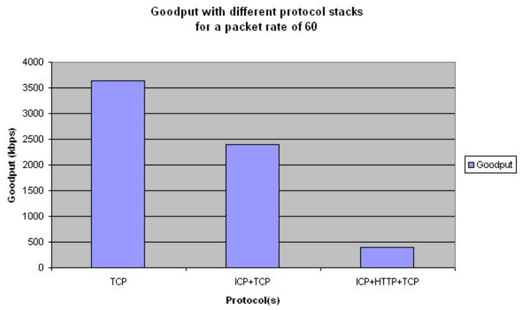

Figure 6.8: SUT goodput at different protocol stacks

By measuring at TCP level, TCP is the only service that is on top of the SUT service. Because TCP is not the bottleneck, the exact performance influence of the TCP service is not further discussed. In the next section, the influence of the HTTP service and ICP service is explained.

6.3.2 ICP and HTTP influence

The workload that is provided at the ICP-SAP is written to (the outputstream of a) CircularByteBuffer, which has a default value in MSP of 20.000 Bytes. The CircularByteBuffer implements the circular buffer producer/consumer model and is thread safe. This means that this buffer functions correctly during simultaneous executions by multiple threads. The workload generator (producer) and the consumer are separate threads. The consumer executes a HTTP POST method which has (the inputstream of the) buffer as body in the method. The HTTP POST allows HTTP clients to upload data to a web server. The mobile host acts a the HTTP client and sends the workload via the POST method to the Surrogate on the surrogate host (web server). After an HTTP session is established between the mobile host and the surrogate host, the connection acts in fact like a TCP connection (the HTTP-chunking protocol overhead is low).

When looking closer to the outputstream that is used to send the workload, it is of type (HTTP) chunked outputstream. This means that the data is packed in fixed size 'chunks' before it is send to the web server. A chunk is created when writing data to the outputstream of the CircularByteBuffer. However, before any data is send, first the HTTP buffer has to be filled to create a chunk. The default value of this buffer is 2KB. Remember the packet size is 1267B thus at least two packets have to be written to the outputstream before any data is actually send. When the HTTP buffer is filled, the client will send the HTTP chunk (containing the application data) via the TCP connection that has been created. When the data provided at the TCP-SAP is greater than the TCP-MSS[4], the data will be fragmented. The MTU discovery meachanism in TCP calculates the optimal MTU based on the path between 2 hosts. Recall, the mobile host is connected to a WLAN while the surrogate host is connected to an Ethernet. Because Ethernet has an MTU of 1500 bytes and WLAN of 2304 bytes, 1500 bytes is the limiting MTU and also the TCP-MSS. Because of this, the HTTP buffer data has to be transmitted in 2 TCP segments. This means that an extra ACK-packet has to be received at the mobile host. And because the second TCP segment of the HTTP buffer is not even half of the TCP-MSS (2048B-1500B) relative overhead compared to the data also degrades performance.

Concluding, using HTTP as an ICP transport protocol introduces delay because of the buffering. As explained at the begin of section 6.3, also the resource problems of the mobile host is a factor in the performance degradation. This is proved by sending the workload from the same computer as the surrogate host which resulted in much lower delays. Because of time constraints, the exact influence of the resource problems could not be measured.

Without using HTTP, thus ICP over TCP, the HTTP buffer is excluded from the stack and thus eleminates the inefficient buffering behaviour. When looking at the influence of ICP on the SUT performance, this can be explained by the CircularByteBuffer and the Keep-Alive mechanism. The CircularByteBuffer acts in fact like a queue in which the workload generator puts the data while the sending entity (a different thread!) reads from this queue. The other factor comes from the Keep-Alive mechanism which takes some network resources but also system resources.

6.4 Location dependency

6.4.1 UT-WLAN and City-WLAN comparison

Next to UT-WLAN, also measurements have been executed in the center of Enschede (City-WLAN). Because location in the network is considered a system parameter

[4] Maximum Segment Size; the maximum number of bytes a TCP packet (segment) can hold

(see chapter 4), this means the SUT has changed. In figure 6.9, the SUT, as part of the

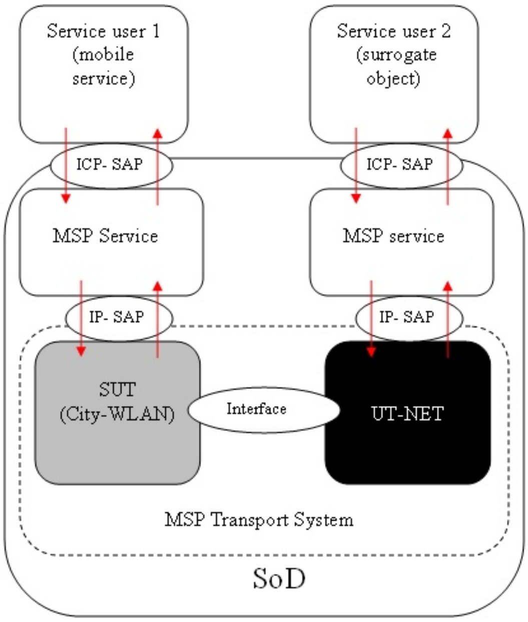

Figure 6.9: City-WLAN as SUT

SoD, is given for the measurements in the city. While the location is changed, the other system parameters and workload parameters were no different compared to the measurements executed at the UT-WLAN.

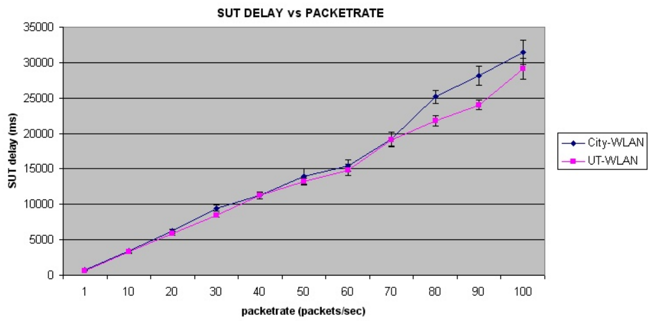

Figure 6.10: SUT delay versus packet rate

When comparing the graphs of the measurements at the City-WLAN and the one of the measurements on the UT-WLAN (6.10 and 6.11), it can be seen that in general the performance of the City-WLAN is less compared to the performance of the UT-WLAN.

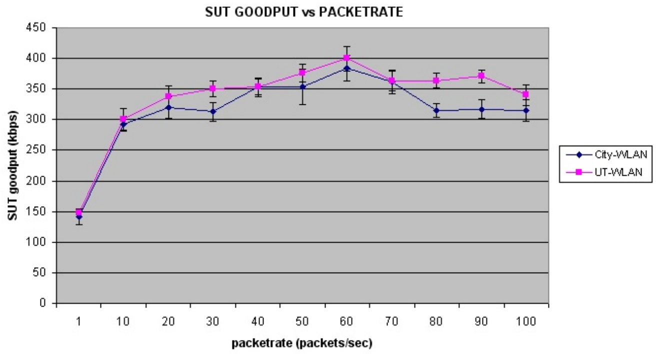

Figure 6.11: SUT goodput versus packet rate

For both locations there seems to be an upper bound on the goodput for a packet rate of 60 packets per second (figure 6.11). For the UT-WLAN this upper bound is 399kbps, while the City-WLAN can reach a maximum of 384kbps.

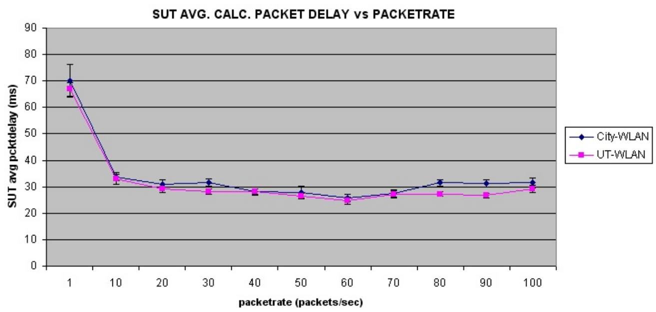

Figure 6.12: Average calculated packet delay of the SUT versus packet rate

In figure 6.12, the average packet delay of the stream in the City-WLAN and UT-WLAN is shown. As can be seen in the graphs 6.10,6.11 and 6.12 the difference between the 2 locations is not very high. The average SUT delay in the City-WLAN is about 5% greater than the SUT delay in the UT-WLAN. For a part, this can be due to a measurement error. Recall, delay consists of 3 parts namely: propagation delay, transmit delay and queue time. The reasons for this difference in performance can be:

- Propagation delay: As the surrogate host is placed at the UT-NET, the measurements at the UT-WLAN are executed at a shorter distance from the surrogate host than the measurements done in the City-WLAN. This means that a signal of the wired AP to the surrogate host, in the city, is larger than a signal from the wired AP to the surrogate host at the University of Twente. The difference in distance to the surrogate host is about five km for those 2 locations. Assuming that the network is for the most part wired, the signal propagation speed is 2/3 of the speed of light. When assuming the average signal travels 6 km because of reflection of these signals, then the propagation delay difference is $6000/200000000 = 0.03$ ms.

- Transmit delay: this is defined as the packet size divided by the bandwidth. The packet size is equally because the workload parameters remained the same. The bandwidth is dependent on the signal strength of which a mobile device is connected to an Access Point. When analyzing the traffic with ITBE[7], the average signal strength at the City-WLAN was between -59dBm and -65 dBm, while at the UT-WLAN these values were -71 en -74 dBm. According to the specification of the access points in [3], these signal strengths are sufficient to get the full capacity (11Mbps). Thus, the bandwidth that is available from the network is also equally for both locations and therefore the transmit delay is also equal.

- Queue time: The difference is that the mobile host at the City-WLAN needs possible more intermediate nodes to reach the gateway to the UT-NET. These intermediate nodes are switches in the wired portions of the WLAN that introduce delay. When analyzing the traffic with ITBE[7], the AP in the city needed 4 switches to reach the gateway (to the UT-NET) while the AP at the university needed 3 switches. This explains why the queue time is larger for the City-WLAN compared to the value at the UT-WLAN.

6.4.2 Performance Map example

In figure 6.13, I have used the Google Maps webservice to visualize the measurements results. I decided to use the average goodput (for all packet rates) at a certain location as a parameter for the visualization. Thus, based on the average goodput at a certain location a color is chosen that represents the value of the goodput. The color is the color of the marker which is placed at the location of the measurements. I choose a light color to represent a high goodput and a dark color for a low goodput. So a user can see directly if there is a place with a better performance than the user's current location. In table 6.1, the SUT goodput classes are given.

Average goodput (kbps)	Color
290-300	black
300-310	grey-black
310-320	grey
320-330	light grey
330-340	white

Table 6.1: SUT goodput classes and their color

The average SUT goodput of the UT-WLAN is 336 kbps, while the average SUT goodput of the City-WLAN is 315 kbps. The result of mapping the average SUT goodput to a color is visualized in figure 6.13.

Figure 6.13: Performance map using Google maps

Chapter 7

Conclusion and Recommendations

This chapter concludes on the work that has been done in this research project (section 7.1) and provides directions for future work (section 7.2).

7.1 Conclusion

The ubiquitous availability of communication infrastructures (e.g. WiFi, GPRS, UMTS) is one of the main driving forces behind the popularity of mobile applications. Traditional mobile applications are un-aware of their context; i.e. are not able to automatically adjust their functionality and/or performance based on the location of the end-user, but next generation mobile applications will be context aware. As stated in chapter 1, the main objective for this project was to perform location aware performance measurements for a Wireless Local Area Network (WLAN) as the mobile part of the MSP transport system. The main research goal for the performance measurements has been stated as:

- Measure the quantitative (uplink) behaviour of the UT-WLAN as the MSP transport system for the Interconnect Protocol.

First, measurements have been executed at the UT-WLAN. For a streaming application, the SUT goodput has been measured at a maximum of 400 kbps for a packet rate of 60 packets/sec. The maximum SUT goodput at the City-WLAN is 384 kbps which also appeared at a packet rate of 60 packets/sec. The difference in average goodput for both locations is about 5%, which is not significant. Factors that are causing this performance difference are the greater propagation delay and queueing delay for the City-WLAN.

The maximum goodput found in both locations, is much less than the nominal capacity of a 802.11b WLAN (11 Mbps). The performance methodology used, prescribes that in this case the measurements process should be revised and repeated. Because of implementation issues, the workload is provided at the SoD (ICP service) rather than at the SUT (IP service). The protocol stack between these MSP services (ICP, HTTP and TCP) degrades the performance seriously. Via TCP a goodput of 3640 kbps is reached, for a packet rate of 60 packets/sec, which is more than 9 times the goodput using ICP. The main factors responsible in the SUT performance degradation are the HTTP buffer and the resource starvation at the mobile device. The default value of the HTTP buffer (2KB) does not match the packet size (1267B). ICP degrades the SUT performance because of the CircularByteBuffer that adds a thread for reading the workload from the buffer. Possible resource problems at the mobile host (e.g. cpu load) could be (partly) responsible for the SUT performance degradation because of adding those protocol layers. However, this side effect could not be measured because of limited time. Concluding, the SUT performance degrades a lot when measured at the ICP service. If the reason for this performance degradation is only the protocol overhead or also limited resources of the mobile host has to be researched.

7.2 Recommendations

The work that has been described in this thesis has helped to understand the bottleneck in the performance of MSP. However, there are many area's in which more research can be done. In this section some proposals for future work are discussed.

- As mentioned in the last section, limited resource of the mobile host are possible responsible for the observed (low) SUT goodput. Therefore, it would be good to research what the exact influence of this is on the SUT performance degradation.

- Use more transport technologies for a better comparison It would be useful to create a map which can display the performance for multiple networks at different locations. Next to WLAN, other communication technologies such as UMTS or GPRS could be used to test the performance at different locations. When this data is available one can choose with this performance map, depending on the application needs, which communication technology is best suited for this application.

- Better integrate performance data with mapping services. To visualize the performance at multiple locations one needs a map of those locations. Nowadays there are multiple webservices that are able to retrieve a map of the environment. For most of them you need the latitude and longitude coordinates of the central point in the map and specify a certain scale. I looked at the following webservices:

  Google Maps: This is a quite simple webservice available via JavaScript that enables one to retrieve a map of the environment based on latitude and longitude coordinates. A while ago also the Google Earth API has been releases which offers even more possibilities.

  Microsoft Mappoint Web Service: this is a webservice with a bit more possibilities than the google maps webservice. It is possible to specify a lot of different properties of the image using their API, which is available in Java. I have created a performance map 'by hand', through calculating the average goodput and create markers on an image retrieved from google maps. However, it would be convenient to dynamically create a performance map using the measurement data and a webservice.

- Integrate performance information in applications. The real benefit of performance information is of course when an application can use this performance information to make optimal use of the available networks for its application needs. An example could be an application that can automatically change it's network based on the performance of multiples networks available. Also a background application could be developed which points to you a location that has a better performance than the current location.

Acronyms

GPRS General Packet Radio Services
http://nl.wikipedia.org/wiki/GPRS

GSM Global System for Mobile Communication
http://nl.wikipedia.org/wiki/GSM

GUI Graphical User Interface
http://en.wikipedia.org/wiki/Graphical_user_interface

HTTP HyperText Transfer Protocol
http://en.wikipedia.org/wiki/HyperText_Transfer_Protocol

IDE Integrated Development Environment
http://en.wikipedia.org/wiki/Integrated_development_environment

IP Internet Protocol
http://en.wikipedia.org/wiki/Internet_Protocol

J2ME Java 2 platform, Mobile Edition
http://en.wikipedia.org/wiki/J2ME

JAR Java ARchive
http://en.wikipedia.org/wiki/JAR_(file_format)

JVM	Java Virtual Machine http://en.wikipedia.org/wiki/Java_VM
LAN	Local Area Network http://en.wikipedia.org/wiki/LAN
MSP	Mobile Service Platform http://janus.cs.utwente.nl:8000/twiki/bin/view/MSP
NAT	Network Address Translation http://en.wikipedia.org/wiki/Network_address_translation
NTP	Network Time Protocol http://en.wikipedia.org/wiki/Network_Time_Protocol
PC	Personal Computer http://en.wikipedia.org/wiki/Personal_computer
PDA	Personal Digital Assistent http://en.wikipedia.org/wiki/Personal_digital_assisent
PE	Protocol Entity
RMI	Remote Method Invocation http://en.wikipedia.org/wiki/Java_remote_method_invocation
SAP	Service Access Point http://en.wikipedia.org/wiki/Service_Access_Point
SOA	Service Oriented Architecture http://en.wikipedia.org/wiki/Service_oriented_architecture
SP	Service Primitive

SUT	System Under Test
SoD	System of Discourse
TCP	Transmission Control Protocol http://en.wikipedia.org/wiki/Transmission_Control_Protocol
UDP	User Datagram Protocol http://en.wikipedia.org/wiki/User_Datagram_Protocol
UMTS	Universal Mobile Telecommunications System http://en.wikipedia.org/wiki/Universal_Mobile_Telecommunications_System
URL	Uniform Resource Locator http://en.wikipedia.org/wiki/Uniform_Resource_Locator
USB	Universal Serial Bus http://en.wikipedia.org/wiki/USB
WLAN	Wireless Local Area Network http://en.wikipedia.org/wiki/WLAN

Glossary

AWARENESS: Dutch collaborative project on context AWARE mobile Networks and ServiceS
Bluetooth: Wireless, short range communication standard
Nomadic mobile service: Software deployed on a mobile device (e.g. a PDA) that can be invoked by service users situated in the Internet
Nomadic service provider: Nomadic (mobile) service provisioning entity: computational device or human
Pocket PC: Handheld-sized computer that is delivered with a specific version of the Microsoft Windows CE operating system.

Bibliography

[1] P.Pawar A.van Halteren, E.A.M.Schoot Uiterkamp. Mobile service platform - realization of nomadic mobile service provisioning. Technical report, University of Twente, 2005.

[2] The awareness project(2005). http://www.freeband.nl/project.cfm?language=en&id=494, 2005.

[3] Cisco aironet 1200 series access point. http://www.cisco.com/en/US/products/hw/wireless/ps430/products_data_sheet09186a00800937a6.html.

[4] Differential gps. http://en.wikipedia.org/wiki/Differential_GPS.

[5] R.G. Golden. *Service and Device Discovery : Protocols and Programming*. Mc Graw-Hill, 2002.

[6] van Beijnum B.J. Hoeksema F.W., van der Veen J.T. A methodical approach to performance measurement, experiments: Measure and measurement specification. Technical report, University of Twente,CTIT, 1997.

[7] Itbe, department for information technology, library and education. http://www.utwente.nl/itbe/en/, 2006.

[8] Itu-t recommendation i.350, general aspects of quality of service and network performance in digital networks, including isdns. Technical report, ITU, 1993.

[9] P. Mtt J. de Heer, R. van Eijk and A. Peddemors. A generic interface for location handling. Technical report, Telematics Institute, 2002.

[10] K Lagerberg J Hendriks, A van Halteren, R Otte, J de Heer, and H Zandbelt. Integrated demonstrator architecture. Technical report, Telematics Institute, 2005.

[11] T. Mansten J. Mannermaa, K. Kalliomaki and S. Turunen. Timing performance of various gps receivers. *Proceedings of the 1999 Joint Meeting of the European Frequency and Time Forum and the IEEE International Frequency Control Symposium*, pages 287–290, April 1999.

[12] Deborah Estrin Jeremy Elson, Lewis Girod. Fine-grained network time synchronization using reference broadcasts. Technical report, University of California, 2003.

[13] Jini project (2005). the jini network technology. `http://www.jini.org/`, 2006.

[14] Java native interface. `http://en.wikipedia.org/wiki/Java_Native_Interface`, 2006.

[15] Elliott D. Kaplan. *Understanding GPS: Principles and Applications*. Artech House, 1996.

[16] A. LaMarca. Device positioning using beacons in the wild. Technical report, Intel Research, 2004.

[17] Leslie Lamport. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, 1978.

[18] YIP Chi Lap. *System clock resolution for different operating systems*. PhD thesis, University of Hong Kong, 2000.

[19] Bruce S.Davie Larry L.Peterson. *Computer Networks, a systems approach*. Addison-Wesley, 2000.

[20] Methodology for testing wireless lan performance with chariot. `http://www.atheros.com/pt/whitepapers/Methodology_Testing_WLAN_Chariot.pdf`, 2003.

[21] D. L Mills. Internet time synchronization: The network time protocol. *IEEE Transactions on Communications*, pages 1482–1493, October 1991.

[22] Mobihealth. `http://www.mobihealth.org/`, 2006.

[23] J. Newmarch. *A Programmer's guide to Jini Technology*. Springer-Verlag, 2000.

[24] Osgi. `http://www.osgi.org/`, 2006.

[25] Placelab. http://www.placelab.org/, 2003.

[26] B.rega of hopf elektronic gmbh; accuracy between dcf77 and gps. http://www.hopf-time.com/en/dcf-gps.htm, 2003.

[27] R.Jain. *The art of computer systems performance analysis: Techniques for Experimental Design, Measurement, Simulation, and Modelling*. 1991.

[28] Jochen Schiller. *Mobile Communications*. Addison-Wesley, 2000.

[29] Sun microsystems:the jini technology surrogate architecture specification. https://surrogate.dev.java.net/doc/sa.pdf, 2003.

[30] Jini surrogate project:the surrogate project. https://surrogate.dev.java.net/, 2005.

[31] Techweb. http://www.techweb.com/encyclopedia/, 2003.

[32] E.A.M. Schoot Uiterkamp. Nomadic position services for a mobile service platform. Master's thesis, University of Twente, 2005.

[33] Uninett. http://www.uninett.no/wlan/throughput.html.

[34] Quartel D.A.C. van Sinderen Vissers C.A., Ferreira Pires L. *The architectural design of distributed systems*. University of Twente, 2000.

[35] Wide area augmentation system. http://en.wikipedia.org/wiki/Wide_Area_Augmentation_System.

[36] K.E. Wac and R.G.A.Bults. Performance evaluation of a transport system supporting the mobihealth banip: Methodology and assessment. Master's thesis, University of Twente, 2004.

[37] Merriam-webster's online dictionary. http://www.m-w.com/, 2003.

[38] Global positioning system. http://en.wikipedia.org/wiki/Global_Positioning_System, 2006.

Appendix A

NMEA - RMC sentence

Example of a NMEA Recommended Minimum Specific sentence: $GPRMC,(1),(2),(3),(4),(5),(6),(7),(8),(9),(10),(11),(12),(CR),(LF)

1. UTC time of position fix, hhmmss.sss format.
2. Status, A = data valid, V = data not valid.
3. Latitude, ddmm.mmmm format.
4. Latitude hemisphere, N or S.
5. Longitude, dddmmm.mmmm format.
6. Longitude hemisphere, E or W.
7. Speed over ground, 0.0 to 1851.8 knots.
8. Course over ground, 000.0 to 359.9 degrees, true.
9. Date, ddmmyy format.
10. Magnetic variation, 000.0 to 180.O.
11. Degrees
12. Checksum, followed by a Carriage Return (CR) and Line Feed (LF)

Appendix B

MSP properties file

oscar.auto.start.1=file:oscar/bundle/shell.jar

file:oscar/bundle/shelltui.jar

file:oscar/bundle/bundlerepository.jar

file:oscar/bundle/MspIOBundle.jar

microedition.connection.pkgs=nl.utwente.msp.io.j2me

nl.utwente.msp.devconnection.keepalive.interval=3000

nl.utwente.msp.devconnection.replytimeout=10000

nl.utwente.msp.devconnection.timeout=60000

nl.utwente.msp.io.chunkoutbuffer.sizebytes=20000

timelocationbundle.surrogatehosturl=jinish://130.89.14.22

timelocationbundle.surrogatejar=http://130.89.14.22/surrogates/TimeLocationSurrogate.jar

timelocationbundle.gps.syncprotocol=tcp

timelocationbundle.gps.gpsserver=130.89.14.22

timelocationbundle.gps.gpsport=2000

workloadgeneratorbundle.transportprotocol=HTTP

workloadgeneratorbundle.surrogatejar=http://130.89.14.22/surrogates/WorkloadGeneratorSurrogate.j

workloadgeneratorbundle.surrogatehosturl=jinish://130.89.14.22

workloadgeneratorbundle.SAMPFREQ=128

workloadgeneratorbundle.PACKETSPERMESSAGE=100

workloadgeneratorbundle.RUNTIME=10

workloadgeneratorbundle.OBSERVATIONWINDOW=100

Appendix C

HTTP POST PDU

In frame 17, the HTTP header is encapsulated in an Ethernet frame. An example of segmentation is shown in frame 19, 20 which together constitute a HTTP buffer (2KB).
Frame 17 (215 bytes on wire, 215 bytes captured)
Ethernet II, Src: Vmware_f5:34:b0 (00:50:56:f5:34:b0), Dst: Vmware_fd:2c:7e (00:0c:29:fd:2c:7e)
Internet Protocol, Src: 130.89.14.22 (130.89.14.22), Dst: 192.168.236.128 (192.168.236.128)
Transmission Control Protocol, Src Port: 1185 (1185), Dst Port: 4756 (4756), Seq: 1, Ack: 1, Len: 161, Data (161 bytes)

0000	00 0c 29 fd 2c 7e 00 50 56 f5 34 b0 08 00 45 00	..)., .PV.4...E.
0010	00 c9 e2 2d 00 00 80 06 1a 69 82 59 0e 16 c0 a8	...-.....i.Y....
0020	ec 80 04 a1 12 94 1d a9 31 e5 09 02 db 4b 50 18	........1....KP.
0030	fa f0 74 60 00 00 50 4f 53 54 20 2f 64 65 76 69	..t'..POST /devi
0040	63 65 73 2f 64 65 76 2d 30 2d 50 6f 63 6b 65 74	ces/dev-0-Pocket
0050	50 43 31 30 30 2f 20 48 54 54 50 2f 31 2e 31 0d	PC100/ HTTP/1.1.
0060	0a 6d 73 67 63 6f 75 6e 74 3a 20 31 0d 0a 55 73	.msgcount: 1..Us
0070	65 72 2d 41 67 65 6e 74 3a 20 4a 61 6b 61 72 74	er-Agent: Jakart
0080	61 20 43 6f 6d 6d 6f 6e 73 2d 48 74 74 70 43 6c	a Commons-HttpCl
0090	69 65 6e 74 2f 33 2e 30 2d 62 65 74 61 31 0d 0a	ient/3.0-beta1..
00a0	48 6f 73 74 3a 20 31 33 30 2e 38 39 2e 31 34 2e	Host: 130.89.14.
00b0	32 32 3a 34 37 35 36 0d 0a 54 72 61 6e 73 66 65	22:4756..Transfe
00c0	72 2d 45 6e 63 6f 64 69 6e 67 3a 20 63 68 75 6e	r-Encoding: chun
00d0	6b 65 64 0d 0a 0d 0a	ked....

Frame 19 (1514 bytes on wire, 1514 bytes captured)
Ethernet II, Src: Vmware_f5:34:b0 (00:50:56:f5:34:b0), Dst: Vmware_fd:2c:7e (00:0c:29:fd:2c:7e)

Internet Protocol, Src: 130.89.14.22 (130.89.14.22), Dst: 192.168.236.128 (192.168.236.128)
Transmission Control Protocol, Src Port: 1185 (1185), Dst Port: 4756 (4756), Seq:162, Ack: 1, Len: 1460, Data (1460 bytes)

Frame 20 (642 bytes on wire, 642 bytes captured)
Ethernet II, Src: Vmware_f5:34:b0 (00:50:56:f5:34:b0), Dst: Vmware_fd:2c:7e (00:0c:29:fd:2c:7e)
Internet Protocol, Src: 130.89.14.22 (130.89.14.22), Dst: 192.168.236.128 (192.168.236.128)
Transmission Control Protocol, Src Port: 1185 (1185), Dst Port: 4756 (4756), Seq: 1622, Ack: 1, Len: 588, Data (588 bytes)

Appendix D

Class diagrams

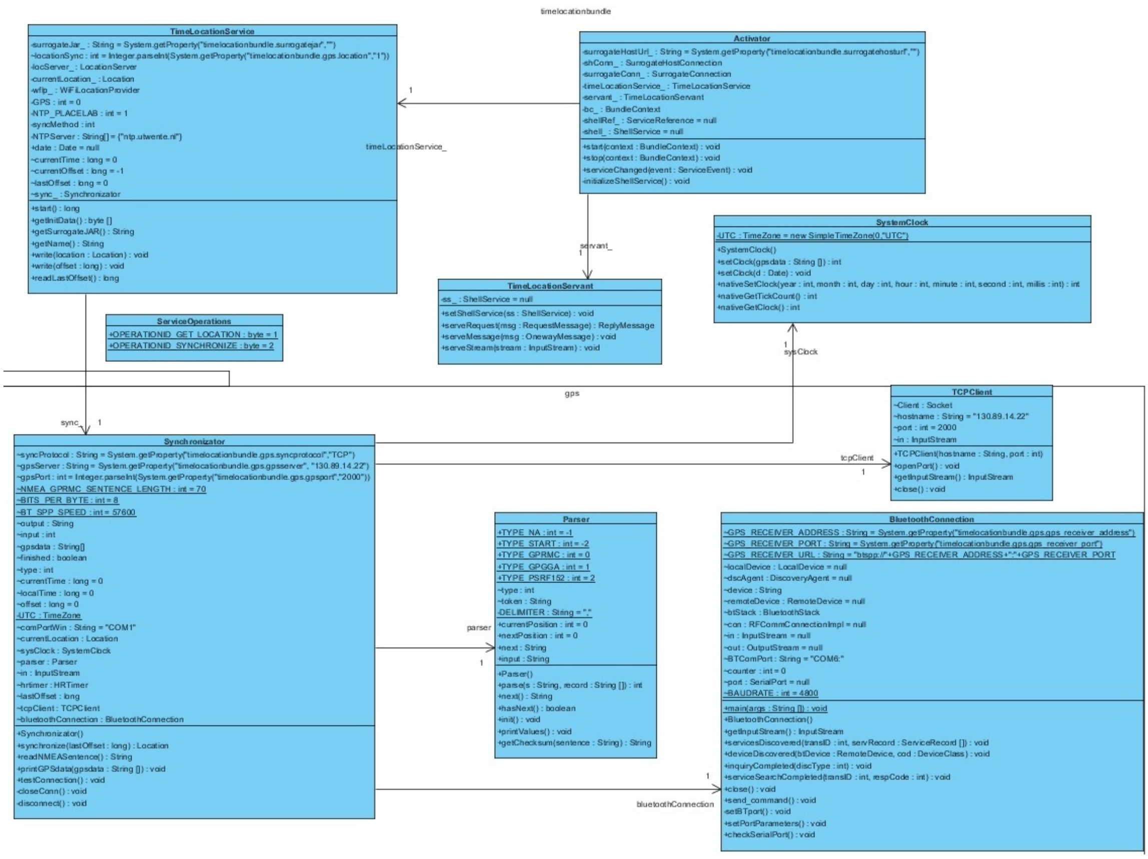

Figure D.1: Class diagram TimeLocationBundle

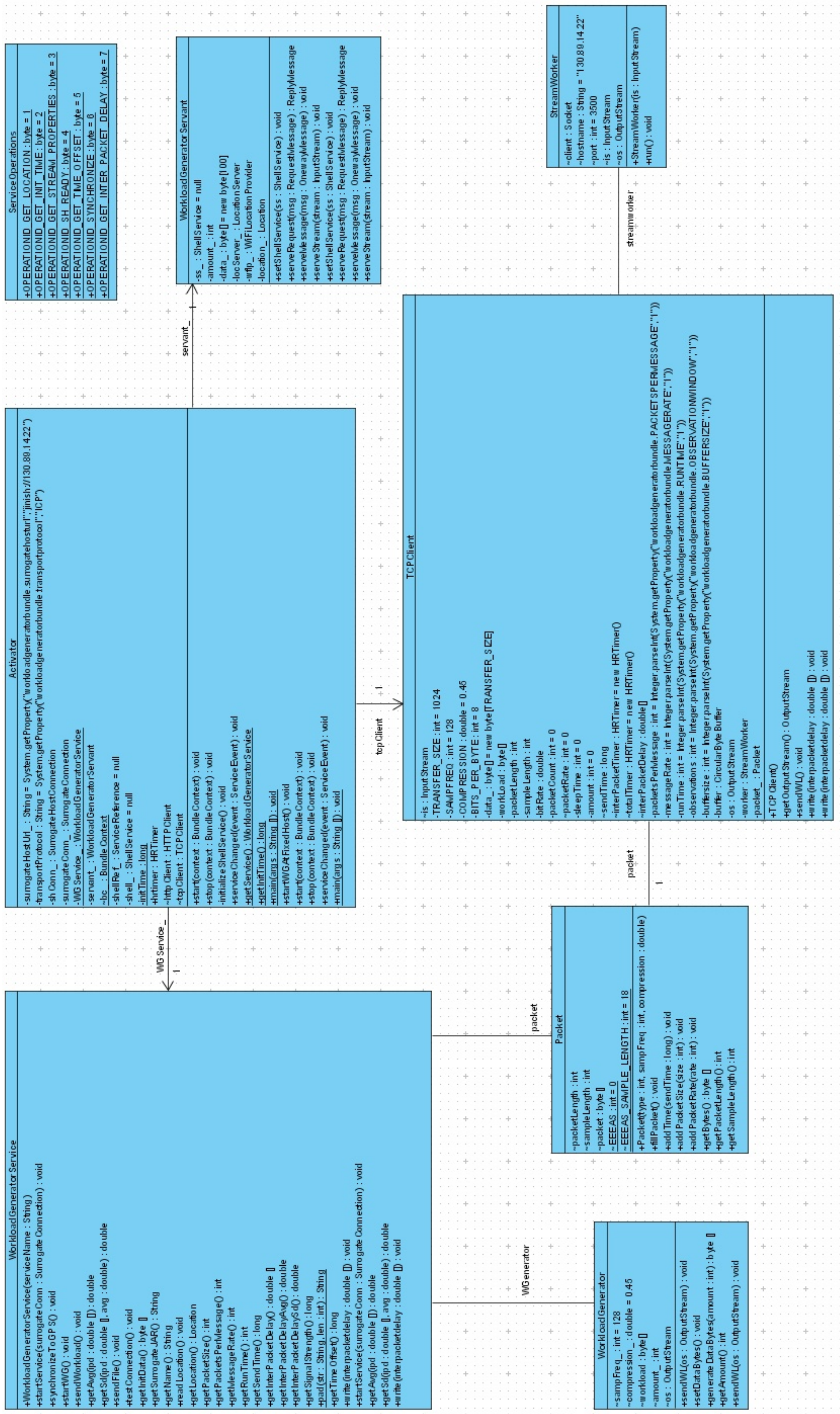

Figure D.2: Class diagram WorkloadGeneratorBundle

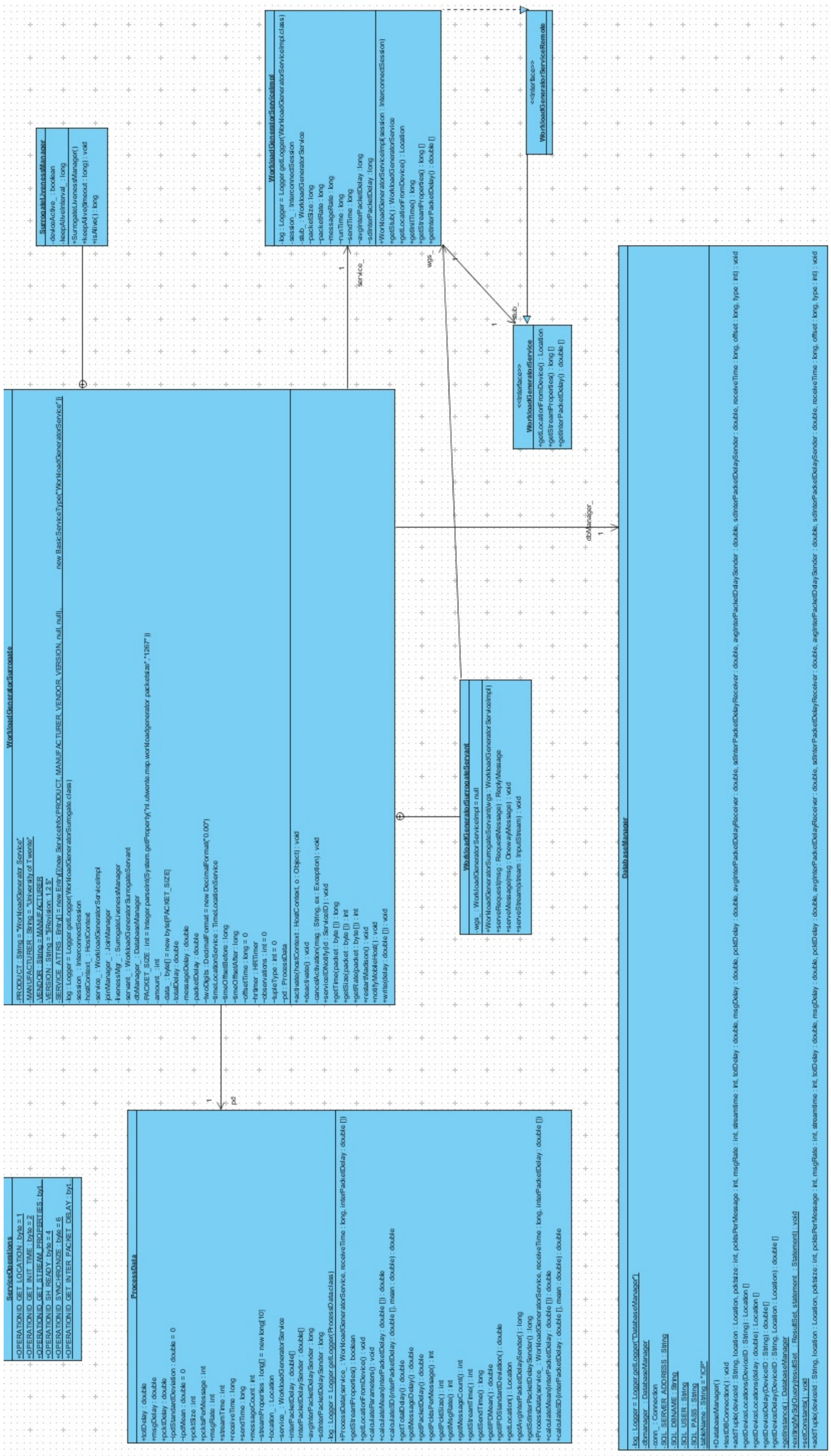

Figure D.3: Class diagram MeasurementFunction