Mobile Internet - Testing of Internet services

From Client to Server and the communication path between

MAGNUS FREDÉN

KTH Microelectronics
and Information Technology

Master of Science Thesis
Stockholm, Sweden 2005

IMIT/LCN 2005-07

Mobile Internet - Testing of Internet services: From Client to Server and the communication path between.

A Master of Science Thesis

MAGNUS FREDÉN

Master of Science Thesis
Stockholm, Sweden 2005

Abstract

In addition to the problems that may occur in the Internet, the people who test mobile internet services confront additional challenges due to the wireless environment. If something goes wrong, it is very problematic for a tester to determine exactly where the error occurred and the reason why it occurred. Is it the software in the telephone that does not work properly, is the GPRS link stable, is the core network of the Internet working as it should, or is there something wrong at the server? The staff at Sony Ericsson has major difficulties determining where the origin of a problem is. They often simply know that something is not working. The major effort is often to detect the source of the actual problem rather than solving it.

The PlayNow service provided by Sony Ericsson will be used as a reference service in this master's thesis project. The PlayNow service is currently used for distributing ring signals. It is built on Internet standards, but depends on GSM's specific features both for distribution and for e-commerce which makes it hard to test using existing web test tools.

This Master's Thesis examines the difficulties in establishing robust IP traffic links between a server and a mobile telephone. It examines the entire communication path between the mobile telephone and the server, and focus on how to detect (potential) errors. Within this project, development of a test tool is included. It supports features necessary for testing the reference service PlayNow.

This report covers many different technologies within the mobile internet such as GSM, GPRS, and mobile IP. Moreover, the normal infrastructure of the Internet will also be considered; as well as how the different parts interact in an environment consisting of a mobile Internet with GPRS. A major part of the project has been evaluation of the communication link between a mobile device and a server. Additionally, some efforts have focused on developing the new test tool.

This report explains the difficulties in finding the origin of problems in the communication path, which can be very difficult to detect. To make troubleshooting easier a test tool has been developed. With help from this newly developed tool, many configuration failures have been discovered and rectified.

Sammanfattning

Utöver de problem som kan uppstå på Internet, konfronteras testare av mobila tjänster av ytterligare utmaningar på grund av den trådlösa miljön. Går något fel kan det vara väldigt svårt att lokalisera var problemet har uppstått och anledningen till varför det uppkom. Är det mjukvaran i telefonen som inte fungerar, är GPRS anslutningen stabil, fungerar kommunikationen över Internet som det ska, eller är det något fel med applikationsservern? De anställda på Sony Ericsson har många gånger stora problem att definiera vad källan till ett problem är. Ofta vet de bara att något är fel. Den stora mödan är ofta att hitta källan till problemet snarare än att lösa det.

Tjänsten PlayNow som tillhandahålls av Sony Ericsson är referenstjänst för detta arbete. För närvarande används PlayNow för distribuering av ringsignaler. Tjänsten bygger på Internetstandards, men är även beroende av GSM specifika delar för både distribution och e-handel, vilket gör det svårt att testa denna tjänst med vanliga testverktyg för webben.

I detta examensarbete undersöks svårigheterna med att etablera robust IP trafik mellan en mobiltelefon och en server och fokuserar på hur potentiella fel kan upptäckas. Inom ramen för detta arbete har ett testverktyg utvecklats. Detta verktyg tillhandahåller funktionalitet för att testa referenstjänsten PlayNow.

Denna rapport behandlar många olika teknologier inom mobilt Internet, såsom GSM, GPRS och Mobil IP. Utöver detta har den vanliga Internet teknologin behandlas. Så som de andra delarna som är inkluderade i mobilt Internet med GPRS. En större del av arbetet har varit utvärdering av kommunikationsvägen mellan klient och server. Utöver detta har en del ansträngning lagts på att utveckla det nya testverktyget.

Denna rapport visar på de svårigheter i att hitta källan till ett problem i kommunikationsvägen, vilka kan vara mycket svåra att hitta. För att göra felsökning enklare har ett testverktyg utvecklats. Med hjälp av detta nyutvecklade testverktyg, många brister i konfigurationen av PlayNow servern har blivit upptäckta och tillrättade.

Table of Content

1 Introduction 1
1.1 Problem description 1
1.1.1 Causes of failures 1
1.1.2 Problem outside of Sony Ericsson's control 2
1.2 This thesis 3
2 Background 4
2.1 What others already have done 4
2.2 Existing test tools 4
2.3 Existing applications for testing a communication link 4
2.3.1 Ping 4
2.3.2 Traceroute 5
2.3.3 AutoMMI 6
3 Technical overview 7
3.1 Mobile infrastructure 7
3.1.1 Global System for Mobile 7
3.1.2 General Packet Radio Service 8
3.1.2.1 GPRS system architecture 9
3.1.2.2 Functionality of a mobile Internet terminal 9
3.1.3 Enhanced Data rate for GSM Evaluation 10
3.1.4 Short Message Service 10
3.2 Internet infrastructure 11
3.2.1 IP protocol 11
3.2.2 UDP 11
3.2.3 TCP 11
3.2.4 HTTP 12
3.2.5 XML 12
3.2.6 XHTML 13
3.2.7 WAP 13
3.2.7.1 Differences between WAP 1 and WAP 2 13
3.2.7.2 User Agent Profile 14
3.2.7.3 WAP Proxy 14
3.2.8 Akamai – Physical distribution of the application servers 15
4 Two different services from Sony Ericsson 16
4.1 PlayNow 16
4.1.1 Communication between the telephone and the PlayNow server 16
4.1.2 Payment with premium SMS 18
4.1.3 The problem with customization parameters 19
4.1.4 Test scenario for a new telephone release 19
4.1.5 The future of PlayNow 20
4.2 Internet Services 21
4.2.1 Test scenario for new telephone releases 21
4.2.2 Problem Statement 22
5 The communication chain 23
5.1 Failures in the communication path 23
5.1.1 Temporary failure 24

5.1.2 Configuration failure ... 24
5.1.3 Application specific problems ... 24
5.2 Testing the communication path ... 24
5.2.1 Test of Telephone ... 25
5.2.2 Test of Wireless link ... 26
5.2.3 Test of external IP network (Internet) ... 26
5.2.4 Test of PlayNow application server ... 27
6 PlayNow Test Center application ... 28
6.1 Communication with the PlayNow server ... 29
6.1.1 A correct response ... 29
6.1.2 A free list ... 29
6.1.3 An error message ... 30
6.2 Different ways of working with the application ... 30
6.3 Graphical user interface and user interaction ... 31
6.3.1 Facilitating simple testing ... 31
6.3.2 Creating test cases for automatic testing ... 32
6.3.3 Creating test cases from a file ... 32
6.4 Programming ... 32
6.4.1 Servlets ... 32
6.4.2 Applets ... 32
6.4.3 Java Script ... 33
6.4.4 HyperText Markup Language ... 33
6.4.5 Java to Java Script ... 33
6.4.6 Java ... 33
7 Conclusion ... 34
8 Future work ... 36
9 References ... 37
Appendix A – Excel file ... 40
Appendix B – Manual for PlayNow Test Center ... 43
Introduction ... 43
Manual Testing ... 43
Automatic Testing ... 44
Setting up test cases in the GUI ... 44
Create a test Case ... 44
Create a combination ... 45
Setting up test cases from file ... 45
How the application interpret the spreadsheet. ... 46
Upload a file ... 47
Fast testing ... 48
Running a test case ... 48
Clear a test case from earlier statistics ... 48
To see ongoing test cases ... 48
To see the results from a test case ... 48
To see the interpretation of an error message ... 49
Administering the database ... 50
Adding parameters to the database ... 50

Removing parameters from database 51
Testing with a mobile telephone 51
Limitations 51
User Instructions 51
Add user agent string 52
Appendix C – Abbreviations 53

Figures

Figure 1: Traceroute program from the command prompt. 5
Figure 2: Mobile Internet from a mobile user's point of view 7
Figure 3: A simple GPRS routing example. The MS is in a visited GPRS network 10
Figure 4: WAP 1 and WAP 2 Protocol Stacks with Proxy. 14
Figure 5: User interface of PlayNow 16
Figure 6: Test scenario for PlayNow prior to a new mobile telephone release 19
Figure 7: One possible extension for the next version of PlayNow 20
Figure 8: Test scenario for "Internet Services" before a new mobile telephone release .. 21
Figure 9: Overall picture of the complete communication path 23
Figure 10: Communication with PlayNow Test Center 28
Figure 11: Download Descriptor for a free ring signal 30
Figure 12: PlayNow Test Center as a proxy 31
Figure 13: The PlayNow Test Center GUI 43
Figure 14: Part of GUI for creating test cases and combinations 45
Figure 15: Admin GUI 50
Figure 16: GUI of the PlayNow Test Center application from a mobile telephone 51

Tables

Table 1: PlayNow Customization Parameters 17
Table 2: Service configuration specification Sweden rev PA6.xls 40
Table 3: A simple Excel spreadsheet 46

1 Introduction

Internet connectivity is becoming more and more important for mobile telephony users. This connectivity is not simply a fun feature of expensive smart phones anymore. Today most mobile telephones provide some sort of Internet connection. For many users it is a requirement to work efficiently. For others this connectivity gives them new features in their gaming and entertainment applications. Sony Ericsson is developing many services where Internet connectivity is a fundamental requirement.

For a normal user there is a huge cloud between his mobile phone and the server it connects to. This cloud consists of a lot of components. For a specific service to work properly all of these components must work correctly.

In the recently created Sony Ericsson department End to End (E2E) solutions, employees are developing new services for the company's telephones. These services range from downloading music and uploading high scores in a java game to many other Internet services. The common factor in these services is a mobile device connecting to the Internet.

1.1 Problem description

There is a continuous need to test these services since changes in the overall environment happen frequently. Many of these changes may affect the services; e.g. some parameters in the server change, server software is upgraded, or a new version of the software is placed in the telephone. Everything must work together with the existing infrastructure.

One major task of this master's thesis project is to come up with a solution of how to handle problems that might occur in an operator's network. The different operators tend to each have their own solutions. Even though the staff at Sony Ericsson have tested the functionality of a service in one operator's network, i.e. Telia, Vodafone, or Comviq, it does not mean that it will work in all other operators' networks around the world. This is because different standards have been deployed in different ways. Even though a standard is meant to standardize, these standards have been interpreted in different ways. Sony Ericsson's experiences are that this problem is greater when it comes to the newly introduced standards, such as Wireless Application Protocol (WAP) [39].

1.1.1 Causes of failures

There are many reasons why a service may not work for the end user. Here are some examples:

- The mobile telephone's software does not support the service.
- The mobile telephone does not have a GSM/GPRS connection.
- Some parts in the GPRS network do not work as needed for this service.
- A WAP proxy does not work as need for this service.
- Network equipment along the communication path, such as routers or switches is defective.

- The application server is defective.
- The server and mobile telephone do not work together. For example, the server does not support a new telephone model.

Of course, all of these problems can not be solved. If a user does not have a connection with a base station in the GSM network (and no other interface for connecting too Internet is available), then it is impossible to connect to the Internet. Additionally, some causes are more likely to occur, but others are more uncommon.

1.1.2 Problem outside of Sony Ericsson's control

There are some serious problems with the services that are out of Sony Ericsson's control. The common reason is that operator's solutions do not work together with a mobile telephone from Sony Ericsson. Here are some examples of problems:

- In PlayNow an operator's HTTP proxy removed everything after the question mark in the URI (See more about the specifics of this URI in the section about PlayNow). This means that the server will receive a request **without** any parameters. The server will not accept this request and replies with an error message.
- An operator's WAP gateway was configured for a maximum transmission size of 90 K byte. However, the latest platform for the Sony Ericsson telephones supports up to 200 K byte. Unfortunately, if the content delivered is greater than 90K byte, it is impossible to download the content.
- The time to establish a connection to the GPRS network takes too long time. Hence, the service timed out before it was able to connect to the server.
- When a mobile terminal tried to connect to the WAP server, the service User Agent Profile file was not in a suitable format for the server. In the WAP standard there is a specification for the format of a User agent profile, but it is not clear how to interpret it. This is a general problem for **all** mobile telephone vendors, not only Sony Ericsson.

All of the problems that can occur in an operator's network are very problematic for Sony Ericsson. Not only is the problem outside Sony Ericsson's control, but troubleshooting is very difficult. Before the cause of the problem is found, it is often very difficult to find it. However, the operators often think the problem lies within Sony Ericsson's terminals, although this may or may not be true.

Sometimes it can be very difficult to know how long a problem has occurred, before it is detected. Every time a server is configured for a new mobile device, regression testing for all older models is not always done.

1.2 This thesis

This thesis examines aspects regarding telephony and computer communications. The reader is assumed to have some knowledge about internet communication techniques and the associated protocols.

This thesis will not go in to detail about all the different protocols. Readers are assumed to know more about TCP/IP than other communication standards; therefore, this report will discuss the other technologies, such as GSM/GPRS in more detail.

2 Background

2.1 What others already have done

Within each specific subsystem in the communication link there have been major efforts by developers and scientists [7, 9, 10, 11, 12, 14, 16]. Many reports have been written about how to increase performance and robustness of the network. However, little has addressed the complete communication path and how to troubleshoot it when something does not work.

2.2 Existing test tools

Today there are several internet testing tools that analyze system behavior and performance. However, when looking at the additional requirements for mobile internet testing there are no satisfactory tools on the market today. Mobile telephone specific features such as SMS and the GPRS link add complexity to the communication path and the services to test.

Some tools for testing the complete IP communication path exist, such as traceroute and ping. However, because mobile networks such as GPRS are not pure IP networks these tools are not always sufficient. Moreover, if the communication link seems to work with traceroute and ping, there is no guarantee that it works with the services developed by Sony Ericsson (see the section 1.1.2).

The different operators use different systems to monitor their networks and servers. From Sony Ericsson's perspective this is **not** helpful. The problem for Sony Ericsson is to find out if a service for a new telephone model works together with the operators' existing networks. Because it is a new telephone model, which has not connected to this network before, there might be a problem that has not been recognized by the operator yet.

2.3 Existing applications for testing a communication link

2.3.1 Ping

A widely used communication link test tool is ping. It is a command line tool and is executed from the command prompt. Ping is a simple network test tool. It sends an ICMP echo packet to a source. The destination host replies to the source address by echoing the same content that the source host sent. If the source host gets a reply from the destination host it knows the communication link between the source and destination works correctly at the IP layer.

It is possible to install a ping program on a smart phone. Most programs are developed for Palm OS and Pocket PC, but there is a program, newTELnet P800 which is developed for Sony Ericsson P800/P900/P910. It is mainly used for Telnet, but it also has ping functionality. With this program it is possible to determine roundtrip time between a server on the Internet and the mobile device.

2.3.2 Traceroute

If the path is broken somewhere there is another tool called traceroute that may be helpful. With traceroute a user can examine the path for IP packets to a given destination. If the link is broken, traceroute will find the failing link. Like ping, it is included in almost all operating systems. Commonly, traceroute can be executed from the command prompt. There are some traceroute programs with a more user friendly GUI, but the underlying technique is based on the same method as the one executed from the command prompt.

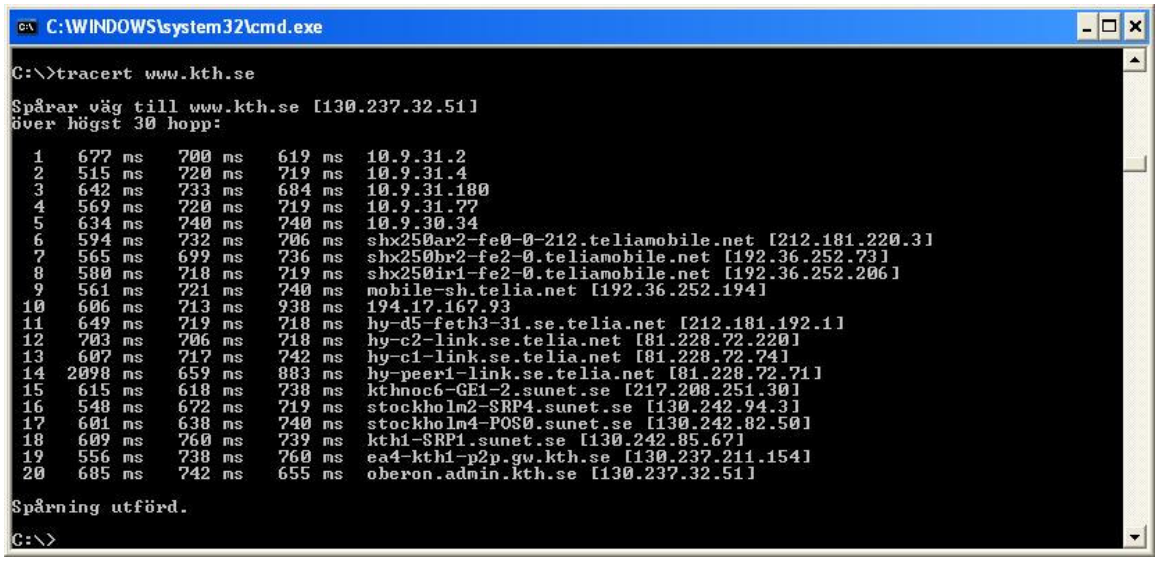


Figure 1: Traceroute program from the command prompt.

Every time a router forwards an IP packet it reduces the time to live field in the IP header by one. If it receives an IP packet with the Time to live field equals to one it throws the packet away and sends an ICMP reply back to the source that the IP packet did not reach its destination.

Traceroute takes advantages of this behavior. If the source (the traceroute program) sends an IP packet to a destination with the time to live field equal to one, the first router that receives the packet discards it because the time to live has expired. The router sends a message to inform the source. This message includes information about the router, including its IP address. If the source sends another IP packet with a time to live equal to two the first router accepts the packet and forwards it. However, the second router along the communication path will not accept the packet because the time to live value is one. The second router sends a reply to the source informing it that the IP packet did not reach its destination. By slowly increasing the time to live value in the IP header traceroute determines the whole IP communication path between the source and destination. This works under the assumption that this path changes slowly.

Ping and traceroute are useful testing tools of a communication link. Using a GPRS telephone, a computer can trace the link between the mobile device and the server. However, the GPRS network look like a single hop to end devices.

Ping and traceroute can be helpful when troubleshooting link errors. However, the problem for Sony Ericsson is often at the application level. Experience has shown that even though the communication link works with ping and traceroute, the service still does not work. See the section about problems outside Sony Ericsson's control (See section 1.1.2). Moreover, ping and traceroute are **not** included in current mobile telephones' operating systems. Therefore a computer must be connected to the mobile telephone in order to test the complete communication path. If the problem is an *unstable* link, where the radio connection is very weak, it is difficult to measure this with ping and traceroute.

2.3.3 AutoMMI

AutoMMI is a tool developed by Tieto Enator for Sony Ericsson. AutoMMI is a tool for controlling a mobile telephone with AT commands. This tests a mobile telephone using a script that executes multiple tests of a mobile telephone. Auto MMI can perform all the same operations as a mobile telephone user.

3 Technical overview

For Internet communication with a mobile device there are several different network architectures that the communication link must pass through. These can be grouped into two different groups:

- Mobile infrastructure
- Internet infrastructure

3.1 Mobile infrastructure

An Internet connection from a mobile device is somewhat complicated. Figure 2 is the user's view of mobile Internet, but reality is much more complex than this.

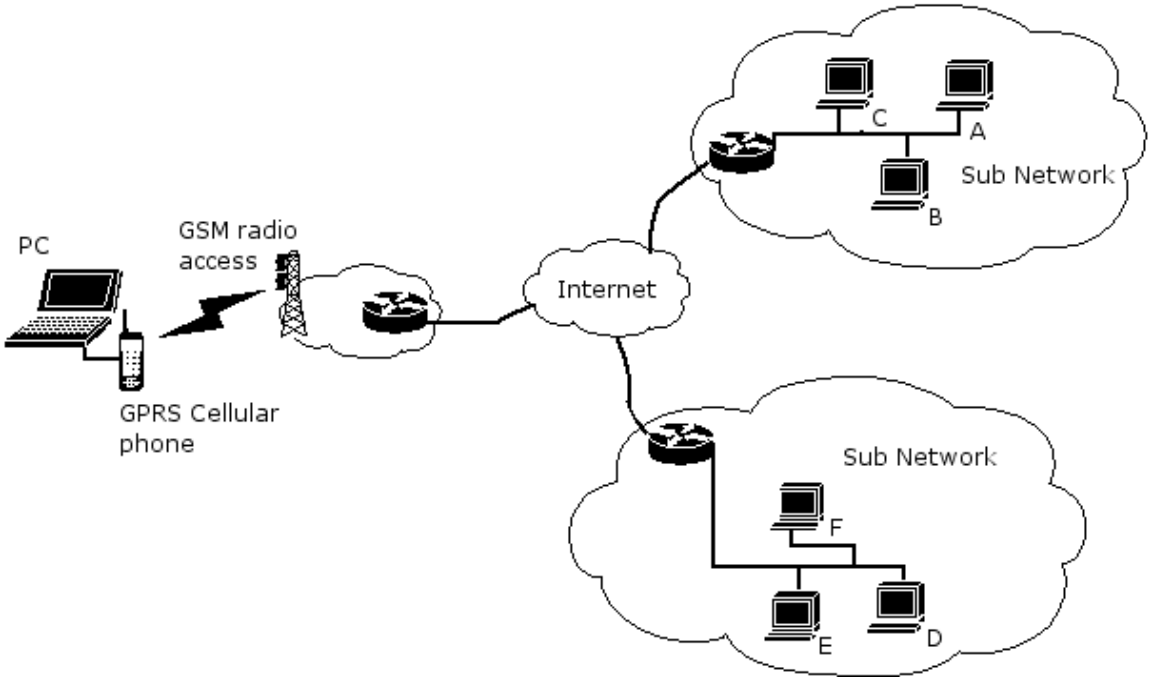


Figure 2: Mobile Internet from a mobile user's point of view

The complexity is due to the wireless environment and the fact that the user is **not** at a single physical location, but can be at multiple locations within the cellular network over time.

3.1.1 Global System for Mobile

The Global System for Mobile (GSM) communication is widely deployed all over the world. Sometimes GSM is called a second-generation mobile telephone system. GSM is a successor to the analog Nordic Mobile Telephony (NMT) system. GSM is based on digital communication using Time Division Multiple Access (TDMA). An active mobile telephone can reserve one or more timeslots out of eight slots in a specific frequency channel. Using one timeslot in each frame the communication link can provide 13 Kbit/s, which is sufficient for transmitting encoded voice data. However, this speed is often insufficient when connecting to the Internet for other applications.

This report will not cover all the details about GSM. Additional information is available in Swedish in a book called “Mobil Radiokommunikation” [1]. There are also numerous articles and books about GSM [1, 6, 16].

3.1.2 General Packet Radio Service

As circuit switched networks were designed for carrying voice traffic, where a steady but limited bandwidth is required, General Packet Radio Service (GPRS), is an extension to the GSM mobile telephone standard. GPRS adds packet switching to the GSM network. GSM’s limited circuit switched bandwidth makes GSM a poor solution for computer communication. Additionally, computer communications’ burst characteristic means that there normally is no need to reserve bandwidth for *continuous* communication. GPRS extends the capability of the GSM standard by providing an underlying packet switched service.

GPRS technology allows one device to use more than a single time slot in the frame. This provides a higher data rate when connecting to a data network (e.g. Internet). Moreover, these timeslots are *only* used when the device communicates; there is no long-term bandwidth reservation. Hence, many devices can share the available timeslots and the useful capacity of the network is increased. Because there is no bandwidth reservation, the user only pays for the amount of sent or received data and **not** for their connect time.

Additionally, GPRS traffic shares timeslots with circuit switched GSM traffic. Since a telephone call is sensitive to interruptions and the operator’s profit for a telephone connection is much greater than GPRS traffic, therefore most operators configure their network to give GSM circuit switched traffic (strict) priority over GPRS traffic. Thus, after the GSM traffic has been assigned its timeslots, then the timeslots available for GPRS are dynamically allocated between active terminals. Therefore the throughput of a GPRS network can vary depending on how the network is used. When more GPRS users exist than free timeslots are available, the available timeslots must be shared between active users.

Depending on radio conditions, GPRS uses different coding schemes. There are four different coding schemes, CS-1 to CS-4. Where CS-4 is used under extremely good conditions and CS-1 is used for poor conditions. However, in most installations only CS-1 and CS-2 are actually used. For more information about this topic see [13].

A mobile terminal is classified according how many time slots it simultaneously can use. A typical value today is “4+1” which means that the mobile telephone is capable of receiving data in *four* timeslots and sending in *one*. Assuming that CS-2 coding is used and the classification is 3+1, a maximum download rate of 40.2 Kbit/s and an upload rate of 13.4 Kbit/s may be available [11].

3.1.2.1 GPRS system architecture

Two major components were added to the GSM network to provide packet switching technology:

- Serving GPRS Support Node (SGSN). This is the main component in a GPRS network. The SGSN provide routing within the SGSN service area and provides the signaling to make mobile IP possible within the GPRS network. The SGSN maintain information about the different GPRS Mobile Stations (MSs) that are currently connected to the network within a given SGSN service area.
- Gateway GPRS Support Node (GGSN). The GGSN is the major element for handling GPRS traffic in conjunction with external networks. The GGSN acts as the interface to an external packet switched network, such as the Internet.

An SGSN provides routing to and from the SGSN service area, which consists of a number of base stations while a GGSN act as the interface to external IP networks. In addition to these two new network elements, some existing parts in the GSM architecture must be upgraded to support GPRS.

- Base Station System (BSS) communicates with the mobile device at the physical and link layer of the OSI model. It consists of one or more Base Transceiver Station (BTS) and Base Station Controller (BSC), as described below.
- The Base Transceiver Station (BTS) consists of the actual radio equipment and some logic for separating GPRS related links from ordinary telephone calls. It communicates with its BCS and the MS. If forms part of the Base Station System (BSS).
- The Base Station Controller (BSC) controls communication via one or more BTSs. It handles all radio connection related details such as channel assignment, cell configuring data, frequency hopping and handovers. The BSC forwards the circuit switched telephone calls to the MSC and the Packet Switched traffic to the SGSN. The BSC is part of BSS.
- The Home Location Register (HLR) contains subscriber information. This includes which service area the subscriber is currently located. The information in the HLR is used by both GSM and GPRS subsystems.
- The Visitor Location Register (VLR) maintains information about currently connected Mobile Stations (MS's). A VLR maintains information about devices **within** the SGSN service area.

3.1.2.2 Functionality of a mobile Internet terminal

One of the big challenges when it comes to *mobile* Internet traffic compared to common Internet traffic is the fact that the communication terminal is not located at a single physical place. In order to solve this, the Home Location Register (HLR) maintains information about all mobile terminals, which belong to a specific SGSN's routing area. When the mobile terminal connects to a new SGSN, the HLR is informed. The HLR acts

as the fixed point for keeping information about each MS. For those familiar with mobile IP, this is similar to a Mobile IP Home Agent.

The GGSN is the gateway to the Internet. The GGSN can implement different address mapping schemes and handles address translations between the GPRS network and the external IP network.

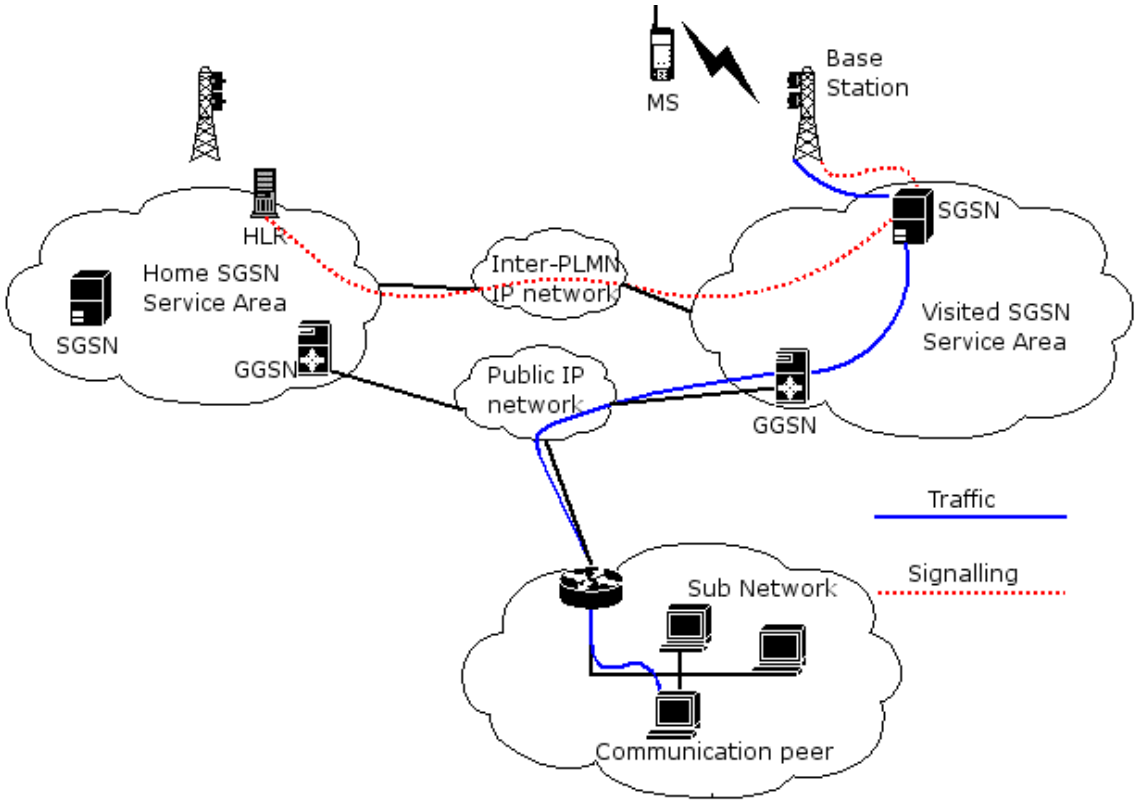


Figure 3: A simple GPRS routing example. The MS is in a visited GPRS network

3.1.3 Enhanced Data rate for GSM Evaluation

The introduction of Enhanced Data rate for GSM Evolution (EDGE) means that a new radio transmission technique is used to transfer data *during* a timeslot. With this technique the, maximum transfer rate is increased by more than three times. The future Sony Ericsson model Z500, will be their first device supporting EDGE technology in Europe. It will support up to 240 kbps bandwidth, which is almost four times the maximum GPRS rate today. According to internal documents at Ericsson Radio Systems [6], EDGE technology can provide data access rates of 473.6 Kbit/s or more. Thus GPRS can provide close to 3G performance. Some people even call GPRS with EDGE a third generation technology.

3.1.4 Short Message Service

Short message service (SMS) enables the user to send short text messages to another (mobile telephone) user. SMS was first developed in order for the operator to send short messages to a mobile device in their own network. The information could be used to alert

the user in some way, for example to notify the user that a new voice mail message exists. Another example is to inform a group of users in a given area about something, such as a traffic accident. In contrast to the original GSM designers' belief, it has become very popular.

Originally SMS was sent via the system's control channels. Thus, an SMS could be delivered **while** a telephone call is active. Each SMS message is limited to 160 characters. The reason for this is that the Signaling system No.7 (SS7) is used for SMS transfer. One packet in the SS7 has a static size of 140 bytes. Using seven-bit ASCII encoding allows for up to 160 characters.

3.2 Internet infrastructure

Communication on the Internet is based on packet switching. This infrastructure generally often only supports "best effort" communication. It is up to the end communication devices to make sure that all data is delivered, if they are concerned about this. Routers forward Internet packets as they arrive.

Within the Internet protocol stack there are a lot of different protocols. However, in this report only the protocols associated with the project will be mentioned. Here is a short description of the Internet protocols used in the services that E2E is concerned with.

3.2.1 IP protocol

The network layer protocol on the Internet is the Internet Protocol (IP). This provides unreliable end to end packet forwarding. On top of IP, TCP or UDP are the most common transport layer protocols.

3.2.2 UDP

The User Datagram Protocol (UDP) adds four more fields in addition to the data. These are source communication port, destination communication port, data length, and a checksum. These fields are used to demultiplex the data to the correct application, check how big the datagram is, and to test if the data has been delivered correctly. The communication is unreliable, i.e. packets may or may not arrive. UDP is often used in real time applications where retransmission is not useful. IP telephony is a typical application where UDP is used. If IP communication is used together with WAP, UDP can be the transport protocol.

3.2.3 TCP

Transmission Control Protocol (TCP) provides a number of mechanisms to make the communication more reliable. Additionally, TCP has mechanisms for controlling the sender's bit rate to avoid congestion and the resulting packet loss. TCP is used by most Internet applications, i.e., Web browsers, FTP, and Mail programs, to provide their reliable communication.

3.2.4 HTTP

The Hyper Text Transfer Protocol (HTTP) is the World Wide Web's application layer protocol. HTTP is used by a client (Web browser) to request the content from a server (web server). HTTP is used by the services PlayNow and Sony Ericsson's "Internet services". Therefore this protocol will be described more in detail.

HTTP Header

An HTTP header might look as follows:

```
GET /index.html HTTP/1.0
user-agent: SonyEricssonK700i/R2E SEMC-Browser/4.0.1 Profile/MIDP-2.0 Configuration/CLDC-1.1
Host: wap.sonyericsson.com
Accept: */*
Authorization: Basic U9ueUyraWNzcr8g29duLVVBds3XYcEQwOTk3MjE47NsRLNTA=
```

The example above shows the fields used when the terminal's PlayNow client connects to the PlayNow server. The HTTP enabled application asks for the file "index.html". The user agent field describes what kind of device, application, and operating system is being used to connect to the HTTP server. In this case it is a Sony Ericsson K700i mobile telephone with software release 1A. The host field is the name of the web site. The Accept field shows the file formats and encoding that the application can handle; in this case */* indicates "any".

In the HTTP standard there are additional optional fields. "W3Schools Online Web Tutorials" [38] describes the HTTP protocol more in detail.

HTTP basic authentication

HTTP provides several types of authentication. One of these is basic authentication. It is based on username and password. The messages that are sent over the network are encoded with uuencoding (Unix-to-Unix encoding) which is based on Base64 Encoding. There is no encryption and it is very easy to decode [35]. It keeps casual users, whom are not authorized from accessing the server's content. In order not to reveal the user's password, the authorization field above has been modified in this example.

3.2.5 XML

EXtensible Markup Language (XML) is a format to *describe* data, unlike Hyper Text Makeup Language (HTML), which describes how data is to be *displayed*. XML is used to transfer information in a structured way. XML is not intended to be specific, rather it is up to the developer to define what the tags (elements of the document which describe the data) stand for. The data is structured in such way that it makes it possible for both humans and machines to interpret the XML document.

3.2.6 XHTML

Extensible Hyper Text Makeup Language (XHTML) is based on XML. XHTML is successor of version four of the HTML standard [40]. XHTML is a way of describing how data is displayed. Web browsers are the most common type of application, which displays XHTML files.

3.2.7 WAP

Wireless Application Protocol (WAP) is a set of protocols specially designed for mobile communication conditions where these conditions exist:

1. The mobile devices have:
   - Weak CPU
   - Little memory
   - Limits on electrical power
   - Limited user I/O
2. The radio link provided has the following characteristics:
   - Narrowband
   - High latency
   - Typically burst errors (not packet loss due to congestion)

Within the WAP standard, there is a transportation protocol called Wireless Transaction Protocol (WTP), which was designed to be more suitable for radio link communication. The WTP protocol was designed to replace the TCP and HTTP protocols between the mobile device and the WAP proxy. Because TCP was designed for communicating in the traditional Internet environment, if packet losses occur TCP believes that the reason is congestion somewhere along the communication link. In wireless communication the most probable reason for packet losses is due to its loss over a radio link. Reducing the congestion window, as TCP does, does **not** reduce the packet loss rate, it only reduces the throughput. WTP uses delayed response to reduce the number of transmissions. The WTP protocol tries to optimize user interaction in order that information can be received when needed.

Within the WAP standard there is a Makeup Language called Wireless Markup Language (WML). It is built on the XML standard and is similar to HTML. The difference is that WML uses binary coding. Binary coding means that the WML script is encoded to a more compact form that is easier for the mobile telephone to decode. The reason for coding is to gain performance when transporting the information over a slow wireless link. Moreover, if the WML script has to be saved in the limited memory of the mobile device it will not take as much space as it would if the script was not encoded. The drawback is that the WML code is not readable in a normal editor.

3.2.7.1 Differences between WAP 1 and WAP 2

WAP 2 is backward compatible with WAP 1. In WAP 2 the ability to show XHTML pages has been included. The standard Internet protocol suite is supported in WAP 2, hence no WAP proxy is needed. However, a WAP proxy can enhance the communication bit rate. Figure 4 shows the protocol stacks for WAP 1 and WAP 2.

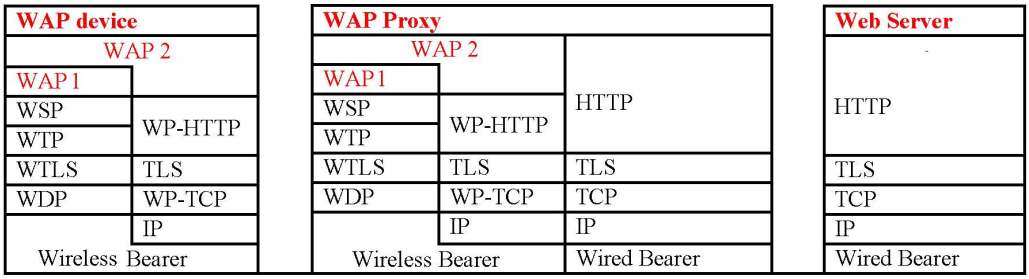


Figure 4: WAP 1 and WAP 2 Protocol Stacks with Proxy.

This figure shows some new concepts not mentioned earlier:

- The Wireless Session Protocol (WSP) provides HTTP functionality and incorporates some features for connection-orientated operations.
- Wireless Transport Layer Security (WTLS) provides functionality similar to TLS, but with poorer security.
- Wireless Datagram Protocol (WDP) acts much as UDP in the WAP protocol stack.
- Wireless Profiled TCP (WP-TCP) provides functionality similar to TCP, but adapted to wireless connection.
- Wireless Profiled HTTP (WP-HTTP) provides the functionality of HTTP. Additionally, WP-HTTP supports compression of responses and establishment of secure tunnels.

3.2.7.2 User Agent Profile

The User agent profile is used to reduce the communication to and from a mobile device. The User agent profile was specified in the WAP 2.0 specification [22] by the WAP Forum [39]. The idea is that a mobile device that connects to a server does not have to send all its attributes. A full User agent profile is approximately 10 Kb. 10Kb is quite a lot to send for a mobile device on a slow wireless communication link. Instead, the mobile device sends an URI to the location where the server can download the mobile device's user agent profile. However, today supplying 10 Kb is not difficult for the WAP server, which is generally located on a high capacity Internet link.

A User agent profile contains all the information concerning a mobile telephone's properties: Display resolution, number of colors the display is able to show, I/O interface, if the mobile is able to play Musical Instrument Digital Interface (MIDI) files, and so on.

3.2.7.3 WAP Proxy

Normally WAP communication goes via a WAP proxy to be able to access HTTP web servers. The proxy supports the mobile device by transforming messages to/from WML format for sending over a low bandwidth communication link. Moreover, the proxy can transform material made for workstations to a more suitable format for a mobile device with limited display area and limited input/output facilities. In the WAP specification

version 1.2 the communication **must** go via a gateway. However, in WAP 2.0 direct communication to the server is allowed.

3.2.8 Akamai – Physical distribution of the application servers

The mobile services from Sony Ericsson are based on physically distributed servers. This means that rather than a single server that handles all communication with the mobile clients there are a set of servers, which act as if they were a single server. Depending on different aspects, such as geographic location, server load and availability, a client will connect to one of many servers that provide the service.

Sony Ericsson uses a distribution server system provided from Akamai. [23] Akamai administers thousands of servers round the world. Many international companies use Akamai for distribution. The main feature of Akamai is that they provide reliable performance via their globally spread server park. The idea is that clients should connect to servers that are located physically near them. By doing this, the connection between the client and the server avoids many bottlenecks, such as peering points and highly loaded backbones. Avoiding these bottlenecks both increases the typical performance and reduces networking costs.

In addition to avoiding these bottlenecks, Akamai utilizes advanced algorithms for DNS lookup. Akamai utilizes information about server load and network stability to direct the clients to an appropriate server.

There are many advantages of using Akamai. First: Distributing the servers can reduce communication link costs. A mobile device in Singapore is likely to get a DNS reply for a server located in Singapore; while a mobile device in Sweden is likely to get a DNS reply for a server in or near Sweden. Secondly: If a server or a communication link to the server does not work, Akamai will redirect the user to another server, which is working. However, the complexity increases if something does not work. Thus, troubleshooting will be more difficult. For example, sometimes one mobile device gets a correct response, while another identical mobile get an error message. The reason for this can be that the two were directed to two different servers.

In some cases the Akamai DNS lookup takes longer than an ordinary DNS lookup. Therefore, the service may appear to be slower to the user than going directly to the destination server.

4 Two different services from Sony Ericsson

In this thesis PlayNow is the reference service. For more general consideration of the mobile internet, Sony Ericsson's service "Internet Services" will also be considered in this report.

4.1 PlayNow

PlayNow is a service for buying ring signals over the Internet via a mobile telephone. By selecting an icon, named PlayNow from the telephone's menu, the user will be presented with ten different ring signals. Before buying the signal the user can listen to it once before he/she decides whether to buy it. If the user decides to buy the ring signal, the telephone sends a premium SMS (a SMS which costs 10 kronor to send) to a SMS payment server. This SMS is sent in the background without notifying the user. This enables the user to be invoiced for the new ring signal. Once the payment has been authorized, a SMS is sent back to the user. When the telephone receives this SMS, the ring signal is released to the user and can be saved in the telephone's memory. Figure 5 shows the user interface for PlayNow. If the user does not buy the ring signal he/she will not be charged.

Figure 5: User interface of PlayNow

Sony Music distributes the ring signals. Every week the list is updated with new ring signals. The idea is that the content in the list shall contain the latest hit music. By frequently changing the content, users' curiosity will be stimulated and they will come back to the site again and again.

4.1.1 Communication between the telephone and the PlayNow server

When the mobile device first tries to initiate a session with the PlayNow server it sends a number of parameters in the URI. These parameters are called "Play now Customization Parameters". These include which language the user uses, the user's home operator and home country, which operator the user is currently connected to and in which country the user's mobile device is located. Information about the telephone model and which

software version is installed in the device are transferred in the User Agent String in the HTTP request header.

Table 1 shows a list of different parameters the mobile device sends to the server.

Table 1: PlayNow Customization Parameters

Parameter	Description	Example	Comments
Lang	ISO 639 language code	sv	Swedish
Mcc	Mobile country code (from SIM card)	240	Sweden
Mnc	Mobile network code (from SIM card)	1(x)	Telia
Cmcc	Current mobile country code (from current network)	240	Sweden
Cmnc	Current mobile network code (from current network)	1	Telia

Here is an example of an URI where the server has the IP address 192.168.0.2:

192.168.0.2?lang=SV&mcc=240&mnc=10&cmcc=240&cmnc=1

The fact that mnc is 10 and cmnc is 1 is **not** a typographical error. Why this is the case will be considered later in section 4.1.3.

Here is an example of a user agent string which is included in the HTTP header:

user-agent: SonyEricssonK700i/R2E SEMC-Browser/4.0.1 Profile/MIDP-2.0 Configuration/CLDC-1.1

The server uses these parameters and user-agent string to determine if the user is authorized to access the server and how to communicate with the mobile device. The server is configured for the different models and releases of telephones and knows how to communicate with them. Knowledge of attributes such as display resolution and I/O capabilities enables the server to adapt to the mobile device; hence information is presented in a suitable way. The server's response is a XHTML formatted reply, which is customized for the mobile device.

PlayNow is intended to be a feature of Sony Ericsson's telephones. To verify that it is a Sony Ericsson mobile telephone connecting to the server, the PlayNow server checks for the word "SonyEricsson" in the user agent field in the request header. Otherwise it sends an error reply. Moreover it uses the HTTP based basic authentication to make sure that only authorized users connect to the service. This might seem a little bit weak, but remember that the content that is sent is protected separately. In the PlayNow specifications, this authentication is based on HTTPS. However, many operators wanted the traffic to be transmitted through their WAP gateway. Unfortunately, it is not possible to use HTTPS (Hyper Text Transfer Protocol over TLS [25]) when the mobile uses WTP. To support all operators' networks, Sony Ericsson *decreased* the security level. With the solution today, it is up to the operators whether the WAP proxy must be used.

For each downloadable object the reply to the HTTP request contains a URI to an Open Mobile Alliance (OMA) [35] Download Descriptor (DD) for the actual ring signal. The DD contains information about the downloadable object. Here is an example:

```
<media xmlns="http://www.openmobilealliance.org/xmlns/dd">
<name> Sara Löfgren - Starkare </name>
<size> 10Kb</size>
<type> audio/midi </type>
<type> application/vnd.wap.drm.message </type>
<description>
Buy this ringtone 10SEK? Rules and conditions apply
</description>
<objectURI>
http://preplay.download.com/starkare.mid
</objectURI>
<installNotifyURI>
rsms:072533?body=Sara
</installNotifyURI>
</media>
```

The DD also contains information about how to obtain the ring signal. The tag <installNotifyURI> contains the number of the SMS payment server and which payment method is to be used (More information about payment methods is in the next section). A general specification for OMA's DD can be found at the Open Mobile Alliance homepage [35].

4.1.2 Payment with premium SMS

In PlayNow there are two different ways a premium SMS payment can be made. The first is called basic SMS. This means that the telephone releases the content without getting a confirmation SMS. The second is called roundtrip SMS. With the later solution the telephone waits for a confirmation SMS from the SMS payment server before it releases the content. By default PlayNow is configured with roundtrip SMS. If there are problems with the reliability of the SMS transport, basic SMS can be configured to handle the payment. This configuration is made in the PlayNow server.

SMS is an unreliable service, which means that problems with the payment may happen. If PlayNow is configured for roundtrip SMS there are two SMS that must be successfully transferred for the user to access the ring signal. It might happen that one of these is lost or delayed. If it is the SMS sent by the mobile device, then nothing happens except that the user gets annoyed. If the reply SMS is lost, then the user pays for the ring signal, but is still unable to download it to the telephones memory. Hence, the user gets very annoyed! If the PlayNow service is configured for basic SMS, then the requirement is limited to only one successful SMS. If this SMS is lost, then the user gets the ring signal without paying. Which method to prefer will not be discussed in this report.

However, loss of an SMS is regarded as a small problem. The probability for an SMS to be received by its recipient is considered as very high. Additionally, the ring signals are not very expensive – a lost SMS means that someone may lose 10 kronor.

The PlayNow server is configured as a trusted site by the telephone. A trusted site has the privilege to instruct the telephone to send a SMS, in the background, without notifying the user. Clearly, this is not the case for other web servers. The PlayNow Server is considered to be a trusted site because the URL is hard coded into the mobile phone and it is launched from the native menu.

4.1.3 The problem with customization parameters

There is a bug in all Sony Ericsson's telephones when they fetch the parameters for home operator (mnc) from the SIM. Despite the fact that an operator code can contain two or three digits the mobile telephone always receives three digits from the SIM card. For Telia, which has an operator code "01", the mobile receives the digits "01X", where X can be anything form 0 to 9.

If a parameter value has a leading zero it is removed before it is included in the URI. The resulting URI after the question mark for Telia with Swedish as language would be: lang=SV&mcc=240&mnc=1X&cmcc=240&cmnc=1. This means that there are ten different combinations that are valid for each operator that has a network code consisting of two digits, which is the majority of the operators.

Sony Ericsson produces a large number of different telephone models, each of which has many software versions. Together with the home operator network and country (mcc and mnc) and the current operator network and country (cmcc and cmnc) there are a lot of combinations to verify that the service works properly. Actually, there are about hundred thousand combinations of these parameters that *should* work! With new software versions and new telephone models the number is continually increasing. Obviously, all of these combinations are impossible to test manually. When the server is upgraded for a new mobile telephone, regression testing is not possible to do manually, due to the many test cases.

4.1.4 Test scenario for a new telephone release

Before a new mobile telephone is released the PlayNow service must be tested to see that everything works properly.

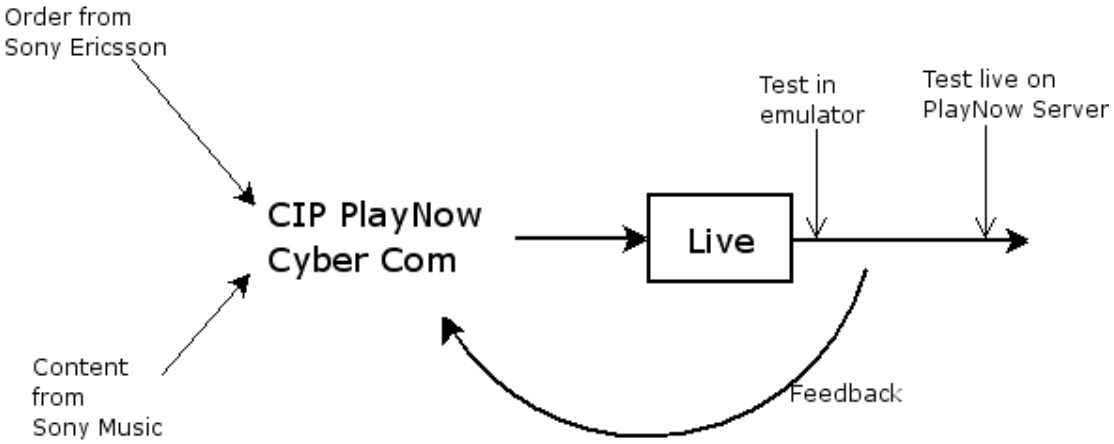


Figure 6: Test scenario for PlayNow prior to a new mobile telephone release

Cybercom, is the provider of the Consumer Information Portal (CIP) servers (i.e., the PlayNow servers), receives a request to upgrade their server for the new mobile telephone model. They make the changes in their production environment. In the beginning tests are done with a simple web emulator to see that the response is correct. Afterwards tests on site are done. This means that different persons located round the world test the service with the new model. These tests are done through different operator's network to see that the service works on the site.

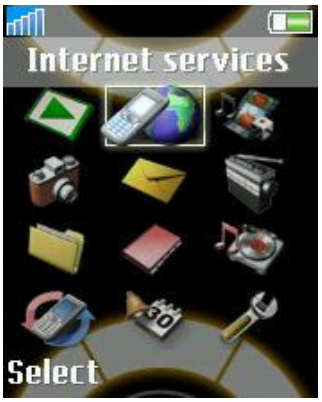
4.1.5 The future of PlayNow

At Sony Ericsson a lot of brainstorming is being done for the next version of PlayNow. Instead of only distributing ring signals, new content types are meant to be distributed with the next version of PlayNow. Full-length music, pictures, video clips, and games are discussed as complementary content types to the ring signals that are provided today. The payment with SMS will still be used for commerce.

Figure 7: One possible extension for the next version of PlayNow

Other related music services are also being discussed for inclusion in PlayNow. So far the PlayNow team at Sony Ericsson is only discussing which different ideas can be implemented, nothing final has yet been decided.

4.2 Internet Services



"Internet Services" which also is called "fun and download" is the name of a Service where a user can browse the Sony Ericsson WAP portal. Here it is possible to download fun stuff like sounds, pictures, animations, themes, and games. Some of these products cost money. Payment are made via premium SMS. The payments are not integrated as in PlayNow, since the Sony Ericsson WAP site is **not** regarded as a trusted site. Therefore the user must explicitly send the premium SMS.

When the mobile device first tries to initiate a session with the Sony Ericsson WAP server it sends a URI to a location where the server can find the user agent profile for the specific telephone model and software version (See section 3.2.7). The server uses the user agent profile to know how to communicate with the mobile device. Knowing attributes like display resolution and telephone model, enable the server to adapt to the mobile device. The information is presented in a suitable way.

4.2.1 Test scenario for new telephone releases

Before a new mobile telephone is released the "Internet services" must be tested to see that everything works properly.

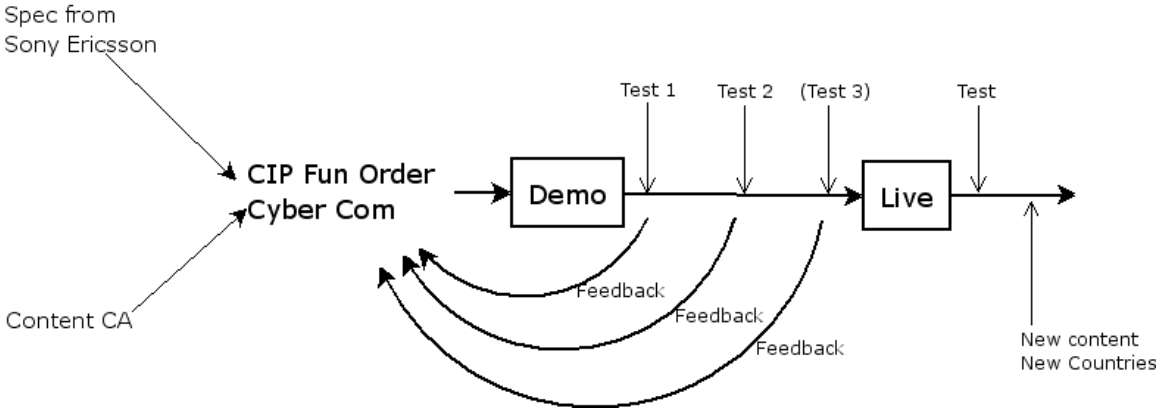


Figure 8: Test scenario for "Internet Services" before a new mobile telephone release

When testing for a new mobile telephone model, a demo platform is used. Usually, there are two or three test scenarios (Test 1, Test2, Test 3) before the service is fully configured for the new telephone. After everything works in the demo environment the configuration changes are made in the live environment.

4.2.2 Problem Statement

The Sony Ericsson WAP portal needs to work together with different telephone models, software versions, and languages. As with PlayNow it must work from many different operator networks – this means tiresome testing for employees round the world. There are many similarities between PlayNow and Internet Services.

5 The communication chain

Figure 9 shows the overall picture of the full communication path for the reference service PlayNow, from the mobile telephone to the servers providing the PlayNow service.

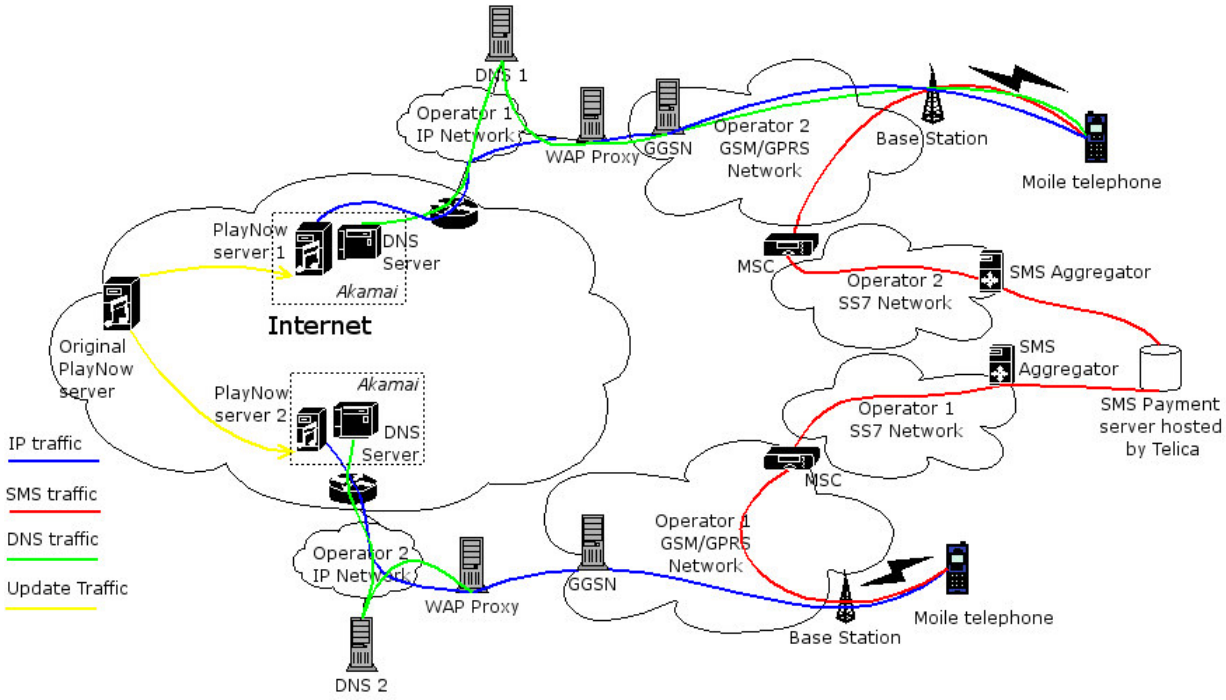


Figure 9: Overall picture of the complete communication path

The communication path can be separated into five different parts:

1. Telephone
2. Wireless link (GSM/GPRS)
3. Public IP network (Internet)
4. PlayNow Server.
5. SMS communication path.

Remember: For the complete communication path to work, all communication links must work in both directions. Additionally, since the PlayNow service is distributed, this means that there can be more than one server handling the requests.

5.1 Failures in the communication path

Some tools on the market can be helpful when testing the communication path. However, none of these are sufficient for complete testing of the complete communication path for PlayNow.

There are different kinds of failures that might occur:

- Temporary failure.
- Configuration failure.
- Application specific problems

They are discussed in more detail below.

5.1.1 Temporary failure

This kind of failure is the most common type of failure. Almost every mobile internet user has experienced this. A failure of this kind is often due to the wireless communication link. When a mobile internet user is located somewhere where the radio signal strength is weak the service can be disrupted. Similar to the situation when making a call with a GSM telephone: The mobile device must have sufficiently good radio contact with the base station.

Another type of temporary failure is when some equipment along the communication path is temporary out of order.

5.1.2 Configuration failure

Sony Ericsson must ensure that their new mobile telephone models work in the different operators' networks. Sometimes it happens that something does not work. Especially with telephone models that use new features that uses the network in ways that earlier telephone models did not. One example of this occurred when the maximum download size in the mobile device exceeded the downloaded size for the WAP proxy (see section 1.1.2.) The offending WAP proxy had to be reconfigured so it could handle the traffic from the new telephone.

5.1.3 Application specific problems

In order to customize the server reply for each mobile telephone models' features, the different services must be configured for these models. If these configurations are not made, or if they have been done incorrect, the server will not respond correctly to the telephone's request. These problems can range from the reply not looking nice in the telephone's browser to the telephone not getting a reply at all. The normal reply to a telephone that does not have a configuration in the server is a list of free ring signals. The reply looks like a normal list, but the content format is MIDI instead of mp3. Thus instead of receiving an error message, the user will *believe* that the service is operational even though they are only getting default context.

5.2 Testing the communication path

There are several different reasons why a mobile service does not work. These can be divided into one of two broad classes:

1. There is something wrong at the application layer.
2. There is something wrong below the application.

Viewed simply, this means that either something is wrong in the communication path between the two communicating devices, or something is wrong with the devices themselves.

To distinguish between the two is not always a simple task, especially when the tester uses only a mobile telephone for testing the service. Trying to browse the Internet can give some hints as to where the problem resides. If the tester can connect to another WAP server and browse the Internet, then he or she knows that the wireless communication link works. If the tester has a computer it is possible to use ping and traceroute for testing for IP connectivity. Thus, it is possible to make sure the server "is alive". If the tester has a computer connected to the Internet via a mobile GPRS device it is possible to test the whole communication path.

If the wireless link works with ping and traceroute, but the service does not work over the same link, then there is a problem. Although this sounds unlikely, it has occurred several times for the staff at Sony Ericsson. When a new telephone model is released for testing the two services, this telephone model has not yet been tested in all the different operators' networks. Unfortunately, the new telephone model might have some characteristics that are not compatible with the operators existing networks. Exactly what the problem is can differ from time to time. Section 1.1.2 illustrates some problems that have occurred.

A real challenge for the Verification Engineers is to find out who is responsible for a given problem. First they must verify that it is not the client or the server that is the source of the problem. As mentioned earlier it is generally the case that the faults are located at the endpoints of the communication chain. Secondly, after eliminating the client or the server as the source of the problem, these engineers must find the origin of the problem. This is difficult, because the staff at Sony Ericsson does not have access to the operator's network management systems. Additionally, the operator can be located somewhere very far from Sweden, where most of the staff from Sony Ericsson is located. Because other telephone models are working flawlessly in the operator's network, the operator does not want to believe that something is wrong in their network.

In the following sections, different characteristics for different parts of the communication path are discussed.

5.2.1 Test of Telephone

The telephone is a very complex device and is a major part in the communication path. There is a web browser inside the telephone that must work according to the WAP and Internet specifications. The GSM/GPRS radio interface must work as specified. All the protocol implementations must be accurate. There are a lot of different factors that **must** work in order for a mobile device to operate correctly.

When a new telephone is received by E2E, they are often prototypes and contain some bugs. When testing new telephones the people that verify the services have several SIM cards from different operators in order to test the service for different networks.

If the same problem occurs in all operators' networks, it is a clear signal that something may be wrong with the telephone. However, the problem could also reside at the server side. For example, because the telephone is not yet configured in the server, the server may respond with an error message (See section 5.2.4).

Most telephone related problems are seen during the development phase of new telephone models. If something suddenly does not work, which has been working earlier, it is seldom a telephone problem.

5.2.2 Test of Wireless link

In order for a service to work the mobile device must have a good radio connection. Thus loss of wireless communication is not an error when connectivity is temporary down, but rather that the user that does not currently have radio connectivity.

In order to connect to the GPRS network the telephone must be configured. Parameter values including: Access Point Name (APN), proxy IP, and parameters for authentication must be configured in the telephone. When configuring a telephone, it is easy for the user to make a typing error when manually entering the parameter values. However, most operators can send the configuration to the mobile via SMS. By visiting the operator's webpage, using your PC, you can specify your telephone model and telephone number. After submitting this information, the configuration SMS will hopefully arrive within a short time.

For a normal GPRS user it is difficult to test the communication links inside the GPRS network. It is possible to set the APN, which determines which GGSN output interface the traffic should be routed through. However, most GPRS operators give their subscribers only one APN value for accessing the Internet. Therefore it is not possible to change this value in order to redirect the traffic within their GPRS network.

There are many different elements that have to work in order to allow wireless communication to/from the Internet to work. If something goes wrong in the wireless link or in the internal GPRS network it is, as a user, not easy to locate the error. Traceroute works on the IP level and the GPRS wireless link will only be seen as one hop in the path as shown by traceroute. However, using traceroute from a mobile terminal makes it possible to determine that the error resides within the GPRS network.

5.2.3 Test of external IP network (Internet)

An IP packet is forwarded though several IP network elements on its way to the server. All of these must work in order for the communication to work. For a communication path from a client to a server there might be a dozen or more IP routers that the packet must traverse.

When testing IP networks, ping and traceroute are excellent tools for testing the communication path. In most cases the path works for the services if it worked with ping and traceroute. Experience from Sony Ericsson indicates that there is still a possibility that some specific communication may not work. However, those problems were due to

the WAP-proxy, which should be regarded as a mobile specific network element. Once an IP packet has reached the public IP network, it is unlikely that some mobile specific traffic should get stuck while other passes through successfully.

5.2.4 Test of PlayNow application server

One big challenge is to determine whether a problem resides in the mobile telephone or the server. Imagine the response "Service not available while roaming" when connecting to the PlayNow service for the first time with a new telephone model. It is impossible to know whether it is the client, the server or both that is at fault. One possibility is that the mobile telephone sends the wrong parameter values to the server. Another possibility is that the server interprets the parameters incorrect and therefore replies with an error message. A third possibility, which is rather small, is that the request from the mobile has been incorrectly manipulated by some network element along the way.

Most problems with the PlayNow service can be addressed from the server side. There are continuous updates of the server with ring signals and telephone models. In addition to this, sometimes there are new countries and operators added to the service. However, every time something is changed in the server there is a risk that something undesirable happens.

There are so many parameter combinations that it is almost impossible to test with a mobile telephone. Moreover, the combinations of countries, operator network codes, languages and User Agent Strings are getting so large that it is impossible to manually test the server with all combinations. What is needed is a tool for automatically testing all the combinations. Here is a statement from an e-mail correspondence between the people testing the service in Singapore. "As we add more models with different UA profiles into the PlayNow program, the required testing is increasing exponentially. For example, the K500i has 3 user agent profiles and 3 operators, there are 9 (3x3) combinations. With 2 models, there will be 18 combinations. This is extremely time consuming and we need to do this regularly due to the instability of the service." This statement points to one of the major problems when testing PlayNow.

Additionally, the physical server is not a single server, but many distributed Akamai servers. This means that it is difficult to test the servers at the IP level.

Testing of the server with the newly developed test tool "PlayNow Test Center" is described in the following sections. A manual of PlayNow Test Center can be found in appendix B.

6 PlayNow Test Center application

The staff at Sony Ericsson has major difficulties troubleshooting the reference service PlayNow. The lack of tools for testing PlayNow makes it impossible to do a full regression testing of the relevant telephones. When making changes in the PlayNow server, they assume that the old configurations still work. Many times, old configurations have simply vanished. Moreover, when a new list of ring signals are configured the only check that they work is via a WAP emulator.

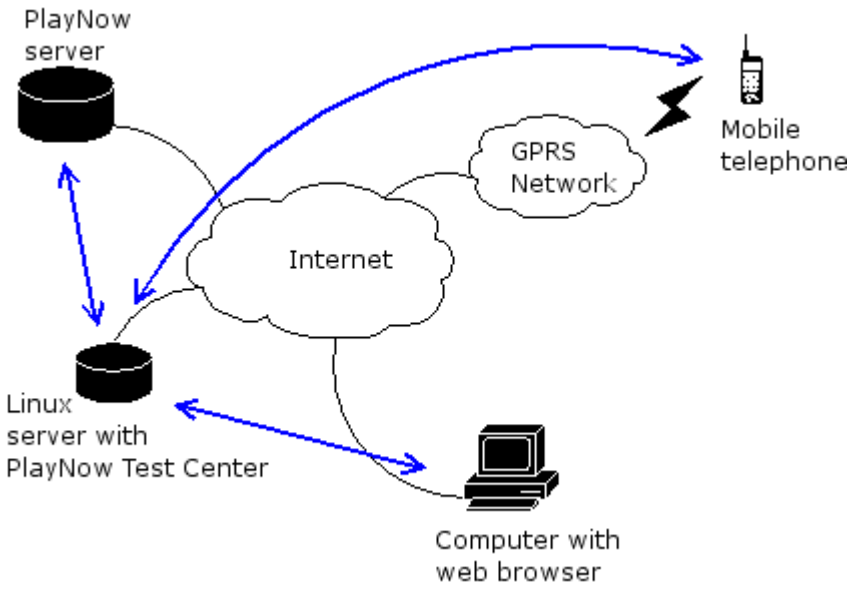


Figure 10: Communication with PlayNow Test Center

The development of PlayNow Test Center is a step forward in simplifying the troubleshooting of the PlayNow service. By testing the server directly via the Internet, it is possible to verify the server's response for different types of requests independent of the wireless part of the communication path. When troubleshooting, it is important to know the server's response, even though the failure resides somewhere other than the server. If the server gives a correct response, at least you can eliminate the possibility that something is wrong with the server.

It is possible to connect to the PlayNow Test Center both with a computer and via a mobile telephone. With the computer it is possible to do automatic testing and with the mobile it is possible to see the response for specific parameter values. We will examine some of the features of the PlayNow Test Center to gain familiarity with this application.

The PlayNow Test Center Graphical User Interface (GUI) is web based in order to distribute the application to the different users. At E2E they have a dedicated Linux server where the application is located. On this Linux server an Apache Tomcat 5.0 web server is installed.

The focus of the application is to generate requests with different parameters, connect to the PlayNow server, and analyze the responses. This can be done both manually and in an automatic way. By utilizing different test cases the application can generate many different combinations of parameters.

6.1 Communication with the PlayNow server

PlayNow Test Center is a web-based application. It is possible to access it from anywhere where an Internet connection is available.

The PlayNow Test Center application connects to the PlayNow server with a combination of parameters. Depending on the parameters there are many different responses that can be generated by the PlayNow server. Basically, the application groups these into three different categories:

1. A correct response. A XHTML response with a list of ten different mp3 ring signals.
2. A free list. A XHTML response with a list of ten different MIDI ring signals.
3. An error message. If the server is not configured for a combination of parameters it replies with an error message.

The PlayNow Test Center application handles these responses differently:

6.1.1 A correct response

If the response seems to be correct, then the PlayNow test center application tries to download the content for every link in the XHTML message. These links should be references to different download descriptors. Each of these contains information about a ring signal. If downloading of the download descriptors succeeds, it parses the URI to determine the location of the ring signal file. It downloads the ring signals and checks the size of the file as compared to the size tag in the download descriptor. This is done for all links in the XHTML message.
If the PlayNow Test Center application succeeds then it generates a “No error” message.

6.1.2 A free list

The XHTML response from the server with a free list looks very similar to a normal response. The difference is the titles of the songs and that the file format of the ring signals are MIDI instead of mp3. However, the download descriptor looks little different.

```
<?xml version="1.0" encoding="UTF-8"?>
<media xmlns="http://www.openmobilealliance.org/xmlns/dd">
  <name>Dream Land</name>
  <objectURI>http://wap.sonyericsson.com/client/preplay/media/68/8135.mp3;jsessionid=Ao3_D-JOg-
WsSFS450dB_DVboXWYASfcpICAmMOH</objectURI>
  <size>70623</size>
  <type>audio/mp3</type>
  <type>application/vnd.oma.drm.message</type>
  <description>This content is free</description>
  <installNotifyURI>http://wap.sonyericsson.com/client/preplay/notify/dm/free;jsessionid=Ao3_D-JOg-
WsSFS450dB_DVboXWYASfcpICAmMOH</installNotifyURI>
  <version>1.0</version>
</media>
```

Figure 11: Download Descriptor for a free ring signal

By checking the tag "<description>" the application is able to check whether it is a free list. If the PlayNow Test Center application finds this tag, then it stops testing and responds: "Reply From Server is a free Page".

6.1.3 An error message

There are a number of different errors that might occur depending on different circumstances. One error message is that the service does not support roaming; another is that the service is temporary out of order. If a severe problem occurs, then the server might not respond at all (The timeout value is 30 seconds). However, all of these incorrect responses are considered a failure by the PlayNow Test Center application. It generates a message "Reply From Server is not valid. ReplySize=X" Where X is the size of the reply in bytes. To see a more detailed description of the error, there is a special page where it is possible to see what each response means.

6.2 Different ways of working with the application

There are different ways of working with the PlayNow Test Center Application. One way, which is the most intuitive, is to select which parameters to send and examine the response from the PlayNow server. In this way it is possible to see the XHTML reply and the different download descriptors for the reply. If the Server replies with an error message it is possible to see this reply (from the server).

A second way is to create test cases from a pool of parameter combinations and then let the application test the different combinations. In this way we create test cases with many different combinations of the parameter values. Thus, generating thousands of different combinations results in a test that can take several hours. The test cases and the results from the tests are stored in a database. The results stored in the database can be used to analyze the server behavior.

A third way is for the application to read an Excel file containing the different parameters. The application analyzes the content and stores the different combinations of parameters in the database. As with the automatic testing method the results are stored in the

database. To get information about how PlayNow Test Center analyze the Excel sheet there is a section in appendix B “How the application interpret the spreadsheet”, which describes this in detail.

A final way is to connect to the PlayNow Test Center with a mobile telephone. In this way it is possible to test the server by selecting different parameter values. The results show the results as seen by in the mobile. This makes it possible to see some of the results from a foreign mobile without visiting the foreign country. This has not been possible to do before. Figure 12 shows when PlayNow Test Center application acts as a proxy between the PlayNow server and the mobile telephone.

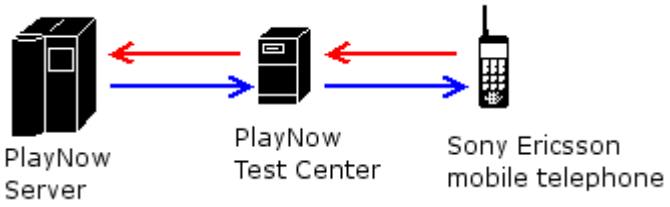


Figure 12: PlayNow Test Center as a proxy

When the PlayNow Test Center server connects to the PlayNow Server it saves the result locally. If the user wishes to see these results, the PlayNow Test Center application redirects the user to the locally saved copy of the result.

6.3 Graphical user interface and user interaction

6.3.1 Facilitating simple testing

There are seven different objects in the graphical interface for manual testing: six dropdown menus and one test button. The six dropdown menus represent the different parameters that are sent to the PlayNow Server. The button is pressed after the different parameters were chosen. When the test is done, the user is redirected to another page where the result is presented.

The GUI is dynamically changed. When the user chooses a value from a dropdown menu the relevant options in other menus are changed. For example: If Sweden is selected from the country menu, then it is only possible to select from the operators represented in Sweden (Telia, Vodafone, and Tele 2). Moreover, the values are changed as if roaming does not occur. PlayNow does not support roaming and it is unlikely that a tester frequently would like to test it. If home country selected is Sweden, the current country is also changed to Sweden. If the user wants to test roaming, then he must change the current country and current operator **after** selecting home country and home operator.

Because of the web interface the user interaction is not very good; however, most users are familiar with such web applications.

6.3.2 Creating test cases for automatic testing

Within the GUI there are objects called test cases and combinations. A test case consists of one or more combinations. A combination represents a combination of parameter values. A combination is created when the user makes lists of parameter values. The application creates all the possible combinations from the lists: Language, user agent string, and the parameter values for operators and networks.

As described in the section 4.1.3 there is a bug in the Sony Ericsson mobile phones. The telephone adds a random character to the operator value. Therefore it is necessary to test all ten combinations that might occur. For example: If Telia has operator code 1, then the possible values passed to the server are 10 to 19. Therefore the checkbox "All ten combinations" is included in the GUI. The user must be aware that the number of test combinations in the test case will be ten times greater.

6.3.3 Creating test cases from a file

Before a change in the PlayNow server takes place an Excel spreadsheet is created to specify which combinations should be configured. To test these combinations the PlayNow Test Center application reads this Excel spreadsheet and creates test cases from it. Appendix A shows an example of a file for the changes to the server configuration for Sweden. On the main web page of the application there is a section for uploading a file to the server. By pressing the "browse" button the user can select a file to upload. To upload the selected file to the server, the user presses the "upload" button. The application tries to find the columns, which are relevant in the Excel spreadsheet. Upon finding cells with the text for the relevant parameter values it finds the row and column of the parameter values. The PlayNow Test Center application assumes that all the values under this cell are parameter values for that specific configuration. The application parses the values in the cells and analyzes the results. If everything seems OK, the application creates a test case with these parameters.

6.4 Programming

The staff at Sony Ericsson wanted the application to be developed in Java with a web based Graphical User Interface (GUI). The programming uses a Model View Controller (MVC) based architecture. The model and the controller are based on java with a MySQL database for data management. The view is based on HTML, Java servlet, Java applet, and Java Script technology.

6.4.1 Servlets

The GUI is based on Java Servlets. Java Servlets are similar to the Common Gateway Interface (CGI), but are based on Java. Via a servlet it is possible to execute classes from java programs.

6.4.2 Applets

Applets are a java program which runs **locally** in the user's web browser. The PlayNow Test Center application uses an applet in the background to communicate with the server.

6.4.3 Java Script

JavaScript is a script language to enable dynamic web pages. In the PlayNow Test Center application Java Script is used to update some objects in the GUI.

6.4.4 HyperText Markup Language

HyperText Markup Language (HTML) is not a programming language, but rather is a markup format to show information via a web browser. HTML is the de facto standard markup language on the Internet. The GUI is based on HTML.

6.4.5 Java to Java Script

Java to Java Script (J2J) is a Java Servlet made by a team from Russia that calls themselves Coldbeans Software [29]. J2J makes it possible for a web browser to execute methods on server side. The technology is based on java script that communicates with an invisible java applet, which is running in the background. The applet forwards the request to the server side object. Using J2J the program can update fields in the web page *without* reloading the whole page. This is helpful, because there are objects in the GUI that are dependent on what is selected in other objects.

6.4.6 Java

Java is the most popular programming language today. There are several reasons why it has gained popularity since its introduction 1995. One reason is its platform independence. Another is its close relation to the Internet. When Netscape 2 (which became the most popular web browser) was introduced it had support for java applets, which is a part of Java. There are lots of sources to learn more about java. A good start is to visit the java home page [32].

7 Conclusion

When testing the communication between a new telephone and an application server, it is difficult to determine where an error is located. The problem is generally that both the telephone configuration and server configuration are new. Therefore either can be the source of the problem. In order to exclude potential sources of errors, a test tool for testing the mobile telephone and the server separately is desirable. A test tool for testing the PlayNow service has been developed within this project. The new application is called “PlayNow Test Center”.

If a failure occurs on the communication path between the mobile device and the server, it can be very difficult to determine *where* the problem is located. The problems described in section 1.1.2 are very difficult to detect if they occur. There is no easy way to expose such problems, especially without access to the network element logs and management system for each operator’s network. Even though these kinds of problems are difficult to detect, one must understand that these problems do not happen very frequently.

Ping and traceroute are useful for testing the communication path at the IP level. These tools can be helpful, especially if the problem resides within the public IP network (Internet). If an error is located within the GPRS network, these tools can only indicate to the tester that the problem resides somewhere within the GPRS network.

If the communication path is unstable it is difficult to troubleshoot. When the problems only appear randomly, it is difficult to recreate the failure. The most common reason for an unstable path is due to the wireless link. If the radio connectivity is weak, a little distortion can interrupt the communication between client and server.

The PlayNow Test Center is an application for testing the PlayNow server. It has been developed within the scope of this thesis. It uses a web based GUI and therefore it is easily distributed to the users. The application tests the PlayNow server configuration with different combinations of parameters. It tries to download the ring signals and check for potential errors. So far the PlayNow Test Center has been very helpful for addressing problems with the PlayNow Server. Despite not having all available combinations tested yet, many errors in the PlayNow server has been revealed. Additionally, it has been helpful when testing the new Z800 mobile telephone. With this test tool all parameter combinations can be tested in order to verify that the server is configured correctly.

There are many additional features that can be implemented in PlayNow Test Center (See section 8 concerning future work), but together with the staff at Sony Ericsson, we have decided that no additional functionality should be implemented within this thesis project.

There are several parameter values that are not possible to test from a mobile telephone. Most people developing these telephones are situated in Sweden, while the telephones they are developing should work globally. Hence, it is impossible to connect via a Chinese telephone and test PlayNow, as it would function in China when the tester is not

located in China. Currently the PlayNow service does not support roaming. However, it is possible to connect to the PlayNow Test Center with a mobile device. In this way the tester can see the PlayNow server reply for any parameter values in his mobile telephone.

One problem, for a tester of the communication path, is that PlayNow is a distributed service. This means that there are multiple PlayNow servers. It may happen that a failure occurs on one server, but not on the others. If there is a bug that rarely occurs in the application software, it is probable that it will only occur in one of the servers. The result may be that only some users get valid responses while others get invalid responses. These kinds of failures are difficult for a tester to detect. The tester might get a connection to a server that works correctly, while a user elsewhere in the world gets a connection to a server which is not working properly.

8 Future work

To examine the communication path deeper does not make sense based on the results in this report. However, there are some improvements of the PlayNow Test Center Application.

During the development of the PlayNow Test Center application there has grown a wish list for improvements of the application. The development of the application has been done in an extreme programming style, where a close correspondence between the developer and the users of the application facilitated. However, not all wishes have been fulfilled. Prioritizing from this list, with importance vs. time has been done.

A more complex verification of the content delivered by the PlayNow server is desirable. Today, the application connects to the PlayNow server and downloads the content. It checks the size of the content versus the size tag in the DD. There is a desire to have the PlayNow Test Center application also verify that the content is a valid. If the content is supposed to be an mp3 file it is desirable that the application can verify that it is a valid mp3 file. It is also desirable to be able to verify that the content matches the description in the XHTML list. The only content that is verified are the ring signals. The PlayNow XHTML page has a number of links to style sheets and pictures, which are **not** currently verified by the PlayNow Test Center application.

An overall improvement of the GUI is desired. The current GUI is web based, which has made it difficult to create a good user interaction. One solution would be to reconstruct the GUI with Java applet or Flash technology.

In the near future the PlayNow application will be upgraded to handle content other than ring signals. As discussed above, this gives PlayNow to be a distributor of games and full-length music. Exactly how this will be implemented is not decided, but it is likely that the PlayNow Test Center must be upgraded to be able to test the new service.

The presentation of statistics by the PlayNow Test Center is made as a table in HTML. Most times this is a good way of presenting information. However, when a test case consists of thousands of test entities this gives a table of thousands of rows. A way to aggregate the results is desirable. One way is to generate a spreadsheet with the result.

9 References

Books

1. Lars Ahlin, Christer Frank, and Jens Zander. **"Mobil Radiokommunikation",** Studentlitteratur 2001, ISBN: 9144019165
2. Richard W. Stevens. **"TCP/IP Illustrated",** Addison-Wesley Longman 1999, ISBN: 0-201-63346-9
3. Luke Welling, and Laura Thomson. **"MySQL Handboken",** Pagina Förlags AB, 2004 ISBN: 91-636-0830-8

Internal Sony Ericsson document

4. Peter Stark. **"Play now Use Cases"** Document Nr: 8/1056-LXE 108 107 Uen
5. Magnus Dahl. **"Pre-play Integrated Solution"** Document Nr: 12/0360-LXE 108 110 Uen
6. Ericsson radio system. "**EDGE IN GSM**", 2001 Document Nr: EN/LZT 123 5776 R2B 2000-12-
7. Sony Ericsson. **"Pre play white paper"** 2004

Papers

8. Amitabh Mishra. "**Performance and Architecture of SGSN and GGSN of General Packet Radio Service**" Bradely Department of Electrical Engineering, Virginia Polytechnic Institute and State University, Blacksburg 2001
9. Götz Brasche and Bernhard Walke. "**Concepts, Services and Protocols of the New GSM Phase 2+ General Packet Radio Service**", Aachen University of Technology, 1997
10. Rajiv Chakravorty and Ian Pratt. "**Practical Experience with HTTP and TCP over GPRS**", University of Cambridge, Computer Laboratory, 2002
11. Rajiv Chakravorty, Joel Cartwright, and Ian Pratt. "**Practical Experience with TCP over GPRS**", University of Cambridge, Computer Laboratory, 2002
12. Rajiv Chakravorty and Ian Pratt. "**WWW Performance over GPRS**", University of Cambridge, Computer Laboratory, 2002
13. W Fetherstone and W Molkdar. "**Capacity benefits of GPRS coding schemes CS-3 and CS-4**", Motorola, UK. 2002
14. Kim Hoyoung, Kim Jinwoo, Lee Yeonsoo, Chae Minhee, and Choi Youngwan. "**An empirical study of the use contexts and usability problems in mobile Internet**", Human Computer Interaction Lab, Yonsei University, Seoul, Korea, 2002
15. Meyer Michael. "**TCP Performance over GPRS**", Ericsson Eurolab Deutschland, 1999
16. G. Peersman, P. Griffiths, H. Spear, S. Cvetkovic, and C. Smythe. " **A tutorial overview of the short message service within GSM**", Computing&Control Engineering JOURNAL, April 2000
17. Rami G Mukhtar, Stephen V Hanly, and Lachlan L. H Andrew. "**Efficient Internet Traffic Delivery over wireless networks**", University of Melbourne, 2003

18. A Saleh, D Avery, and M Siddiqui "**An application framework for mobile Internet services**", Wireless technology planning, Bell, Canada, 2004
19. Akamai Technologies, Inc. "**Why Performance Matters**", Akamai Whitepaper 2002
20. Wireless Application Protocol Forum "**WAP Architecture, Version 12-July-2001**" Architecture Specification WAP-210-WAPArch-20010712, 2001
21. Wireless Application Protocol Forum "**Wireless Application Protocol"** White Paper, June 2000
22. Wireless Application Protocol Forum "**Wireless Application Protocol, WAP 2.0",** Technical White Paper, January 2002

Web pages

23. **"Akamai"** http://www.akamai.com/Accessed regularly
24. "**DELI**", http://cocoon.apache.org/2.0/develoPing/deli.html Accessed regularly between 2004-10-25 and 2004-12-01
25. **"Encryption An Introduction - Apache HTTP Server"** http://httpd.apache.org/docs-2.0/ssl/ssl_intro.html Accessed 2004-02-22
26. "**GPRS White Paper**", http://www.vocal.com/data_sheets/gprs5.html Accessed 2004-11-14
27. **"HttpClient - HttpClient Tutorial"** http://jakarta.apache.org/commons/httpclient/tutorial.html Accessed regularly
28. "**Introduction to WAP**". http://compnetworking.about.com/library/weekly/aa123000a.htm Accessed regularly between 2004-10-25 and 2004-12-01
29. **"J2J development tool"** http://coldjava.hypermart.net/servlets/j2j.htm Accessed regularly.
30. **"Jakarta POI - Java API To Access Microsoft Format Files"** http://jakarta.apache.org/poi/ Accessed regularly between 2005-01-15 and 2005-02-04
31. **"Java 2 Platform Standard Edition 5.0 API Specification"**, http://java.sun.com/j2se/1.5.0/docs/api/ Accessed regularly.
32. "Java Technology" http://java.sun.com/ Accessed regularly.
33. **"**JavaScript HTML DOM Objects**"** http://www.w3schools.com/js/js_obj_htmldom.asp Accessed regularly
34. **"Jericho HTML Parser (au.id.jericho.lib.html)"** http://jerichohtml.sourceforge.net/api/au/id/jericho/lib/html/package-summary.html#package_description Accessed regularly between 2004-12-01 and 2005-01-05.
35. "**OMA home**", http://www.openmobilealliance.org Accessed 2004-10-29.
36. **"Open Source HTML Parsers in Java"** http://java-source.net/open-source/html-parsers Accessed 2004-12-20 Accessed regularly between 2005-01-15 and 2005-02-04
37. "**Using Login Authentication**", http://java.sun.com/webservices/docs/1.2/tutorial/doc/Security5.html Accessed regularly between 2004-11-20 and 2005-01-01

38. "**W3Schools Online Web Tutorials**", http://www.w3schools.com/ Accessed regularly.
39. **"WAP Forum"** http://www.wapforum.org/ Accessed regularly.
40. **"XHTML 1.0 The Extensible HyperText Markup Language (Second Edition)"** http://www.w3.org/TR/xhtml1/#xhtml Accessed regularly.

Appendix A – Excel file

Part of the Excel file – Service configuration specification Sweden rev PA6.xls. The red text (The six last rows in the spreadsheet) shows the new parts which are not included in version PA5 of the document.

Table 2: Service configuration specification Sweden rev PA6.xls

					CIP PlayNOW		
Operator name	**Network code**	**Sub-network code**	**Country**	**Devices**	**UserAgents/softwar eversions**	**Languag e variant**	**Languages**
Telia SE	240 01	10 => 19	Sweden	K700i	SonyEricssonK700i/R2A, SonyEricssonK700i/R2E, SonyEricssonK700i/R2J, SonyEricssonK700i/R2L, SonyEricssonK700i/R2N, SonyEricssonK700i/R2T, SonyEricssonK700i/R2V, SonyEricssonK700i/R2Y, SonyEricssonK700i/R2AA, SonyEricssonK700i/R2AC, SonyEricssonK700i/R2AE	EU4	English Swedish Finnish Norwegian
Vodafone SE	240 08	80 => 89	Sweden	K700i	SonyEricssonK700i/R2A, SonyEricssonK700i/R2E, SonyEricssonK700i/R2J, SonyEricssonK700i/R2L, SonyEricssonK700i/R2N, SonyEricssonK700i/R2T, SonyEricssonK700i/R2V, SonyEricssonK700i/R2Y, SonyEricssonK700i/R2AA, SonyEricssonK700i/R2AC, SonyEricssonK700i/R2AE	EU4	English Swedish Finnish Norwegian
Comviq	240 07	70 => 79	Sweden	K700i	SonyEricssonK700i/R2A, SonyEricssonK700i/R2E, SonyEricssonK700i/R2J, SonyEricssonK700i/R2L, SonyEricssonK700i/R2N, SonyEricssonK700i/R2T, SonyEricssonK700i/R2V, SonyEricssonK700i/R2Y, SonyEricssonK700i/R2AA, SonyEricssonK700i/R2AC, SonyEricssonK700i/R2AE	EU4	English Swedish Finnish Norwegian
Telia SE	240 01	10 => 19	Sweden	K500i	SonyEricssonK500i/R2J, SonyEricssonK500i/R2L, SonyEricssonK500i/R2N, SonyEricssonK500i/R2T, SonyEricssonK500i/R2V, SonyEricssonK500i/R2Y, SonyEricssonK500i/R2AA, SonyEricssonK500i/R2AC, SonyEricssonK500i/R2AE	EU4	English Swedish Finnish Norwegian

Vodafone SE	240 08	80 => 89	Sweden	K500i	SonyEricssonK500i/R2J, SonyEricssonK500i/R2L, SonyEricssonK500i/R2N, SonyEricssonK500i/R2T, SonyEricssonK500i/R2V, SonyEricssonK500i/R2Y, SonyEricssonK500i/R2AA, SonyEricssonK500i/R2AC, SonyEricssonK500i/R2AE	EU4	English Swedish Finnish Norwegian
Comviq	240 07	70 => 79	Sweden	K500i	SonyEricssonK500i/R2J, SonyEricssonK500i/R2L, SonyEricssonK500i/R2N, SonyEricssonK500i/R2T, SonyEricssonK500i/R2V, SonyEricssonK500i/R2Y, SonyEricssonK500i/R2AA, SonyEricssonK500i/R2AC, SonyEricssonK500i/R2AE	EU4	English Swedish Finnish Norwegian
Telia SE	240 01	10 => 19	Sweden	S700i	SonyEricssonS700i/R3B, SonyEricssonS700i/R3C, SonyEricssonS700i/R3F, SonyEricssonS700i/R3G, SonyEricssonS700i/R3H, SonyEricssonS700i/R2AA, SonyEricssonS700i/R2AE	EU4	English Swedish Finnish Norwegian
Vodafone SE	240 08	80 => 89	Sweden	S700i	SonyEricssonS700i/R3B, SonyEricssonS700i/R3C, SonyEricssonS700i/R3F, SonyEricssonS700i/R3G, SonyEricssonS700i/R3H, SonyEricssonS700i/R2AA, SonyEricssonS700i/R2AE	EU4	English Swedish Finnish Norwegian
Comviq	240 07	70 => 79	Sweden	S700i	SonyEricssonS700i/R3B, SonyEricssonS700i/R3C, SonyEricssonS700i/R3F, SonyEricssonS700i/R3G, SonyEricssonS700i/R3H, SonyEricssonS700i/R2AA, SonyEricssonS700i/R2AE	EU4	English Swedish Finnish Norwegian
Telia SE	240 01	10 => 19	Sweden	K300i	SonyEricssonK300i/R2AE	EU4	English Swedish Finnish Norwegian
Vodafone SE	240 08	80 => 89	Sweden	K300i	SonyEricssonK300i/R2AE	EU4	English Swedish Finnish Norwegian
Comviq	240 07	70 => 79	Sweden	K300i	SonyEricssonK300i/R2AE	EU4	English Swedish Finnish Norwegian
Telia SE	240 01	10 => 19	Sweden	"Clara"	Vodafone/SonyEricssonV800/R1A001	EU4	English Swedish Finnish Norwegian

Vodafone SE	240 08	80 => 89	Sweden	"Clara"	Vodafone/SonyEricssonV800/R1A001	EU4	English Swedish Finish Norwegian
Comviq	240 07	70 => 79	Sweden	"Clara"	Vodafone/SonyEricssonV800/R1A001	EU4	English Swedish Finish Norwegian

Appendix B – Manual for PlayNow Test Center

Introduction

PlayNow Test Center is an application for testing the functionality of Sony Ericsson PlayNow application server. It is possible to do both manually and automatic testing of the PlayNow server. Additionally, it is possible to test from a mobile telephone and from a normal web browser on a computer. There are two web addresses to PlayNow Test center:

1. http://192.16.134.148:8080/PlayNowTestCenter/J2J?PlayNowTestCenter.htm
2. http://192.16.134.148:8080/PlayNowTestCenter

The first address is for testing the PlayNow server from a computer. The second address is for testing from a mobile telephone.

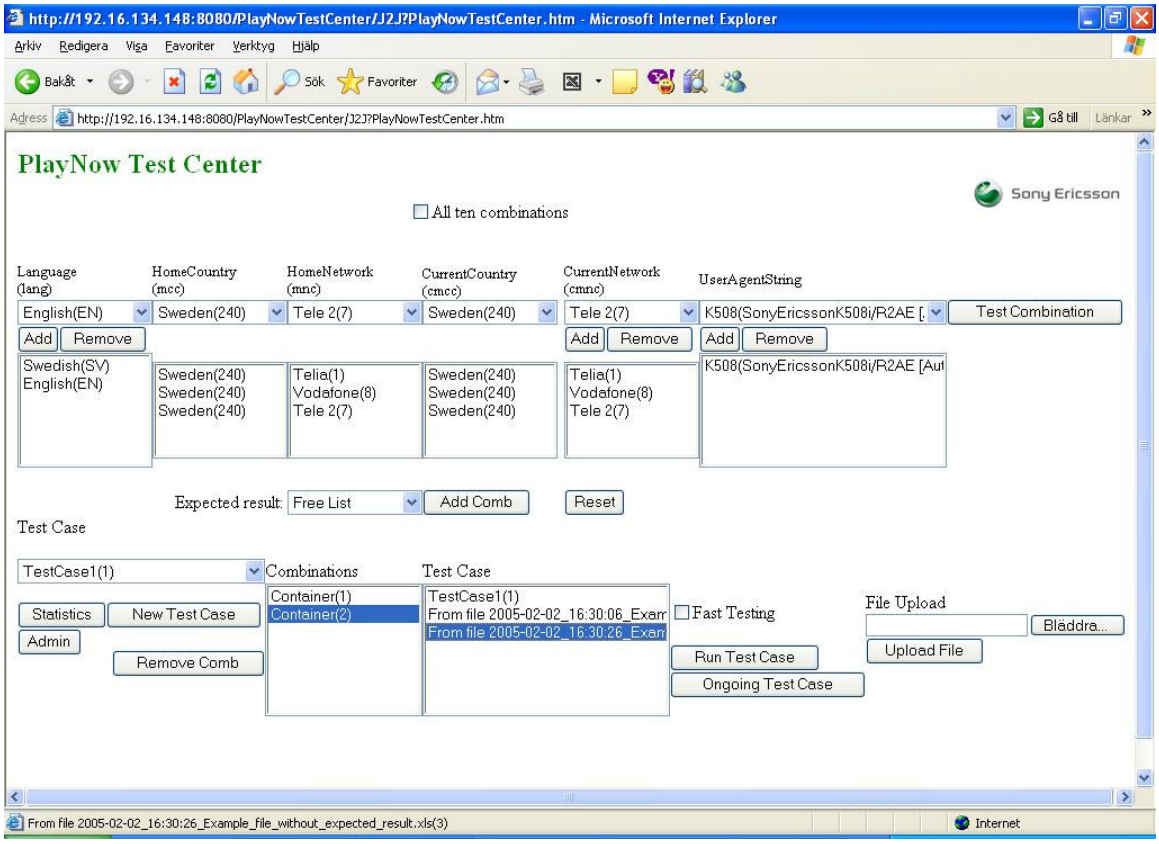


Figure 13: The PlayNow Test Center GUI

Manual Testing

To manually test the PlayNow server, select the parameter values from the six dropdown lists and press the button “Test Combination”.

When selecting a country, the dropdown menu for the operator changes to values that are valid for that country. For example: When selecting Sweden the values “Telia”, “Tele2”, and Vodafone appears in the dropdown menu for operators. Since roaming is not allowed in PlayNow the values of CurrentCountry(cmcc) and CurrentNetwork(cmnc) change when the values HomeCountry(mcc) or HomeNetwork(mnc) is changed. The value changes to the same value as the one selected. If the desire is to test roaming, then the values for CurrentCountry(cmcc) and CurrentNetwork(cmnc) must be selected after the values for HomeCountry(mcc) and HomeNetwork(mnc) are chosen.

When “Test combination” is pressed the application tries to connect to the PlayNow server. Depending on the reply from the server, the response will be different:

- A correct response: The reply from the application is the response expected from the PlayNow server. There are links to toggle through the different download descriptors.
- A free list: The response is a similar to a correct response, but the application only download one download descriptor, that of the first melody in the list.
- An error message: If the reply is an error message, the server’s reply is displayed.

Automatic Testing

The PlayNow Test Center application offers the possibility to configure test cases for automatic testing. It is possible to setup test cases manually or create test cases direct from a file. When setting up test cases manually, the application creates different combinations for the selected parameter values. When testing from a file, the user uploads a specially formatted Excel spreadsheet and creates different combinations for the values in the cells in the spreadsheet.

Setting up test cases in the GUI

For automatic testing, it is possible to create test cases. A test case consists of one or more combinations. A combination is a collection of parameters, which can be described by the lists under the drop down menus for the parameter values in the GUI (See figure 14).

Create a test Case

To create a new Test Case, just press the button “New test case”, then a small window will appear. Type the name of the test case and press “Ok”. The new test case will appear both in the “Test Case” dropdown and in the list with the same name.

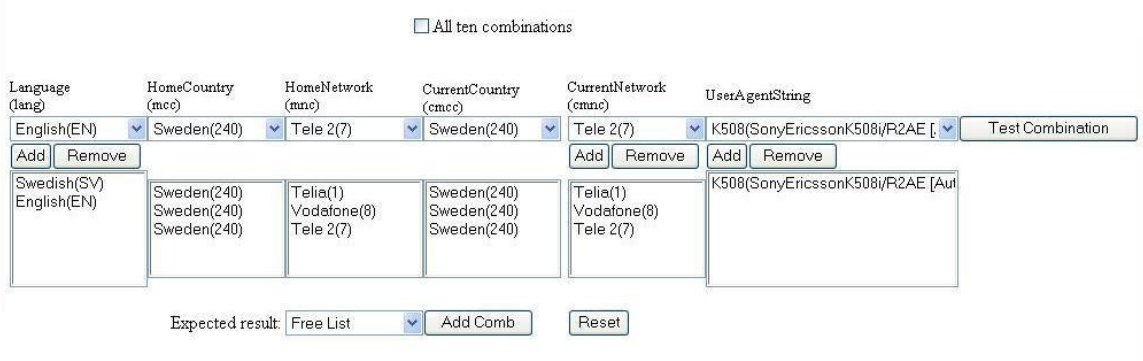


Figure 14: Part of GUI for creating test cases and combinations

Create a combination

Figure 14 shows the part of the GUI for creating a combination of parameters. The program groups the parameter values into three groups. Language is one group, the four parameters Home Country, Home Network, Current Country, and Current Network create a second parameter group, and the user agent string is the third group. When pressing the button named “Add Comb”, parameter values for all combinations of these groups are created and are saved in the database. A representation of the combinations will appear in the list called combinations. In the example in the figure, the number of parameter combinations is 2 * 3 * 1 = 6. If a combination is selected in the list named “Combinations”, the parameter values will appear in the lists of respective parameters. By selecting the different combinations in the list, it is possible to see the different combinations of test cases.

When pressing the button “Reset” all values in the parameter lists in the GUI will be cleared. The Reset button only has effect within the GUI. Nothing is changed in the database.

There is a checkbox called “All ten combinations”. It is used to create sub-network codes. For example: Telia has network code 01 in Sweden, but in reality the parameter value can be 01X, where X is anything from zero to nine. If the checkbox is selected when pressing the “Add Comb” button, the application will automatically create parameter values for all the different sub-networks as well. The number of combinations in the figure would be 6 *10 = 60 if the checkbox is selected when pressing the “Add Comb” button.

The drop down menu called “Expected result” is used to specify which result the user expects when testing this combination. By choosing one of the options “No Errors”, “Free List”, or “Error” it is easier to verify which test cases produced an expected result.

Setting up test cases from file

The files that can be read by the PlayNow Test Center application are based on files named “Service configuration specification XX.xls”, where XX is a country name followed by a sequence number of the document. In appendix A there is an example of such a file for Sweden. In these Excel files there are columns named “Network code”,

"Sub-network code", "UserAgents/softwareversions", and "Languages". These title cells must be located within the first ten rows in the sheet. The application searches for them to find which columns it should look at for the parameter values. Moreover, the parameter values must be directly below the title cells.

Table 3: A simple Excel spreadsheet

		CIP PlayNOW		
Network code	**Sub-network code**	**UserAgents/softwareversions**	**Languages**	**Expected result**
240 01	10 => 19	SonyEricssonK700i/R2A, SonyEricssonK700i/R2E, SonyEricssonK700i/R2J, SonyEricssonK700i/R2L, SonyEricssonK700i/R2N, SonyEricssonK700i/R2T, SonyEricssonK700i/R2V, SonyEricssonK700i/R2Y, SonyEricssonK700i/R2AA, SonyEricssonK700i/R2AC, SonyEricssonK700i/R2AE	English Swedish Finnish Norwegian	No Errors
240 01		SonyEricssonK300i/R2AE	English Swedish Finnish Norwegian	Free List

Table 3 is a simple Excel spreadsheet with the required format for the application.

How the application interpret the spreadsheet.
As mentioned earlier the program finds the title row where the titles for the parameters are located.

Network code
The Network code should be a string with a "space" inside. The value before the "space" is treated as the country code and the value after the "space" is treated as the network code for that country.

Sub-network code
The value for the "Sub-network code" is not processed other than to see whether there is something in that field. If something is written, the program interprets it as all the different Sub-network codes that should be tested. Otherwise, the program only tests one combination of the sub-network code.

UserAgents/softwareversions
A cell for "UserAgents/softwareversions" may consist of many different values. The application assumes a "space" or a "new line" separates these. If a comma character

separates the values, the application assumes these commas are not part of the UserAgents/softwareversions-values.

Languages
A cell for "Languages" may also consist of different values. The application assumes that a "new line" character separates these. There are languages that consist of more than one word (i.e. "Latin American Spanish") and therefore the application does **not** interpret a space as a character separating the values. If a comma character separates the values, then the application assumes that these commas are not part of the Language-values. Note: It is important that the language value is in the database. The application uses these values for connecting the database to get the correct language code. If the language is spelled incorrectly or if it is not in the database, then it is not possible for the application to generate the correct URI.

Expected result
There is an additional optional field "Expected result". For each row it is possible to specify an expected result for the test. If a cell is empty or the column "Expected result" does not exist, then PlayNow Test Center treat it as if the expected value is "No Errors".

Calculation of example from Table 3
When creating combinations of parameters from these rows inside a spreadsheet, the PlayNow Test Center application creates all possible combinations of these values. In the example in Table 3 the first row will generate 1*10*11*4 = 440 different combinations of parameters. The second row will generate 1*1*1*4 = 4 different combinations. All together, the application is anticipated to create 444 different combinations.

Upload a file
Uploading files to the database follows these steps:

- To upload a file press the "Browse…" button – a browser window will open.
- Select the file to upload. Once selected, the location of the file should be displayed in the text box named "File Upload".
- Press the button named "Upload File". Now the application tries to interpret the uploaded file. If the file is in the correct format, you should be redirected to a page with the information that the file has been loaded to the database and how many combinations of parameters that the content in the file generated. If the file does not meet the criteria mentioned earlier, the user will be notified of the fault.
- Click on the link "Back" to go back to the main page.

After returning to the main page, a new test case has been generated for the uploaded file. The test case is named "From file" plus the date and time when the test case was created followed by the file name of the file. For example: "From file 2005-01-01_12:00:00_ Service_configuration_specification_Singapore_PA6.xls (1)". The value between parentheses is the ID number of the test case in the database.

Fast testing

Under normal testing the application downloads all melodies for every combination of parameters. However, if the application is to test thousands of different combinations, it takes several hours, or even days to test. The part that takes the most time in the testing scenario is the downloading of the ring signals. When using “Fast Testing”, the application connects to the PlayNow server. If the reply is a correct response (No Error) and if the application was able to download all the ring signals, then it saves the links to the download descriptors. When connecting a second time to the PlayNow server, with another parameter combination, and receiving a correct response, it compares the links in the second reply with the links saved from the first connection, which have already been tested. If the links in the second reply are the same as the links form the first connection, the application assumes that these links still are correct. Without downloading the download descriptors and ring signals the application replies with a “No error” message.

By choosing “Fast testing” the test speed increases from approximate 150 combinations of parameters per hour to 1200 per hour.

Running a test case

When a test case is created, it is simple to run it. Simply select the test case in the list named “TestCase”, select whether “Fast Testing” should be used and press the button named “Run Test Case”. An alert window will appear with the text "Auto test started". Depending on how many combinations of parameters are in the test case it takes various amounts of time. Approximately 150 combinations per hour or with the “Fast Testing” option this can be increased to 1200 combinations per hour.

Clear a test case from earlier statistics

To remove earlier statistics from a test case, select test case and press the button “Clear Stat”. Before all statistics for the selected test case is removed, a warning with the option to cancel the operation is displayed.

To see ongoing test cases

To see which test cases are currently running, click on the button named “Ongoing test cases”. A window, with information about which test cases are currently running, is displayed.

To see the results from a test case

To see statistics from a test case, select the test case in the dropdown menu named “Test Case” and press the button “Statistics”. The application will redirect the browser to a page where statistics are shown. This page shows only the tests which failed. However, there is a link to a page where all test cases can be reviewed.

The statistics are shown as one row for each combination of parameters. The column shows the time when the test was completed. The following column shows the part of the URI, which contains the parameter values. The next column is the user agent string. The

last column is the result from the test. It is possible to see the result from an ongoing test case as described above.

To see the interpretation of an error message

If a test fails and the PlayNow server replied with an XHTML format error message. The messages in the view for the statistics are “Reply From Server is too small. ReplySize=X” Where X is the size of the reply in bytes. To see a more detailed description of the error, there is a link to a page where it is possible to see the correspondence between the size of the message and the XHTML formatted response.

Administering the database

When the button “Admin” is pressed the browser is redirected to a webpage where the database can be administered. Figure 15 shows the Admin GUI.

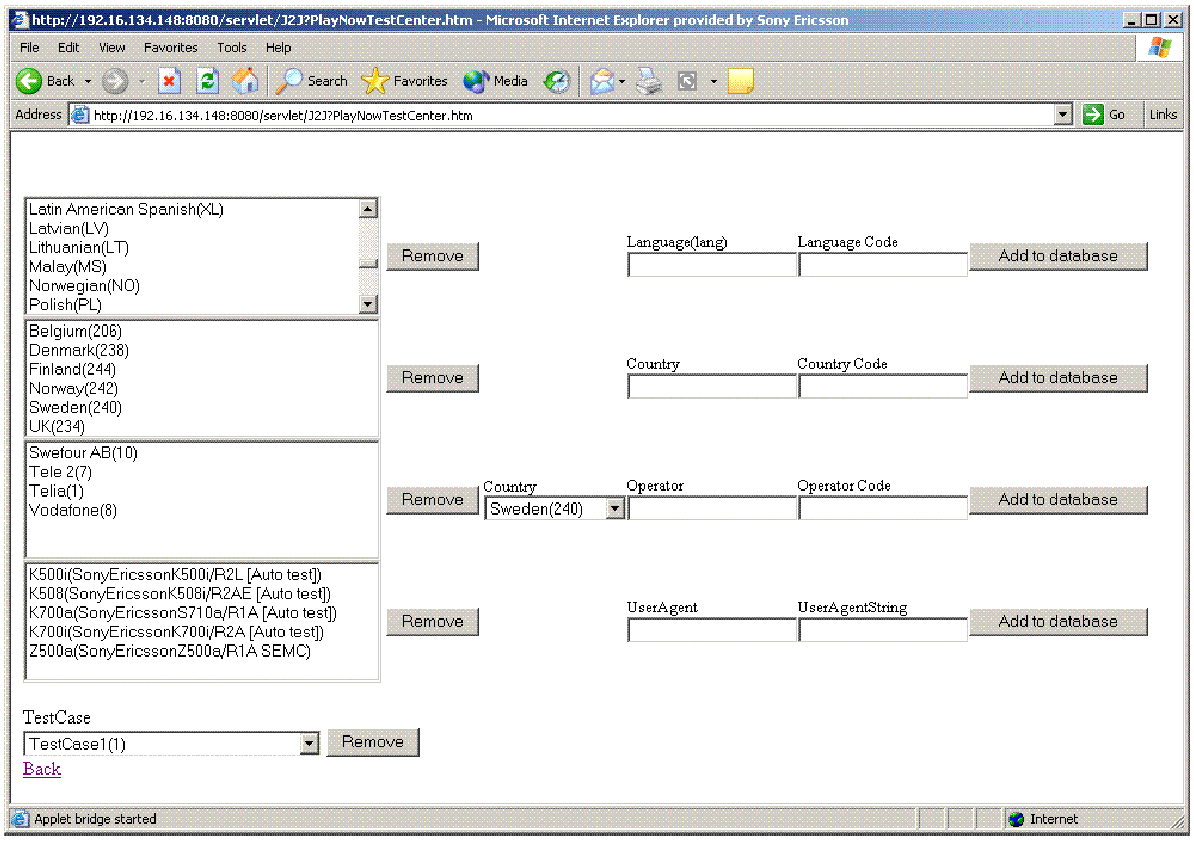


Figure 15: Admin GUI

From this page it is possible to add and remove items from the database.

Adding parameters to the database

There are four different kinds of parameters that can be added to the database:

1. Languages,
2. Countries,
3. Operators, and
4. User agent strings.

The same method is used to add languages, countries, and UserAgentStrings. To add a value to the database there are two textboxes to fill in. The first is the Name or the description of the parameter. The second is the value, which is going to be used by the application. It is important that the second field is correct!

To add an operator there is one additional object to take into consideration. All operators have a reference to a country. When adding an operator, a country **must** be selected in the drop down menu “Country”.

Test cases can be added from the main page.

Removing parameters from database

All the parameters that can be added to the database can also be removed. Additionally, test cases can be removed via this web page as well. To remove a parameter value from the database, simply select the parameter and press the corresponding “Remove” button.

To delete an operator the country must be selected in order to get the list of operators for that country. If a country is removed, the operators connected to it are also removed.

When removing a test case, all combinations connected to that test case are also removed.

Testing with a mobile telephone

The PlayNow Test Center has a section for testing from a mobile telephone. With this functionality it is possible to test the PlayNow server wherever the tester is.

Limitations

The GUI is simplified to support manual testing only. Additionally, because of the limitations of the browsers in mobile telephones the GUI is static and not constructed dynamically as was in the main GUI in the PlayNow Test Center, country codes and network operator codes have to be manually entered into their respective textboxes. The parameter values for languages and User Agent String are chosen from dropdown menus. Home country (mcc) and home operator (mnc) are Compulsory while Current Country (cmcc) and Current network (cmnc) is optional. If the optional values are left empty, the application assumes that no roaming will be tested.

User Instructions

Figure 16: GUI of the PlayNow Test Center application from a mobile telephone

When browsing to the Mobile PlayNow Test Center the user first comes to the main page (a). Fills in the form (b). When submitting (c), the browser is directed to a page where the result is shown (d). If the result is a “No Errors” message, a button named “See XHTML

Page" is displayed. When selecting this, it is possible to see the actual result in the mobile display (e). The idea is that a tester in Sweden with a Swedish SIM card can verify the result for any parameter values.

When testing with the mobile telephone with Mobile PlayNow Test Center it does not test that all the ring signals can be downloaded. The reason is that it may take as long as one minute for the PlayNow Test Center to download the ring signals. On a bad day it can take so long time that the mobile telephone will timeout.

Add user agent string

In the main page there is a link to a page for adding user agent strings to the database. As described in the section for administering the database (section 6) there are two textboxes to fill in. The first is the Name or the description of the parameter. The second is the value, which is going to be used by the application. Press the button "Add to DB" to submit the values to the database.

Appendix C – Abbreviations

API	Application Programming Interface
BSMS	Basic SMS (a PlayNow SMS payment scheme)
BSS	Base Station System
BTS	Base Transceiver Station
BSC	Base Station Controller
CA	Certificate Authority
CDMA	Code Division Multiple Access
CGI	Common Gateway Interface
CIP	Consumer Information Portal
Cmcc	Current Mobile Country Code
Cmnc	Current Mobile Network Code
CSS	Cascading Style Sheets
DB	Database
DD	Download Descriptor
DRM	Digital Rights Management
DUA	Download User Agent
GGSN	Gateway GPRS Support Node
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communication
GUI	Graphical User Interface
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
Mcc	Mobile Country Code
MIDI	Musical Instrument Digital Interface

MMI	Man-Machine Interface
Mnc	Mobile Network code
MS	Mobile Station
MT	Mobile Terminal
MVC	Model View Controller (MVC)
NMT	Nordic Mobile Telephony
OMA	Open Mobile Alliance
RDF	Resource Description Framework
RSMS	Roundtrip SMS (a PlayNow SMS payment scheme)
SGSN	Serving GPRS Support Node
SIM	Subscriber Identification Module
SQL	Structured Query Language
TE	Terminal equipment
TLS	Transport Layer Security
UAProf	User Agent Profile
UI	User Interface
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
VLR	Visitor Location Register
W3C	World Wide Web Consortium
WAP	Wireless Application Protocol
WML	Wireless Markup Language
WP-HTTP	Wireless Profiled HTTP
WP-TCP	Wireless Profiled TCP
WSP	Wireless Session Protocol

WTLS	Wireless Transport Layer Security
WTP	Wireless Transaction Protocol
XHTML	eXtensible Hypertext Markup Language
XHTML MP	Extensible HTML Mobile Profile
XML	eXtensible Markup Language

IMIT/LCN 2005-07

www.kth.se