Computer Science Honours Project Proposal

David Brown

March 2, 2012

1 Principle Investigator

David K. Brown
2 St Aidens Avenue,
Grahamstown, 6139
078 632 2520
Email: g09b0279@campus.ru.ac.za
Supervised by: James Connan

2 Project Title

The proposed project is titled:
AN SMS-Based Student Response System

3 Statement of the Problem

Other than with pen and paper, there is no *cost-effective* and efficient means of taking roll call or holding quizzes in large classes. For a roll call, calling out names is slow and tedious and passing around a list makes it simple for students to cheat by signing their friends names for them [9]. For a quiz, passing out multiple-choice answer sheets or collecting written answers wastes even more valuable lecture time. In an environment where all the students have computers and can answer online quizzes, this problem is easy to solve. Answering the quiz will automatically verify that the student was in attendance. In a lecture theatre, however, the only form of network communication students have is via their cellphones and lecturers cannot expect their students to spend their own airtime answering questions via SMS at normal network rates. The purpose of this project is to investigate whether it is possible to set up an Student Response System that makes use of SMS, at a reasonable price, and to determine whether this system would be a viable option to solve the problem at hand.

4 Objective of the Research

The Objectives of this project are as follows:

1. To develop a modular network architecture that will provide a platform for creating an SMS-based Student Response System.
2. To expose the functionality of the system as a web service.
3. To develop a prototype client application to make use of the exposed web service
4. To determine whether this system could be used to solve the issues defined in the problem statement above.
5. To develop an application for Android that can be used to answer quizzes over the network and eliminate human error in answer formats.

Future extensions to this work could include adding greater functionality to the web service and providing ways, over and above SMS, to answer quizzes.

5 History and Background

According to [8], there are now 5.9 billion people in the world with cellphones. This translates to 87% of the world's population. Despite this, sales of mobile devices are still on the rise and growth in this area is being led by smartphones. SMS remains the most popular form of communication with 8 trillion SMS messages being sent during 2011.

The OpenBTS project is an attempt at developing a Unix application that makes use of a software radio (such as the USRP designed and developed by Ettus Research) to present a GSM air interface that can connect calls between 2G GSM handsets using a PBX (such as Asterisk). This cellular network can be installed and operated at about a 10th of the cost of present-day technologies [3].

The history of OpenBTS begins in mid 2007 when Kestrel Signal Processing, Inc. began writing an implementation of a GSM basestation with the goal of creating a new kind of inexpensive, light-weight cellular network that could be used to connect rural and sparsely populated areas to existing cellphone infrastructure. David Burgess and Harvind Samra were the initial developers of the project and contributed most of the code for layers 1 and 2 of the GSM stack before the first public release of the source code [4].

Previous approaches to building such a network have paired OpenBTS with a Universal Software Radio Peripheral (USRP). A USRP is a piece of equipment that basically allows a general purpose PC to operate as a high bandwidth software radio. Together with the GNU Radio framework, which provides the signal processing run-time and processing blocks, and Asterisk, which is used to interface the GSM calls, these products can be used to build a private cellphone

network [5]. According to Herman Maritz, a former Stellenbosch student, who did a similar project for his Masters degree, his setup gave a network range of about 50 m [6].

One of the problems with this setup, however, is the onboard clock of the USRP. It makes registering a phone on your network challenging. According to Maritz [7], *you have to try and try again*. This is because the default onboard clock of the USRP runs at 64MHz. GSM clocks are derived from 13MHz, which means that good clock speed for a host would be one that is at a multiple of 13. For this reason, the USRP needs to be reclocked using an external clock. The clock suggested by gnuradio.org is ClockTamer [2].

Another way to provide SMS functionality to a system is via a GSM modem. The modemd is connected to the PC via a USB cable and AT commands are used to communicate with it.

6 Approach

Before beginning the project, the required hardware will have to be ordered. While waiting for this hardware to arrive, I will begin reviewing as much existing literature on the topic as possible. During this phase I will enhance my knowledge on OpenBTS, GNU Radio, the USRP, Asterisk and the steps involved in setting up the network. One aspect I will also be investigating is the legal issues that will be involved in setting up a private network. Checking for alternative methods for solving the problem defined in the problem statement will also form part of this stage. As such, research methods of using the GSM modem to satisfy the project objectives. I will also install and familiarize myself with Ubuntu.

When the equipment arrives, I will have a sound understanding of the steps required to set up the network. I will immediately set up the hardware and start installing the software. Installing GNU Radio will require installing its dependencies, getting and installing the C++ boost libraries, and then installing GNU Radio itself. Once these are installed I will add permissions for it to work with the USRP. After this I will test that GNU Radio and the USRP are working correctly together [5].

Once the USRP is working properly with GNU Radio, I will install OpenBTS. To do this I will first install its dependency (Asterisk). I will then install OpenBTS itself and configure it. This configuration will entail setting the Mobile Country Code to that of South Africa, setting the Mobile Network Code to a number between 0 and 99, and defining the frequency band that OpenBTS will operate within. Once this is done, the Absolute RF Channel will be set depending on the frequency band chosen [5].

With OpenBTS configured, I will set up Asterisk. Amongst other things, this will require the SIM Card IMSI. Once Asterisk is set up, OpenBTS can be tested [5].

Once OpenBTS is tested and found working, it will be time to connect the USRP to the external clock. For this step, I will send the USRP to the

electronics department for them to do the soldering. When it returns, I will make a few small software changes to GNU Radio. At this point, the network should be running smoothly.

With the network up and running, I will setup the GSM modem to run concurrently with it. Once this is operational, I will start working on the applications that will be required by the system. These applications will include the main Lecture Theatre System that will read in SMSs for class roll calls and quizzes as well as the mobile application for Android that will allow students with Android phones to have a simple interface to answer quiz questions with.

Should I have trouble at any of the stages and not be able to implement the network using the USRP, my fallback plan will be to use an SMS Gateway to try and implement the Lecture Theatre System. Should this need arise, I will use the open-source WAP and SMS Gateway: Kannel [1].

Once the implementation of the network and lecture system has been completed, I will write and submit my thesis.

7 Requirements and Resources

- Hardware Requirements:

1. USRP
2. 2x RFX900 Daughterboard
3. 2x VERT900 Antenna
4. External Clock (ClockTamer)
5. Unlocked cellular phone
6. SIM Card (preferably with option to select network)
7. GSM modem (Huawei E220)

The USRP, RFX900 boards and VERT900 antennas can all be acquired from Ettus Research [5, 6]. Together with ClockTamer, this hardware will cost $1824 (US).

- Software Requirements: [5]

1. OpenBTS
2. GNURadio
3. C++ Boost 1.37
4. Linux - Ubuntu

All the software is open-source and free.

8 Progression Time-line

Deadline	Activity
02 March 2012	Project proposal submitted
09 March 2012	Hardware ordered
16 March 2012	Project website online, Ubuntu installed
23 March 2012	Familiarized with Ubuntu
30 March 2012	All references for literature survey acquired
06 April 2012	Investigation into network setup completed
13 April 2012	Investigation into alternative methods completed
20 April 2012	Begin setting up the GSM Network
21 May 2012	GSM network online with SMS capability
21 May 2012	USRP sent to electronics dept.
28 May 2012	Literature survey and Plan Of Action submitted
24 July 2012	Seminar Series 2: oral presentation (15 minutes)
24 August 2012	Sever-side application completed
7 September 2012	Android application completed
17 September 2012	Short paper submitted
29 October 2012	Seminar Series 3: final oral presentation
02 November	Project completed
05 November 2012	Research website completed
21 November 2012	Final research oral examination

References

[1] Kannel: Open Source WAP and SMS gateway. Online, 2010. Available from: `http://www.kannel.org/`.

[2] GNU Radio - OpenBTSClockModifications - gnuradio.or. Online, 2011. Available from: `http://gnuradio.org/redmine/projects/gnuradio/wiki/OpenBTSClockModifications`.

[3] The OpenBTS Project. Online, 2012. Available from: `http://openbts.sourceforge.net/`.

[4] Burgess, D. Low Cost Cellular Networks with OpenBTS. Online, March 2010. Available from: `http://timreview.ca/article/332`.

[5] Loula, A. Openbts installation and configuration guide v0.1, May 2009.

[6] Maritz, G. S. H. Engineering Fun » OpenBTS (My own cellphone network). Online, 2009. Available from: `http://students.ee.sun.ac.za/~gshmaritz/?p=77&cpage=1`.

[7] Maritz, G. S. H. OpenBTS fix: Connect External Clock to USRP - Herman Maritz. Online, 2009. Available from: `http://gshm.posterous.com/openbts-fix-connect-external-clock-to-usrp`.

[8] MOBITHINKING. Global mobile statistics 2012. Online, February 2012. Available from: http://mobithinking.com/mobile-marketing-tools/latest-mobile-stats.

[9] TAKEMOTO, T. Online, 2007. Available from: http://moodle.org/mod/forum/discuss.php?d=22526&parent=309133.