Terminal Based Monitoring of Voice Call Quality in Mobile Networks

Liam Hartley and Liam Kilmartin

Communications and Signal Processing Research Unit,
Department of Electronic Engineering,
NUI, Galway,
IRELAND.
Phone: +353 (0) 91 49 2771
Fax: + 353 (0) 91 49 4511
E-mal: {liam.hartley,liam.kilmartin}@nuigalway.ie

Abstract

Recent technological advances have allowed great enhancements in the quality of service provided to subscribers by mobile telephony network operators. A key requirement in enhancing a subscriber's experience of utilising the services provided by such networks is the ability for the network operator to gather reliable information relating to the customer's experience when attempting to invoke the services offered by the network. Traditional methods include "drive testing" the network in order to gather sample information concerning the ability to access network services and subsequent experiences while utilising them, or the use of advanced network simulation tools to predict characteristics concerning radio coverage and likely resultant air interface characteristics. While these methods have served their purpose, it can be said that they are no longer as efficient as they have been in the past. This paper outlines a system for real time monitoring of call parameters by placing a small monitoring application on the subscriber's terminal to enable regular monitoring of various call parameters during voice calls. On call termination the subscriber is asked to rate the quality of the call by submitting a "call score" based on the Mean Opinion Score (MOS) rating. This rating and the collected information is then transmitted over a wireless connection to a network server to facilitate real-time quality analysis relating to subscriber's experiences in accessing the voice telephony service provided by the network.

Keywords

Mobile Telephony Networks, Service Levels, Customer Experience Monitoring

1 Introduction

The increase in market penetration of wireless devices in recent years has encouraged network operators to invest heavily in the quality and range of services offered to their subscribers. Consequently these subscribers have become accustomed to much improved network coverage and an expectation of a higher level of quality leading to a desire for ever improving network performance. This has resulted in network operators deploying functionality within the network that provides a capability to quantify the quality of service being experienced by their subscriber base or to specific subgroups within their subscriber base (e.g. larger corporate clients).

2 Monitoring Methodologies

Monitoring of service level offerings to customers can be implemented by a number of different approaches. However it is important that, regardless of what form this monitoring takes, it can be implemented network wide as well as in smaller sub-sections of the network. Traditional monitoring methods such as "drive testing" are often implemented locally while more comprehensive methods such as those based on monitoring parameters on the A_{bis} interface [1] are implemented network wide to enable automated continuous monitoring of the network-wide air interface.

These methodologies have traditionally focused on network based monitoring (e.g. a "probe" implemented in the Base Station Subsystem (BSS)) as opposed to the subscriber's terminal monitoring system outlined in this paper. This system proposes the deployment of a small monitoring application on subscriber devices in order to automatically sample relevant network parameters during voice calls. On call termination these parameters are transmitted via a wireless connection to network based server application to allow for real-time evaluation and storage of the information.

2.1 Traditional Network Monitoring Methods

Traditional Monitoring methods have focused on a combination of "drive testing" and network simulation. However there are a number of issues associated with such approaches. "Drive tests" for example can be quite cumbersome to plan and undertake, as vehicles and engineers have to be assigned to test specific routes at particular times. This in itself leads to limitations in the ability to generalise from the recorded results as such tests are completed when the network has a specific loading and, hence, it is questionable whether deductions can be reliably made regarding the ability to predict the performance of the air interface from the recorded results under other loading conditions.

An addition limitation relates to the fact that a delay can exist between finalising cell upgrades or modifications and the completion of a follow up "drive test" (conducted specifically to analyse the impact of these changes on the air interface's performance). A final and possibly more significant limitation with "drive testing" is the fact that it is route specific and as such there is no allowance for random paths or indoor areas to be easily tested.

2.2 Air – Interface Methods

In order to achieve a more accurate view of the network interface, it is important for network operators to have access to as much network information as possible. This means that while "simulated" information is good, "real" data is better. In recent times, new methods have been proposed and developed which try to achieve this by sampling "real-time" network parameters on a continuous basis. A number of solutions have been proposed to achieve this such as those based on placing "probes" on the A_{bis} interfaces to allow for continuous monitoring of the entire network for all subscribers.

As these methods monitor network behaviour based on "real" network transactions associated with customers interactions associated with subscriber interactions with the network, they allow every user of the network to effectively "drive test" the network. This leads to a major increase in both the quality and the quantity of data network operators have regarding dropped calls, handover failures and in call signalling levels etc. all of which are useful indicators of a customers experience in the network.

One such solution [2] employed by a number of Europe's leading GSM operators monitors customer experience by collecting key information from the mobile network. This information can then be processed by a number of software applications allowing operators measure and quantify service levels offered to customers.

An alternative solution [3] does concentrate on field based monitoring. However the solution is limited to monitoring performed by network operatives. This system requires two laptop computers with specialist software packages and a number of additional hardware accessories. A software GUI is responsible for simultaneous transmission and reception of a predefined voice file. The received (degraded) file is compared to the original file by using the standard suite of intrusive quality evaluation algorithms (e.g. PESQ [4] etc.)

Further software allows the generation of a number of graphs to allow easy and accurate analysis of the operator's network service offering.

The "Terminal based" monitoring system this paper proposes, provides a "simpler" solution to the problem in that it requires no change to existing network infrastructure and no additional hardware, Instead an automated application executes on the subscriber's handset and is responsible for "sampling" relevant parameters during voice calls thus making the subscriber the effective "test drivers" of the network. The installed application records the call parameters during mobile originated and mobile terminated calls. This information along with additional subscriber feedback is automatically transmitted from the handset device to a network server for processing, before it is stored in a network database for future analysis. As with other interface monitoring systems the collected data is "reliable" and it can form the basis of an analysis to allow the operator to ascertain a view of service levels that subscribers experience during voice calls.

Opinion Score Rating	Level of Distortion
5	Imperceptible
4	Just Perceptible but not Annoying
3	Perceptible and Slightly Annoying
2	Annoying but not Objectionable
1	Very Annoying and Objectionable

Table 1: Mean Opinion Score (MOS) Ratings

As both voice quality and air interface parameters are equally important in determining the quality experienced by the subscriber, the "drive tester" is prompted by the monitoring application to enter a "score" for the call quality once monitoring has completed. This rating is based on the Mean Opinion Score (MOS) [5], a commonly used subjunctive rating scheme as per Table 1.

This additional feature is one of the benefits of device side monitoring, as it not only allows an

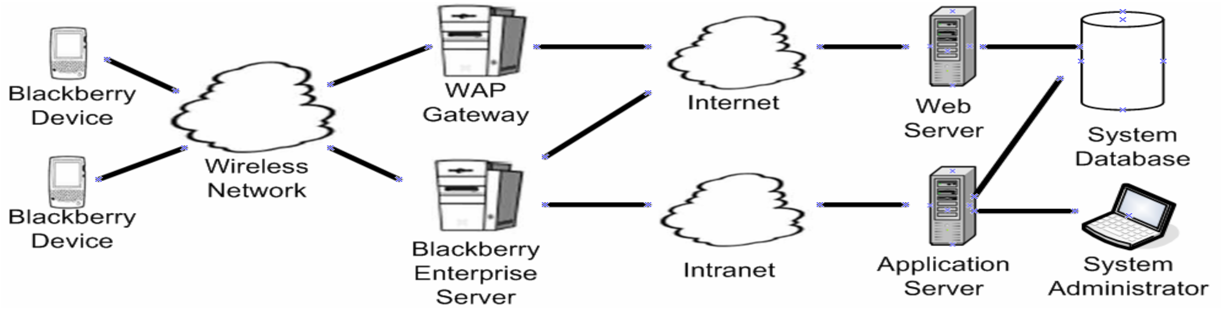

Figure 1:Overall Blackberry System Architecture

operator access to what they perceive as a network's quality, but also to the quality from a user's perspective, thus ensuring as accurate a portrayal of the entire network as is possible.

Additionally, as a network consists of a large number of subscribers (covering both urban and rural areas), the ability to generalize data recorded during "drive tests" can be questionable. While "drive testing" may have its advantages in urban areas, the "subscribers to area" ratio in rural areas is much lower and therefore limits the efficiency of scheduling a "drive test" in such areas. The proposed system also addresses the problem that arises as a result of the relatively small number of "drive tests" that are possible in comparison to the large number of subscribers in a network, by offering all device users the option of being a "drive tester".

3 System Implementation

Traditionally, network operators have used a number of Key Performance Indicators (KPI's) to provide a measurement of a network's service offering [6]. Operators rate and define quality of service differently; however, overall service quality is generally calculated based on a combination of the following parameters

- Network Coverage
- Call Blockage
- Dropped Call Rate
- Voice Quality

The first three parameters are easily measured using existing software tools available from a number of vendors. Voice quality however is a more complicated metric to rate. Traditional methods for rating call quality have often focused on Bit Error Rate (BER) analysis, however the system outlined in this paper proposes use of a "call score" value based on the MOS rating, provided by subscribers on call termination.

The prototype system, utilizes the Blackberry architecture [7] developed by Research in Motion (RIM), hence this architecture will form the basis of the following description of the system.

The terminal based Java application is responsible for recording a number of call parameters during any incoming or outgoing voice transmission, along with prompting the subscriber for the call score at the end of the call. Once all the information has been collected, transmission of the stored data is completed over a socket connection to a destination Internet based server, via a Blackberry Enterprise Server (BES) or a WAP Gateway. After processing, the data is stored in a SQL based database to allow for later retrieval and analysis.

The monitoring system that has been developed utilizes a number of signed RIM API's in conjunction with the JAVA2 Mobile Environment (J2ME) and it can be divided into two separate modules as follows:

- Client (Device) Application
- Server (Operator) Application

3.1 Client (Device) Application

The subscriber device enters network-monitoring mode on commencement of a voice call. During the call, the following parameters are measured at a configurable time interval.

- Sample Timestamp
- Received Signal Strength Indicator (RSSI)
- Cell Identifier
- Channel Number (AFCHN)

The sampling of additional parameters accessible via the terminal specific programming API's can be easily integrated into the client application. Ideally the transmit and receive power levels would be sampled, however, as the available API's do not allow for this it is not currently possible. As a result, the Received Signal Strength Indicator (RSSI) level is recorded as part of the monitoring process. The RSSI value is based on a logarithmic scale which can be directly related to the receive power level on a subscribers device.

The reason the Frequency channel number (ARFCN) is recorded in conjunction with the Cell Identifier is to allow analysis of the situation where a number of intracell handover's occur during a call. Intracell handovers can represent localised interference of the air interface, so a call utilising more than one frequency channel while in the same cell may be indication of localised interference. The use of a cell identifier as the basis of location information ensures that an accurate portrayal of network coverage is possible.

Once the voice call is terminated the subscriber is prompted to rate the call quality using the Mean Opinion Score (1 – 5) and this selection is appended to the stored information (including the subscriber's MSISDN and the terminals IMEI) along with a code representing the reason for the termination of the previous call. A background thread is then initiated and is responsible for the transmission of the information to the database server.

Depending on the device configuration, the information is transmitted to the server via a socket connection over the GPRS network using either a WAP Gateway or a Blackberry Enterprise Server (BES), which has the Mobile Data Services (MDS) feature, installed. The MDS Feature is required to allow socket creation between the handheld device and the application server. Corporate clients will normally access the network via a BES (Figure 2) while standard network subscribers are more likely to utilize the WAP Gateway (Figure 3) configuration.

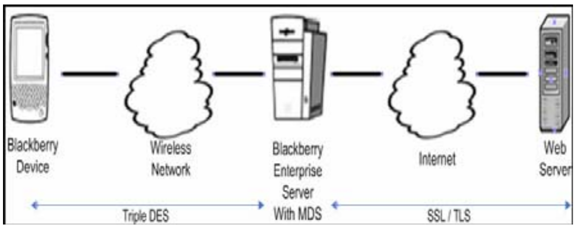

Figure 2: Blackberry Connection utilizing BES Server

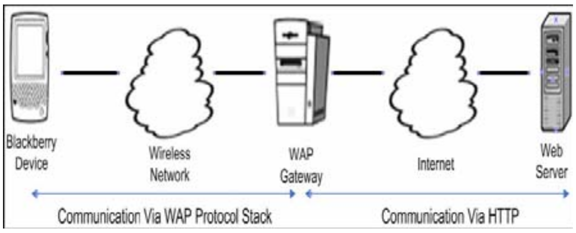

Figure 3: Direct Socket Connection utilizing WAP Gateway

3.2 Server (Operator) Application

Once information is received at the server from the mobile terminal based subscriber application, it must be analyzed in order to ensure the database avoids corruptions arising from invalid submissions. A unique call identifier is generated based on the time at which the air interface parameter sampling procedure commenced on the mobile terminal. A MySQL database is used to store the submitted information while PHP and HTML web interfaces allow an administrator to analyze the data and to generate graphs based on specific search criteria. These graphs provide instant access to stored information, making it immediately apparent if there is a problem in a particular area of the network or with a particular device. A number of graphing options are supported these include:

- Plotting RSSI values for a single call (Figure 4) or a number of voice calls.
- Plotting "User Score" values for a particular cell or MSISDN (Figure 5).
- Plotting "User Scores/RSSI" values for a particular call termination reason.

All plots can be time relative, or time independent, depending on the selected parameters, this allows for accurate graphical representation of selected parameters over a specific time frame thus enabling operators to trouble shoot faults more accurately and efficiently.

Call information can also be based on a particular voice call, range of voice calls or a particular cell, as required, to allow efficient and easy access to the valuable real-time call statistics thus ensuring network issues can be identified and subsequently resolved with minimal difficulty.

Traditionally channel holding time (the duration of time which a terminal is allocated a channel in a particular cell during a voice call) can be difficult to readily determine in practice. However, with the proposed system, it is possible to calculate the real-time channel holding time to the nearest specified time interval. This allows network operators find locations were excess "cross handovers" are occurring, as well as providing further information on network utilisation and possible faults.

Typically, the Java application is installed on the target blackberry device's using the standard Blackberry desktop software packaged with all Blackberry devices. However the Blackberry architecture provides the capability for application deployment using an Over the Air (OTA) installation mechanism. System administrators store the application's jar and jad files on an accessible server, users are then able to automatically download the application at their convenience. This provides an efficient and easy way for mass deployment of the application to large subscriber groups.

4 SYSTEM IMPLEMENTATION ON OTHER TERMINAL OS

While the Java based RIM (Blackberry) operating system is in widespread use, there are currently a number of additional operating systems that are growing in popularity for use on mobile terminals. Windows Compact Edition (Windows CE) is a Microsoft Operating System, which has been designed primarily for deployment on smart phones.

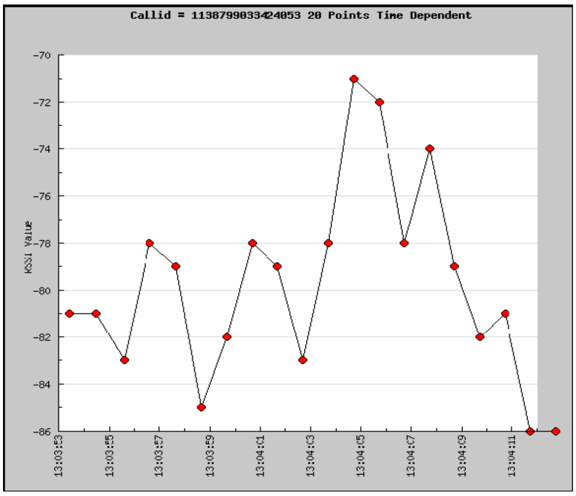

Figure 4: Sample Time Dependent Plot:

Call_id = 1134564663200107

It provides a number of programming interfaces that allow development across a wide number of areas. Traditional device manufacturers such as Nokia, Ericsson and Motorola have developed their own operating systems for use on their devices, while also contributing to the development of the Symbian Operating System [8]. The Symbian OS drives standards for the interoperation of data-enabled mobile devices with mobile networks, content applications and services to its licensees.

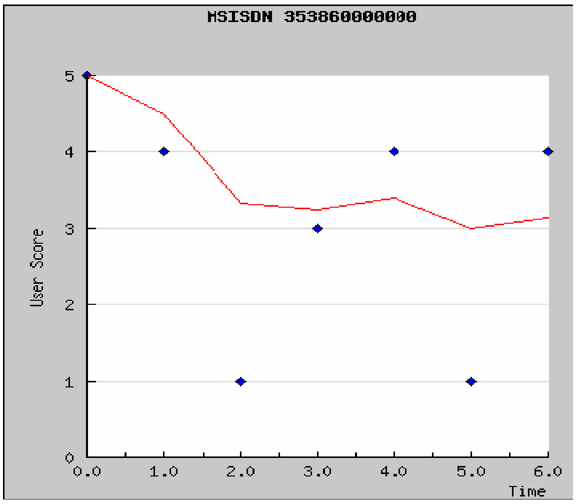

Figure 5: Sample User Score plot:

MSISDN = 353860000000

A set of standard application programming interfaces (APIs) across all Symbian phones and the advanced computing and communications capabilities of Symbian Operating Systems enable the development of advanced services by developers.

While the system outlined in section 2 has been implemented successfully using the RIM Blackberry architecture, it is possible to implement a similar system for a Symbian based target platform. There are a number of differences between the two systems not least the fact that, while RIM applications are developed exclusively in Java, Symbian devices use both JAVA and C++. The C++ programming language is required to access all the native operating system parameters while requiring all Symbian devices to support Java Midlets ensures Java compatibility. There is a major drawback to this approach however, as while midlets can provide additional functionality; their capabilities are severely restricted by what is known as the “Sandbox Problem”.

The development of the system based on the Blackberry architecture avoided the ‘Sandbox Problem’ due to the fact that RIM provide access to the core terminal API by what are known as “signed APIs”. However this is not the case with the Symbian OS, which only supports Java applications via the use of Midlets (which are very restricted in terms of the terminal functionality which they can access).

One possible solution to this problem in a Symbian OS environment would be for the developer to write native C++ code that can be installed on the Symbian device and for this local application to use local socket connections to access the Java Midlet and the core development classes as required [9]. This solution alleviates the need for a complete redevelopment of the application using the C++ programming language, which is currently the only other option available to developers.

As Symbian is an open and standardized operating system for data enabled mobile handsets, it provides a perfect platform for additional implementation of the device side call monitoring mechanism. The Symbian OS supports additional device location information through the JSR – 179 Location API for J2ME. [10] Location information similar to this is only supported by the Blackberry OS in more recent versions, which have not yet been widely implemented on devices. The JSR – 179 API enables a mobile device to obtain information about its present geographic location and orientation in addition to the ability to access a database of known landmarks stored in the terminal. However as JSR – 179 forms part of the J2ME API it is limited by the “Sandbox Problem” and the workaround outlined in the previous section. With the introduction of the Symbian V8.0 Operating System, the ETEL 3rd party API has been extended to incorporate nearly all services required by independent developers that were previously inaccessible. For instance with Symbian 8.0 it is possible to access information on:

- Device IMEI (International Mobile Equipment Identity)
- Subscriber IMSI (International Mobile Subscriber Identity)
- Signal Strength and Network Availability
- Location Area Code and Cell Identifier

These enhancements make it possible to extract similar information using the Symbian OS as is extracted by the developed application on Blackberry devices.

5 CONCLUSIONS AND FUTURE WORK

The system outlined in this paper is based on an architecture that will allow an improvement in the volume, relevance and quality of information relating to the level of service provided to subscribers in mobile networks. The system is easily implemented and is both scaleable and reliable. It is also possible for the system to be used to provide network operators with real time real information about customers call quality in a particular area at a particular time, or over a period of time, thus helping to ensure that the highest level of customer experience is achieved and maintained.

The stored information can be graphically accessed by the network operator and analyzed using on a wide range of input parameters thus providing an easily understood accurate graphical representation of the collected information. Finally the proposed system is capable of monitoring the number of dropped calls and the reason for the disconnection which, enhances the system's capabilities to provide readily available information on the service provided in individual cells or groups of cells and to individual subscribers or subgroups of subscribers.

The system, as it has been implemented, allows for a number of future enhancements and developments. For example, current trends mean that more specific location information, such as that provided by GPS hardware, is likely to be a feature of future releases of both the RIM and Symbian operating systems. This improvement in location information would enhance the accuracy of the collected data, allowing location information of sampled data to be accurate to a couple of meters thus aiding more accurate network planning and monitoring based on the information captured by the system.

Additionally, the use of automated non-intrusive speech quality monitoring techniques such as ITU-T standard P.563 (Single Ended Method for Objective Speech Quality Assessment in Narrow-Band Telephony Networks) [11] , could further enhance the benefits provided by the system. Traditional intrusive perceptual voice quality estimation techniques, much like drive testing has proven very successful in the past, however non-intrusive techniques offer the possibility of providing a MOS value based on the analysis of the actual speech waveforms during the call (perhaps in addition to the subscriber entered MOS value). However none of the current operating systems used on mobile devices provide the direct access to the on-system DSP functionality which would be required to implement such speech processing algorithms in real-time.

Finally the addition of data mining techniques for the purpose of analyzing the large amounts of data, which would be stored on the system database, would greatly enhance the overall value of the system. This would ultimately lead to greater accuracy in predicting the outcome of completed network enhancements before they are finalised and implemented.

6 ACKNOWLEDGEMENTS

The authors would like to acknowledge the support of O2 (Ireland) during this project.

VII REFERENCES

[1] J.O. Neilsen, G.F. Pedersen, C. Solis, "In Network Evaluation of Mobile Handset Performance," in Proceedings of (IEEE) Vehicular Technology Conference 2000. IEEE VTS-Fall VTC 2000. 52nd Vol. 2, pp 732 – 739, September 2000

[2] Brad Smith, "Managing the Customer Experience," Wireless Week, http://www.wirelessweek.com/article/CA622740.html, 1st July 2005

[3] "Voice Quality Testing, User Guide", *GL Communications Inc.* November 2005

[4] ITU – T Recommendation P.862, "Perceptual Evaluation of Speech Quality (PESQ), an objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs," *International Telecommunications Union*, February 2001

[5] ITU – T Recommendation P.800.1, "Mean Opinion Score (MOS) Terminology," *International Telecommunications Union,* March 2003

[6] "Optimising Service Quality in GSM/GPRS Networks", Infocus, http://infocus.telephonyonline.com/ar/telecom_optimizing_service_quality/index.htm, 19th Sept. 2001

[7] Blackberry Enterprise solution architecture, Research In Motion (RIM), http://www.blackberry.com/products/software/server/architecture/index.shtml

[8] J. Stichbury, "The Symbian OS Explained: Effective C++ programming for smart phones," *Symbian Press*

[9] A. Gupta, M. de Jode, "Extending the reach of MIDLets: How MIDLets can access native services," Version 1.1, *SymbianConsortium*, http://www.symbian.com/developer/techlib/papers/MIDlet_Native

_Services_Framework/MIDlet_Native_Services_Framework_v1.1 .zip, June 2005

[10] Java Specification Request – 179, "Location API for Java 2 Micro Edition," Version 1.0, http://jcp.org/en/jsr/detail?id=179, September 2004

[11] ITU – T Recommendation P.563, "Single ended method for objective speech quality assessment in narrow band telephony applications," *International Telecommunications Union*, February 2005