Journal of Mobile Multimedia, Vol. 2, No. 1 (2006) 002-022

GPRS-BASED MOBILE TELEMEDICINE SYSTEM

BRYAN WOODWARD, M. FADLEE A. RASID, LAWRENCE GORE, PAUL ATKINS

Loughborough University, U.K.
b.woodward1@lboro.ac.uk

Received October 6, 2005
Revised February 27, 2006

An important emerging issue in mobile telemedicine, or m-Health, is how best to exploit the mobile communications technologies that are now almost globally available. This paper describes the design of a telemedicine system to transmit a patient's biomedical signals to a hospital for monitoring or diagnosis, using a Bluetooth-enabled mobile telephone networked to the General Packet Radio Service (GPRS). The system can transmit from one to eight biomedical signals, typically including the electrocardiogram, blood pressure, temperature and oxygen saturation. The design of a database server that allows access to the received data by clinicians is also briefly described. The complete system has been tested successfully using several different data-enabled "smart" mobile telephones running on the Series 60 development platform. The tests were carried out while stationary and while travelling at high speed in a car.

Key words: telemedicine, cellular networks, mobile communications, General Packet Radio Service (GPRS), Bluetooth
Communicated by: A Ganz & R Istepanian

1. Introduction

Although the common medical scenario of face-to-face consultations between a clinician and a patient will never be replaced, there are medical cases that can be managed more efficiently by using wireless telemedicine. Potential mobile applications include remote routine check-ups, emergency and rescue situations, and sports science physiological measurements. Medical services can now be delivered to any location within the coverage area of cellular networks, therefore routine monitoring can be done while the patient is at home, travelling, at work or at leisure. This paper presents the design and implementation of a mobile telemedicine system that allows the up-link transmission of multi-channel biomedical signals from a patient to a hospital, using a mobile phone on a commercial cellular network. It includes a patient-interface processor with a Bluetooth link to a mobile telephone using the General Packet Radio Service (GPRS). The prototype processor transmits data for any or all of the four most common parameters used in medical monitoring, i.e. electrocardiogram (ECG, from which the heart rate can be derived), blood pressure, temperature and oxygen saturation ($Sp0_2$). An embedded Bluetooth interface between the processor and a mobile phone enables signals to be transmitted from a patient to a hospital, or to a clinician on the move. A clinician can then monitor the received data, make an assessment, and inform the patient about the outcome. The design of the processor has been 'future-proofed' so that it can be easily upgraded, e.g. with high-bandwidth Third Generation (3G) networks. The software is coded such that any future upgrades will require only minimal changes. The complete system includes a hospital database server, which allows large amounts of data to be securely stored after its acquisition. [1, 2].

2. System Design

2.1 Overview of the Mobile Telemedicine System

The system contains the following links, as shown in Figure 1:

(i) a short-range Bluetooth link from the processor ('patient hardware') to a mobile phone;
(ii) a mobile cellular network (GPRS) to a base station, then to a clinician on the move ('doctor browser') or via a Packet Data Network (PDN) based on an Internet Protocol (IP), e.g. Internet Service Provider (ISP) to a hospital server;
(iii) a local area network to a clinician, e.g. hospital local area network (LAN);

(iv) the PDN/GPRS networks from the hospital server to a doctor browser.

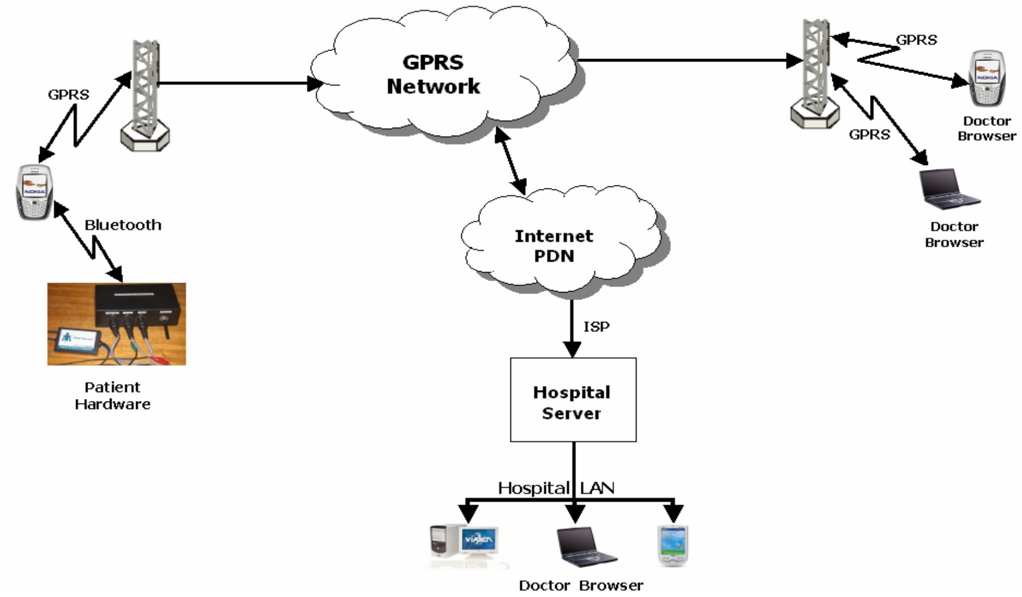


Figure 1 Multi-channel GPRS-based mobile telemedicine system

2.2 *Mobile Telemedicine Processor Module*

The prototype processor module has a proprietary operating system designed to support the acquisition of data from biomedical sensors [3]. The module samples the signals from sensors on the patient and transforms the digitised data into a Bluetooth packet structure for short-range transmission to a mobile phone. The telemedicine processor module comprises three sub-systems, as shown in Figure 2:

(i) *Biomedical sensors unit:* The unit receives multi-channel inputs from an ECG signal conditioning module, a temperature sensor, a blood pressure meter and a pulse oximetry finger probe. The sampling frequency is determined by the ECG bandwidth, which for monitoring applications is in the range of 0.5 to 50Hz [4, 5]. To limit the amount of multi-channel data to be handled by the limited Random Access Memory (RAM) capacity of the microcontroller, the sampling frequency is 150Hz, which corresponds to the minimum ECG sampling frequency for commercial ambulatory and monitoring systems.

(ii) *Microcontroller*: The main component of the processor module is an 8-bit Atmel T89C51AC2 microcontroller with an 18.432MHz crystal clock. This device also supports Atmel's FLIP (Flexible In-System Programmer), which allows compiled programs to be downloaded from a desktop computer. The main feature of the microcontroller is the on-chip memory capacity that includes a 32 kB Flash memory for program code, a 1 kB RAM for storing temporary variables, and a 2 kB EEPROM for retaining preset values while the power is disconnected. There is also a 2 kB on-chip Flash for the Atmel boot loader and for the Application Programming Interfaces (APIs) required for In-System Programming (ISP). The microcontroller also has an 8-channel analogue-to-digital converter (ADC) with 10-bit accuracy; four channels are configured as analogue inputs, and four are available as digital input-output connections. The microcontroller's interface to the Bluetooth Communication Module is a Universal Asynchronous Receiver Transmitter (UART).

(iii) *Bluetooth Communication Module:* A Bluetooth BlueWAVE RS232 PCB communication module, conforming to the Bluetooth version 1.1 standard, provides short-range wireless connectivity with a mobile phone [6]. The Bluetooth module is specified as power class 1 category with the highest output power of 100mW (20 dBm) and a typical operating range of 100m. Since the Bluetooth module is a slave module in a *piconet*, it is only capable of one active connection at a time. The mobile phone becomes the master device and controls the connection.

The connection uses the Asynchronous Connection-Less (ACL) Bluetooth link, intended to support data applications.

2.3 Sensors

(i) *Electrocardiogram:* The electrocardiogram (ECG) signal is acquired by a commercially available Vernier EKG/ECG Sensor module, which uses three silver-silver chloride (AgCl) electrodes [4]. This is a practical solution for remote monitoring by mainly non-medically trained personnel. The module has an instrumentation amplifier with a gain of 1000, i.e. it can amplify a 1mV typical body potential to 1V. The first electrode (green clip, negative) is placed on the inside of the right elbow; the second electrode (red clip, positive) is placed on the inside of the left elbow; the third electrode (black clip, ground) is placed on the right wrist to provide a reference point for the ECG baseline.

(ii) *Blood Pressure:* Another analogue input of the processor module is set to read non-invasive blood pressure (NIBP), i.e. systolic (maximum) pressure and diastolic (minimum) pressure. For the prototype system, this parameter was not measured directly but simulated using a potentiometer input.

(iii) *Temperature:* The temperature is acquired by measuring the voltage across an LM35 precision centigrade temperature sensor (National Semiconductors, Santa Clara, California). The output voltage, which increases linearly with temperature between 2º C to 150º C, is fed to an analogue input in the processor module.

(iv) *Oxygen Saturation:* Oxygen saturation values are obtained with an SpO_2 Finger Probe (Viamed, West Yorkshire) as an analogue input to the processor. Oxygen saturation is indicated as a percentage and normal values for a healthy individual are in the region of 97% to 99%.

2.4 Power Consumption

The processor module has two power supply options. One is a standard plug-pack wall socket using a Pro-Power IC Universal Regulator (Model PP500R), supplying 600mA at 9V supply voltage. The second is a 9V alkaline battery, as used during trials, which allows over an hour of continuous operation, i.e. equivalent to 6 or 7 10-minute transmission sessions. The module has physical measurements of 155 (length) x 90 (width) x 55 (height) mm and weighs about 300g. The average current drain of the microcontroller board is about 8.6mA when active, 6.8mA when idle, and less than 160 μA in power-down mode. For future developments, the ZigBee standard will be considered to optimise the needs of remote monitoring with very low power consumption (Zigbee Alliance, http://www.zigbee.org). The Bluetooth module is powered from the microcontroller board via a 5V DC source, which is regulated to 3.3V.

2.5 Mobile Phone

The most important features of the mobile phone are its Bluetooth connectivity, software development capability, and GPRS multi-slot class. The class defines the behaviour of mobile phones in terms of their operation, as specified by the European Telecommunications Standards Institute (ETSI) [7]. The majority of GPRS mobile phones are of Class B, which supports both circuit-switched and packet-switched services but the traffic exchange (either data or voice) must be only one at a time. During development, the model chosen was a Nokia 6600 Class B GPRS, multi-slot Class 6 phone. It was among the earliest available models, with two simultaneous up-link timeslots, the highest number available. It runs on the Symbian operating system and supports software application development with Java technology on a development platform that is widely available on data-enabled “smart” mobile phones, known as its Series 60 platform (Nokia Corporation, http://www.series60.com).

The GPRS mobile phone class and operation mode supported by mobile network operators determine the type of services and data available. The Class 6 multi-slot specification indicates that the Nokia 6600 phone has a maximum of four active time slots, which are the total number the phone can operate simultaneously for both up-link and down-link transmissions. The phone is capable of either three down-link and one up-link timeslots or two down-link and two up-link timeslots, the latter being the configuration used for patient data transmission. Since Class B supports either data or voice operations, a patient cannot make a voice call while simultaneously transmitting medical data. If the patient interrupts with a voice call, data transmission is suspended until the voice call is terminated.

The decision to use the Vodafone GPRS network was because it provides a roaming capability in over 29 countries, which in principle allows the system to be deployed in different parts of the world without any modification. The coverage of the Vodafone GPRS network is currently available over most of the UK and to over 99% of the UK population (Vodafone UK Ltd, http://www.vodafone.co.uk).

In the network architecture, the Serving GPRS Support Node (SGSN) and the Gateway GPRS Support Node (GGSN) are implemented together as a Combined GPRS Support Node (CGSN), and only supports the CS1 and CS2 type of GPRS channel coding schemes. CS2 is the most commonly used scheme because it provides better coding, while CS1 is mainly used during network congestion.

The connection configurations required on the mobile phone are specific to the Vodafone GPRS network settings. The main part is to configure the Access Point Name (APN), which is the point of entry to an external Packet Data Network from a GPRS network, as shown in Figure 1 [8]. The Nokia 6600 is configured to work with the Internet, therefore the GPRS connection is configured to work with the Vodafone Internet GPRS access point, which routes all the data packets from the mobile phone to the server at the hospital. When a secure connection is established, an acknowledgment message is sent from the server to the patient's mobile phone. Once a clinician has been identified to the server, the patient's data can be displayed using the internal LAN. Alternatively, a clinician who is away from the hospital may receive a patient's data via the GPRS network, either on a mobile phone, a laptop or a personal digital assistant (PDA).

3. Telemedicine Processor Module Software

The telemedicine processor module software was designed using Keil 8051 Microcontroller Development Tools (Keil Software, http://www.keil.com), which supports either 8051 assembler or C programming as a means of designing, simulating and debugging the program code for the microcontroller. The FLIP software is used to download the final program code into the Flash memory of the microcontroller. A Microsoft HyperTerminal on a standard desktop computer is used to communicate with and configure the Bluetooth module via a serial interface [6]. The main program for the processor module is implemented in assembly language. The program controls the whole operation of the module, including the transmission of medical data via the Bluetooth link. The program executes the same routine continuously until the system is reset, then the program reinitialises. This adds an additional safeguard should the system crash, for example due to a hardware fault. Initialisation involves setting up the clock register, the timer and interrupts, as well as the baud rate for the internal UART connection shown in Figure 2.

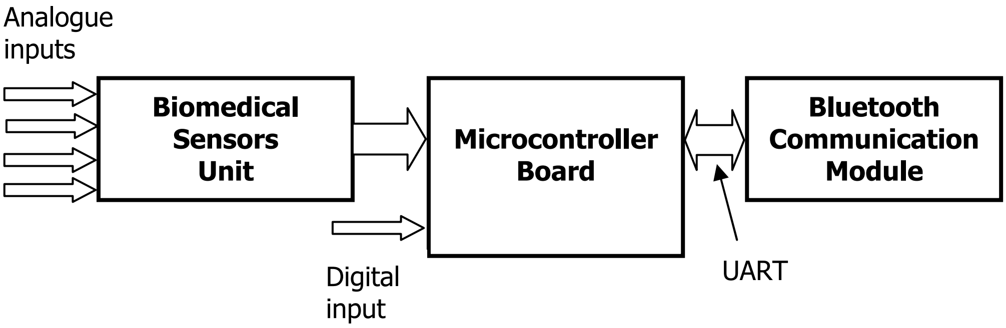


Figure 2 Mobile telemedicine processor module

The processor module performs routines sequentially on each of the four enabled channels of the microcontroller. These first identify the channel number and perform an analogue-to-digital conversion. The digitised data are then wrapped into a packet structure in hexadecimal format, as shown in Figure 3. The packet header (3 bytes) first indicates Channel 1 and the start of a new data stream (4 bytes). The trailer (1 byte) signifies the end of a packet and the whole packet. The packet is then sent via the UART port at 9600-baud rate to the Bluetooth module. This rate easily supports the required sampling rates, including the highest sampling rate of 150 Hz, with 10 bits per sample, for the transmission of the ECG. The same procedure is repeated for Channels 2, 3 and 4, and a synchronisation character is generated after the Channel 4 trailer.

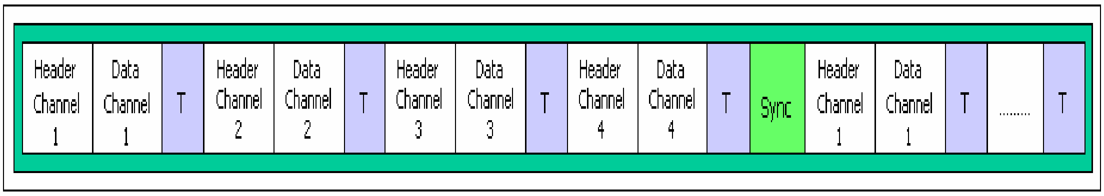


Figure 3 Digitised data output format of telemedicine processor

Standard link establishment protocols are required to make a connection between the mobile phone (the Master in a Bluetooth piconet) and the telemedicine processor module (the Slave). The module is set to a specific password, which the phone requests as one of the security features implemented to ensure authorised use of the system. When paired, a virtual serial port connection is established and data can then be transmitted in both directions.

3.1 GPRS and IP Protocols

GPRS is a telecommunications service that provides access to the Internet using the existing infrastructure of the Global System for Mobile Communications (GSM). A GPRS network allows the transmission and reception of data in an end-to-end packet transfer mode based on standardised network protocols supported by the GPRS bearer services. GPRS supports applications with external packet data networks based on Internet Protocol (IP) and X.25 network protocols [9]. IP-based protocols are widely used in the Internet, and are continuously being developed for providing better performance in wireless environments. IP is used as the network layer protocol for the GPRS backbone, for example to connect to a Serving GPRS Support Node (SGSN) and a Gateway GPRS Support Node (GGSN). A new protocol is defined in the backbone network, known as GPRS Tunnelling Protocol (GTP) [10]. Tunnelling is a two-way point-to-point process of transferring encapsulated data through a network from the point of encapsulation (the mobile phone, which adds 48 bytes of overhead to each packet) to the point of decapsulation (the hospital server). The new protocol is built on top of an IP network to handle mobile terminal mobility, and to support registration and authentication procedures. Each registered GPRS user who wants to exchange data packets with the IP network gets an IP address. The address can be either dynamic (e.g. the user's IP address is allocated from a pool of unused IP addresses every time the subscriber activates the access to an IP network) or static (e.g. a certain IP address is permanently allocated to a particular subscriber). GGSN is the first node that processes IP packets from a mobile terminal to an external Packet Data Network (PDN), for example an Internet host [11]. It means that GGSN is the first hub for every IP packet regardless of packet destinations.

In most commercial GPRS implementations, a mobile terminal is dynamically assigned a private IP address [12]. Therefore a Network Address Translation (NAT) server is required to translate the IP addresses of the packets delivered between the private IP address domain of the GPRS network and the public IP address domain of the external packet data network. The Internet Assigned Numbers Authority (IANA, http://www.iana.org) has reserved three blocks of IP address space for private networks; these are 10.0.0.0 to 10.255.255.255, 172.16.0.0 to 172.31.255.255, and 192.168.0.0 to 192.168.255.255. An example of the dynamic private IP address assigned during a particular connection is 10.16.6.124. The IP address of the application server is a static IP address, which is not presented here for security reasons.

The digitised medical data received by the mobile phone via its Bluetooth link is encapsulated into the required segments and frames based on GPRS radio access protocols. Details of the data flow process and the relevant transformations, which add overheads at each protocol layer, are available elsewhere [13]. Each final data block for transmission consists of a header, data and spare bits. The size of each block depends on the GPRS channel coding schemes (referred to as CS1, CS2, CS3, or CS4) for the transmission over the radio interface. The application program on the mobile phone is set to transmit 1036 bytes of medical data per IP packet, which is equivalent to 28 set of samples (37 bytes per sample) of multiplexed data from four analogue input channels. The choice of IP packet size is a balance between the overheads added by the protocols for each IP packet and the limited buffer size of the mobile phone processor. If the IP packet is too large, it cannot accommodate all the data before transmission. If it is too small, more overheads are added for the total number of IP packets required for transmission. The size of the IP packets also has a dominant effect on the performance of the GPRS network transmission. A small IP packet, around 100 bytes or less, has a substantially lower throughput compared with larger IP packet size [14]. The minimum number of blocks needed, rounded up to the nearest whole block, is:

No. of RLC Blocks = [Total User Data (bytes) + Protocol Overheads (bytes)] / RLC Data Unit (bytes)

The overheads for each block depend on the channel coding scheme, which for the Vodafone GPRS network is CS1 and CS2. For CS1, the overheads when transmitting 1036 bytes of data are:

$$\text{Overhead \%} = \frac{\text{Protocol Overheads (bits)} + [\text{ No. of RLC Blocks x RLC/MAC Overheads (bits)}]}{\text{Total User Data (bits)}} \times 100 = \frac{(55 \times 8) + (55 \times 21)}{1036 \times 8} \times 100 = 19.24\%$$

For CS2, the corresponding overheads are 17.80%. Clearly, the overheads for these schemes reduce the transmission performance, but for medical monitoring this is not a significant practical problem, as shown by the field test results described later.

3.2 Mobile Phone Application Software

Development of the mobile phone application software required specifications for the Java 2 Platform (Sun Microsystems, http://java.sun.com), the Symbian version 7.0 operating system for the mobile phone and the Series 60 platform (Nokia Corporation, http://www.forum.nokia.com).

The Sun Java platform architecture consists of three editions, the Java 2 Platform Standard Edition (J2SE), the Java 2 Platform Enterprise Edition (J2EE) and the Java 2 Platform Micro Edition (J2ME). The one adopted here is the J2ME platform, which only supports Java functionality meant for applications with limited processing power, i.e. for portable devices, such as PDAs and mobile phones. Currently, J2ME comprises the Connected Limited Device Configuration (CLDC), which defines a standard for mobile phones, and the Connected Device Configuration (CDC), which defines a standard for larger memory devices such as PDAs. The CLDC is defined in terms of a Mobile Information Device Profile (MIDP), which specifies the architecture and Application Programming Interface (API) needed for “mobile information devices”, which share common features such as small memory, small screen size and limited network bandwidth. Java specifications are given by standards known as Java Specification Requests (JSRs), of which there are about 70 currently available for J2ME. The most relevant is JSR 82, which standardises a set of Java APIs for Bluetooth applications. A MIDP-compliant mobile phone is conveniently programmed with an application development called a MIDlet. While a full account of the software development for the Series 60 platform is unnecessarily complex for this paper, it is appropriate to present a brief overview.

The Series 60 architecture comprises three functionality layers, namely Applications, User Interface and Enablers layers. The Enablers layer is for application execution and supports either

Symbian C++ or Java virtual machine environment. The Java environment in the Series 60 platform supports the Bluetooth API (JSR 82) for J2ME applications. Development based on J2ME specifications allows the application to be compatible with many mobile phone models manufactured by LG Electronics, Nokia, Sendo, Panasonic, Siemens and Samsung, which together sell millions of devices based on the Series 60 platform, including the Nokia 6600 used for this research.

There are four major software components, or "toolkits", needed to establish a development platform for J2ME applications. These are a Java 2 RunTime Environment (J2RE), an Eclipse Software Development Kit (SDK), a J2ME Wireless Toolkit, and an EclipseME plug-in for the Eclipse SDK (Sun Microsystems Inc, http://developers.sun.com; Eclipse Foundation, http://www.eclipse.org). The J2RE provides the Java virtual machine and all the minimum classes needed to run and execute a Java 2 program on a computer [15]. The SDK provides a universal platform for application development with multi-language capability, such as Java and C++. The Eclipse platform is therefore used to produce a Java-based, J2ME application for the mobile phone program in the mobile telemedicine system. The J2ME, as already mentioned, creates mobile applications based on CLDC and MIDP specifications and allows emulation and performance optimisation for mobile applications development [16]. It is integrated with the Eclipse SDK to create MIDlets for J2ME applications. The final main component is the EclipseME plug-in that helps to integrate the wireless toolkit to the Eclipse development environment and performs all the necessary configurations required in the Eclipse platform to develop J2ME MIDlets (Eclipse Foundation, http://eclipseme.org).

The first step in setting up the development environment is to install the J2RE, followed by the Eclipse SDK version 3.0.1, then the J2ME wireless toolkit using the Eclipse platform, which automatically integrates the wireless toolkit features into the Eclipse platform. The last part is to install the EclipseME plug-in via the Eclipse platform to configure it for developing J2ME MIDlets. The importance of adopting J2ME MIDlets is that the source code is compiled on a desktop computer but the MIDlet itself runs on a mobile phone. Therefore, the integrated emulator on the Eclipse platform provides standard mobile phone operations on the desktop. The emulator on the wireless toolkit, which appears as a generic mobile phone interface, allows the Eclipse platform to run the pre-verified MIDlet as an emulated J2ME MIDlet. If the MIDlet runs as expected, the next stage is to package the MIDlet into a Java Archive (JAR) to deploy the application into the real mobile phone. An additional file is also created, known as an Application Descriptor file, which describes the contents of the MIDlet JAR. Since the MIDlet is used with a Series 60 mobile phone model, the Application Descriptor file shows the CLDC 1.0 configuration and MIDP 2.0 profile that were configured to work with the Series 60 Platform.

4. Client-Server Features

In the context of the mobile telemedicine system, a patient from a given location becomes a client when accessing the telemedicine server at a hospital. The transport protocol is based on connection-oriented interactions, which means that a connection must be established between both ends of a transmission before either side starts to transmit any data. The overall client-server model of the system is shown in Figure 1, in which the patient hardware is the client side connecting to the hospital server via the GPRS network. The hospital server, shown in Figure 4, consists of a Telemedicine Application Server (Server A), which controls the telemedicine connections, and a MySQL Telemedicine Database Server (Server B), which stores the data processed by the Application Server. The Application Server is also a client (Client B) to the Database Server at the same time. The transmitted medical data are available to clinicians as and when required by accessing the database server as a client, either by connecting to the hospital LAN locally or via the GPRS network while on the move.

A client-server program has been developed by using a control property known as Windows Sockets (Winsock), which is part of the Visual Basic 6.0 toolbox called ActiveX Controls. Winsock defines a network programming interface that allows client-server applications to communicate using a TCP transport protocol. A communication control property of Visual Basic 6.0 provides a serial communication control to allow the client program to read input data from the telemedicine processor module, via a Bluetooth link to the phone, from where it is transmitted on the GPRS network to a server program at the hospital. A browser program is also produced for clinicians to view the data from the telemedicine database. These application programs, which allow interaction with a database, are possible with the database technology supported by Visual Basic 6.0, which can also generate a graphical user-interface [17].

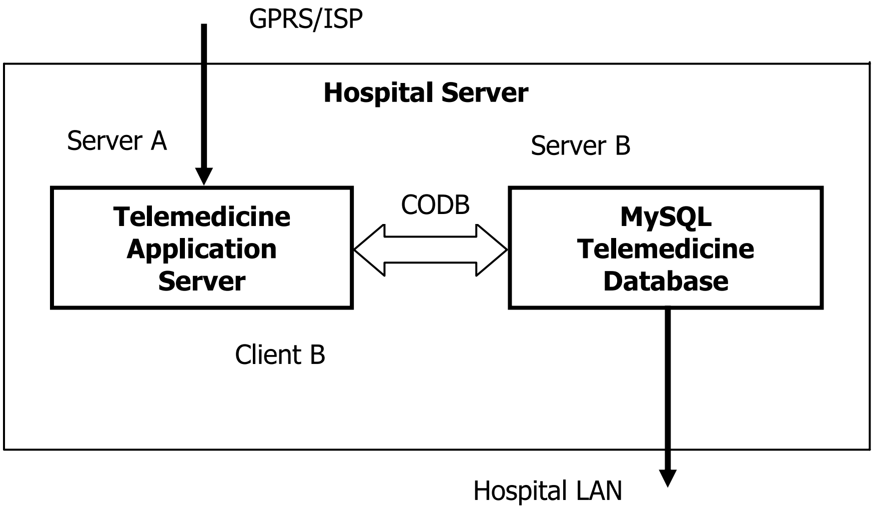


Figure 4 System architecture of the doctor browser interface

4.1 Telemedicine Application Server

The main function of the Telemedicine Application Server is to control telemedicine sessions. The server controls the client connection request from a patient user-interface program and responds accordingly to accept the request and establish a secure point-to-point link with the patient. It also communicates with the MySQL Telemedicine Database to store the time and date of a session and to create an entry in the database to store the multi-channel medical data [18, 19]. The Application Server also provides a graphical user-interface for healthcare providers to monitor and to administer the mobile telemedicine system. Some of the administrative functions include adding a new patient to the system, updating or deleting existing patient information and monitoring the number of sessions performed by a patient. A full description of the functioning of the Telemedicine Application Server is provided elsewhere [18]. In the interests of brevity it will not be further described here.

4.2 Telemedicine Database

The MySQL server consists of two main databases. One holds the user account and administrative information, such as the usernames, passwords, host location and user privileges. The administrative privilege is for the system administrator, and user privileges vary according to the level of authority or level of usage authorised. For example, a clinician may be granted with privileges to update, delete and retrieve data. A patient may only be granted limited access to retrieve data. The other is a database for access by clinicians to view the stored data when they need to. The database consists of three tables, as shown in Figure 5. The Patient Information Table provides details such as a unique patient identification number (patient ID), name, address, age and sex. The Sessions Table provides information on telemedicine sessions created by a patient and consists of session number, patient ID, time and date. The Signal Sets Table stores the medical data transmitted during each session and consists of session number, patient ID, and data from four channels.

A ‘primary key’ is needed to identify uniquely a row of data elements in a table. The patient ID is a primary key in the Patient Information Table, which means that there can be no duplication of patient ID. A combination of session number and patient ID is the primary key for both the Sessions Table and the Signal Sets Table. There can be many sessions for the same patient ID in the Sessions Table but no duplication of session number for the same patient ID. Exact conditions also apply to the Signal

Sets Table to avoid duplication or redundancy of stored data. The patient ID in the Sessions Table is a 'foreign key', i.e. the patient ID in the Sessions Table is the primary key of the Patient Information Table. The Signal Sets Table stores the multi-channel data instead of using the Sessions Table. Although both tables have the same session number and patient ID fields, a separate table for the data is essential to minimise the fields and entries in the Sessions Table, since it is retrieved extensively in the Telemedicine Application Server as well as the Doctor Browser application program.

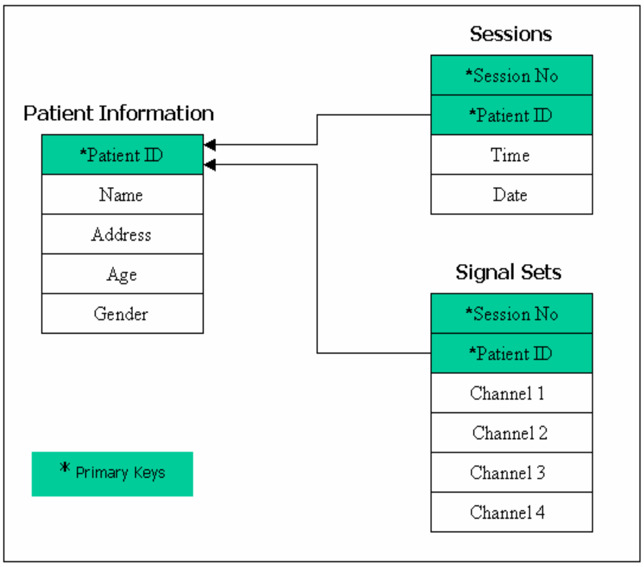
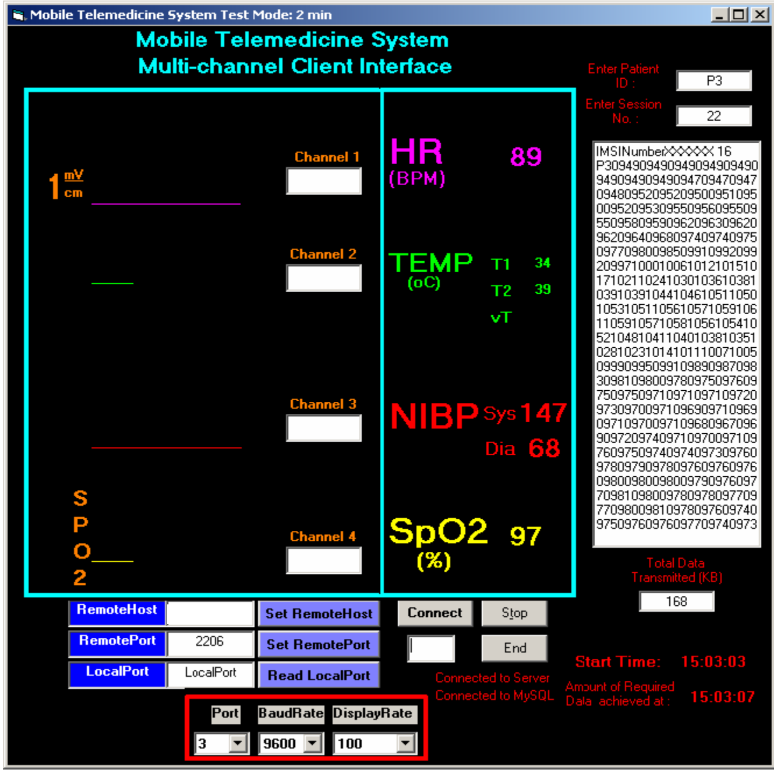


Figure 5 Relationships between telemedicine database tables



HR (BPM) 89

TEMP (oC) T1 34 T2 39 vT

NIBP Sys 147 Dia 68

SpO2 (%) 97

Enter Patient ID : P3

Enter Session No. : 22

```
IMSINumberXXXXXX 16
P3094909490949094909490
94909490949094709470947
09480952095209500951095
00952095309550956095509
55095809590962096309620
96209640968097409740975
09770980098509910992099
20997100010061012101510
17102110241030103610381
03910391044104610511050
10531051105610571059106
11059105710581056105410
52104810411040103810351
02810231014101110071005
09990995099109890987098
30981098009780975097609
75097509710971097109720
97309700971096909710969
09710970097109680967096
90972097409710970097109
76097509740974097309760
97809790978097609760976
09800980098009790976097
70981098009780978097709
77098009810978097609740
97509760976097709740973
```

Total Data Transmitted (KB) 168

RemotePort 2206

LocalPort LocalPort

Connected to Server
Connected to MySQL

Start Time: 15:03:03

Amount of Required Dala achieved at : 15:03:07

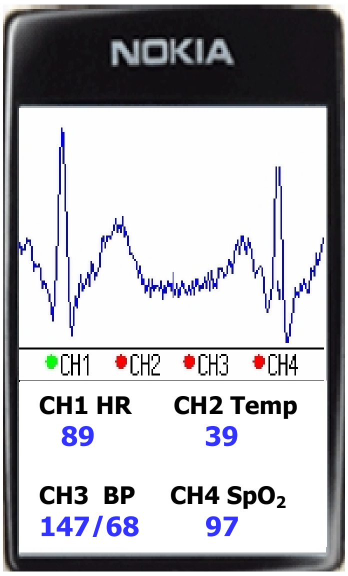
Port 3 BaudRate 9600 DisplayRate 100

Figure 6 Hospital server's patient interface

4.3 Patient Interface

The hospital server's patient interface program provides a graphical display that is used to facilitate a connection to the mobile telemedicine system. The interface comprises sections that perform various functions of the program, as shown in Figure 6. The patient (or paramedic or clinician) enters a password to establish a connection with the server. The patient interface program is acknowledged once the server has granted the connection request. An option appears on the interface waiting for the patient to start the telemedicine session. The data display section is where the decoded data from the hardware is displayed in both a graphical format and a numerical format, with each channel displayed on a separate window. Once the server receives and recognises the end-of-transmission characters, a message is sent back to the client program to acknowledge that all the data transmitted have been received and the server closes the point-to-point connection with the patient.

For the convenience of the patient, paramedic or clinician at the sending end, the data to be transmitted can be displayed on the mobile phone [20]. The application program in the mobile phone reads the multi-channel medical data from the telemedicine processor module via its Bluetooth link, then transmits it to the Telemedicine Application Server at the hospital via the GPRS network. Due to the limited buffer capacity and the much smaller screen on a mobile phone than on a desktop or laptop computer, there is a practical limit as to what can be displayed. Figure 7 shows an ECG signal displayed on the mobile phone, together with numerical values for the heart rate (Ch. 1: 89 beats/minute), temperature (Ch. 2: 39.0 degrees Celsius), non-invasive blood pressure (Ch. 3: 147/68 mm Hg), and oxygen saturation (Ch. 4: 97%).



CH1 HR 89

CH2 Temp 39

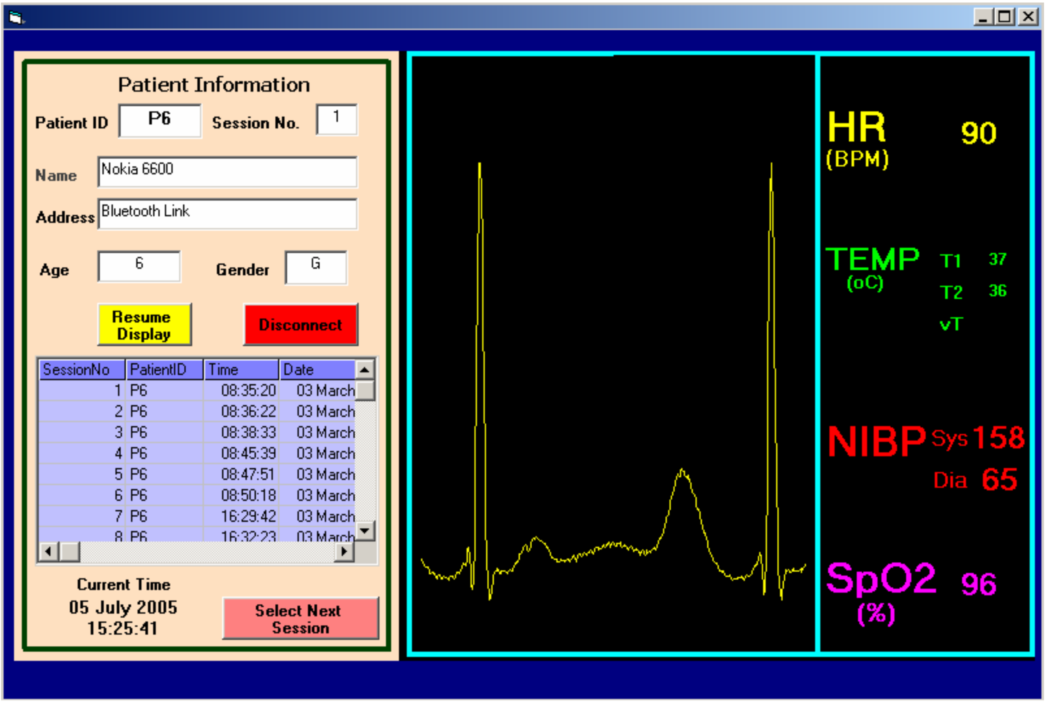
CH3 BP 147/68

CH4 SpO_2 97

Figure 7 Mobile phone browser interface, showing an ECG signal and numerical data on four channels. Real-time data viewed at the sending end by the patient; stored data viewed via the server by a clinician

4.4 Doctor Browser Interface

The doctor browser interface program produces a graphical display to allow a clinician to access data from mobile telemedicine sessions. The program allows a clinician to retrieve patient information, session information and medical data from the telemedicine database, and to exercise administrative privileges. The browser program can be used with a laptop or portable devices such as a mobile phone or a PDA. The clinician may use the browser program from either a fixed-IP network (a LAN, Dial-up internet or Broadband internet from a service provider) or the GPRS network while on the move. The architecture of the doctor browser interface program is shown in Figure 1. The browser developed for the system has two user-interfaces, one for a program running on a mobile phone, shown in Figure 7, and one for a program running on a laptop, shown in Figure 8. The main advantage of the browser program is to allow a clinician to use a portable device to access information from the mobile telemedicine system while on the move. Further details are available elsewhere [18].



Patient Information

Patient ID: P6
Session No.: 1
Name: Nokia 6600
Address: Bluetooth Link
Age: 6
Gender: G

SessionNo	PatientID	Time	Date
1	P6	08:35:20	03 March
2	P6	08:36:22	03 March
3	P6	08:38:33	03 March
4	P6	08:45:39	03 March
5	P6	08:47:51	03 March
6	P6	08:50:18	03 March
7	P6	16:29:42	03 March
8	P6	16:32:23	03 March

Current Time
05 July 2005
15:25:41

HR (BPM) 90
TEMP (oC) T1 37 T2 36 vT
NIBP Sys 158 Dia 65
SpO2 (%) 96

Figure 8 Laptop browser interface, showing ECG signal stored in the telemedicine database

4.5 Security Issues

Security is an issue of concern in mobile telemedicine applications and is the subject of ongoing debate in medical circles. While the main security emphasis is to protect the entire server host network against all types of possible attacks, i.e. eavesdropping, altering, playback, and denial of service, a further measure is to encrypt the medical data to ensure a secure connection and to protect medical confidentiality. The Bluetooth link between the telemedicine processor module and the patient's mobile phone implements the 128-bit authentication key and 8-128 bit encryption key security [21]. In addition, the 15-digit International Mobile Subscriber Identity (IMSI) number is used to identify a patient making a mobile phone connection because it is unique to the Subscriber Identity Module (SIM) card present in the mobile phone. The IMSI number is composed of three parts:

(i) Mobile Country Code (MCC), which consists of three digits to identify the country of residence of the mobile subscriber (234 for the UK);
(ii) Mobile Network Code (MNC), which consists of two or three digits to identify the home mobile network of the mobile subscriber (15 for the Vodafone network);
(iii) Mobile Subscriber Identification Number (MSIN), which identifies the subscriber within a mobile network (not presented here for reasons of confidentiality).

An International Mobile Equipment Identity (IMEI) number, which uniquely defines a particular mobile phone, is also available to provide a secondary identification of the patient. The combination of IMSI and IMEI numbers produces a secure and reliable patient identification feature. The data transmission between the patient's phone and the hospital are protected by the security protocols implemented in the GPRS network. The three-tier security, with the A3 algorithm for user authentication, A8 ciphering key generating algorithm and A5 ciphering algorithm for data encryption security features in the GPRS network, ensures a secure session for mobile telemedicine applications.

As the system presented here is a prototype whose functionality is the main issue, any legal implications of medical data digitisation, compression, encryption, transmission and storage are not discussed.

The security of the Telemedicine Application Server, the Telemedicine Database Server and the browser programs is implemented based on the MySQL server security model. In general, MySQL implements security based on Access Control Lists for all connections, queries and other operations [17, 19]. Clinicians and other medical staff gain access on a privilege basis, which ensures that they perform only operations deemed permissible by the system administrator.

5. Performance Analysis and Technical Results

5.1 Performance of the GPRS Network

The allocation of physical channels in the GPRS network is done dynamically according to "capacity on demand" in each radio cell. The flexibility of GPRS channel allocations means that the network does not require permanently allocated packet data traffic channels [22]. Several studies have investigated GPRS performance based on a network simulation model, an analytical model and a practical evaluation approach [23-25]. The results of these studies are used as benchmarks in evaluating the performance of the mobile telemedicine system in terms of the GPRS network performance and focuses mainly on the data throughput, error control techniques and packet delays.

The data throughput performance depends on the choice of coding scheme, e.g. CS1 allows a data rate of 9.05 kbps, while CS4 allows the highest data rate of 21.4 kbps. In practice, the network dynamically switches among coding schemes to achieve the highest throughput [14]. As most GPRS operators use only CS1 and CS2, we observed much lower uplink throughputs than the theoretical values during the field tests. This was due to variations in data packet sizes, mobile channel conditions and slot contentions from multiple users; none of these variables is user-controlled.

Packet loss and packet delays depend on channel effects, such as shadowing and multi-path fading. Forward Error Correction (FEC) and Automatic Repeat Request (ARQ) are two error control techniques implemented in the GPRS network [25]. In FEC, redundant bits are added to information bits to detect and correct any channel-induced errors. In ARQ, error control is achieved through the re-transmission of corrupted data packets. Channel conditions determine the coding scheme to be used and therefore the number of packets required in the data transmission. The Carrier-to-Interference ratio (C/I) indicates the Bit Error Rate (BER): if C/I indicates a channel with a high BER the RLC protocol has to use strong coding redundancy to assure good error resilience and requires more packets for the transmission. If C/I represents a low BER the coding redundancy can be smaller and fewer packets are required to transport the same IP packet. The coding scheme adopted for a particular GPRS data transmission, based on the C/I ratio, provides channel protection at the physical layer. In addition, unsuccessful error correction at the physical layer is later handled by the ARQs at a higher layer. One important aspect of ARQ is the influence of re-transmission time on the packet delay [23]. Due to the ARQ implementation at the RLC layer, packets are rarely lost, just delayed. The combination of FEC and ARQ techniques improves the GPRS link in terms of packet error and packet loss. Packet loss does occur in both the down-link and up-link directions, but the incidence is relatively rare and hard to quantify.

At the application layer, it is difficult to distinguish between a packet that appears to be lost due to large delays and a truly lost packet. The packet delays in GPRS data transmission are reflected by the time taken to complete a transmission. In the mobile telemedicine application, it is important to receive securely all the transmitted medical data, which are stored in flexible, predetermined time slots. Delays are less critical than for real-time applications because in most scenarios the data are viewed off-line.

The GPRS performance is limited by processing and queuing delays in each protocol stack; buffering delays caused by the data collection for packet formation; delays while accessing radio resources; delays caused by the re-transmission of lost radio blocks and packets; and delays due to mobility management. Delays in the base station sub-system onwards and in the GPRS backbone network are small relative to delays in the radio interface, which are directly related to the size of the transmitted packets [26]. This theoretical overview on possible sources of transmission delays in the GPRS network is relevant in explaining the outcome of the mobile field tests described below.

5.2 Mobile Field Tests

Since the prototype of the mobile telemedicine system was tested with the Vodafone GPRS network, the coding schemes adopted were assumed to be CS1 and CS2. As detailed below, the data were transmitted in two ways to the Telemedicine Application Server in our laboratory.

(i) *Single-channel transmission* of a clinical ECG record from the MIT-BIH arrhythmia database stored as a file in a laptop computer [27]. The record was MIT-BIH record 103 for a male patient. The original lead configurations used were a modified limb lead II (MLII) and a modified lead V2, obtained by placing the electrodes on the chest. The laptop transmitted the data via a Bluetooth link to a mobile phone, used as a modem, and hence to the GPRS network. This was a convenient arrangement for initial tests because it was not necessary to attach sensors to a person and therefore the telemedicine processor module was not required.

(ii) *Four-channel transmission* of real-time medical data from a patient. The data were obtained from the four analogue inputs of the telemedicine processor module, then transmitted via its Bluetooth link to a mobile phone and hence to the GPRS network. The ECG signal was obtained by placing the electrodes on the patient with the Lead I ECG configuration. The other three input signals were from a temperature sensor, a potentiometer to simulate blood pressure, and another potentiometer to simulate oxygen saturation. This arrangement allowed a practical assessment of the system's capability for various "real world" medical scenarios.

Mobile tests were performed to observe any variation in the performance of the prototype system based on the mobile environment, the total amount of data, the multi-slot classes and the time of day. The performance was measured in terms of the total time taken to complete a transmission, referred to as completion time, the average throughput of a GPRS transmission, and verification that the total data received matched the total data transmitted. The mobile tests represent how the system might be used in telemedicine sessions and include possible variations in mobile environments that reflect patient mobility situations. The performance of the system was investigated in four different mobile environments that reflect possible patient mobility patterns in the real world. The tests were performed in an indoor location, and in a car in a town centre, along country roads and along motorways at high speed. These variations in mobile channel environment represented differences in path delays, mobile speed and noise or interference conditions of the GPRS network. An attempt was then made to compare the experimental and theoretical results.

The variation in the total data reflects the duration of the recorded signals being transmitted. The ECG signals from the MIT-BIH database were compiled in a data format such that half the recording was from one lead and half from the other. For example, the 84 Kbytes data sample represents 60 seconds of recorded ECG signals, with 30 seconds of signals from the MLII lead and 30 seconds of signals from the V2 lead. The data size variations listed in Table 1 for the mobile tests with the patient were 'short' (14 Kbytes of data), 'medium' (84 and 168 Kbytes) and 'long' (337 and 506 Kbytes).

The variation in terms of the GPRS multi-slot classes was achieved by using different mobile phone models during the tests. The three models tested were expected to produce some variation in the measured values of completion time and average throughput. The models and their corresponding multi-slot capabilities used in the mobile tests are listed in Table 2, in which the last column, showing the multi-slot capability, represents the number of down-link channels followed by the number of up-link channels, e.g. '4+1' means 4 down-link and 1up-link.

Since the allocated GPRS radio resources are based on subscriber requests and the resources may be decreased due to slot contention from multiple subscribers, the variation in terms of the time of day was also taken into consideration. The availability of GPRS radio resources during peak hours (12 noon and 4 p.m.) and off-peak hours (9 p.m. and during a weekend) may therefore affect the average throughputs and the completion time.

Data Size (Kbytes)	**ECG Duration**
14	10 s
84	60 s
168	2 min
337	4 min
506	6 min

Table 1 Data size variations used in the mobile tests

Mobile Phone Model	**Mobile Class**	**Multi-slot Capability**
Sony Ericsson T610	Class 8	4+1
Nokia 6600	Class 6	3+2
Nokia 6630	Class 10	4+2

Table 2 Mobile phone models used in the mobile tests

In general, test parameters were varied one at a time. For example, the data size was varied to indicate different monitoring durations while maintaining the other parameters constant. The completion time and average throughput were then measured, and any packet loss was verified. The measured values were stored in the telemedicine database for further analysis. Acknowledgement messages were sent by the server to the mobile phone on receiving a complete data set. A summary of the different mobile environments, parameters varied and the type of medical data is shown in Table 3.

Mobile Environment	**Parameter Variation**	**Medical Data Type**
Indoor	Data Size Multi-slot Classes Time of Day	MIT-BIH ECG Records Real Multi-channel Data
Town/Urban	Data Size Multi-slot Classes	MIT-BIH ECG Records Real Multi-channel Data
Countryside	Data Size Multi-slot Classes	MIT-BIH ECG Records Real Multi-channel Data
Motorway	Data Size Two Multi-slot Classes	MIT-BIH ECG Records

Table 3 Parameters varied in each mobile environment

The average throughput of the transmission was also displayed by the Telemedicine Application Server in real-time. The incoming data arrivals in the server were averaged over fixed 5-second intervals that enabled variations in average throughput to be observed. Since the calculation for the average throughput was based on the incoming user data (the payload of IP packets), the protocol overheads involved in the GPRS network and packet data network transmission were not taken into account. Therefore, the average throughput measured by the server estimated the effective data rate, which was lower than the total data rate for the GPRS network. In addition, the incoming multi-channel medical data from a patient were also displayed on the Telemedicine Application Server, together with its corresponding analogue signal displays.

In the first mobile test set-up with the laptop the procedure was as follows:

(i) the patient interface program sent a connection request to gain access into the Telemedicine Application Server in the laboratory;

(ii) once the point-to-point connection with the server was established, the server sent a notification to the patient interface to notify successful connection;

(iii) the patient interface then started the data transmission and was set to transmit the digitised ECG data stored in the laptop. The start time of the transmission was recorded from the first packet being transmitted; the end time was recorded when all the data were received by the server. The patient interface then received an acknowledgement from the server upon successful transmission of all the ECG data. These procedures were repeated for sending different data sizes representing different monitoring durations. For the variation in mobile environments, the set-up was tested at different locations, at different mobile speeds and with phones having different multi-slot classes.

The second set-up for the mobile tests was to transmit real-time four-channel medical data from a patient using a mobile phone. Four analogue inputs of the telemedicine processor module were connected and the digitised data from the processor module were then transmitted to the mobile phone. The patient interface program on the phone handled the Bluetooth link establishment with the processor module and the GPRS data transmission to the Telemedicine Application Server. The start time was recorded when the first packet from the patient was received by the server. The end time was recorded when the end of transmission characters were received by the server, i.e. as soon as the patient pressed the phone's 'stop session' option.

A field test table was created in the database to store the total data transmitted, date, start time and end time of the transmissions. The completion time of a transmission was the duration between the start time and the end time recorded. It is acknowledged that the method used did not give a precise measurement of the transmission time of the data packets. However, the same method was consistently used throughout the mobile tests to estimate the transmission time. The estimated throughput for the transmission was produced by dividing the total data transmitted over the completion time recorded. The system performance results based on variations in the test parameters are presented according to the set-up used and the data transmitted.

5.3 Transmission of Single-Channel Data

In the first mobile tests, single-channel data were obtained from a CD-ROM on which the MIT-BIH Arrhythmia Database was stored. These tests were performed using two mobile phones:

(i) GPRS Class 8 (4+1 slots) Sony Ericsson T610, i.e. with one up-link transmission slot;
(ii) GPRS Class 6 (3+2 slots) Nokia 6600, i.e. with two up-link transmission slots.

For brevity, and to avoid the repetition of similar graphs, only a selection of the test results with the Class 6 mobile phone is presented here. Two types of plots were produced:

(i) Completion time in seconds (defined earlier) as a function of total data in Kbytes (i.e. 1024 bits);
(ii) Throughput in bits per second as a function of total data in Kbytes.

In general, and as expected, the completion time increased with the increase in data size. Among the four mobile environments, the indoor environment had the fastest completion time for the transmission of all the data sizes. There was no significant difference in the completion time among the four mobile environments for data transmissions of less than 100 Kbytes. The transmissions of data sizes larger than 100 Kbytes performed under the countryside mobile environment were far more distinct, requiring a longer completion time compared to the other mobile environments.

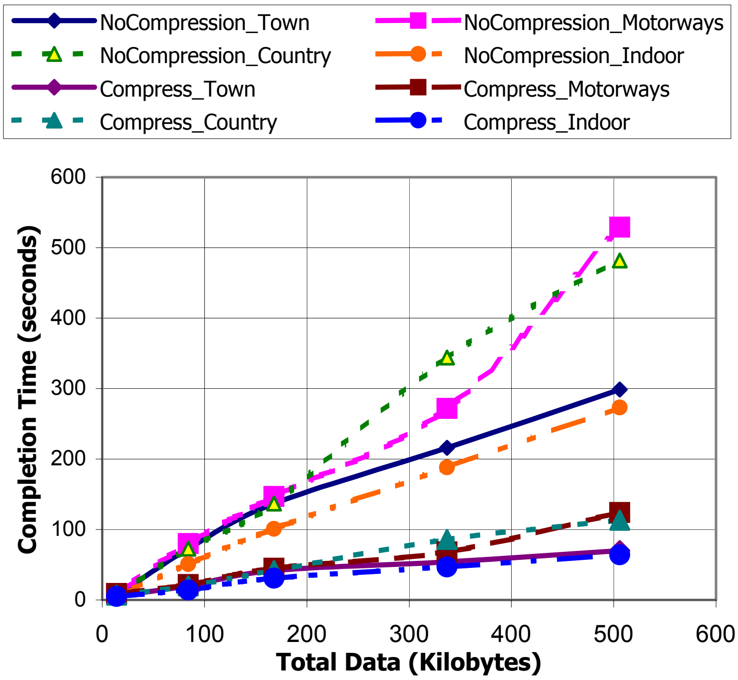


Figure 9 Completion time for various data sizes and different mobile environments, GPRS Class 6 mobile phone, with and without compression

Figure 9 shows the total completion time for various data sizes, for the four environments, using the Class 6 mobile phone. The expected pattern of increasing completion time with data size increase was consistent in all the tests. This was also true for the Class 8 phone, although not shown here. However, the range of estimated throughputs for both tests was much higher than the theoretical GPRS throughputs, even under ideal conditions. For example, the maximum possible throughput for the Class 6 phone with CS2 coding is 26.8 kbps (2 x 13.4 kbps per timeslot). Thus, with this throughput, the transmission of 337 Kbytes of data would require a completion time of at least 103 seconds for the indoor environment, yet from the results it was only 47 seconds. Theoretically, this was only possible with the implementation of a data compression mechanism during the transmission. After a thorough investigation, it turned out that the commercial software used for accessing the GPRS network on the laptop interface was defaulted to a compression mode. Users therefore have no information on the exact compression ratio implemented. Figure 9 shows that with the compression mode off a completion time for 337 Kbytes of data was now 188 seconds for the indoor environment, which is closer to the theoretical minimum of 103 seconds for two up-link timeslots with CS2 coding. Further, Figure 9 shows that with compression, the completion times for all four mobile environments were close together for data sizes of up to around 200 Kbytes (fastest for the indoor environment) before starting to spread out for larger data sizes. The completion times were 60-65s for the indoor and town environments, and 110-120s for the country and motorway environments. The reason for these variations is not clear, although it may be a function of the network cell sizes and the locations of base stations. Without compression, the total completion times for various data sizes were all significantly longer.

Figure 10 illustrates the estimated throughputs for the Class 6 phone. With compression, the throughputs were 30-40 kbps for the country and motorway environments and 45-65 kbps for the indoor and town environments. Without compression, the throughputs were 7-10 kbps for the country and motorway environments and 10-15 kbps for the indoor and town environments. A high throughput of about 24 kbps was recorded for the indoor environment, which was close to the maximum possible throughput of 26.8 kbps with CS2 coding.

The results match the expectation of the performance difference when two different GPRS mobile classes were used. The completion times for the Class 6 phone were approximately half those for the Class 8 phone and correspondingly the average throughput for the Class 6 phone was almost double that for the Class 8 phone.

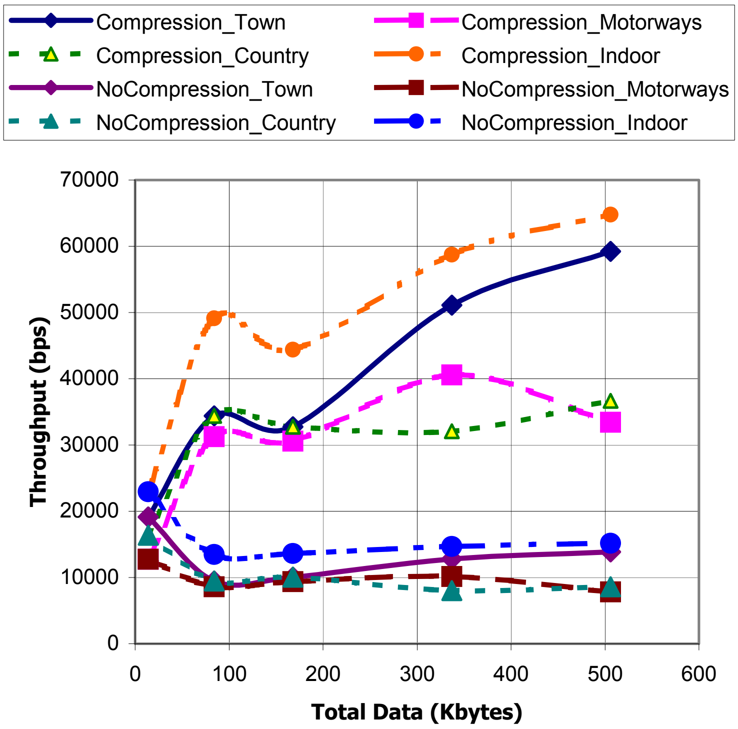


Figure 10 Throughputs for various data sizes and different mobile environments, GPRS Class 6 mobile phone, with and without compression

5.4 Transmission of Four-Channel Data

In the second mobile tests, data on four multiplexed channels were obtained directly from a patient and transmitted using a Nokia 6630 Class 10 (4+2) mobile phone, i.e. with two up-link transmission slots. The indoor tests were performed in the laboratory of our department. The outdoor tests took place around a town centre, then on country roads, from stationary to cruising speeds, mostly in rainy weather.

The variation in the total amount of data transmitted from the patient represents short, medium and long duration telemedicine sessions. For the short duration session, 20 to 50 Kbytes of data were transmitted; for the medium duration session, 90 to 110 Kbytes; and for the long duration session, 200 to 220 Kbytes. The choice of the total amount of data transmitted was made on the basis that these would be reasonable data lengths for monitoring sessions. The actual total data are clearly variable, depending on the specific medical requirements.

Figure 11 shows the completion time for the transmissions of four channels of data for both indoor and outdoor mobile test environments. The indoor completion time was marginally faster than the outdoor time for most of the transmissions, although the reason for this is not clear. The completion time for the indoor transmissions increased linearly with the increase in total data size.

Figure 12 shows the estimated throughputs in kbps for the different sizes of data blocks. The throughputs for the indoor tests were almost constant, with only a small spread of 5.5-6.5 kbps. The throughputs for the outdoor tests were generally lower than those for the indoor tests and were typically 4-6 kbps. These throughputs observed for the transmission of real multi-channel medical data from the patient for the indoor and the outdoor environments were much lower than the possible up-link throughputs for the Class 10 mobile phone, which is capable of up to 26.8 kbps for up-link transmissions.

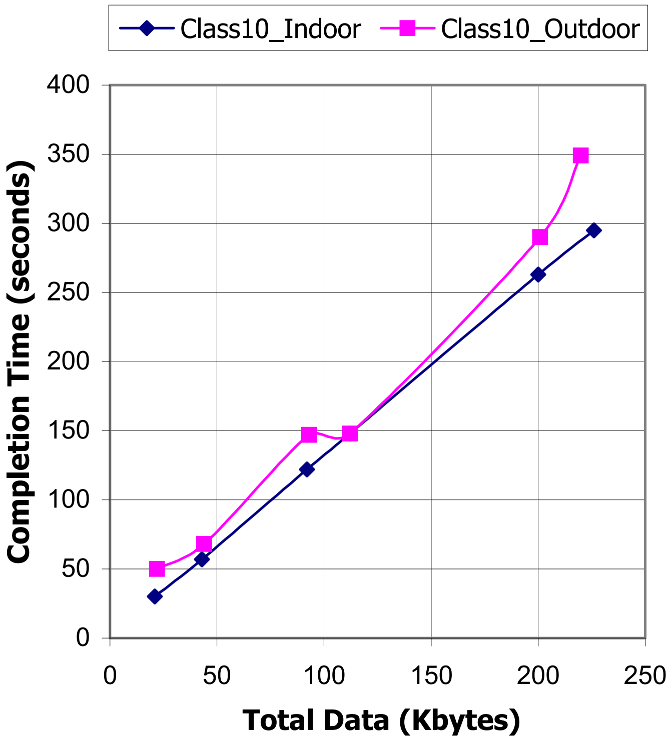


Figure 11 Completion time of four-channel data, GPRS Class10 mobile phone

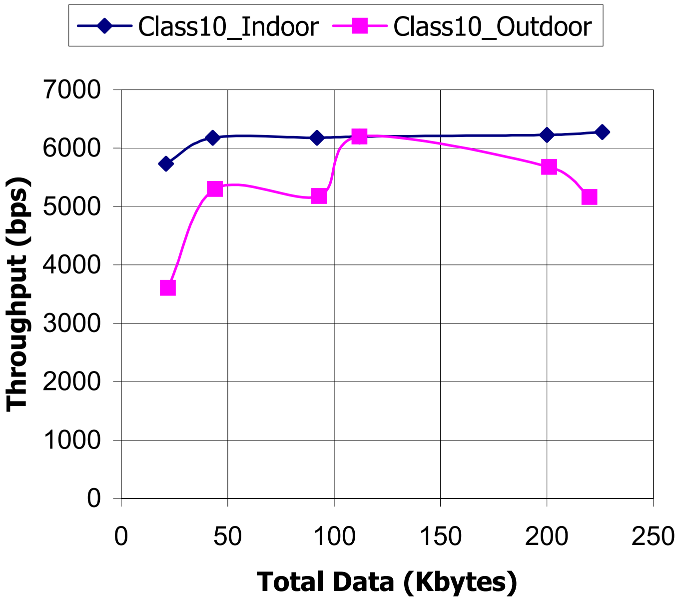


Figure 12 Throughputs of four-channel data, GPRS Class10 mobile phone

The significantly low throughput observed in the tests was most likely due to the packet size transmitted by the mobile phone. Since the GPRS throughput is directly related to the size of IP packets set for a transmission, small packets have substantially lower throughput compared with larger packets. Based on observations of incoming IP packet sizes on the Telemedicine Application Server during the indoor tests, the average payload size for each packet was between 200 to 300 bytes. This showed that the mobile phone was transmitting a small amount of data per packet.

The size of IP packets transmitted by a mobile phone is controlled by its hardware configuration. The J2ME platform on the phone is responsible for handling incoming data from the Bluetooth link and preparing the data for GPRS transmission. The MIDP profile in the J2ME platform includes networking support and defines the socket stream connection interface. The sending buffer size option informs the low level networking code about the intended usage patterns that the application will use with the socket connection. The phone may then adjust the buffer sizes to account for better throughput based on the data from the current network information. Since the preliminary tests yielded acceptable data transmission, no optimization of the buffer size was attempted.

6. Discussion

The mobile tests performed have shown successful transmissions of single-channel and four-channel medical data via the GPRS network. The tests in four different environments reflected a variety of possible patient mobility scenarios. As expected, the performance in the indoor mobile environment had faster completion times and higher throughputs than other mobile environments. The mobile test interfaces and procedures were designed to best estimate the completion times and data throughputs of the GPRS network. The consistency in the measurement techniques throughout the field tests is essential in producing results that reflect the general pattern in the mobile telemedicine system performance. The mobile tests also showed consistency with the expected theoretical results in the behaviour, throughputs and data quality of the GPRS network. However, it is acknowledged that more precise and accurate measurements on the GPRS network are possible with the use of commercial and portable air interface test tool solutions for verification, maintenance, and troubleshooting of mobile networks. These tools are normally used by mobile network operators for maintenance and optimisation. For example, the TEMS™ Pocket GSM and the TEMS™ Investigation 6.0 test tools from Ericsson are capable of giving details and precise GPRS traffic performance measurements such as the data throughput for up-link and down-link, the percentage of the data blocks re-sent/erroneously decoded, GPRS status details and other performance parameters (Ericsson, http://www.ericsson.com).

With the randomness of radio resources availability in the GPRS network, the results obtained in these tests may vary slightly if all the tests were to be repeated. Due to the flexible allocation of GPRS physical channels, based on the common pool of available resources in a radio cell at any particular time, there is some variation in throughputs of the mobile telemedicine system. The results may also be influenced by non-ideal radio conditions such as traffic congestion, frequency interference, poor radio coverage and weather conditions. The highest average throughput observed in the mobile tests was only between 16 to 20 kbps under the indoor environment, using the two up-link timeslots mobile class, although transmission with a small data size (about 15 Kbytes) did give a throughput of around 24 kbps. With the limitation of only the CS1 and CS2 coding schemes implemented by most GPRS network operators, the uplink throughputs of the mobile telemedicine system can only reach the maximum theoretical data rate of 26.8 kbps.

Although the transmission delays and data throughputs vary according to the mobile environment of a remote patient, the mobile telemedicine system has significantly demonstrated the real-time possibility with the display of incoming signals at the Telemedicine Application Server. As the IP packet network only offers a 'Best-Effort' service and does not retain the timing information, integration with a Real-Time Transport Protocol may improve the real-time capability, as used in video communications over GPRS [28].

With the successful up-link transmission of multi-channel physiological signals via the GPRS network, the mobile telemedicine system has demonstrated the practicality of integrating healthcare delivery with wide area connectivity using a mobile phone. The system also reflects the role of mobile technologies in medical connectivity developments for monitoring and on-the-move m-Health applications [29].

7. Conclusions

In this paper we have presented a novel telemedicine system for transmitting four channels of biomedical signals from a patient to a hospital via a mobile phone. A processor module samples signals from up to four sets of sensors attached to the patient and transmits the digitised data over a Bluetooth link to a mobile phone that is configured to work with the Vodafone GPRS network in the UK and controls the network connection with a hospital server. The main reason for adopting the commercially available GPRS network is that it is currently available over most of the UK. The second factor is the availability of many GPRS mobile phone models with large memories and sophisticated displays, and with different multi-slot classes. Although high-bandwidth Third Generation (3G) cellular networks are readily available, the availability of the 3G networks and 3G mobile phones is still limited at this stage.

The data transmission includes the IMSI number, which is unique to the SIM card in the mobile phone and is part of the security features developed for the system, although clinical security has not been a main issue in this research. The data are made available to a clinician, as and when required, from the server's database. The telemedicine system has been tested in a variety of environments, including indoors and in a car driving in a town centre, on country roads and on a motorway. In all these scenarios, the system has been shown to be capable of transmitting a variety of data block sizes without any detectable errors. The tests have thus shown the capability of the GPRS network to transmit medical data reliably and at rates that are acceptable for the transfer of four channels. The clinical integrity of the received data has not been analysed, although the sampling rates of four typical signals have been chosen to be commensurate with accepted values [30] . The next stage in the work is to miniaturise the telemedicine processor module and use it to carry out clinical trials with the Da Vinci health technology network (http://www.davinci-net.org).

References

1. Woodward, B., Istepanian, R.S.H. and Richards, C.I., Design of a telemedicine system using a mobile telephone, IEEE Transactions on Information Technology in Biomedicine, (2001), 5 (1), 13-15.
2. Woodward, B. and Rasid, M.F.A., Wireless telemedicine: the next step?, Proc. 4th Int. IEEE EMBS Special Topic Conference on Information Technology Applications in Biomedicine, (Birmingham, UK, 2003), 43-46.
3. Gore, L., Biomedical data telemetry from a mobile patient, M.Eng. Project, Loughborough University, UK, 2004.
4. Tompkins, W.J. and Webster, J.G. Design of Microcomputer-Based Medical Instrumentation. Prentice-Hall, 1981.
5. Webster, J.G., Medical Instrumentation: Application and Design, 2nd edn. Houghton Mifflin, 1992.
6. Wireless Futures UK Limited, Wireless Futures BlueWAVE RS232 PCB DCE Terminal Integration Guide, Document Version 2.1, Firmware Version 1.2, Milton Keynes, UK, 2003.
7. M. Pecen and A. Howell, Simultaneous voice and data operation for GPRS/EDGE: class A dual transfer mode, IEEE Personal Communications, (2001), 8 (2), 14-29.
8. 3GPP, Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Numbering, addressing and identification, TS 23.003 version 6.6.0 Release 6, March 2005.
9. 3GPP, General Packet Radio Service (GPRS); Service Description; Stage 2 (Release 6), 3GPP TS 23.060, Dec 2003.
10. Park, J.H., Wireless internet access of the visited mobile ISP subscriber on GPRS/UMTS network, IEEE Transactions on Consumer Electronics, (2003), 49 (1), 100-106.
11. Tsao, S.L., Improving packet delivery in GPRS/UMTS networks, Proceedings of 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, (Lisbon, Portugal, 2002), 2024-2028.
12. Lin, Y.B., Lo, Y.C., and Rao, H.C.H., A push mechanism for GPRS supporting private IP addresses, IEEE Communications Letters, (2003), 7 (1), 24-26.
13. Bettstetter, C., Vogel, H.J. and Eberspacher, J., GSM Phase 2+ General Packet Radio Service GPRS: Architecture, Protocols, and Air Interface, IEEE Communications Surveys and Tutorials, (1999), 2(3), 2-14.

14. Cai, J. and Goodman, D., General Packet Radio Service in GSM, IEEE Communications Magazine, (1997), 35 (10), 122-131.
15. Berg, D.J. and Fritzinger, J.S., Advanced Techniques for Java Developers. New York, USA: John Wiley & Sons, 1999, ISBN: 0471327182.
16. Knudsen, J., Wireless Java: Developing with J2ME, 2nd ed. New York, USA: Apress, 2003, ISBN: 1590590775.
17. Connolly, T.M. and Begg, C.E., Database Systems: A Practical Approach to Design, Implementation and Management, 4th ed. Addison-Wesley, 2005.
18. Rasid, M.F.A., Multi-channel GPRS-based mobile telemedicine system with Bluetooth and J2ME interfaces, Ph.D. Thesis, Loughborough University, 2005.
19. Norman, M., Database Design Manual: Using MySQL for Windows. Springer-Verlag, 2004.
20. Atkins, P.G., Displaying medical signals on a mobile phone, M.Eng. Project, Loughborough University, UK, 2005.
21. Krishnamurthy, P., Kabara, J. and Anusas-Amornkul, T., Security in wireless residential networks, IEEE Transactions on Consumer Electronics, (2002), 48 (1), 157-166.
22. Lin, P. and Lin, Y.B., "Channel allocation for GPRS," IEEE Transactions on Vehicular Technology, vol. 50, no. 2, pp. 375 - 387, March 2001.
23. Graja, H., Perry, P. and Murphy, J., "A statistical analysis of IP packet delay and jitter in cellular networks," Proceedings of 15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*,* Barcelona, Spain, pp. 1881-1885, 5-8 September 2004.
24. Vornefeld, U., "Analytical performance evaluation of mobile internet access via GPRS networks," Proceedings of European Wireless 2002: Next Generation Wireless Networks: Technologies, Protocols, Services and Applications, Florence, Italy, 25-28 February 2002.
25. Premkumar, K. and Chockalingam, A., "Performance analysis of RLC/MAC and LLC layers in a GPRS protocol stack," IEEE Transactions on Vehicular Technology, vol. 53, no. 5, pp. 1531-1546, September 2004.
26. Kilpi, J. and Mannersalo, P., "Performance analysis of GPRS/GSM from the single user point of view," ILIAS-2001 Project, VTT Information Technology, Tampere, Finland, February 2002.
27. MIT-BIH Arrhythmia Database CD-ROM, Harvard-MIT Division of Health Sciences and Technology, Cambridge, Massachusetts, May 1977.
28. Fabri, S.N., Worrall, S., Sadka, A. and Kondoz, A., Real-time video communications over GPRS, Proceedings of First International Conference on 3G Mobile Communication Technologies, (London, UK, 2000), 426 – 430.
29. Istepanian, R.S.H., Jovanov, E. and Zhang, Y.T., Guest editorial introduction to the special section on M-Health: Beyond seamless mobility and global wireless health-care connectivity, IEEE Transactions on Information Technology in Biomedicine, (2004), 8(4), 405-414.
30. Tompkins, W. J. (Ed), Biomedical Digital Signal Processing, London, U.K: Prentice Hall, 1993.