The Scalable Architecture of a Continuous ECG Monitoring System

by

Xiaonan Wu

A thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfillment of
the requirements for the degree of
Master of Computer Science

Ottawa-Carleton Institute for Computer Science
School of Computer Science
Carleton University
Ottawa, Ontario, Canada

January 2004

©Copyright

2004, Xiaonan Wu

National Library
of Canada

Bibliothèque nationale
du Canada

Acquisitions and
Bibliographic Services

Acquisitions et
services bibliographiques

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 0-612-89872-5

Our file Notre référence

ISBN: 0-612-89872-5

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this dissertation.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de ce manuscrit.

While these forms may be included in the document page count, their removal does not represent any loss of content from the dissertation.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

Abstract

Patient monitoring is becoming an important and vital part of any health care system. With the support of sophisticated telecommunications, particularly wireless technology, wireless patient monitoring systems have received increasing attention. The challenge is to create a patient monitoring system that is capable of providing continuous, reliable and real time monitoring and data acquisition services while removing any restrictions on patient location. We present a scalable wireless telemedicine system that is capable of simultaneously monitoring a large number of patients by acquiring continuous patient ECG data. In this thesis, we describe an efficient technique for overcoming temporary “surges” in the environment, where a large number of patients (beyond the normal capacity of the monitoring system) must be monitored. Our solution is based on modifications to the data transmission pattern over the wireless network to temporarily extend the service capacity of the system, and a server-side cluster with load balancing and fault-tolerance mechanisms to provide long term scalable and reliable service.

Acknowledgements

I would like to express my sincere thanks to my supervisor, Professor Doron Nussbaum, for his valuable advice, encouragement and effort on this thesis. I feel very fortunate to have had his supervision and am extremely grateful for the learning experience he has given me.

Special thanks goes to Xuebin Li for his great help during this study. My thanks also goes to Oliver Crosby for his collaboration during system implementation, and David Jackson for his proof reading for my thesis.

I would like to dedicate this thesis to my mother Jinghua Huang, my father Ruobin Wu, and my sister Ying Wu. Without their endless love, I could not have gone through everything I have faced in my life.

Table of Contents

1	Introduction.....	1
1.1	Motivation.....	1
1.2	Thesis Contributions	3
1.3	Thesis Organization	4
2	Telemedicine and ECG Background	5
2.1	Telemedicine.....	5
2.2	ECG Monitoring	8
2.3	ECG Signal	10
2.4	Related Work	18
3	The Scalable System Architecture	23
3.1	Challenges in Wireless Monitoring	23
3.1.1	Limitations of Mobile Computing	23
3.1.2	ECG Data Transmission	24
3.1.3	Scalability	25
3.1.4	Security	26
3.2	The Architecture Overview.....	27
3.2.1	Central Servers.....	27
3.2.2	ECG Mobile Units	28
3.2.3	A Wireless Network.....	29
3.2.4	An Operational Scenario.....	29
3.3	Design of System Components.....	32
3.3.1	Hub Server Components.....	32
3.3.2	Data Server Components	36
3.3.3	Mobile Unit Components.....	39
3.3.4	Communication between Components	43
3.4	Handling Challenges in Wireless Monitoring.....	45
3.4.1	ECG Data Transmission	45
3.4.2	“Surge” Protection	46
3.4.3	Scalability and Load Balancing	47
3.4.4	Power Saving	47
3.4.5	Reliability.....	48
4	Experimental Results and Analyses.....	51
4.1	Motivation for Simulation.....	51
4.2	Approaches to Simulation Implementation	52
4.2.1	Simulation of Wireless Network.....	52
4.2.2	Simulation of Mobile Units	53
4.2.3	Calculations of Concurrent Connections	62
4.3	Simulation Environment Setup	63
4.4	Connection Capacity of One Server	65
4.4.1	Transmission without Intervals.....	66
4.4.2	Transmission with Different Intervals	67

4.5	Real Data Transmission via CDMA1x	72
4.6	Experimental Results Analysis	73
5	Conclusions and Future Work	79
5.1	Conclusions.....	79
5.2	Future Work	80
	References.....	82

List of Figures

Figure 2-1	The Placement of ECG Electrodes	11
Figure 2-2	ECG signal	12
Figure 2-3	ECG Signals of Lead I, II and III.....	13
Figure 2-4	ECG Signals of Lead aVR, aVL and aVF	13
Figure 2-5	ECG Signals of Lead V1, V2, V3, V4, V5 and V6	14
Figure 2-6	Three Orthogonal Directions	15
Figure 2-7	ECG Segments	17
Figure 3-1	Overview of a Wireless Monitoring System.....	28
Figure 3-2	ECG Monitoring Mobile Unit Prototype	29
Figure 3-3	System Component Collaboration Diagram	31
Figure 3-4	System Message Sequence Diagram	32
Figure 3-5	Hub Server Component Diagram.....	33
Figure 3-6	Hub Server Task Dispatcher Component Diagram	34
Figure 3-7	Hub Server Controller Component Diagram	35
Figure 3-8	Hub Server Replicator Component Diagram.....	36
Figure 3-9	Data Server Component Diagram.....	37
Figure 3-10	Service Manager Component Diagram.....	38
Figure 3-11	Data Server Controller Component Diagram	39
Figure 3-12	Mobile Unit Component Diagram	40
Figure 3-13	ECG Signal Display on the Mobile Unit	41
Figure 3-14	Mobile Unit GUI Settings.....	42
Figure 4-1	Simulation of Wireless Network.....	53
Figure 4-2	Maximal Concurrent Connections (Method 1).....	55
Figure 4-3	Average Concurrent Connections (Method 1).....	56
Figure 4-4	Maximal Concurrent Connections (Method 2).....	59
Figure 4-5	Average Concurrent Connections (Method 2).....	60
Figure 4-6	Calculations of Average Concurrent Connections.....	63
Figure 4-7	Effects of Different Simulation Methods on Concurrent Connections.....	64
Figure 4-8	Real-time Maximal Users	67
Figure 4-9	Effects of Different Intervals on Maximal Concurrent Connections of One Data Server.....	70
Figure 4-10	Effects of Different Intervals on Average Concurrent Connections of One Data Server.....	71
Figure 4-11	Effects of Serialization on Maximal Concurrent Connections of One Data Server.....	76
Figure 4-12	Server Capacity Increase.....	78

List of Tables

Table 2-1	ECG Segment.....	16
Table 4-1	Maximal Concurrent Connections (Method 1).....	54
Table 4-2	Average Concurrent Connections (Method 1).....	54
Table 4-3	Maximal Concurrent Connections (Method 2).....	58
Table 4-4	Average Concurrent Connections (Method 2).....	58
Table 4-5	Effects of Different Simulation Methods on Concurrent Connections.....	64
Table 4-6	Effects of Different Intervals on Maximal Concurrent Connections of One Data Server.....	69
Table 4-7	Effects of Different Intervals on Average Concurrent Connections of One Data Server.....	69
Table 4-8	ECG Data Transmission via CDMA.....	73
Table 4-9	Server Capacity Increase.....	78

1 Introduction

1.1 Motivation

The availability of sophisticated telecommunications can lead to a change in the conceptual design of healthcare delivery. Telemedicine has been defined as “the usage of communications and information technology to deliver health care services and health information over large and small distances” [Gov98]. Telemedicine systems, which are based on existing telephone network lines, are already available and being used, for example to enable a doctor to monitor a patient remotely for home care or emergency application [Gott95] [Coy95]. However, when using this technology, a patient (a user in this case) is confined to an environment that consists of a wired network. Wireless communication technology and wireless network have become more common in recent years providing new applications with flexibility. Mobile telemedicine takes telemedicine one step further and relaxes the restriction imposed by a wired environment. Moreover, it may enable for the first time the provision of personalized and customized health services regardless of patient location. Thus, the focus is on continuous patient monitoring in an unknown environment where the patient is free to roam.

Telemedicine has been used in various medical fields to transfer different kinds of medical data, such as tele-ophthalmology [Yog00], tele-ultrasound [Ars00], tele-radiology [Tak01], tele-psychology [Sch00], and tele-cardiology [Ist98]. In particular, the focus here is on cardiac patients who require continuous monitoring of their heart activities, such as ElectroCardioGram (ECG) monitoring.

Remote ECG monitoring can be very useful in the follow-up of out-of-hospital patients receiving drugs in order to assess the efficacy of therapy and the presence of potentially harmful conditions for the heart. In particular, it is a convenient way of providing low-cost ECG monitoring for the elderly. The usage of ECG in emergency cases is also critical at times, as the transmission of ECG signals to the receiving hospital can reduce response time and even enable the resuscitation of sudden cardiac arrest victims. With the help of intelligent agents, remote consultation sessions can be greatly enhanced and costs can be lowered. Data sharing among specialists and database acquisition for research purposes can also be very helpful in improving health services [Hus97].

The objective of this thesis is to investigate and demonstrate the incorporation of wireless and small devices technology into medical monitoring, in particular to address wireless communication issues in order to improve the scalability and reliability of a long-term continuous monitoring system.

1.2 Thesis Contributions

In this research we focused on continuous monitoring of heart patients. We have designed and developed a scalable wireless telemedicine system that is capable of simultaneously monitoring a large number of patients [Nuss03].

In designing the system we focused on the trade off between real time data transmission and scalability issues in a long-term continuous wireless monitoring system. The main objectives are summarized as follows:

- Improve the system scalability by dynamically adjusting the transmission pattern and easily adding/deleting servers.
- Provide continuous and almost real-time monitoring and data transmission over a long period of time, which can be fed into other services, such as real-time analysis and post acquisition analysis, compression, and storage.
- Save mobile unit battery life to relieve most restrictions on patient roaming, while maintaining long-term continuous monitoring.

1.3 Thesis Organization

In Chapter 2, we give an overview of telemedicine and medical monitoring. We briefly introduce the topics of heart monitoring and Electrocardiograph, and discuss other work on telemedicine. In Chapter 3, we present an ECG telemedicine system for heart patients who have been discharged from hospital, including high level overview and detailed design of the system components. We first discuss the challenges of wireless continuous monitoring. Then, an overview of the system architecture is presented, followed by a detailed description of the system infrastructure and components. In Chapter 4, we discuss the results of our experiments on the system, including experiment setup and analysis. We first discuss the motivation for our experiments and approaches. Then, we describe how the experiments were set up. Finally, we present and analyze the experimental results. In Chapter 5, we summarize our work and suggest directions for future research.

2 Telemedicine and ECG Background

In this chapter, we give an overview of telemedicine in relationship to medical monitoring and in particular ECG monitoring. We start with an overview of telemedicine, and continue with an introduction of ECG. We also discuss previous related work in this chapter.

2.1 Telemedicine

Telemedicine is a new but rapidly growing research area, and by incorporating advanced technology, it has had an impact on many aspects of health care. These impacts have ranged from the reduction of care deliver costs, to improvements in health services, to increases in the life quality of patients and others who are in need. Improvements to patient life quality include the following:

- Reduction in the number of days a patient is hospitalised;
- Interaction with a patient in his/her own environment (e.g. at home);
- Prolongation of the stay of acute patients at home, as opposed to hospitalisation, using remote surveillance and monitoring.

As a result of these important effects, telemedicine has received more and more attention in recent years. As a key word, telemedicine was found in 154 references in 1994, and these increased to 400 in 1995 [Joh99]. In 1996, a literature review on telemedicine was found in over 4,400 references [Pic99]. By 2003, the number of references that have telemedicine as a keyword has increased significantly to 517,000 [Google].

One important aspect of telemedicine is the transfer of various medical data, such as photographs, medical images, and sounds, from one site to another for diagnostic or archival purpose. This has led to the development of a large variety of dedicated telemedicine systems.

- Tele-CT

In a computed tomography (CT) telemedicine system [Tak01], the mobile unit consists of a van that houses a spiral computed-tomography machine and various telecommunications equipment. The unit allows medical examination, CT scanning, and the on-line two-way transfer of image data/teleconferencing to a medical center for consultation with various specialists via satellite communication.

- Tele-Ophthalmology

A tele-ophthalmology consultation system was developed linking a hospital and an ophthalmology center [Yog00]. Patients' eyes were screened for glaucoma using a hand-held camera. Optic disc images were compressed and stored together with patient information on a laptop computer. The images were then transmitted to the ophthalmology center using either a mobile phone or a satellite phone.

- Tele-Ultrasound

The purpose of the ultrasound telemedicine application is to create an electronic archival system for ultrasound images, as well as for retrieval of these images through a wireless medium at remote locations [Ars00]. The application is designed to capture an ultrasound image, and save it in a digital format on the hard drive of a laptop. It also includes an option for doctors to annotate voice and text to the image file. Then, all three media (image, voice and text) are transmitted as a single object through a wireless medium, which comprises the unique approach of this particular telemedicine application. Finally, these files are stored on a central server for future analysis or archiving.

- **Tele-Cardiology**

Many telemedicine researchers have focused on tele-cardiology. For instance, in a cardiac disease telemedicine system [Hoo01], telemedicine has been applied to ECG signal acquisition, the management and rehabilitation of recently discharged patients, and peer-to-peer consultation where remote expertise can facilitate diagnosis. A Web-based longitudinal ECG monitoring application is used to collect clinical data from the patient's home [Mag98]. A central database acts as a central repository where this clinical information is stored. A Web browser provides access to the patient's records and clinical data. ECG data are analyzed on the Web server. An optimal patient care strategy with reminders and suggestions for action is then sent to the doctor and patient by email. Note that most tele-cardiology systems have been applied to ECG monitoring.

2.2 ECG Monitoring

ECG is a representation of the heart's electrical activity recorded from electrodes on the body surface [Medlib]. ECG monitoring is used to determine the health level of a patient suffering from heart malfunction or irregular functioning of the heart. One application of this technology is the monitoring of patients with pacemakers.

Generally there are three types of ECG monitoring – clinical monitoring, event monitoring, and long-term monitoring [Heart]:

- Clinical monitoring usually takes place in a hospital or a clinic. Caregivers use an ECG monitor to acquire 1-5 minutes of ECG signal from patients for instant ECG analysis purposes. A major advantage of clinical monitoring is that patients can have instant diagnoses from caregivers and they can be treated immediately if they are in need. However, this type of ECG monitoring will not detect abnormal heart conditions that are not continuously symptomatic, and may not be observed during the 5 minute ECG signal sample collected by the caregivers.
- Event monitoring does not record every heartbeat. Instead, event monitoring only records approximately 120 seconds of a patient's heart rhythm. When the patient experiences the symptom of interest, he/she presses a button that freezes the recording, which is then transmitted by telephone to an interpreting centre. A major advantage of event monitoring is that it can be used to look for the occurrence of transient symptoms. Furthermore, this type of monitor is light and does not consume too much power. However, patients who have cardiac ischemia (a decrease in the blood supply to the heart caused by constriction or obstruction of the blood vessels) may not actually experience symptoms during their ischemia episodes. This condition is called "silent ischemia". Event monitoring will not detect silent

ischemia, since the condition is not symptomatic and patients will not know when to record it.

- In long-term monitoring, a patient is sent home and resumes normal activities while continuous ECG data is acquired for 24 hours or more. The ECG data is recorded or transmitted to hospitals for further analysis or archiving. The greatest advantage of long-term monitoring is that it enables cardiologists to view and analyze the patient's heart activities over a long period of time, day and night. Abnormal heart activities which occur during that time will be detected or identified. Even heart diseases that are not symptomatic, such as silent ischemia, can often be diagnosed using long-term monitoring.

2.3 ECG Signal

Cardiologists can analyze ECG to obtain information on the heart rate, heart rhythm, conduction abnormalities, history (e.g. a prior heart attack), coronary artery disease, the abnormal thickness of the heart muscle, and more, all of which are important in determining the condition of the heart. In this section, we demonstrate how the ECG signal is captured and analyzed.

- **12-lead ECG**

The ECG essentially reads the electrical impulses that stimulate the heart to contract. The heart, which is like a pump, is moved by the heart

muscle. The heart muscle moves when it is negatively or positively charged. Changes in electric potential are detected by electrodes attached to the patient's body, such as on the right arm (RA), the left arm (LA), the left leg (L Leg) and on the chest (V1, V2...V6) (see Figure 2-1).

Figure 2-1 The Placement of ECG Electrodes

The changes, which are detected by electrodes, are drawn by mechanical pencils on a continuously moving paper tap. When the positive wave moves toward a positive electrode there is an upward recording of the ECG and a negative deflection downward (see Figure 2-2).

Figure 2-2 ECG signal

An ECG lead is a continuous ECG signal drawing, which is done by a mechanical pencil. A lead represents changes in electric potential between a pair of electrodes or at one of the electrodes that are placed at a specified location on the body surface (see Figure 2-1). In clinical practice, 12 electrocardiograph leads are usually used in the diagnostic ECG. They are three Standard Limb Leads, three Augmented Limb Leads, and six Precordial Leads.

- **Standard Limb Leads**

The three standard limb leads are usually designated as Leads I, II and III. They are all bipolar (i.e., they detect a change in electric potential between two electrode points) and detect an electrical potential change in the frontal plane (see Figure 2-1). Lead I is between the right arm and left arm electrodes, the left arm being positive. Lead II is between the right arm and left leg electrodes, the left leg being positive. Lead III is between the left arm and left leg electrodes, the left leg again being positive. (see Figure 2-3)

Figure 2-3 ECG Signals of Lead I, II and III

- **Augmented Limb Leads**

Augmented limb leads are three unipolar leads which are formed by the same three electrodes that form the standard limb leads. These three leads are referred to as aVR (right arm), aVL (left arm) and aVF (left leg), and also record a change in electric potential in the frontal plane. Figure 2-4 shows ECG signals responding to Leads aVR, aVL and aVF.

Figure 2-4 ECG Signals of Lead aVR, aVL and aVF

- **Precordial Leads**

These six unipolar leads, each in a different position on the chest, record the electric potential changes in the heart in a horizontal plane (see Figure 2-1). Each lead records the electrical variations that occur directly under the electrode. Figure 2-5 shows ECG signals responding to Leads V1, V2, V3, V4, V5, and V6.

Figure 2-5 ECG Signals of Lead V1, V2, V3, V4, V5 and V6

Leads I, II, III and Leads aVR, aVL, aVF are in the frontal plane, while Leads V1, V2, V3, V4, V5, and V6 are in horizontal plane. Therefore, 12-lead ECG provides spatial information about the heart's electrical activity in three approximately orthogonal directions: right – left, superior – inferior, and anterior – posterior (see Figure 2-6).

Figure 2-6 Three Orthogonal Directions

Among the 12 leads, Leads I, II and III are most important for heart disease diagnoses. Usually, ECG waves and intervals in Lead II are more typical and more easily recognized than those in other Leads. Therefore, if only one lead is displayed, Lead II is usually the one chosen.

- **ECG Segment**

An ECG of a single lead is divided into a number of ECG segments (see Figure 2-7). The purpose of each segment is given in Table 2-1.

Table 2-1 ECG Segment

Name	Description
P wave	the sequential activation (depolarization) of the right and left atria
QRS complex	right and left ventricular depolarization (normally the ventricles are activated simultaneously)
U wave	origin for this wave is not clear - but probably represents "after depolarizations" in the ventricles
PR interval	time interval from onset of atrial depolarization (P wave) to onset of ventricular depolarization (QRS complex)
QRS duration	duration of ventricular muscle depolarization
QT interval	duration of ventricular depolarization and repolarization
RR interval	duration of ventricular cardiac cycle (an indicator of ventricular rate)
PP interval	duration of atrial cycle (an indicator of atrial rate)

Figure 2-7 ECG Segments

Cardiologists diagnose heart disease in a patient by recognizing and analyzing ECG segments, such as the P wave, QRS complex and U wave. The ECG signal is drawn on a paper that moves at a standard speed of 25mm/sec. Thus, the X-axis is used to measure time. As illustrated in Figure 2-7, there are 200 microseconds between two bold vertical lines, and 40 microseconds between two non-bold vertical lines. The Y-axis is used to measure voltage. There is 0.1 mV between two non-bold longitudinal lines. For example, to obtain the heart rate in beats/minute, caregivers count the number of QRS complexes, from the beginning of one QRS complex to the beginning of the last QRS complex, within 30 continuous bold vertical lines (which represent 6 seconds), and multiply by 10.

2.4 Related Work

Telemedicine systems, which are based on wired networks, have enabled doctors to remotely monitor a patient at home or in the emergency room [Gott95] [Coy95]. Advances in mobile computing technology are today providing a flexible and affordable means of communication, which has triggered the use of wireless technology in telemedicine [Wood01] [Ist01] [Her01]. These papers provide different designs of telemedicine systems using wireless communication.

Current designs of ECG monitoring systems must take into account the continuous acquisition and transmission of large amounts of data. In cases where an ECG analysis application does not require high-resolution ECG, the amount of data can be reduced by using lossy data compression methods, thus reducing transmission time and data storage space. However, for high-resolution analysis, lossless ECG compression must be used in order to avoid signal distortion. In this case, the ECG data size is still large. Therefore, one of the most important issues related to ECG monitoring is the data transmission between a client and a central server, especially when real-time and reliable transmission is required. Due to the usually unsteady communication channels used by wireless networks, the transmission issue in the wireless ECG telemedicine system is critical.

Chao et al. propose a framework in which a large volume of ECG data is filtered on site, and only the information required by doctors is transferred over the Internet [Cha01]. The main aim of this framework is to support interoperability and to optimize the monitoring process by reducing unnecessary data transmission. Patients and their monitoring systems, doctors, and database systems are connected via the Internet. Within this framework, ECG customized software is migrated from the Central Monitoring System (CMS) to the remote PC, on demand, in order to collect and filter ECG data on remote sites. When a patient complains of pains in the chest, a remote ECG machine and a PC are used to carry out the acquisition and analysis of the ECG data. The ECG analysis software resident on the remote patient's PC transforms the data into diagnostic information, and identifies abnormal situations. In this framework, because customized software processes data on remote patient sites, significant CPU and memory resources are required on patient sites. This requirement may not be fulfilled by a Pocket PC that has fewer CPU and memory resources than a PC.

Dafonte et al. propose a task manager framework [Daf02]. The system includes a control mechanism that manages the tasks and their distribution between the connected computers. An agent resides in each computer. Every agent must know at each moment all the other agents that are dispersed over the network, and efficiently detect appearing or disappearing agents. The task manager is implemented through a system of distributed agents. The intelligent agent

technology controls and distributes tasks according to their priority and to the availability of resources between the interconnected computers. The task manager knows how to efficiently use the resources that are available in the network and the transmission capacity of the connection lines between the agents. Each agent examines the free resources of the computer by which it is executed locally and sends this information to all the other agents of the network. The information is dynamic because it is periodically examined and distributed to the other agents. This framework is a good solution for monitoring in Intensive Care Units (ICU). Since it is essential to guarantee acceptable response times for the execution of tasks in ICUs, tasks are distributed from one busy site to another available site to ensure they are finished as quickly as possible. However, in order to calculate the time lag of a communication line between two agents, this framework must send a series of packets to verify communication time between two agents, consuming significant network bandwidth for maintenance. For a long-term monitoring system, response times for execution are more flexible than are those in an ICU. It is not necessary to consume so much network bandwidth for maintenance. Therefore, this framework is not suitable for a long-term wireless monitoring system.

Another mobile computing architecture for patient monitoring in a battlefield environment was proposed by Bukhers et al. [Buk96]. This paper introduces a mobile transaction manager (MTM) responsible for transactions created by mobile hosts within its area, and a global transaction manager (GTM)

responsible for re-scheduling global transactions. When the MTM receives a transaction, it searches its mobile host log table to find an appropriate base station to submit the transaction to. If the MTM fails to submit the transaction to a particular base station because the base station has been damaged during an attack, the MTM will instead submit the transaction to GTM. Then, the GTM submits the transaction to another appropriate site for execution. The GTM in this architecture is not necessary in a civilian environment because base stations are not susceptible to damage by an enemy force. Since the assumption that base stations are susceptible to damage by an enemy force is not necessary in a civilian environment, the GTM in this architecture is useless in most cases.

A monitoring system often needs to handle many client requests simultaneously. When client requests are much greater than the service capacity of one server, it is better to design a system whose architecture consists of multiple servers. This enables the system to spread client requests among server nodes. Many previous papers have proposed solutions concerning multiple server architectures. For example, Anderson et al. developed the MagicRouter [And96], which is a packet-filter-based approach to distributing network packets in a cluster [Mog87]. The MagicRouter acts as a switchboard that distributes requests for Web service to the individual nodes in the cluster. To do so, it requires that packets from a client are forwarded by the MagicRouter to the individual server chosen to service the client's TCP connection. It also requires that packets from the server be forwarded, by the MagicRouter, on their way back to the client. An

architecture slightly different from that of the MagicRouter is described by Dias et al. [Dia96]. They present a “TCP Router” which acts as a front-end, forwarding requests for Web service to the individual back-end servers of the cluster. Law et al. present another architecture, which uses a TCP-based switching mechanism to implement a distributed proxy server [Law97]. The motivation for this work was to address the performance limitations of client-side caching proxies by allowing a number of servers to act as a single proxy for the clients of an institutional network. The architecture uses a centralized dispatcher to distribute client requests to one of the servers in the cluster representing the proxy. The function of the centralized dispatcher is similar to that of the MagicRouter. These architectures focus on the multiple server architectures mainly used for Web browsing, whose services in response to client requests last a shorter duration. However, they are not suitable for continuous data transmission in a long-term monitoring system with large data transmission whose services must last much longer.

3 The Scalable System Architecture

In this chapter, we start by discussing the challenges of a wireless long-term monitoring system. Then we give a high level description of our system and its components. Finally, we discuss how the system provides solutions to the challenges.

3.1 Challenges in Wireless Monitoring

The development of wireless networking technology opens the door to a flexible communication environment and continuous access to network services regardless of locations. Therefore, a wireless ECG monitoring system is more promising than a conventional monitoring system based on a wired network. In designing a scalable wireless ECG monitoring system we encounter many challenges. In the following sections, we briefly discuss the major challenges.

3.1.1 Limitations of Mobile Computing

In spite of the rapid development of mobile computing technology, the technical challenges to mobile computing are not insignificant [For93].

- **Connection Continuance**

Wireless communication is more susceptible to interruption than traditional wired network communication. Most noticeable interruption occurs when a mobile unit is disconnected from the server. The result of a session disconnection is a loss of transmitted data. In case of long-term monitoring, it is vital that the system be able to obtain all ECG data of each patient.

- **Power Consumption**

One of the objectives of using mobile wireless units is to enable free roaming over a long period of time. Thus, continuous wireless monitoring requires that the power supply of a mobile unit be sustained. Therefore, the system must address the issue of electrical power consumption by the mobile unit. The battery, which is the power source, is the heaviest component of the mobile unit. There is a trade-off with respect to patient quality of life – between carrying a heavy device and the need to recharge the power supply more frequently.

3.1.2 ECG Data Transmission

A monitoring system must be able, as much as possible, to provide real-time data about the patient. The real-time data can then be used to detect any

problems that the patient may have and provide assistance by remote caregivers, for example, by immediately dispatching an ambulance.

In most cases, a mobile unit acquires a single ECG lead which is captured at a frequency of 100 - 400 Hz, 8-12 bits/sample. To meet real-time transmission rates, the network bandwidth must be at least 4,800bps. In some situations, doctors may want to detect high-resolution ECG signals, requiring that 8-lead ECG signals be captured at a frequency of 1 KHz using 16 bits/sample. This necessitates a real-time transmission rate of 128 Kbps.

Long-term monitoring requires that ECG data continue as long as necessary. In addition, the number of monitored patients may be large. The continuous vast amount of ECG data is a challenge for ECG data transmission. The system must be able to monitor a large number of patients for a long period while maintaining all the information for real-time analysis and post acquisition analysis. Thus, the system must be able to address large volumes of data for a long period of time.

3.1.3 Scalability

A monitoring system must be able to scale to ensure that large numbers of patients can be effectively monitored simultaneously.

- **Load Balancing**

In order to get better transmission performance when monitoring a large number of patients with vast data, load balancing issues should be considered.

- **“Surge” Protection**

In particular, the system must be able to cope with periods of times in which a sudden high load is imposed on the system, typically beyond the normal capabilities of the system; the system must be able to accommodate this temporarily large volume of patients. For example, this problem may arise when a large number of cardiac patients request the monitoring service at the same time.

3.1.4 Security

There are other issues related to the system that we are not addressing in this thesis, particularly security issues such as data encryption, patient authentication, and service authentication. The system uses a public wireless network to transmit patient data. Therefore, security tools, e.g. authentication, public/private key encryption, should be used to protect patients' privacy and ensure data authenticity.

3.2 The Architecture Overview

To overcome the above challenges, in this thesis we present a scalable architecture of a wireless continuous monitoring system. The system consists of central servers, ECG mobile units, and a wireless communication network (see Figure 3-1).

3.2.1 Central Servers

Central servers consist of a cluster of Hub Servers (HS) and Data Servers (DS) which usually reside at the cardiology department of a hospital or a cardiologist's office. The role of the DS is to obtain the transmitted patient data, store it, and simultaneously retrieve the data for analysis and diagnostic purposes, either by an automatic system or by a cardiologist. An HS acts as a mediator between mobile units and DSs.

Figure 3-1 Overview of a Wireless Monitoring System

3.2.2 ECG Mobile Units

The ECG mobile unit (MU) is carried by a patient 24 hours a day. As shown in Figure 3-2, it consists of three off-the-shelf components: an ECG signal capture device, a Pocket PC – iPaq 3970 with 400MHz processor, 64 MB RAM and a wireless communication card – Sierra AirCard 555. The ECG signal capture device has two electrodes that are attached to the patient and provide a single ECG lead. The role of the Pocket PC is to interact with the user, to acquire the data from the ECG device through a serial port and to prepare data for transmission. The Sierra AirCard 555 is used to connect servers wirelessly.

Figure 3-2 ECG Monitoring Mobile Unit Prototype

3.2.3 A Wireless Network

Depending on the wireless network that is used, the ECG data can be transmitted via GSM/GPRS/CDPD/CDMA regardless of the location of the patient within the coverage of the network. In our system, we use CDMA (the bandwidth can reach up to 144Kbps).

3.2.4 An Operational Scenario

In order to elaborate how the system works, we present an operational scenario below. In addition, a collaboration diagram and a message sequence diagram of

data servers, hub servers and mobile units are provided in Figure 3-3 and Figure 3-4.

- ECG signal is acquired by the ECG capture device and is transferred by a serial connection to the Pocket PC.
- After acquiring ECG data, the mobile unit needs to upload ECG data to servers in the hospital. The MU transmission controller sends the hub server a connection request message to request a data server that can receive the ECG data.
- Upon receiving a connection request, the task dispatcher (TD) in the hub server determines a suitable DS according to the latest workload information of DSs. The HS controller gets the workload information from DSs periodically.
- Once a DS is selected, it is assigned to the mobile unit and the HS replies to the MU with a connection permission message containing the IP address and port number of the DS.
- The mobile unit connects to the DS and sends an initialization message. After receiving DS instruction, the MU starts to transmit the ECG data. There are two options for bi-directional interaction between the mobile unit and the DS: (i) the mobile unit maintains a continuous connection with the DS; (ii) the mobile unit reconnects to the DS each time it sends the ECG data.
- The DS receives, processes and stores the ECG data. Each DS has a service manager (SM) and DS controller. The SM is responsible for establishing

connections with MUs for receiving ECG data. The DS controller gets MU information and sends instructions to MUs.

Figure 3-3 System Component Collaboration Diagram

Figure 3-4 System Message Sequence Diagram

3.3 Design of System Components

In this section, detailed descriptions of system components are given.

3.3.1 Hub Server Components

An HS consists of three main components – Task Dispatcher, HS Controller and HS Replicator. In addition, there are two assistant components: HS List and DS Task List (see Figure 3-5).

Figure 3-5 Hub Server Component Diagram

Task Dispatcher (TD) implements a dynamic task dispatch algorithm. TD keeps a list of all latest DS workload information. When receiving an ECG data transmission request message from an MU, it queries the DS task list and selects an DS with the lightest workload according to the dynamic dispatched algorithm. The Socket Connection establishes connection with the MU, receives a connection request and sends back connection permission with the IP address of the selected DS and an updated HS list (see Figure 3-6).

Figure 3-6 Hub Server Task Dispatcher Component Diagram

HS Controller functionality varies between central HS and non-central HS. Central HS Controller periodically polls the workload information from all DSs through Socket Connection (see Figure 3-7). It updates the DS task list. The central HS periodically polls the aliveness of all other HSs and sets the latest timestamp of the aliveness of the HS as the timestamp of HS list.

Non-central HS Controllers report aliveness to the central HS upon polling from the central HS. In case the central HS accidentally goes down, non-central HS Controllers have a central HS election algorithm. When a non-central HS has not received any information from the central HS for a certain period of time, the central HS election algorithm takes effect.

Figure 3-7 Hub Server Controller Component Diagram

The central HS periodically polls the latest DS workload information. The central HS Replicator replicates these server states to other HSs. The roles of the HS Replicator in a central and a non-central HS are different. The central HS Replicator periodically sends out the DS task list and HS list while each non-central HS Replicator receives these data and updates its own DS task lists and HS lists though the Socket Connection (see Figure 3-8).

Figure 3-8 Hub Server Replicator Component Diagram

3.3.2 Data Server Components

The Data Server is responsible for receiving ECG data from remote Mobile Units, storing ECG data, and processing ECG data according to different medical purposes.

A data server consists of four main components: Service Manager, DS Controller, DS Processor, DS Storage and Socket Connection to connect with MUs and HSs (see Figure 3-9).

Figure 3-9 Data Server Component Diagram

The Service Manager (SM) establishes connections with MUs. Each connection serves one MU. When an ECG data message is received, it is sent to the DS Data Storage and DS Data Processor.

The SM maintains its workload information by Connection Service when new connections come in or leave, and saves the information in a Connection Counter. When the SM receives a workload poll message from HS, it replies

with a message to report its workload. The SM also updates the list of connected MUs (see Figure 3-10).

Figure 3-10 Service Manager Component Diagram

The Interval Calculation component adjusts the ECG data transmission interval according to the number of connections from SM. After each adjustment, DS Controller broadcasts the new transmission mode to all connected MUs according to a list of connected MUs which is updated by Service Manager. Remote instructions from DS are sent to MUs by Socket Connection component.

Figure 3-11 Data Server Controller Component Diagram

3.3.3 Mobile Unit Components

An MU consists of these main components: GUI, MU Settings, ECG Data Acquisition, and MU Transmission Controller (see Figure 3-12).

Figure 3-12 Mobile Unit Component Diagram

The GUI provides a user-friendly interface for users to configure MU Settings and displays ECG signal (see Figure 3-13).

Figure 3-13 ECG Signal Display on the Mobile Unit

The MU Settings component maintains MU configuration options, such as the mobile unit ID, server address list and ECG acquisition rate, etc., which are manually set by users or automatically updated by applications (see Figure 3-14).

Figure 3-14 Mobile Unit GUI Settings

ECG Data Acquisition gets ECG signals from an ECG signal capture device through a serial port and saves ECG data into a buffer. The MU Transmission Controller (TC) selects an HS from the HS list by using a static balancing algorithm, so that MUs can be balanced within the HS cluster. The TC sends a connection request to the HS. After obtaining connection permission, it fetches ECG data from the ECG Data Buffer and composes ECG stream messages. The TC also receives ECG data transmission intervals from the DS, so that the MU can adjust the ECG data transmission interval and transmit data using this time interval. MU connections to servers are handled by the MU Socket Connection component.

3.3.4 Communication between Components

Each communication among the HS, DS and MU requires a message that is commonly understood. To support all communications in the entire system, different kinds of messages are defined and implemented for different communication purposes. Since the medical information is critical, a reliable connection is necessary. In our system reliable TCP/IP sockets are used to implement communications rather than UDP sockets [Wal00].

A Hub Server needs to serve many Mobile Units at a given time, as does a Data Server. Generally, a server is able to provide socket services in the following modes.

- **Sequential mode**

The socket server accepts one connection request from a socket client and builds a session with this client. During this session, the socket server serves only this client. After the interaction with the client is completed, the socket server accepts a new client's socket connection request. All clients are served sequentially and there is no concurrency. This mode does not provide concurrent services.

- **Multiple processes mode**

When the socket server accepts a connection request from a socket client, it forks a child process to work for this socket connection. The parent process keeps accepting connection requests and forking child processes while child processes serve client message requests simultaneously. When one message needs a long time to be processed on the server side, it uses CPU time slices of the serving child process only. It does not use CPU slices of other child processes which serve other clients. This mode can fulfill the concurrent requirements of our system, but it is a costly solution because forking means concurrent child processes consume greater resources, such as memory and CPU time.

- **Multiple threads mode**

When a socket server accepts a connection request from a socket client, it creates a thread to serve this socket connection. The main thread continues to accept connection requests and create threads while all other threads serve the client message requests concurrently. This mode fulfills the concurrent requirements of our system and its resource consumption is much lower than the multiple processes mode due to the benefits of multithreading over multiprocessing. Therefore, this mode is used in our system.

In addition to socket communications, there are many other functional components with concurrency requirements. In our system, POSIX Threads

[But96] are used to implement concurrency. Since concurrent components in the same application share critical resources, POSIX Threads Mutex is used to synchronize the access to the shared critical resources.

3.4 Handling Challenges in Wireless Monitoring

In this section we discuss how the system overcomes the challenges noted in Section 3.1. The important issues, which we address for the most part in this long-term continuous wireless monitoring system, are summarised as follows.

3.4.1 ECG Data Transmission

With the development of wireless technology, the memory spaces in Pocket PCs and the wireless bandwidth have increased significantly.

Usually, a portable ECG unit provides 1 record lead ECG data. The ECG frequency ranges from 100 Hz to 400 Hz, and the resolution is 8 - 12 bits/sample. Therefore, the ECG data acquisition rate is 800bps - 4,800bps (bits per second). When the ECG data needs to be transmitted in real-time, the transmission rate is the same as the acquisition rate. Wireless communication bandwidth has a wider range from 9.6 Kbps to 54 Mbps [For93]. Assuming that no error occurs during data transmission, all of them can potentially fulfil the

requirement of real-time ECG data transmission. In reality, this is not the case. Thus, our system uses CDMA (bandwidth can reach up to 144Kbps) in order to compensate for occasional losses of communication. When communication is lost, the data are saved locally by the Pocket PC and are sent at a later time. Once communication is recovered, current data are sent and the saved data are interlaced between the real time transmissions.

3.4.2 “Surge” Protection

When the system becomes overloaded with users or when there is a sudden increase in the number of users, there is a risk that the system will grind to a halt and collapse. Since a patient monitoring system must provide reliable continuous data acquisition services, it must provide a mechanism against failure even when the number of mobile units that attempt to connect to the DS exceeds the capacity of the DS.

The system addresses this issue and overcomes potential shutdown by sacrificing to a certain extent the real time aspect of the data acquisition. When a surge occurs, the DS instructs mobile units to change their transmission pattern. For example, instead of fully maintaining the real time aspect of the system, the DS instructs the mobile units to send the acquired ECG data at disconnection intervals of two seconds or more.

In this way, as shown in the experimental results in Chapter 4, the resource consumption of the server decreases significantly, allowing the server to overcome its limits on simultaneous connections.

3.4.3 Scalability and Load Balancing

Assuming that no "surge" occurs, scalability issues are addressed by a variety of load distribution methods. Thus, when there is more than one DS available, a dynamic load-distributing algorithm can be used to achieve superior performance. In order to achieve load balancing, the DSs in the system communicate with the HS and report their latest workloads. When accepting a connection request from the MU, HS selects a suitable DS by using a load-distributing algorithm, and assigns the data transmission task of the ECG mobile unit to the designated DS.

3.4.4 Power Saving

Although power management in most Pocket PCs can save power by turning the LCD off after a certain period of idle time, power is generally consumed when data is being transmitted to the DS. If the mobile unit maintains the connection with the data server and sends ECG data as soon as they are acquired, the power is consumed continuously at a high rate.

More power is consumed when a portable PC sends data than when it does not [For93]. Thus, a mobile unit that is continuously connected to the DS consumes more power than a mobile unit that connects to the DS at a predefined time interval (e.g. 4 seconds). Moreover, if the disconnection period is greater than the transmission time, more power savings can be obtained.

3.4.5 Reliability

The network communication of the monitoring system is based on TCP. TCP is a connection-oriented and reliable service [Wal00]. Therefore, we only address high-level communication reliable issues.

In our ECG telemedicine system, central servers are responsible for services for all mobile units, such as communication connection, data transmission, data storage and analysis. The impact of central server failure is much greater than that of the failure of a mobile unit, therefore, we mainly focus on reliability of server side. Our system consists of two clusters, a data server cluster and a hub server cluster, to ensure reliability on server side. Each hub server keeps a list of all hub servers and a list of all data servers. Replication among the hub server cluster keeps all of them updated.

When a mobile unit needs to connect with a hub server, DNS is usually one of the most efficient methods of distributing incoming connections across a cluster of servers. By using DNS, mobile units only need to know one host name, enabling it to connect to one of all the individual IP addresses mapping to this host name. However, Embedded C++, which is used to implement applications in mobile units, can not resolve the host name to an IP address. Therefore, mobile units must keep a hub server IP list locally. When a hub server is down or a new hub server is up, the mobile units need to obtain the updated local hub server list.

There is a central hub server within the hub server cluster. The central hub server periodically polls the aliveness of all other hub servers. A new hub server should inform the central hub server that it has joined. Whenever any server joins or leaves, the list of hub servers is updated and replicated among the cluster of the hub server with a time stamp. The central server periodically broadcasts its aliveness. If the central server leaves, the first one of the rest of the hub servers is set as the new central hub server, replacing the old one. The hub server list is sent to mobile units along with the time stamp.

When a patient mobile unit sends a request message to a hub server, it adds its time stamp of the hub server list to the command message. The hub server checks if the time stamp is up to date. If it is out of date, the hub server updates the hub server list on the mobile unit and then sends the response message to the

request. Using this mechanism, all live mobile units obtain the updated hub server list.

The central hub server also periodically polls the workload information of all data servers and replicates the data server list within the hub server cluster.

When a patient mobile unit is registered into the ECG telemedicine system, it initially receives one hub server address. It then builds connection to the hub server and requests the latest hub server list.

To upload ECG data to central servers, a mobile unit attempts to connect to the data server which is assigned to the mobile unit by a hub server. If the data server goes down, the mobile unit will fail to build a connection to the data server. Then, the mobile unit will request HS for another data server address. Upon successful connection to the data server, the mobile unit begins transmitting the ECG data.

4 Experimental Results and Analyses

In this chapter, we first discuss the motivation for simulation and approaches of simulation experiments. Then, we describe how we set up the simulation experiments. Finally, we present and analyze the results of the experiments.

4.1 Motivation for Simulation

In order to get test results of connection capacity for one server, we need to simultaneously connect many mobile units to a single server. However, the cost of each mobile unit and the wireless connection service fee is too high. For example, the price of one iPAQ Pocket PC 3975 is around \$500, a Sierra Wireless AirCard is \$500 and the wireless connection service fee without limitations on data quantities is \$100/month. If we need 1000 mobile units for the test, the cost becomes enormous. Due to budget limitations, we did not have enough mobile units for this test. Therefore, we ran simulated-client applications in one computer to simulate mobile units and simulate the wireless data transmission in an LAN environment.

4.2 Approaches to Simulation Implementation

In this section, we discuss how to simulate wireless network communication in an LAN, how to simulate many mobile units on a few computers, and how to examine the experimental data.

4.2.1 Simulation of Wireless Network

Let T_{sleep} be the “sleeping time”, D be the transmission data, $S_{wireless}$ be the wireless network speed, and S_{LAN} be the LAN speed. In the simulated experiment, we simulate wireless network communication in a local network by sleeping for T_{sleep} when sending ECG data according to the data amount D , the local network transmission speed S_{LAN} and the wireless network transmission speed $S_{wireless}$. In our experiments, we assume the wireless network speed is CDMA (100K bps) and the LAN speed is 100M bps. When a certain amount of data D is transmitted via the wireless network, it takes time $T_{wireless} = D / S_{wireless}$. We simulate the wireless transmission time by transmitting data D via LAN, by which it takes time $T_{send} = D / S_{LAN}$, and sleeping for T_{sleep} during data transmission. T_{sleep} is calculated as in Formula 4-1, illustrated in Figure 4-1.

Formula 4-1:
$$T_{sleep} = T_{wireless} - T_{send}$$

Figure 4-1 Simulation of Wireless Network

Therefore, the simulated transmission time is $T_{simulation} = T_{send} + T_{sleep}$, which is the same as $T_{wireless}$.

4.2.2 Simulation of Mobile Units

Method one: To calculate the number of connections that a single server can simultaneously serve, 100 simulated-client applications are run in a computer to simulate 100 mobile clients communicating with the same sever. We use 1, 2,...,10 computers to simulate 100, 200,...,1000 mobile units respectively. When we simulate 100 clients, we run 100 simulated-client applications in one computer. When the number of clients is gradually increased from 100 to 200, 300, 400, 500, 600, 700, 800, 900 and 1000, the number of computers also gradually increases from 1, to 2, 3, 4...9, and 10. We test 10 cases in which clients maintain connections with the server (real-time) or connect with the server once at a time interval of 1 sec., 2 sec., 4 sec., 8 sec., 16 sec., 32 sec., 200 sec., 400 sec., or 1000 sec. The maximal simultaneous connections are shown in

Table 4-1 and Figure 4-2, and the average simultaneous connections as in Table 4-2 and Figure 4-3.

Table 4-1 Maximal Concurrent Connections (Method 1)

Clients	Maximal Number of Concurrent Connections									
	real time	1S.	2S.	4S.	8S.	16S.	32S.	200S.	400S.	1000S.
100	100	43	38	27	23	32	30	93	86	100
200	200	73	48	35	33	51	58	129	172	198
300	300	91	64	48	44	55	87	156	259	291
400	400	92	89	57	63	69	97	251	321	379
500	500	118	102	81	64	70	109	293	352	480
600	600	144	122	90	71	86	116	340	443	584
700	700	150	123	95	63	79	118	396	560	667
800	800	163	143	101	88	96	123	378	584	756
900	900	168	149	112	90	90	127	388	701	871
1000	1000	183	155	123	107	109	135	491	769	954

Table 4-2 Average Concurrent Connections (Method 1)

Clients	Average Number of Concurrent Connections									
	real time	1S.	2S.	4S.	8S.	16S.	32S.	200S.	400S.	1000S.
100	100	13	12	9	7	10	15	47	48	50
200	200	21	16	13	12	17	27	82	95	100
300	300	30	25	20	19	22	39	102	144	150
400	400	37	41	24	22	22	52	163	185	194
500	500	45	45	35	25	25	60	171	218	238
600	600	54	48	40	30	30	62	195	246	281
700	700	62	56	43	32	32	67	205	271	314
800	800	69	61	46	33	42	69	189	294	348
900	900	73	71	57	39	39	71	196	322	384
1000	1000	77	75	57	47	47	74	199	327	405

Figure 4-2 Maximal Concurrent Connections (Method 1)

Figure 4-3 Average Concurrent Connections (Method 1)

Method Two: In this method, we still use 10 computers to simulate 100, 200...1000 clients respectively. All 10 computers are involved in all of the test cases, but each computer runs 1/10 of the total number of clients on it. When the number of clients gradually increases from 100 to 200, 300, 400, 500, 600, 700, 800, 900 and 1000, the number of simulated-client applications in each computer also gradually increases from 10 to 20, ... 90, and 100. In this way, we get the maximal simultaneous connections as shown in Table 4-3 and Figure 4-4, and the average simultaneous connections as in Table 4-4 and Figure 4-5.

Table 4-3 Maximal Concurrent Connections (Method 2)

Clients	Maximal Number of Concurrent Connections									
	real time	1S.	2S.	4S.	8S.	16S.	32S.	200S.	400S.	1000S.
100	100	45	35	62	86	96	99	100	100	100
200	200	138	64	105	126	172	188	200	200	200
300	300	227	106	151	174	234	257	300	300	300
400	400	291	158	187	228	301	283	396	400	400
500	500	445	237	223	265	327	330	485	500	500
600	600	529	304	264	269	393	363	574	595	540
700	700	535	442	335	302	396	449	661	688	700
800	800	547	500	417	354	457	513	743	740	800
900	900	555	531	450	368	424	506	836	831	900
1000	1000	568	538	532	369	504	551	921	968	1000

Table 4-4 Average Concurrent Connections (Method 2)

Clients	Average Number of Concurrent Connections									
	real time	1S.	2S.	4S.	8S.	16S.	32S.	200S.	400S.	1000S.
100	100	12	10	12	33	45	43	41	49	50
200	200	42	22	22	34	65	83	86	93	93
300	300	128	34	32	38	67	105	126	130	124
400	400	184	61	44	42	75	117	161	162	154
500	500	290	111	66	51	78	141	198	195	165
600	600	332	185	107	53	85	158	226	225	195
700	700	362	289	158	66	88	190	256	250	255
800	800	366	335	225	75	104	209	284	263	289
900	900	377	340	293	97	116	213	310	292	301
1000	1000	378	356	352	135	121	235	349	306	324

Figure 4-4 Maximal Concurrent Connections (Method 2)

Figure 4-5 Average Concurrent Connections (Method 2)

In a multi-tasking operating system, the operation system (OS) divides processing time among multiple applications (tasks) by giving each task a “slice” of the CPU’s available work time. In this way, there is still only one task executing at a time. However, the tasks share the processor because the OS grants a particular percentage of the CPU’s time to each task. As a result, multiple applications appear to be running simultaneously on a single CPU. Since we run many simulated-client applications on each computer, the fewer applications run on one computer, the greater the number of time slices an application receives. If a simulated-client application has more time slices to communicate with the server, the simulated-client applications in one computer are more similar to real concurrent clients.

We performed the two simulation methods 10 times respectively. In the first method, each computer runs 100 simulated-client applications each time, and for 10 test cases the number of computers increases from 1 to 2... up to 10, while in the second method 10 computers are involved each time and for 10 test cases, simulated-client applications running on each computer increase from 10 to 20... up to 100. We compared the experimental results. The first method simulates less concurrent clients than does the second method. Obviously, the second simulation method is more similar to the real concurrent situation than is the first one.

4.2.3 Calculations of Concurrent Connections

There are two methods of calculating the average number of concurrent connections during a test. The first is to get the arithmetic average of all the numbers of connections. Let S_i be the i -th number of the connections. The average number of connections is calculated as in Formula 4-2. However, the average number is not related to time when using this calculation method.

Formula 4-2:
$$\text{Average} = (S_1 + S_2 + \dots + S_{n-1} + S_n) / n$$

The second method is to obtain each duration time D_i of each connection number N_i and to sum the product of each connection number and its duration time. Then, let the sum of all products be divided by the total duration time, according to Formula 4-3, also illustrated in Figure 4-6, and we derive the average number.

Formula 4-3:

$$\text{Average} = (D_1 * N_1 + D_2 * N_2 + \dots + D_{n-1} * N_{n-1} + D_n * N_n) / (D_1 + \dots + D_n)$$

Figure 4-6 Calculations of Average Concurrent Connections

In this way, we calculate the average number relative to time. The latter method is more valuable in emulating the impact of the time interval in our experiment. Therefore, we use the second method to calculate the average number in our experiments.

4.3 Simulation Environment Setup

Within a single server (the same IP address), we have an experiment to test the differences in communication capability when using multiple listening ports and a single listening port.

The experimental results are shown in Table 4-5 and Figure 4-7. Note that using multiple ports to communicate does not improve connection capability over using single port. Therefore, in our experiments, a server listens to only one port.

Table 4-5 Effects of Different Simulation Methods on Concurrent Connections

Maximal Number of Concurrent Connections				
Clients	4Kbits/second		32Kbits/8seconds	
	1 port(4Kb/s)	10 port(4Kb/s)	1 port(32Kb/8s)	10 port(32Kb/8s)
100	42	57	16	16
200	64	66	27	34
300	71	91	39	32
400	87	99	34	40
500	122	142	44	51
600	139	162	47	57
700	158	164	58	50
800	162	169	56	52
900	181	185	61	73
1000	186	199	73	75

Figure 4-7 Effects of Different Simulation Methods on Concurrent Connections

In a multi-tasking operating system, the tasks share the processor because the OS grants a particular percentage of the CPU's time to each task. As a result, multiple applications appear to be running simultaneously on a single CPU. However, there is still only one task executing at a time. If the total number of simulated clients is fixed, the greater the number of computers used to run the simulated-client applications, the fewer the simulated-client applications that will run on each computer. Furthermore, the fewer the applications running on one computer, the more frequently an application can get time slices. If a simulated-client application can obtain more time slices, the simulation will be more similar to the real situation. Therefore, in order to improve our simulation we use more computers (25) and use the second simulation method to simulate mobile units.

4.4 Connection Capacity of One Server

To examine the simultaneous connection capacity of one server, we allow clients to keep connecting with the server or transmit data with disconnection for different intervals.

4.4.1 Transmission without Intervals

In this experiment, a client acquires ECG data in 400 Hz and tries to transmit the ECG data in as close to real-time as possible.

We examine six modes. In the first mode, once the client acquires one ECG sample, it sends out a message with this ECG sample. From the second mode to the sixth mode, after the client has 50, 100, 200, 300, 400 samples respectively in the buffer, it sends out a message with these ECG sample data together.

Figure 4-8 show the experimental results. Note that when clients send out a message as soon as they obtain one ECG sample, the server can maximally service 17 simultaneous connections. With clients increasing the number of samples per message, i.e. prolonging the time between 2 messages, the connection capacity of the server also increases. When clients are sending out a message once with 400 samples together, the capacity increases to accommodate up to 665 simultaneous connections.

Figure 4-8 Real-time Maximal Users

4.4.2 Transmission with Different Intervals

When a surge occurs, there is a risk that the data server will not be able to handle all the data, and may even shut down. In order to overcome potential shutdown, the DS instructs the mobile units to change their transmission patterns by transmitting data with different disconnection intervals, sacrificing somewhat the real time aspect of the data acquisition. To compare the different numbers of simultaneous connections, we have 10 cases in which clients maintain connection with the server or connect with the server once at a time interval of 1 sec., 2 sec., 4 sec., 8 sec., 16 sec., 32 sec., 200 sec., 400 sec., or 1000 sec. The

mobile ECG unit acquires 1-lead ECG data at a frequency of 400Hz, with the sample resolution of 10 bits/sample. The data acquisition rate is the same as 4K bps. 1-second ECG data is 4K bits; 2-second ECG data is 8K bits; 4-second ECG data is 16K bits; 8-second ECG data is 32K bits; 16-second ECG data is 64K bits; 32-second ECG data is 128K bits; 200-second ECG data is 800K bits; 400-second ECG data is 1.6M bits; and 1000-second ECG data is 4M bits. Each connection lasts long enough to send the whole message with the acquired ECG data.

To simulate the randomly distributed arrival of connections, we let simulated-client applications start randomly during the time interval. We get the maximal concurrent connections as shown in Table 4-6 and Figure 4-9, and the average concurrent connections as shown in Table 4-7 and Figure 4-10.

Table 4-6 Effects of Different Intervals on Maximal Concurrent Connections of One Data Server

Clients	Maximal Number of Concurrent Connections									
	real time	1S.	2S.	4S.	8S.	16S.	32S.	200S.	400S.	1000S.
100	100	9	8	8	8	6	8	6	5	12
200	200	20	13	11	11	10	12	9	8	20
300	300	88	23	15	15	13	14	12	11	26
400	400	125	54	20	18	17	17	13	16	33
500	500	145	78	28	20	21	18	19	20	40
600	600	172	121	55	23	26	20	23	22	47
700	700	274	223	49	28	33	26	23	26	54
800	800	341	240	78	29	39	26	25	28	66
900	900	352	276	94	31	45	28	30	32	67
1000	1000	368	308	129	40	48	34	31	36	66

Table 4-7 Effects of Different Intervals on Average Concurrent Connections of One Data Server

Clients	Average Number of Concurrent Connections									
	real time	1S.	2S.	4S.	8S.	16S.	32S.	200S.	400S.	1000S.
100	100	2	2	2	1	1	2	1	1	2
200	200	6	4	4	3	3	4	3	3	4
300	300	53	7	6	5	5	5	5	5	6
400	400	97	16	8	7	7	7	7	7	8
500	500	113	35	10	9	9	9	9	9	10
600	600	124	73	14	11	14	11	11	11	12
700	700	136	105	18	13	16	13	13	12	14
800	800	157	115	29	15	21	15	15	14	16
900	900	166	131	45	17	29	18	16	16	16
1000	1000	194	178	67	20	28	19	18	18	18

Figure 4-9 Effects of Different Intervals on Maximal Concurrent Connections of One Data Server

Figure 4-10 Effects of Different Intervals on Average Concurrent Connections of One Data Server

4.5 Real Data Transmission via CDMA1x

In addition to the above simulation experiments, we also tested the effects of different intervals on Pocket PC with real data transmission via CDMA1x, whose wireless service provider is Telus®.

In this experiment, an ECG mobile unit acquires 1-lead ECG data with 400Hz and 2 Bytes/sample, and transmits ECG data to the data server using different time intervals of 1 second, 2 seconds, 4 seconds, 8 seconds, 16 seconds, 32 seconds, 64 seconds, 128 seconds, 256 seconds and 512 seconds. The total acquired ECG data are the same (2560-second data) for each time interval, i.e. the total ECG data to be sent is $2 \text{ Bytes} * 400 * 2560 = 2048 \text{ k Bytes}$. The experimental results are shown in Table 4-8. The total actual data transmitted with shorter intervals is greater than when using longer intervals.

Table 4-8 ECG Data Transmission via CDMA

Interval (s)	ECG acquired duration (s)	ECG Data to be Sent (kB)	Data Sent (kB)	Data Received (kB)	Total Actual Data Transmitted (kB)
1	2560	2048.00	3194.88	122.16	3317.04
2	2560	2048.00	3328.00	110.09	3438.09
4	2560	2048.00	3051.52	95.56	3147.08
8	2560	2048.00	2795.52	81.04	2876.56
16	2560	2048.00	2928.64	83.46	3012.10
32	2560	2048.00	2949.12	84.63	3033.75
64	2560	2048.00	3020.80	84.46	3105.26
128	2560	2048.00	2652.16	80.41	2732.57
256	2560	2048.00	2949.12	88.77	3037.89
512	2560	2048.00	2949.12	85.45	3034.57

4.6 Experimental Results Analysis

We observe in the results of the transmission with different intervals experiment (see Table 4-6 and Figure 4-9) that the number of concurrent connections in real-time transmission is the largest. When the time interval is 1 second or 2 seconds, the number of concurrent connections decreases a lot. As the time interval increases up to 8 seconds, the number of concurrent connections decreases more. This is because the number of concurrent connections is related to the data amount and transmission time, which are affected by the following factors:

- The message header: In our system, the message header is 60 bytes (480 bits). The transmission pattern with a 1 second interval has to send more data than the pattern with a 10 second interval.
- TCP/IP packet header: To transmit a fixed amount of data, the shorter the packet size, the greater the TCP/IP packet header overhead.

When the time interval is 400 seconds or more, the number of concurrent connections slightly increases. This result shows that the number of concurrent connections is also affected by other factors:

- TCP/IP transmission error rate: The transmission speed is influenced by the error rate. The greater the error rate, the lower the transmission speed. [Wal00]
- The serialization of the simulation experiment: Although multiple simulation applications appear to be running simultaneously on a single CPU, there is still only one task executing at a time. The greater the number of applications running on one computer, the longer it takes to finish one task.

The communication overhead for each message includes the message header and TCP/IP packet header. For a certain amount of data, the shorter the message size, the more number of messages. Thus, the greater the communication overhead will be. Therefore, when the time interval increases, the number of

concurrent connections decreases, as shown in Figure 4-9 and Figure 4-10. When the time interval is 1s or 2s, the number of concurrent connections is much greater than when the time interval is more than 8s; adjusting time interval from 1s to 8s can significantly improve capacity. When the time interval is more than 8 seconds, the improvement is slight.

The wireless communication is more susceptible to disconnection when the message size increases. The greater the error rate, the lower the transmission speed. The transmission speed will therefore decrease. As a result, when the time interval is 400 seconds or more, the concurrent connection number increases again. In addition, the experiment is done by simulating many clients in one computer. When the message size increases, it takes longer to complete the transmission of the message. The longer the transmission time, the larger the number of concurrent connections will be. Comparing Figure 4-9 and Figure 4-11, obviously using 10 computers for simulation, the number of concurrent connections is greater than when using 25 computers for simulation. The reason is that the greater the number of computers used to run the simulated-client applications, the more frequently an application can get CPU time slices. If a simulated-client application can obtain more time slices, it can finish the transmission faster, thus leading to fewer concurrent connections.

Figure 4-11 Effects of Serialization on Maximal Concurrent Connections of One Data Server

In the real data transmission experiment (see Table 4-9), the total data actually transmitted is greater than the acquired ECG data, because the transmission overhead (including the message header and TCP/IP header) is added when ECG data are transmitted. As the time interval increases from 1 second to 512 seconds, the message size increases as well. An analysis of this reveals that the smaller the message size, the greater the transmission overhead will be. Therefore, the total amount of data actually transmitted with shorter intervals is greater than when using longer intervals.

From the experimental results of Section 4.4, we see that the server connection capacity has been significantly increased by transmitting data using intervals. The capacity of servers is affected by different server resources, such as CPU speed, memory, network bandwidth, and system configuration. Therefore, the exact quantity of one server's capacity is not a useful measure that can be applied as a general rule. Instead, we examine the number of times the server capacity can be increased. Assuming the capacity of a single server is predetermined, by changing the real-time transmission mode to transmission with a 1 second interval, the server capacity is increased at least 2 times, while the real-time aspect of data transmission is sacrificed somewhat (see Table 4-9 and Figure 4-12). By increasing the time interval from 1 second to 200 seconds, the server capacity can be increased much more; the maximum is 32 times. Therefore, we can overcome the limitation of server connection capacity by adjusting the time interval.

Table 4-9 Server Capacity Increase

Maximal Concurrent Connections	Server Capacity Increase								
	Increased Times								
	1S	2S	4S	8S	16S	32S	200S	400S	1000S
100	11.11	12.50	12.50	12.50	16.67	12.50	16.67	20.00	8.33
200	10.00	15.38	18.18	18.18	20.00	16.67	22.22	25.00	10.00
300	3.41	13.04	20.00	20.00	23.08	21.43	25.00	27.27	11.54
400	3.20	7.41	20.00	22.22	23.53	23.53	30.77	25.00	12.12
500	3.45	6.41	17.86	25.00	23.81	27.78	26.32	25.00	12.50
600	3.49	4.96	10.91	26.09	23.08	30.00	26.09	27.27	12.77
700	2.55	3.14	14.29	25.00	21.21	26.92	30.43	26.92	12.96
800	2.35	3.33	10.26	27.59	20.51	30.77	32.00	28.57	12.12
900	2.56	3.26	9.57	29.03	20.00	32.14	30.00	28.13	13.43
1000	2.72	3.25	7.75	25.00	20.83	29.41	32.26	27.78	15.15

Figure 4-12 Server Capacity Increase

5 Conclusions and Future Work

In this chapter, we outline the conclusions of the thesis and suggest future research directions.

5.1 Conclusions

In this thesis, we designed and developed a scalable wireless telemedicine system. The system is capable of addressing scalability issues by dynamically adjusting transmission patterns and by adding/deleting servers.

Our solution to scalability and surge protection issues is based on changing the data transmission intervals, which reduces the amount of resources that must be consumed by the system. We showed that it is an efficient mechanism for handling “surges” in the potentially vast environment of concurrent patient monitoring. Although this solution does somewhat affect the real time performance of the system, it does provide a reliable solution. We believe that, in most cases, this delay would be acceptable. Also, by using the load distribution method, which takes into account the latest work load information of each data server, the system achieves superior communication performance.

Furthermore, the wireless monitoring system can increase the quality of life of patients by freeing the patient to roam within the range of the wireless network environment, instead of being confined to the monitoring station. By modifying the transmission time intervals, the mobile unit battery power is saved. This increases the duration of continuous monitoring possible between battery charges, allowing the patient to be away from home for a longer period of time.

5.2 Future Work

This research could be further extended in the following respects:

- Real-time ECG data analysis could provide better health services for patients. With real-time analysis, abnormal heart activities can be detected as soon as the ECG data are acquired. The patient is able to get immediate treatment once heart disease begins to occur (or a heart attack strikes).
- In critical cases, a priority could be set in order to determine which mobile monitors need to alter their transmission patterns. Criteria for setting priorities could be based on the ECG real-time analysis results or other real-time information.
- The system uses a public wireless network to transmit patient data. Therefore, security methods such as encryption and authentication should be provided to ensure data safety and patient privacy.

- Wireless service fees are more expensive than wired services, and some wireless service providers charge their customers according to the amount of data transmitted. In addition, the bandwidth of a wireless network is lower than that of the wired network. In order to reduce transmission time and service costs, ECG data could be compressed in a mobile unit before it is sent to a server. Different compression algorithms could be chosen according to the different requirements of ECG resolution.

References

- [And96] E. Anderson, D. Patterson, and E. Brewer, “The MagicRouter, An Application of Fast Packet Interposing”, 1996,
<http://www.cs.berkeley.edu/~eanders/projects/magicrouter> .
- [Ars00] B. R. Arslan, A. Kazmi, P. C. Pedersen, K. Pahlavan, and M. Tulgar, “A User-Friendly Approach to Telemedicine: Wireless Ultrasound”, Proceedings of IEEE Engineering in Medicine and Biology Society Workshop on Biomedical Information Engineering, pages 151-154, Chicago, U.S.A., 2000.
- [Buk96] O. Bukhers, S. Mortonand, M. Mossman, “Mobile Computing Architecture for a Battlefield Environment”, International Symposium on Cooperative Database Systems for Advanced Applications, pages 130-139, Kyoto, Japan, 1996.
- [But96] D. R. Butenhof, “Programming with POSIX Threads”, Addison-Wesley, 1996.
- [Cha01] K. M. Chao, R. Anane, J. Plumley, N. Godwin, R.N.G. Naguib, “A mobile agent framework for telecardiology”, Proceedings of 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pages 3484-3487, Istanbul, Turkey, 2001.
- [Coy95] G. Coyle, L. Boydell, and L. Brown, “Home Telecare for the Elderly”, Journal of Telemedicine Telecare, pages 183–185, Vol. 1, 1995.

- [Daf02] C. Dafonte, A. Gomez, A. Castro, B. Arcay, "Intelligent agents technology applied to tasks control in ICU telesupervision", Proceedings of the Second Joint 24th Annual International Conference of the IEEE Engineering in Medicine and Biology Society and the Annual Fall Meeting of the Biomedical Engineering Society, pages 1861- 1862, Houston, TX, U.S.A., 2002.
- [Dia96] Daniel M. Dias, Willian Kish, Rajat Mukherjee, and Renu Tewari, "A Scalable and Highly Available Server". In Digest of Papers. COMPCON 1996. Technologies for the Information Superhighway, pages 68-74, February 1996. IEEE Computer Society Press.
- [ECG] <http://endeavor.med.nyu.edu/courses/physiology/courseware/ekg>
- [For93] G.H. Forman, J. Zahorjan, "The Challenges of Mobile Computing", UW CSE Tech Report #93-11-03 from <ftp.cs.washington.edu> .
- [Gar99] Felix C. Gartner, "Fundamentals of Fault-Tolerant Distributed Computing in Asynchronous Environments", ACM Computing Surveys, pages 1-26, Vol. 31, No. 1, March 1999.
- [Google] <http://www.google.com> .
- [Gott95] M. Gott, Telematics for Health: The Role of Telemedicine in Homes and Communities. Oxford, U.K. Radcliffe Med. Press, 1995.

- [Gov98] Government of Canada, Industry Canada, Telehealth Industry: Part 1: Overview and Prospects, Second Competitive Frameworks, March 1998.
- [Ham03] K. Hameed, "The application of mobile computing and technology to health care services", Telematics and Informatics, pages 99-106, January 2003.
- [Heart] <http://heartdisease.about.com/cs/ekgecg/a/ECG.htm#b> .
- [Her01] A. I. Hernández, F. Mora, G. Villegas, G. Passariello, and G. Carrault, "Real-Time ECG Transmission Via Internet for Nonclinical Applications". IEEE Transactions on Information Technology in Biomedicine, pages 253-257, Vol. 5, No. 3, September 2001.
- [Hoo01] G.S. Hooper, P. Yellowlees, T.H. Marwick, P. J. Currie, and B.P. Bidstrup, "Telehealth and the Diagnosis and Management of Cardiac Disease", Journal of Telemedicine and Telecare, pages 249-256, Vol. 7, 2001.
- [Hus97] J. L. Huston, "A Telemedical Record Model", Journal of Telemedicine and Telecare, pages 41-44, Vol. 3, No.1, 1997.
- [Ist01] R.S.H. Istepanian, B. Woodward, C. I. Richards, "Advances in telemedicine using mobile communications", Proceedings of 23rd Annual International Conference of the IEEE Engineering in

Medicine and Biology Society, pages 3556-3558, Istanbul, Turkey, 2001.

- [Ist98] R.S.H. Istepanian, "Modeling of GSM-Based Mobile Telemedical System", Proceedings of 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pages 1166-1169, Hongkong, China, 1998.
- [Ist99] R.S.H. Istepanian, "Telemedicine in the United Kingdom: Current Status and Future Prospects", IEEE Transactions on Information Technology in Biomedicine, pages 158-159, Vol. 3, No. 2, June 1999.
- [Jam01] A. James, Y. Wilcox, and R.N.G. Naguib, "A Telematic System for Oncology Based on Electronic Health and Patient Records", IEEE Transactions on Information Technology in Biomedicine, pages 16-17, Vol. 5, No. 1, March 2001.
- [Jin99] J. Jing, A. Helal, and A. Elmagarmid, "Client-Server Computing in Mobile Environments". ACM Computing Surveys, pages 117-157, Vol. 31, No. 2, June 1999.
- [Joh99] E. Johnson, "Telemedicine: An annotated bibliography, Part I", http://tie.telemed.org/old_Bibliographic.asp?ID=2203&FIELDN_AME=AuthorID
- [Law97] K.L.E. Law, B. Nandy, and A. Chapman, "A Scalable and Distributed WWW Proxy System", Nortel Limited Research Report, 1997, <http://citeseer.nj.nec.com/context/8463/0>.

- [Leo00] A. Leon-Garcia, and I. Widjaja, Communication Networks, pages 588-590, McGraw-Hill 2000.
- [Mag98] F. Magrabi, N. H. Lovell, and B.G. Celler, “Web Based Longitudinal ECG Monitoring”, Proceedings of 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pages 1155-1158, Hongkong, China, 1998.
- [Medlib] <http://medlib.med.utah.edu/kw/ecg/intro.html> .
- [Mog87] J. Mogul, R. Rashid, and M. Accetta, “The Packet Filter: An Efficient Mechanism for User-level Network Code”, Proceedings of SOSP’87: The 11th ACM Symposium on Operating System Principles, 1987. <http://citeseer.nj.nec.com/mogul87packet.html>
- [Nuss03] D. Nussbaum and X. Wu, “An Architecture of a Scalable Wireless Monitoring System”, Proceedings of 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pages 1386 – 1389, Cancun, Mexico, 2003.
- [Pat02] C.S. Pattichis, E. Kyriacou, S. Voskarides, M.S. Pattichis, R. Istepanian, C.N. Schizas, “Wireless Telemedicine Systems: An Overview”, IEEE Antenna’s and Propagation Magazine, pages 143-153, Vol.44, No.2, April 2002.
- [Pic99] J. Picot, “Towards a Methodology for Developing and Implementing Best Practices in Telehealth”. The Impact of Telemedicine on Health Care Management, IOS Press, 1999.

- [Res02] D. Reske, Z. Moussavi, “Design of a web-based remote heart-monitoring system”, Proceedings of the Second Joint 24th Annual International Conference of the IEEE Engineering in Medicine and Biology Society and the Annual Fall Meeting of the Biomedical Engineering Society, pages 1847- 1848, Houston, TX, U.S.A., 2002.
- [Sch00] H. Schulze, and K. Irmscher, “Mobtel – A Mobile Distributed Telemedical System for application in the Neuropsychological Therapy”, Trends in Distributed Systems: Towards a Universal Service Market, Munich, Germany, 2000,
<http://citeseer.nj.nec.com/schulze00mobtel.html> .
- [Shi92] N.G. Shivaratri, P. Krueger, and M. Signhal, “Load Distributing for Locally Distributed Systems”, IEEE Computer, pages 33-44, December 1992.
- [Tak01] M. Takizawa, S. Sone, K. Hanamura, and K. Asakurs, “Telemedicine System Using Computed Tomography Van of High-Speed Telecommunication Vehicle”, IEEE Transactions on Information Technology in Biomedicine, pages 2-9, Vol. 5, No. 1, March 2001.
- [Wal00] Jean Walrand and Pravin Varaiya, High-preformance Communication Networks, 2nd Edition, Morgan Kaufmann.
- [Wood01] B. Woodward, R.S.H. Istepanian, and C.I. Richards, “Design of a Telemedicine System Using a Mobile Telephone”, IEEE

Transactions on Information Technology in Biomedicine, pages
13-15, Vol. 5, No. 1, March 2001.

- [Yog00] K. Yogesan, I.J. Constable, W. Morgan, and D.Y. Soebadi,
“International Transmission of Tele-ophthalmology Images”,
Journal of Telemedicine and Telecare, pages 41-44, Vol. 6, 2000.