Implementation of a Real-time Multi-Channel Gateway Server in Ubiquitous Integrated Biotelemetry System for Emergency Care (UIBSEC)

Gyeongwoo Cheon, Il Hyung Shin, Min Yang Jung, and Hee Chan Kim, *Member, IEEE*

***Abstract*—We developed a gateway server to support various types of bio-signal monitoring devices for ubiquitous emergency healthcare in a reliable, effective, and scalable way. The server provides multiple channels supporting real-time N-to-N client connections. We applied our system to four types of health monitoring devices including a 12-channel electrocardiograph (ECG), oxygen saturation (SpO_2), and medical imaging devices (a ultrasonograph and a digital skin microscope). Different types of telecommunication networks were tested: WIBRO, CDMA, wireless LAN, and wired internet. We measured the performance of our system in terms of the transmission rate and the number of simultaneous connections. The results show that the proposed network communication strategy can be successfully applied to the ubiquitous emergency healthcare service by providing a fast rate enough for real-time video transmission and multiple connections among patients and medical personnel.**

I. Introduction

Recently, studies of telemedicine and ubiquitous healthcare(U-healthcare) have shown that patients can have a chance to get better medical service through remote monitoring of their health condition by medical doctors. With the help of the recent development of wireless communication such as CDMA (Code Division Multiple Access) and WIBRO (Wireless Broadband), limitations of communication bandwidth, location, and time have been overcome and researchers have started to propose diverse health monitoring devices for U-healthcare services [1], [2].

There have been many studies of the considerations for successful U-healthcare services; target disease, patients' ages, and sensor modules for measuring bio-signals. However, there are few studies, to our knowledge, on the optimized network communication strategy for better management of the transmitted data. One strategy is to use a peer-to-peer network communication and in terms of implementation and data handling, it has many advantages. Because it is based on one-to-one connection, however, it is hard to be applied to emergency situation with a large number of patients. Also, one-to-one connection is hard to handle situations flexibly which are necessary to change data format or protocol simultaneously for devices using different data types.

Previously, we have developed a network server system, the Ubiquitous Integrated Biotelemetry System for Emergency Care (UIBSEC) supporting only one patient-to-one caregiver connection for emergency healthcare services [3]. In order to solve the fore mentioned problems, we need to upgrade the UIBSEC to a multi-channel gateway server. The new server is expected to meet the following requirements. It should be able to overcome the limitation of peer-to-peer communication method by supporting simultaneous network communications among diverse network devices. By incorporating a data processing capability over simple data communication function in this server, we should be able to handle the acquired data more systemically, through which the data from different types of telemedicine devices can easily be transferred to a medical information system.

In this paper, we propose a real-time N-to-N network communication strategy and implement it as a gateway server to be applied to our UIBSEC. Performance of the proposed network server system was evaluated with four different types of health monitoring devices of a 12-channel ECG, a SpO2, and a portable ultrasonograph and a hand-held digital microscope.

II. Materials and Methods

A. Requirements of Gateway Server

Functional specifications of the proposed gateway server are given as follows;

First, the gateway server should be able to recover from malfunction by oneself. Because the server mediates all clients' data streams, malfunction of server is able to result in disconnection of all clients and contamination of data stream. Therefore, the server should have modules for not only data mediation but also monitoring these data mediating modules.

Second, it should be designed to handle exceptions which are caused by the unstable network condition. Because

Manuscript received October 9, 2009. This research work was supported by the Nano/Bio Science & Technology Program (2008-00863) of the Ministry of Education, Science and Technology (MEST), Republic of Korea.

Gyeongwoo Cheon is with Interdisciplinary Program, Biomedical Engineering Major, Graduate School, Seoul National University, Seoul, Korea (email: malignant82@melab.snu.ac.kr)

Il Hyung Shin is with Interdisciplinary Program, Biomedical Engineering Major, Graduate School, Seoul National University, Seoul, Korea (email: lepagats@melab.snu.ac.kr)

Min Yang Jung is with Department of Computer Science, Johns Hopkins University, Baltimore, MD, USA (email: myj@jhu.edu)

Hee Chan Kim is with Department of Biomedical Engineering, College of Medicine and Institute of Medical and Biological Engineering, Medical Research Center, Seoul National University, Seoul, Korea (email: hckim@snu.ac.kr)

wireless network, especially, can be influenced easily by surroundings, bandwidth of the network varies severely and unexpected disconnections occur frequently. Therefore, the server should be robust to precarious network condition.

Third, the server should be able to manage multiple communication channels simultaneously so that each channel can be used for different purposes. To maintain independence of each channel, the channels are operated by different processes.

Fourth, the server and clients use TCP/IP networks. Because the transmitted data have medical information, the network communication should guarantee integrity. Because TCP/IP networks support only one-to-one connection, we use a Server-Client model to control data flow for N-to-N connection. It is also possible to change protocol during mediation of the data flow because all data pass through the server.

Fifth, the network server should permit raw data communication. The term, 'raw data' means data which do not have protocol. By permitting raw data communication, the server can be used by any U-healthcare devices using TCP/IP networks.

Sixth, the gateway server and client module should use single socket. There are many embedded system devices which can use only one socket. Cellular phone is one example. Therefore, to include these devices, the gateway server use single socket for each client.

B. Organization of Network Framework

Our network framework consists of client module and the gateway server.

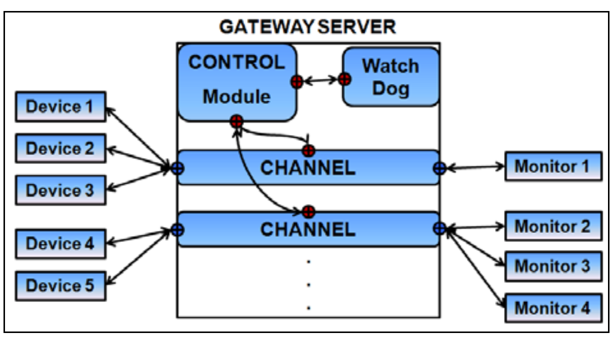

Fig. 1. Organization and data flow of the system. Device can be various health monitoring devices and monitor means medical caregiver.

1) Client Module:

Client module is used in terminal devices and is supplied in the form of dynamic link library (DLL). Visual C#.NET™ language is used. By abstracting network communication part from an interface program, library users can only use `Open`, `Close`, `Write`, and `Read` functions and `Write` function is divided into `WriteSerial` and `WritePacket` again. `Open` function is to connect client to the gateway server. `Close` function is to disconnect client from the server. `WriteSerial` function is to send continuous data such as ECG data and `WritePacket` function is to send data which need be separated by independent meaning such as each frame data in video stream. Although TCP/IP guarantees the data integrity, the specific size of receiving data can be different from sending data because of internal buffering of socket. Thus, `WritePacket` function is necessary to guarantee that the raw data user can receive the data which is the same size as sending data. `Read` function is to receive data from the server. Clients can send next data packet only after receiving the message, which confirms that server receive the client's data. By using this way, transmission rate can be controlled in terms of network condition. Data buffering is done during the interval for effective network communication [4].

2) Server Module:

C# is also used to develop the gateway server. The server is divided into three modules: Channel module, Control module, and Watch-Dog module. Each module is operated by independent process to prevent mutual interference.

Channel module carries out actual mediation of data stream among clients according to the client's level of send and receives. Multiple channel modules can be created and each channel has a different port number.

Control module manages channels. The module gets reporting messages from the channels periodically. By these messages, control module can confirm each channel's status. If the control module fails to receive the reporting message from channel or receives a message reporting abnormal status of channel, the control module terminates the target channel and restarts it. The clients who access to the channel during this process try to reconnect automatically and data is buffered in client module.

Watch-Dog module monitors the control module. By periodic report message, watch-dog module can monitor the condition of the control module. Since we assume that it is possible that the control module is able to fall into malfunction state because it carries out many complex functions, watch-dog module is needed to guarantee the stability of the control module.

C. Performance test and Applications

1) Server Performance Test:

The transmission rate for various bio-signals is very different and the number of users is also very different according to a specific purpose and time. Therefore, it is necessary to check how fast our server can mediate data and how many clients our server can accommodate. The maximum transmission rate is measured in several specific periodic times, and then, we test how many one to one communication pairs are permissible at the transmission rate. We executed this test in local network to remove the influence of specific network condition because at first, we wanted to check the performance of the server. We classified the test into two categories, 'Serial type', and 'Packet type' in terms of the client functions: `WriteSerial` and `WritePacket`.

2) Application Test:

We prepared three applications to test whether our gateway server operates reliably in actual condition. Each case uses different devices, networks, and transmission types.

i) ECG Data

Patients' ECG is measured by 12-channel ECG devices and then transmitted to hospital. HiCare Dr.U (Elbio, Korea) is used for main device and WIBRO is used for communication. The required minimum transmission rate is 6Kbyte/s because the sampling rate is 500Hz in each channel. Serial type transmission is used because ECG data is continuous. Clients for monitoring can be plural and use wired internet.

ii) SpO2 Data

Embedded system is used for main device and CDMA is used for communication. SpO2 of patients is measured in every second and transmitted in each ten minutes. Because we cannot use network library, raw data communication is used. Clients for monitoring can be plural and use wired internet.

iii) Video Data

Video data is a important bio-signal to diagnose patient's condition [5]. And in this application, three type video streams - patient view, ultrasonograph, and magnified skin image - are transmitted using WIBRO. Measuring devices are TiTAN™ (SonoSite®) for ultrasonograph and DigiMicro USB microscope (DigiMicro) for magnified skin image. These measuring devices are connected to laptop or HiCare Dr.U. The size of patient view is 320 by 240 pixels and 640 by 480 pixel sizes are used for ultrasonograph and magnified skin image. Color depth is 24bit RGB. The frame rate is 20fps and H.264 format [6] is used to reduce data size. Transmission rate varies depending on contents used.

III. Results

A. Server Performance Test Result

The system resource for test is that Intel® Core™ Quad CPU Q6600 2.40GHz 3.25GB of RAM and operating system is Microsoft Windows XP Professional server pack3.

In fixed condition, which is specific system resource and operating system, the maximum transmission rate is mainly dependent to algorithm of the framework and the maximum number of one-to-one pair is related to the load which is necessary to operate the whole programs.

According to the fig. 2 and fig. 3, serial type is faster than packet type and the transmission rate of serial type has small variation in the range of 400~420Kbyte/s. On the other hand, in the range of 80~140ms, the maximum transmission size of packet type is fixed in 31.5 Kbyte. In this range, the transmitted data cannot be buffered efficiently because we set the maximum buffering size to 63Kbyte and the size larger than 31.5 Kbyte can only be buffered once. Instead, the maximum number of one-to-one pairs increases in this range (fig. 3) because less system load is imposed to the computer in this time range.

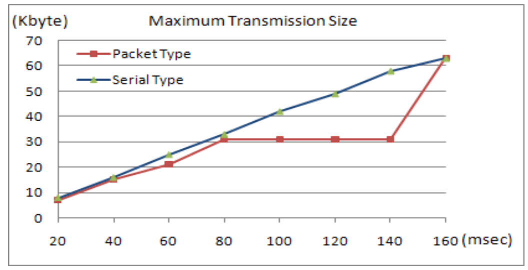

Fig. 2. Maximum Transmission Data Size per specific transmission periods.

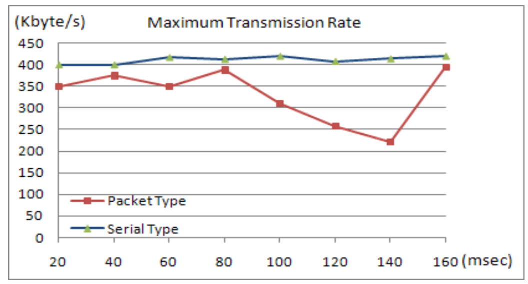

Fig. 3. Maximum Transmission Rate per specific transmission period.

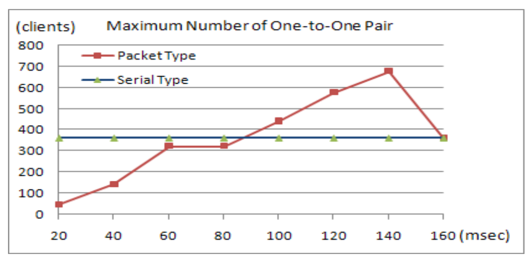

Fig. 4. Maximum Permissible number of One-to-One client pairs.

B. Application Test Results

1) ECG module

In this application, we can use our library in client program. Therefore, data stream was fully managed by our system. The transmission rate, 6Kbyte/s, was in the range of WIBRO bandwidth. Although there was an unexpected disconnection problem where WIBRO could not cover, the problem was able to be overcome in the range of buffering.

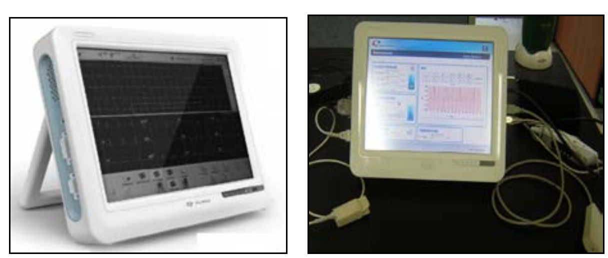

Fig. 5. ECG device (HiCare Dr. U) and application program

2) SpO2 module

In this application, we should use raw data communication because our client library cannot be used in the embedded system. Because data flow cannot be fully controlled, slow transmission rate is recommended in this module. We used only 20 bytes/min transmission rate including supporting header data. The system operated stably.

Fig. 6. SpO2 device and application program

In video application, the measured average transmission rates were 30 Kbyte/s for patient view, 112 Kbyte/s for ultrasonograph, and 185 Kbyte/s for magnified skin image. Even though the captured image sizes of ultrasonograph and magnified skin image are the same, the encoded data sizes are different because of the inequality of information. We could control the data size by changing fps (frame per second), Bit rate [7], and GOP size (key frame rate). There was a disconnection problem remarked previously in ECG application. Also, there was a data loss problem while transmitting any two or more combination of video data due to bandwidth limitation. Theoretically, WIBRO can transmit 5Mbit/s or 640Kbyte/s (downlink) but in our experiments, the maximum rate was around 240Kbyte/s.

Fig. 7. Ultrasound device and application program

Fig. 8. USB digital microscope devices and application program

Despite of wireless network limitation, we were able to confirm that our gateway server can mediate data flow reliably in the three applications. Our network framework minimized the effect of instability of the networks.

IV. DISCUSSION

We developed a gateway server to support various types of bio-signal monitoring devices for ubiquitous emergency healthcare in a reliable, effective, and scalable way. The server provides multiple channels supporting real-time N-to-N client connections. Especially, multiple medical care givers were able to monitor a number of patients simultaneously through multiple channels supporting N-to-N connections.

In the case of wireless internet, transmission of heavy data stream such as video data may suffer from a data loss problem due to the bandwidth limitation of wireless network compared with the wired network and the problem can be worse in the hetero-bandwidth networks.

The function—changing protocols—is not implemented yet although we proposed several possible methods for it. The reason is that we need much more discussion about data interlocking between our framework and the medical information system in hospital in order to utilize these telemedicine data for actual medical care.

The conclusion to be drawn here is that our network framework is expected to separate data managing part from data acquisition part in health monitoring devices. This means that our system gives an opportunity to concentrate on data management and we will be able to use our system as a basis for future studies to collect many types of bio-signals using different networks.

ACKNOWLEDGMENT

This research work was supported by the Nano/Bio Science & Technology Program (2008-00863) of the Ministry of Education, Science and Technology (MEST), Republic of Korea.

REFERENCES

[1] Jaemin Kang, Honggu Chun, Il Hyung Shin, Sang Do Shin, Gil Joon Suh and Hee Chan Kim, "Preliminary evaluation of the use of a CDMA-based emergency telemedicine system", *Jounal of Telemedicine and Telecare*, Vol.12, No.8: pp.422-427, 2006

[2] Jaemin Kang, Il Hyung Shin, Yoonseo Koo, Min Yang Jung, Gil Joon Suh, and Hee Chan Kim, "HSDPA (3.5G)-Based Ubiquitous Intergrated Biotelemetry System for Emergency Care", Proceedings of the 29^{th} annual International Conference of the IEEE Engineering in Medicine and Biology Society, article number 4353126, pp.3665-3670

[3] Min Yang Jung, Jaemin Kang, Il Hyung Shin, Yoonseo Koo, and Hee Chan Kim, "Implementation of Relay Server for the Ubiquitous Integrated Biotelemetry System for Emergency Care (UIBSEC) based on 3.5G HSDPA Technology", Proceeding of 29^{th} annual International Conference of the IEEE Engineering in Medicine and Biology Society, article number 4353901, pp.6713-6716

[4] Prasad, R.S., Jain, M., Dovrolis, C. "Socket Buffer Auto-Sizing for High-performance Data Transfer", (2004) Journal of Grid Computing, 1, pp. 361-376.

[5] Meade B, Barnett., "Emergency care in a remote area using interactive video technology: a study in prehospital telemedicine", *Journal of Telemedicine and Telecare* 2002; 8: 115-7 New York: McGraw-Hill, 1964, pp. 15-64.

[6] http://en.wikipedia.org/wiki/H.264/MPEG-4_AVC(last accessed: April 20, 2009)

[7] http://en.wikipedia.org/wiki/Bit_rate(last accessed: April 20, 2009)