논문 2008-45CI-3-13

u-헬스케어 시스템과 네트워크 트래픽 매니지먼트 시스템의 데이터베이스 설계 및 구현

(Database Designs for u-Healthcare System and Magi Network Traffic Management System)

용 기 택*, Teemu H. Laine*,이 채 우*

(Ki-Tak Yong, Teemu H. Laine, and Chae-Woo Lee)

요 약

본 논문에서는 u-healthcare system과 트래픽 모니터링 및 분석 시스템인 MAGI의 시스템에 대해 설명하고 각각의 데이터베이스를 분석한다. u-healthcare system은 생체 신호 센서 네트워크와 모바일 게이트웨이를 이용하여 환자의 생체 신호를 환자의 장소와 시간에 관계없이 모니터링하고 분석하는 시스템이다. MAGI는 기존의 트래픽 모니터링 툴의 문제점을 보완한 실시간 트래픽 모니터링 및 분석 시스템이다 MAGI 실시간 트래픽을 분석을 통하여 다이나믹한 방화벽을 구현 할 수 있다. 본 논문은 구현한 시스템의 분석은 물론 데이터베이스의 스키마 및 저장된 데이터를 자세히 분석 하였다. 또한 본 논문은 이론적인 전개와 실제 테스트를 통한 정성적인 성능 분석을 통하여 시스템의 적절성을 평가하였다.

Abstract

In this paper we present two databases for two different systems each having novel implementation ideas: a u-Healthcare system with a mobile gateway and MAGI network monitoring system. u-Healthcare system is capturing and monitoring vital sign data without restrictions of location and time using body sensor network and a mobile gateway. MAGI is a system which has a dynamic firewall function and solves problems of existing traffic monitoring tools. In addition to presenting the design principles behind these two systems, we describe the respective database schemas together with detailed information of the data stored within the databases. We will also show and discuss performance measurements and calculations for both systems. Based on the performance data we will discuss the systems' suitability for their intended uses.

Keywords : database, sensor network, traffic monitoring, zigbee, bluetooth

I. Introduction

Every day modern people deal with large amounts of information. It is important to make a distinction between necessary and unnecessary information, and save the information useful to us. For storing the important information in digital format, a digital system called database is often used; it is a collection of data which is saved to serve our information needs. Databases are used in various fields, including sensor networks and traffic analysis - the fields related to this paper.

Databases can be divided to categories according to their characteristics and form. These categories are as follows.

Hierarchical Database: It is a database which has the form of tree branches. It is an assembly of record types and links, so when a record is

* 정회원, 아주대학교 전자공학과
(Department of Electronics, Ajou University)

※ 본 연구는 지식경제부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음 IITA-2008-C1090-0801-0014(2008년도 사업)

접수일자: 2008년4월26일, 수정완료일: 2008년5월6일

removed, all child records connected with that particular record are also removed. The tree has a simple structure; it helps constructing a database and searching data becomes simpler as well. However, the procedures of inserting and deleting data are rather complicated and make multi-connection harder due to the tree's characteristics.

Network Database (Mesh type): It does not have a tree structure type as in Hierarchical Database; codes are connected by the mesh network. Various searching is available, and records have connection to their owner and members. Network database is difficult to program due to the complexity of the structure. Since it is also connected with the entire record, keeping the independence is challenging.

Relational Database: It is a model simplified of hierarchical database and network database. The most attractive characteristic of this kind of database is that it is modeled in form of relational tables. A relational database is realized by using tables and as a result, making queries and converting it to other databases is easy. It can also express different connections; such as 1:1, 1:N, and N:M (N and M are unequal).

Object-oriented database: It was originally developed in the late 1980s and is based on object-oriented programming. Information stored in the database is represented as objects which can be manipulated directly from appropriate object-oriented programming languages. As a result, handling data as objects is easy and processing complicated data can be done simply. However, object-oriented database is hard to use in real situation because this database is more of a conceptual form and not much used.

Object Relational Database: It was originally developed in the late 1990s with an aim to import the object-oriented database concept to relational database. As processing of multimedia data is becoming more and more popular, this kind of databases are used more widely.

Databases can also be categorized into real-time processing and batch processing by the order of data processing. In the real-time processing model a database system is processing data as soon as it arrives. Real-time processing is useful for example in real-time booking services and inquiring bank accounts which need speedy response. On the other hand, batch processing processes data all at once after the data has been gathered. Examples of systems using batch processing are score calculations and electric charges.

Ubiquitous healthcare is the latest trend in bringing healthcare systems available regardless of patients' location and time. With a ubiquitous healthcare system it is possible to monitor various vital signs of a patient such as ECG and heart rate this is particularly useful for patients living in rural areas far away from medical facilities. Common method in building a ubiquitous healthcare system is by using a wireless sensor network and appropriate gateway device which mediates the communication between the wireless sensor network and a more accessible network such as the Internet. In order to get maximum benefit from the captured sensor data, they should be stored in a centralized database for global access.

Network monitoring systems are used to monitor the status of a network in order to discover problems caused by overloaded or crashed devices, broken connections, and slow nodes. Without network monitoring tools, work a network administrator would become significantly more difficult or even impossible. Network monitoring systems can send alarms to the network administrator if one of the predetermined thresholds is exceeded. In order to utilize captured network traffic data for analysis and prediction, they should reside in a database accessible by network monitoring and analysis tools[1].

In this paper we present two relational databases for two different systems each having novel implementation ideas: a u-Healthcare system with a mobile gateway and MAGI network monitoring system. In addition to presenting the design principles behind these two systems, we describe the respective database schemas together with detailed information of the data stored within the databases. We will also show and discuss performance measurements and calculations for both systems. Based on the performance data we will discuss the systems' suitability for their intended uses.

This paper is composed as follows. u-Healthcare system and MAGI network traffic monitoring system will be introduced on chapter 2. In chapter 3 will be presented the implementation of our u-Healthcare system's database and network traffic monitoring system's database in detail. And in chapter 4 we present observations of our system's performance and discuss improvements. Finally, the chapter 5 concludes the findings.

Ⅱ. Background

In this chapter, we introduce u-Healthcare System with a mobile gateway and network traffic monitoring system. Both systems use a relational database.

1. u-Healthcare system with mobile gateway

Ubiquitous, or omnipresent, computing systems are currently under active research. The availability and affordability of ubiquitous mobile devices and components for smart environments allow researchers to build new systems rapidly with relatively low costs. Ubiquitous computing systems have been employed for example in education[3~4], military[5~6], transportation and supply chains[7~8], and tourism[9]], [10]. Common characteristics for most of the ubiquitous systems is the high mobility of the users, utilization of sensor technologies such as RFID, and invisibility of the surrounding technologies. The major advantage of ubiquitous systems is that users are able to access contextual and potentially individualized information regardless of location and time.

Wireless sensor networks achieve environment monitoring and controlling through use of small devices of low cost and low power. Such networks are comprised of several sensor nodes, each having a microprocessor, sensor(s), actuator and wired/wireless transceiver inside a small device. The sensor nodes provide intelligence information and useful services for common human life through negotiation over ZigBee protocol with their neighbor nodes. Wireless sensor networks can be utilized widely in application fields such as medical care, military tactics, home networking, environment monitoring and so on. Recently, in the medical field sensors to evaluate different types of vital signs (heart rate, blood pressure, body temperature, etc) have been developed. These body sensors are small in size, light-weight and wearable, thus they enable wireless health monitoring regardless of location and time. Wireless body sensors are in the key role when ubiquitous healthcare (or u-Healthcare) systems are being built.

Mobile devices have recently developed into highly potential client platforms for ubiquitous applications. Screen sizes and input methods especially in high-end devices meet now the input and output requirements of most mobile applications. At the same time processing and storage capacity of devices have grown to run 3D graphics and high quality videos. High-speed Internet access (3G, Wi-Fi) is available in most of the modern mobile devices and GPS is becoming a standard feature. Furthermore, Bluetooth is already a de facto feature available in most of the mobile devices today. With these features at hand, we sought to incorporate a Bluetooth-based mobile gateway into a ubiquitous healthcare system built upon a wireless sensor network. The aim of the mobile gateway is to mediate communication between a wireless network and other network such as the Internet. The data is stored in a centralized database for global access.

The main objective of the u-Healthcare system is to allow capturing and monitoring of vital sign data without restrictions of location and time. While bodily sensors are portable and can be moved around easily, the sensed data must be collected and stored somewhere for further processing. In case of an

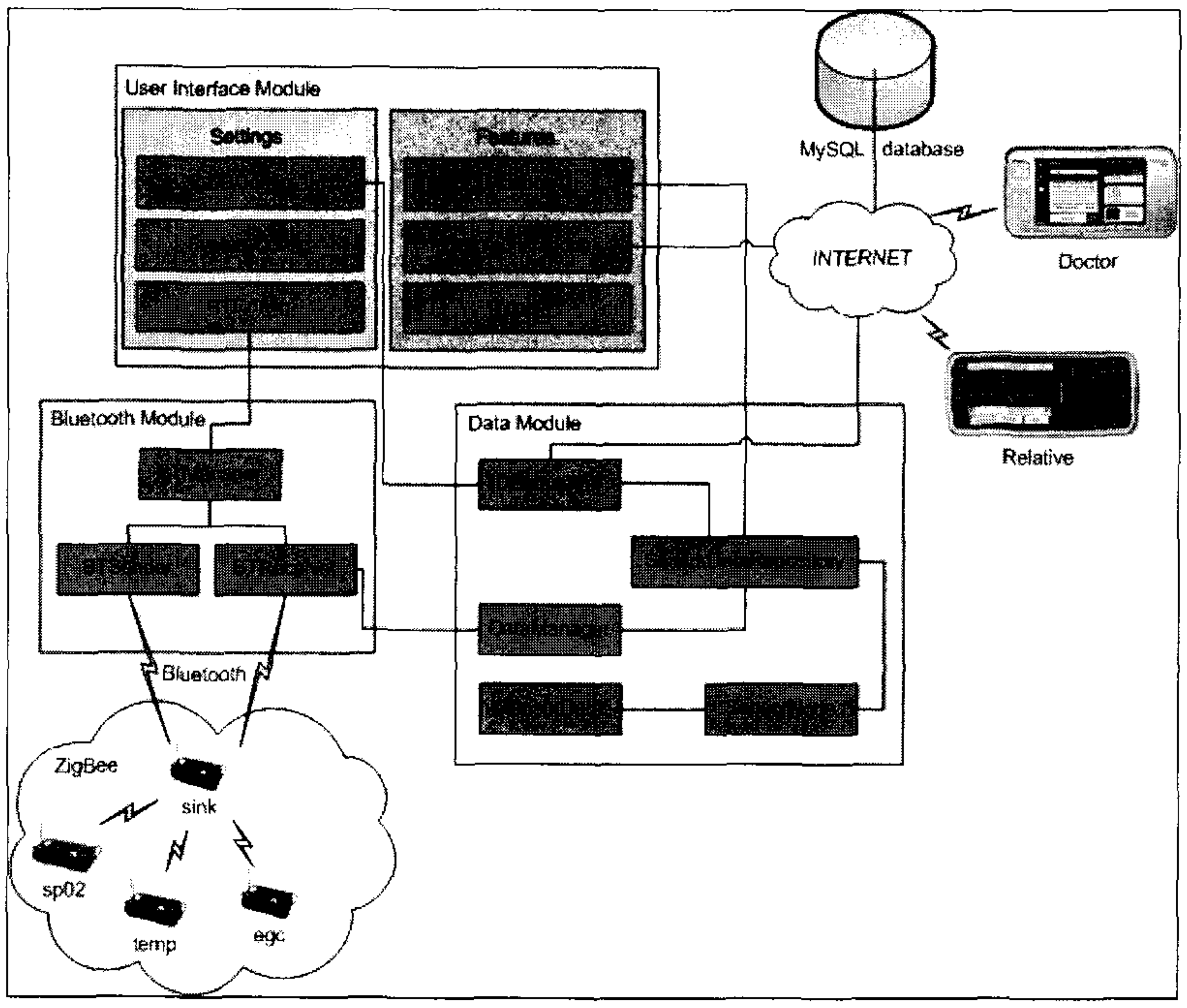


그림 1. u-헬스케어 시스템을 위한 모바일 게이트웨이 아키텍쳐
Fig. 1. Architecture of the Mobile Gateway for u-Healthcare Systems.

ordinary hospital environment, sensors can be conveniently wired to a computer which in turn does the data processing. However, if the target is moving, a mobile data processor is needed. For this purpose we designed a mobile gateway for the u-Healthcare system which allows location-agnostic sensing and monitoring. Once the data has been processed and stored in a centralized repository, it can be retrieved by other clients (mobile and stationary) that can monitor and analyze the data further in almost real time.

From technical perspective, the mobile gateway has a modular structure as depicted in figure 1. Three main modules are User Interface Module, Bluetooth Module, and Data Module. After the vital sign data has been captured by the nodes of wireless sensor network and aggregated by the sink node, the data is transferred to the BTReceiver component which first checks the validity of the data and then forwards it to the Data Module. Data module then parses the data and stores it both locally (for caching) and remotely. Data Module has been design so that it can be easily extended to utilize any type of sensors having any types and amounts of sensor data. After the data has been parsed and stored successfully, the user interface component is notified and the data is visualized in real-time on the screen of the mobile device. Similar visualization can be done by the doctor over the Internet by using the data stored in the centralized database. In addition to visualization, the User Interface Module has various tools for facilitating communication between the patient and a doctor or any other relevant party. Furthermore, patient is able to adjust the behaviour and performance of the system through various settings, e.g. he can modify the sensing period of sensor nodes or start/stop sensing process of any node.

u-Healthcare system with a mobile gateway can be used in various contexts. For example, a personal trainer can monitor athlete's physical status with a mobile device through medical sensors attached to athlete's body. Soldiers in war or training can have medical sensors attached on their bodies and medics can monitor soldiers' health on mobile devices through collected vital signs. In addition, medical sensors can be attached to patients residing at places which are far from medical facilities. In such cases, vital signs collected from the sensors are transmitted to hospital through a mobile device which is connected to the Internet. Caregivers such as doctors and nurses in the hospital receive the vital sign data through the Internet. The caregivers can thus constantly monitor a patient's status, and if it changes, they can immediately respond to the situation appropriately. Medical students can utilize the same system for educational purposes.

The development of u-Healthcare system with mobile gateway is a part of the SensorPlanet program initiated by Nokia Corporation[2] and the research is conducted as a joint effort between Ajou University in South Korea and University of Joensuu in Finland. Aim of the SensorPlanet program is to gather sensor data from various research projects into a global database. The data can be then used for further analysis, mining, visualization, machine learning, etc. Research challenges declared by SensorPlanet regard volume of the collected data, adaptation and configuration of the devices, integration of wireless sensor networks and other networks, power management, and user interaction. Our u-Healthcare system concentrates primarily on the network integration and user interaction aspects, yet power consumption issues are addressed as well.

2. MAGI

Recently, the internet has become one of the most important ways of communication in our lives. As a result, the amount of network traffic is larger than before. A thing that has a great influence on the stability of networks is the excessive increase of the traffic. Therefore, a real-time system tool which can analyze traffic of a network is very important. But the existing network monitoring tools do not have database or have a poor database system. Therefore they are not suitable for analyzing long-term activities of a network. Furthermore, the most existing traffic monitoring tools show merely simple results, e.g. the amount of traffic[1].

MAGI is a system which has a dynamic firewall function and solves the problems of existing traffic monitoring tools. MAGI is activated as a bridge on the network and it monitors all traffic of the network. The analyzed traffic is stored in a database in real time. The stored results can be accessed without restriction over the Internet, and even another system can inquire the results of the analysis. It can make various firewall rules based on the analysis result. MAGI's structure combines software and hardware; this differs from the existing monitoring tools which are usually developed in software form.

In the software part, the web user interface differs from the existing monitoring systems by allowing traffic analysis and firewall management, as well as monitoring of MAGI's hardware part. Because the user interface can be accessed by PHP with MySQL, it has an advantage that the data analysis is simple for the user. The hardware part is independent from the network environment. Also, management of the hardware can be applied to diverse environments by modifying the interface based on ncurses[1].

The example of measuring network traffic using MAGI is shown in figure 2. All the packets flowing from external network to internal network are monitored.. In the case of installing MAGI between a router and hub all client connections within the network can be managed. Each client is very simple because the firewall of that client can be managed by MAGI. It is also possible to install MAGI to different network configurations (hub and client, client and

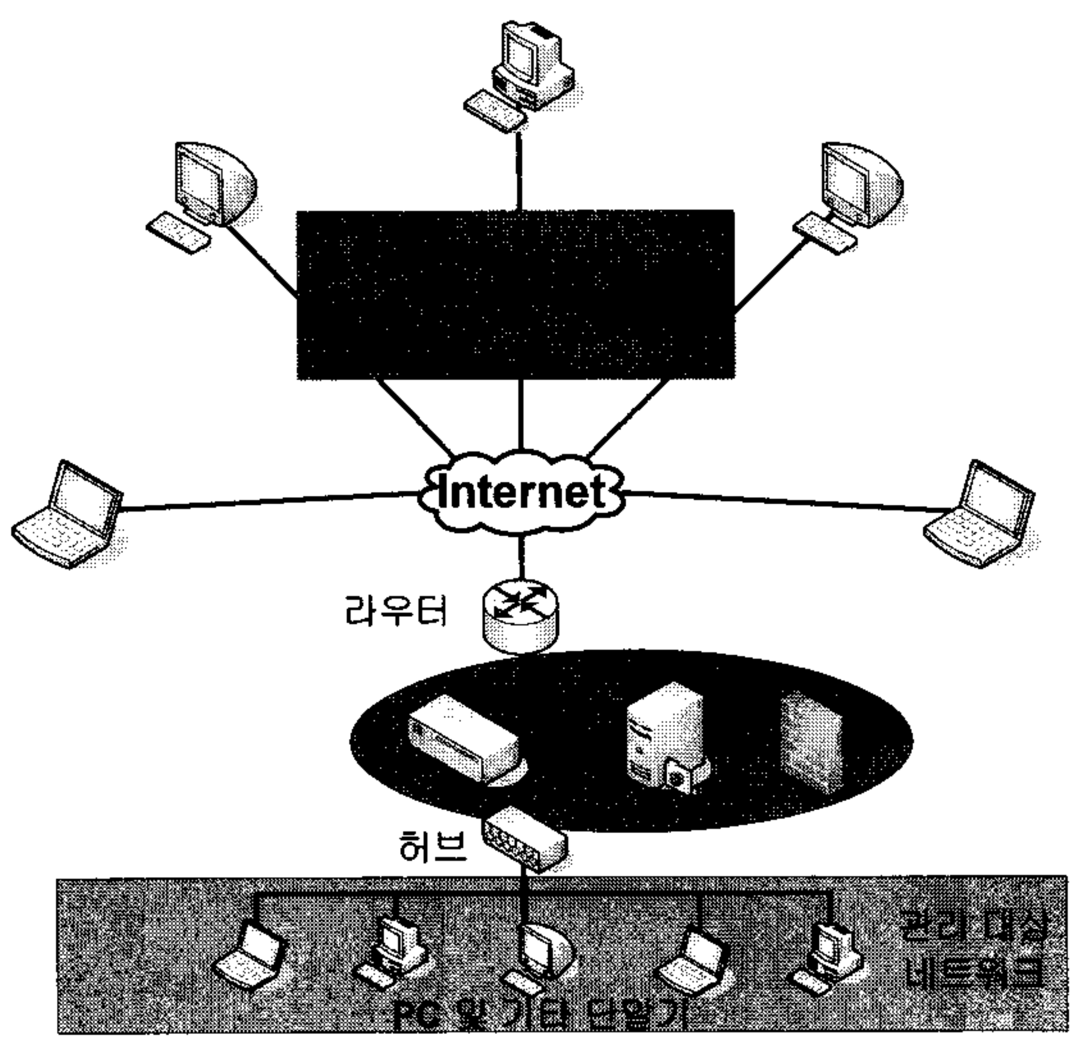


그림 2. 네트워크 트래픽 매니지먼트를 위한 네트워크 구성의 예
Fig. 2. Example of network organization for network traffic management.

client and so on). This is because MAGI can work in bridge mode.

Ⅲ. Database Implementations

Databases of u-Healthcare system and MAGI are materialized with MySQL. MySQL is an open-source replational database engine which has been recently acquired by Sun Microsystems. It is easy to set up and use, and it is supported on various operating systems. It also has different application program interfaces(API) for C, C++, Eiffel, JAVA, Perl, PHP, and Python script, so that it is highly portable. As a result of being open-source, MySQL can be used free of charge.

We used MySQL in u-Healthcare system and MAGI system based on LAMP configuration. LAMP stands for Linux operating system, Apache web-server, MySQL, and PHP script language. A database of each system is composed as follows.

1. Database for u-Healthcare system

A database of u-Healthcare system consists of four tables; Location table which stores position information at which the data is sensed, Sensor_data table which stores the sensed data, Sensor_data_type

표 1. 환자의 위치 정보 테이블
Table 1. Table of patient's location.

location			
PK	location_id	int	unsigned
	name	varchar	utf8_unicode_ci
	description	varchar	utf8_unicode_ci
	longitude	float	
	latitude	float	

표 2. 측정된 데이터 테이블
Table 2. Table of sensed data.

sensor_data			
PK	sensor_data_id	int	unsigned
	type_id	int	unsigned
	target_id	int	unsigned
K	timestamp	bigint	unsigned
	value	int	unsigned
	location_id	int	unsigned

표 3. 데이터 타입 정보 테이블
Table 3. Table of data type.

sensor_data_type			
PK	type_id	int	unsigned
	name	varchar	utf8_unicode_ci
	description	varchar	utf8_unicode_ci

표 4. 환자 정보 테이블
Table 4. Table of patient information.

target			
PK	target_id	int	unsigned
	name	char	utf8_unicode_ci
	age	int	unsigned
	gender	char	utf8_unicode_ci
	location	char	utf8_unicode_ci
	notes	char	utf8_unicode_ci

table which stores the type of the sensed data, and Target table which stores patients' information. Tables 1-4 display the database schema of the u-healthcare system.

Location table is made up with five fields; location_id, name, description, longitude, and latitude. It is connected with the sensor_data table in order to register patients' current position. The field location_id is the primary key, and is used for a fast search. Since its data are simple numbers, it has an integer type. Name field stores name of the location, and description denotes further details of the location. Since these two fields are for keeping characters, they are of varchar type. Longitude and latitude fields used for storing GPS information of patient's position. The GPS feature is not yet implemented.

Sensor_data table is the most important table of the u-Healthcare system. It is the central table and other tables are connected to it. Sensor_data_id field is a unique ID for each row of sensed data. Type_id is a field informing what kind of sensor is used. Type_id is a secondary key to the ID field of the sensor_data_type table. The target_id field stores a unique ID of the target, i.e. patient. This can be for example social security number. The moment of time at which a data is captured is stored in the timestamp field. This is a millisecond value of Unix time type. We use the timestamp field as an index key to make searching more efficient. Value field stores the sensed data. It has the data range from -10,000 to +10,000, and integer type. Location_id refers to the location table.

Sensor_data_type stores information of different sensor data types. The sensor_data table refers to the type_id primary key for each stored row of data. Name field is the name of the sensor, and description field contains detailed description of the sensor. The target table stores patients's (or any other target's) information. It has six fields that are denote id, name, age, gender, location, and additional nodes of the target. The location field does not refer to location table; it merely provides information of the target's residence. Target_id field is the primary key and it is referred to from the sensor_data table.

As we examine the u-Healthcare system's database structure, it has 1:3 virtual relationship. It means they have no real relation with other table. But sensor_data table uses other data table's data when it needs. This structure is organized by each table characteristics; sensor_data table is updated in constantly but other tables have more static data.

2. Database for MAGI

A database of MAGI consists of etc_info, general, ip, and resource tables, as presented in tables 5-8.. These four tables are used for system information, protocol analysis and port analysis which are shown on the MAGI web user interface. The MAGI database uses index, so tables are not directly connected. The reason why MAGI's database does not use relations is because the system is designed to store huge amounts of data, and relations between tables would decrease the system performance. Therefore, in case of MAGI, every table is made up with parallel formations according to different usage. Even though there are no particular connections between tables, MySQL is able to access to any table and obtain data from others by SQL queries.

표 5. etc_info 테이블
Table 5. etc_info table.

etc_info			
index	time	int	unsigned
	ttl_0	int	unsigned
	ttl_1	int	unsigned
	ttl_2	int	unsigned
	ttl_3	int	unsigned
	ttl_4	int	unsigned
	ttl_5	int	unsigned
	ttl_6	int	unsigned
	ttl_7	int	unsigned
	unicast_traffic	int	unsigned
	unicast_count	int	unsigned
	broadcast_traffic	int	unsigned
	broadcast_count	int	unsigned

표 6. general 테이블
Table 6. general table.

general			
index	time	int	unsigned
	p_type	int	unsigned
	in_traffic	bigint	unsigned
	in_packet	bigint	unsigned
	out_traffic	bigint	unsigned
	out_packet	bigint	unsigned

표 7. ip table
Table 7. ip table.

ip			
index	time	int	unsigned
	ip_type	int	unsigned
	src_ip	int	unsigned
	dst_ip	int	unsigned
	src_port	smallint	unsigned
	dst_port	smallint	unsigned
	in_traffic	bigint	unsigned
	in_packet	bigint	unsigned
	out_traffic	bigint	unsigned
	out_packet	bigint	unsigned

표 8. resource 테이블
Table 8. resource table.

resource			
index	time	int	unsigned
	cpu_use	int	unsigned
	mem_use	bigint	unsigned
	disk_use	bigint	unsigned
	traffic_in	bigint	unsigned
	traffic_out	bigint	unsigned

Etc_info table is for storing the result after analyzing TTL size and casting method of packet of monitored traffic. Time field records the time packet captured at pcap, and it is used to draw a graph on the web interface. It is also used as index in order to make searching faster because it has non-duplicated data. The monitored packet is classified by TTL size and stored in fields from ttl_0 to ttl_7. Each ttl_# has 32 intervals of TTL size; 0~31 for ttl_0, 32~63 for ttl_1, 64~95 for ttl_2, and so on. The ttl_7 has 224~256. Unicast_traffic and unicast_count fields store the amount of the traffic and the number of the unicast packets the among monitored traffic. The amount of traffic is stored in bytes. Broadcast_traffic and broadcast_count fields store the amount of traffic and the number of the broadcast packets among the monitored traffic.

Ethernet frame's protocol analysis is stored in general table. It consists of time, p_type, in_traffic, in_packet, out_traffic, and out_packet fields. Time field is also same as time field of other tables. It is used as index of the table and the monitored time. The protocol type of the packet; udp, tcp, or icmp, is recorded at p_type field. In_traffic and out_traffic fields store the amount of in-and-out traffic. In_packet and out_packet fields store the number of in-and-out packets.

The ip table stores data regarding port and ip analysis information. Time field keeps the time of packet arrival and it is also used as table index. Src_ip and dst_ip fields store the source and destination ip addresses, and src_port and dst_port fields store the source and destination port numbers of in-and-out packets. In_traffic and out_traffic fields store amount of come-and-out traffic. Finally, in_packet and out_packet fields keep the number of in-and-out packets. Therefore we know the amount of in-and-out traffic and packets for each ip address.

Resource table holds information from Linux kernel captured by the monitoring system. Time field records the capture time and it is used index on this table. Cpu_use, mem_use and disk_use fields record usage information of CPU, memory and hard disk, respectively. Traffic_in and traffic_out fields store the total amount of traffic which flows through MAGI.

Ⅳ. Performance Analysis

1. u-Healthcare system

The sink node of the wireless sensor network in the u-Healthcare system uses the Bluetooth Option Module from Hanbak Electronic Co. This option module has an embedded FB155BC Bluetooth module from Firmtech Co. Ltd. The data transmission speed between the FB155BC module and a host ZigBee node can be adjusted from 1200bps to 115,200bps. On the other hand, the data transmission speed in a Bluetooth network by using the given components can be as high as 1 Mbps. In our test setting we use an ECG sensor which generates data at rate 19200bps. Considering the maximum speed of the data between a FB155BC module and the sink node (i.e. 115,200bps), we can calculate the theoretical maximum of concurrently operating ECG sensors in a single-sink wireless sensor network to be $115200 \div 19200 = 6$. This value can be increased by using multiple sink nodes each of which communicate with the mobile gateway over Bluetooth. As the maximum speed of Bluetooth data transmission is 1Mbps and the maximum data rate of a single sink node is 115,200bps, the gateway can theoretically be connected to 8 different sink nodes. However, a Bluetooth server can serve at most 7 slaves. From this knowledge we can compute the maximum number of ECG sensors if there are 7 sink nodes in the wireless sensor network: $7 \times 6 = 42$.

The aforementioned figures are based on assumptions of theoretical maximum transmission rates. In practise, interference to the wireless signal and other effects reduce the actual transmission speed. At this phase of development we cannot verify what exactly is the maximum number of ECG sensors in the proposed system. ECG sensors generate fairly large amount of data compared to most environmental sensors and many medical sensors including glucose level monitor and body temperature sensors. This means that the u-Healthcare system may be able to host a larger number of sensors than 42, depending on the sensor types.

2. MAGI

If we calculate the theoretical performance, it is nearly 140Mbps. This numerical value can be obtained as follows. MTU size is about 1500 bytes except the length of header in ethernet level. and database can save one packet for each record. Accordingly we obtain (1500 × 10000) ÷ 1024 ÷1024 = 14Mbps. In this formula, 10000 is amount of packet analysis process during one second in our benchmarking test. This theoretical value is calculated when we process the whole incoming packet to MAGI. Because of characteristics of network traffic, duplicate packets are generated frequently. For this reason we use an AVL Tree to compress replica packets. As a result, we can reduce the number of packets by 10 times than before. According to this, we obtain the theoretical performance as 140Mbps(14Mbps × 10).

In real case, howereve, MAGI system used CPU resources of 80% when traffic amount was near 50Mbps, and maximum CPU resources when traffic amount was near 65Mbps. Low performance is due to slow hard disk. Our hard disk drive had low speed(DMA-66, 5400rpm) which can handle data at 21.87MB/sec. Therefore we benchmarked the performance between another hard drive(DMA-100, 7200rpm) and the original hard drive. As a result, the performance of DMA-100 model is almost two times higher compared to that of DMA-66 model. Table 9 shows the benchmarking results. Especially in this MAGI system, disk drive's performance is very important because database is extremely active.

표 9. 하드디스크 성능 비교 결과
Table 9. Benchmarking result.

하드디스크 종류	속도
UltraDMA 66 + 5400rpm	66MB in 3.02 sec = 21.87 MB/sec
UltraDMA 100 + 7200rpm	157MB in 3.01 sec = 52.15 MB/sec

According to this result, if we change the disk drive, we can obtain performance of about 130Mbps.

Ⅴ. Conclusions

Data management is very important in the presented u-Healthcare system and MAGI system. To store data properly, we constructed database systems by using MySQL. This paper presented the two realized systems and their databases. In the case of MAGI, it needs very high performance because sometimes network infrastructure is working faster than the current speed of MAGI. Hence, in the future research, we will develop a system which has more optimized database structure and higher hardware specifications compared to current implementation of MAGI. In case of u-Healthcare system, even though our u-Healthcare system operates well for small-scale wireless sensor networks having high-rate sensors such as ECG, in the future we will concentrate our research efforts to remove the bottleneck caused by the connector between ZigBee node and the Bluetooth module. Furthermore, the database design of the u-Healthcare system will be adjusted to serve well any kind of sensor environments, not just medical sensors.

참 고 문 헌

[1] K. T. Yong, C. W. Lee, "Implementation of a Real-time Network Traffic Monitoring" JCCI 2008, April 2008.
[2] Nokia Corporation, SensorPlanet Website. http://www.sensorplanet.org/.
[3] C. Islas-Sedano, T.H. Laine, M. Vinni and E. Sutinen, "Where is the answer? - The importance of curiosity in pervasive mobile games", Proc. of the FuturePlay 2007 Conference, 2007.
[4] Ogata, H. and Yano, Y. Context-aware support for computer-supported ubiquitous learning. Proc. of the 2nd IEEE International Workshop on Wireless and Mobile Technologies in Education, 2004, 27-34.
[5] Adkins, M., Kruse, J. and Younger, R. Ubiquitous Computing: Omnipresent Technology

in Support of Network Centric Warfare. In Proc. of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02), 2002.

[6] M.S. Ju and S.S. Kim, A Study on the Use of Ubiquitous Technologies in Military Sector. International Journal of Contents, 2 (2), 2006, 6-9.

[7] Blythe, P. RFID for road tolling, road-use pricing and vehicle access control. IEEE Colloquium RFID Technology, 123(1999).

[8] Kärkkäinen, M. Increasing efficiency in the supply chain for short shelf life goods using RFID tagging. International Journal of Retail \& Distribution Management, 10(31), 2003, 529-536.

[9] Lehner, F., Nsekabel, H. and Seibold, M. Mobile Systems in Tourism: Existing Solutions and Design of an Ubiquitous Platform. Proc. of the Internet and Multimedia Systems and Applications, 2004.

[10] Y.P. Huang, T.W. Chang and W.P. Chuang, An Interactive Handheld Device-Based Guide System Using Innovative Techniques. Proc. of the Second International Conference on Innovative Computing, Information and Control, 2007, 217-217.

저 자 소 개

용 기 택(정회원)

2008년 아주대학교 전자공학과 학사 졸업.

2008년 3월～현재 아주대대학교 전자공학과 석사 과정

<주관심분야 : Traffic Engineering, Sensor Netwrok>

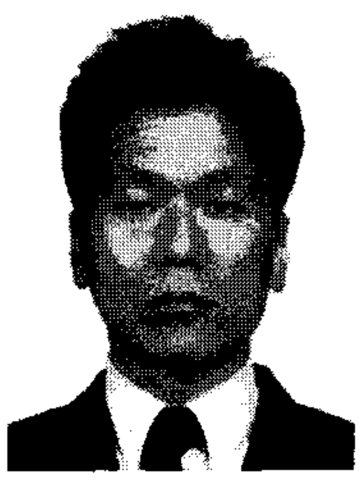
Teemu H. Laine

2001년 Kemi-Tornio Polytechnic 학사 졸업.

2007년 Department of Computer Science and Statistics, University of Joensuu 석사 졸업.

2007년 3월～현재 University of Joensuu 박사 과정.

<주관심분야 : mobile learning, pervasive computing, handheld technologies>



이 채 우(정회원)

1985년 서울대학교 제어계측 학사 졸업.

1988년 한국과학기술원 전자공학과 석사 졸업.

1995년 University of Iowa 박사 졸업.

1985년 1월～1985년 12월 (주)금성통신 연구원.

1988년 9월～1999년 3월 한국통신 선임연구원.

1999년 3월～2001년 9월 Lucent Technologies Korea 이사.

2001년 9월～2002년 2월 한양대학교 겸임교수

2002년 3월～현재 아주대학교 전자공학과 부교수.

<주관심분야 : 광대역 통신망, Ubiquitous networking, Traffic Engineering>