웹 기반의 센서네트워크 질의 및 데이타 관리
(A Web-based Sensor Network Query and Data Management)

황 광 일 † 엄 두 섭 ††
(Kwang-il Hwang) (Doo-seop Eom)

요 약 최근 무수한 센서 노드들로 구성된 무선 센서 네크워크는 물리적인 환경을 감독하고 분석하는데 많은 기여를 할 것으로 기대된다. 그러나, 센서 네트워크는 매우 많은 노드들이 통신에 참여하고 전력이 제한된다는 점에서 기존의 네트워크와 매우 차별된다. 이러한 독특한 특징 때문에 센서네트워크의 데이타 질의를 포함한 센서 네트워크 데이타 관리는 주목할 만한 연구분야가 되고 있다. 또한 인터넷의 활발한 보급과 사용의 편리함 때문에 독립적인 한 네트워크에 대한 감독과 관리에 있어 인터넷을 통한 원격의 웹기반 네트워크 관리기술을 사용하는 것이 일반적으로 고려되고 있다. 그리하여 본 논문에서는 센서 네트워크 데이타 질의 및 관리를 보다 효율적으로 하기 위한 웹기반의 센서 데이타 질의 서버와 이를 기반으로 한 게이트웨이의 구조를 제안하고, 제안된 웹기반의 센서 게이트웨이의 구현관련 세부사항과 그 결과를 기술한다. 제안된 웹기반 게이트웨이는 크게 두 부분으로 구성된다. (인터넷 관련 처리부분과 센서 데이타 처리부분) 센서 데이타 처리부분은 인터넷으로 전달된 사용자의 질의와 해당하는 데이타를 다양한 센서 네트워크 (평면적 또는 계층적)에 적용 가능한 센서 데이타 질의 및 데이타 관리를 대행하는 역할을 하며, 인터넷 관련 처리부분에서는 센서 네트워크와 인터넷 사이의 순조로운 데이타 교환을 위한 모듈화된 게이트웨이 기능을 공급한다.

키워드 : 게이트웨이, 질의 서버, 센서 데이타 관리, Sensor Querying, 무선 센서 네트워크

Abstract Wireless sensor networks consisting of hundreds to thousands of nodes are expected to be increasingly deployed in coming years, as they enable reliable monitoring and analysis of physical worlds. These networks have unique features that are very different from traditional networks, e.g., the numerous numbers of nodes, limitation in power, processing, and memory. Due to these unique features of wireless sensor networks, sensor data management including querying becomes a challenging problem. Furthermore, due to wide popularization of the Internet and its facility in use, it is generally accepted that an unattended network can be efficiently managed and monitored over the Internet. In particular, in order to more efficiently query and manage data in a sensor network, in this paper, the architecture of a sensor gateway including web-based query server is presented and its implementation detail is illustrated. The presented web-based gateway is largely divided into two important parts: Internet part and sensor network part. The sensor network part plays an important role of handling a variety of sensor networks, including flat or hierarchical network architecture, by using internally layered architecture for efficiently querying and managing data in a sensor network. In addition, the Internet part provides a modular gateway function for favorable exchange between the sensor network and Internet.

Key words : Gateway design, Query Server, Sensor Data Management, Sensor Querying, Wireless sensor networks

1. Introduction

Recent advances in wireless communications and micro-electro-mechanical system (MEMS) technologies have enabled the development of low-cost, low-power and small-size wireless sensor nodes.

† 학생회원 : 고려대학교 전자컴퓨터공학과
brightday@final.korea.ac.kr
†† 정 회 원 : 고려대학교 전자공학과 교수
eomds@korea.ac.kr
논문접수 : 2006년 3월 30일
심사완료 : 2006년 9월 2일

Wireless sensor networks consisting of hundreds to thousands of nodes are expected to be increasingly deployed in coming years, as they enable reliable monitoring and analysis of physical worlds. These networks are very different from traditional networks; they are composed of a large number of nodes that are densely deployed, produce very large amounts of data, and are limited in power, computational capacities, and memory. Such unique properties of sensor networks lead to a specific data management method [1] in which users query the sensor network and each node in the network responds with its sensing data. Therefore, sensor data management including querying becomes a challenging problem. Furthermore, due to wide popularization of the Internet and its facility in use, it is generally accepted that an unattended network can be efficiently managed and monitored over the Internet.

In order to achieve the web-based sensor data management, a connection between a sensor network and the Internet should be preceded. Therefore, a gateway placed between a wireless sensor network and the fixed Internet is necessary, which plays an important role of connecting two networks.

The design problem of this sensor gateway does not mean just a connection from a sensor network to the web. A sensor gateway should satisfy the following basic requirements; sensor data aggregation, flexible query and data management between the Internet and sensor network, and efficient access by the Internet.

In this paper, in order to more efficiently query and manage data in a sensor network, the design of a sensor gateway including web-based query server is presented and its implementation detail is illustrated. The presented web-based gateway is composed of largely two important parts: Internet part and sensor network part. The sensor network part plays an important role of handling a variety of sensor networks, including flat [2] or hierarchical [3] network architecture, by using internally layered architecture for efficiently querying and managing data in a sensor network. In addition, the Internet part provides a modular gateway function for favorable exchange between the sensor network and Internet.

The remainder of this paper is organized as follows. In the next section, several literatures relevant to our research are presented. And then, the design architecture of the presented web-based sensor gateway, including a query server designed with several layers to flexibly and efficiently manage the network, is presented. Finally, we conclude the paper by showing its implementation details.

2. Related Work

Query and data management is regarded as one of the most important functions in wireless sensor networks. Therefore, several literatures, such as [5, and 7-9], are proposed to transmit the query to the appropriate nodes. In addition, [4] and [6] present the approach to In-network Query processing through the sensor database. However, most of the traditional schemes do not address sensor data query and management through the Internet. Web-based network management gives the administrator the ability to configure and monitor the network over the Internet by means of a web browser [12-13]. Since a sensor gateway plays a role connecting between a wireless sensor network and the Internet, the gateway should have the ability to serve as a web server [10-11]. In order to facilitate user's query and monitor by the Internet, the presented gateway includes a compact web server function, which enables to transfer user query to sensor network through the gateway and to configure and monitor the sensor network by the user's Internet web browser by using java applet and java script as well as HTTP [14]. In addition, the layered architecture of the presented gateway reduces dependency on a specific sensor network protocol. Thus, various types of routing protocols [2-3] for sensor networks can cooperate with the presented gateway. Employment of a layered architecture in designing the gateway was largely motivated from the WAP gateway for connection between wireless devices and the fixed Internet in [15].

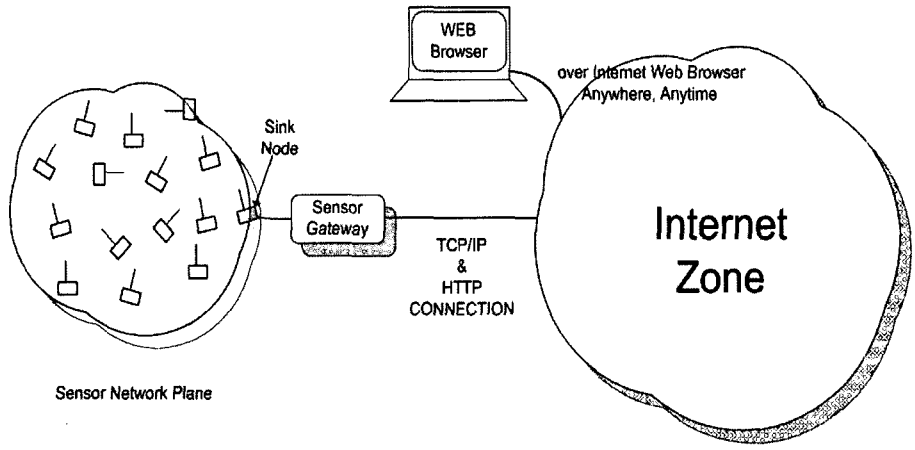

Fig. 1 Architecture of a wireless sensor network connected to the Internet

3. A Proposed Sensor Gateway Architecture

Fig. 1 illustrates the architecture of a sensor network connected with the Internet. A sensor gateway is placed on the route to connect a sensor network to the fixed Internet, and it must provide a flexible delivery of query and response between two networks. We have developed a sensor gateway to achieve this goal. As the gateway executes a web server internally, users can get easily web pages for management by trying to access to the gateway using its IP address or domain name. The web query received from user's web browser is sent to the gateway.

In order to process and transmit sensor query, three functional modules are newly defined in the presented gateway. The DAP (Data Analysis Process) is a module to analyze the data received from the user's web browser, and the SQML (Sensor Query Management Layer) plays a role of managing and translation of sensor query. In addition, by SDAL (Sensor Data Aggregation Layer), all queries are sent to the sensor network through the sink node connected directly to the gateway and all responses about the user query from the sensor network are aggregated. How the user's web data is processed with the gateway is described in Subsection 3.A, and the functions of the web server in the gateway is described in Subsection 3.B.

3.1 Proposed Gateway Architecture

The design of gateway to connect between a sensor network and the Internet (web) is of very importance. The design problem does not mean just the connection from a sensor network to the web. A sensor gateway should satisfy the following basic requirements; sensing data aggregation, flexible query management between the web and a sensor network, and efficient access by the Internet.

As shown in Fig. 2, we have defined the new layered architecture in the sensor gateway, to efficiently connect between a sensor network and the Internet, and process and transmit sensor query. There are largely two parts in the gateway; the one is an internet access part including TCP/IP stack and a small web server, and the other is a sensor network access part having SDAL, SQML and DAP. Accordingly, the data received from the user's web browser through internet access part is handed to the sensor network management part. After processing in each layer of the part, sensor network management part, the processed data is delivered to the sink node physically connected to the gateway through RS-232C, USB, or other interfaces. Then the sink node disseminates the data to the sensor network.

Since the presented gateway provides common interface among modular components, the default web server application can be replaced by a new gateway application by a user.

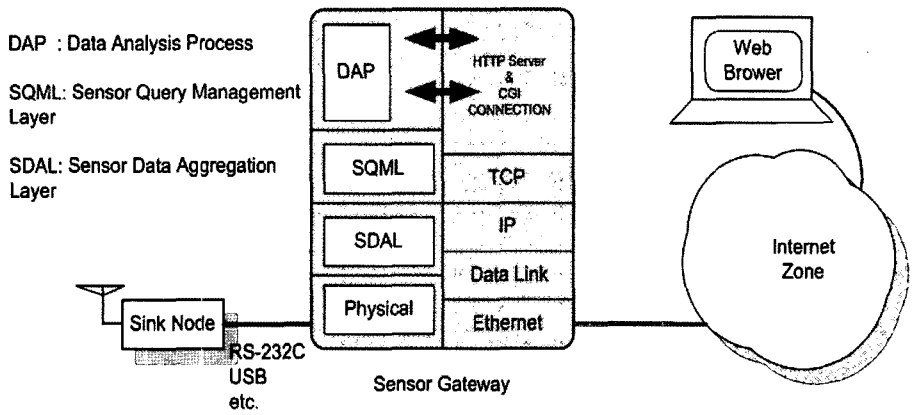

Fig. 2 Internal architecture of our sensor gateway

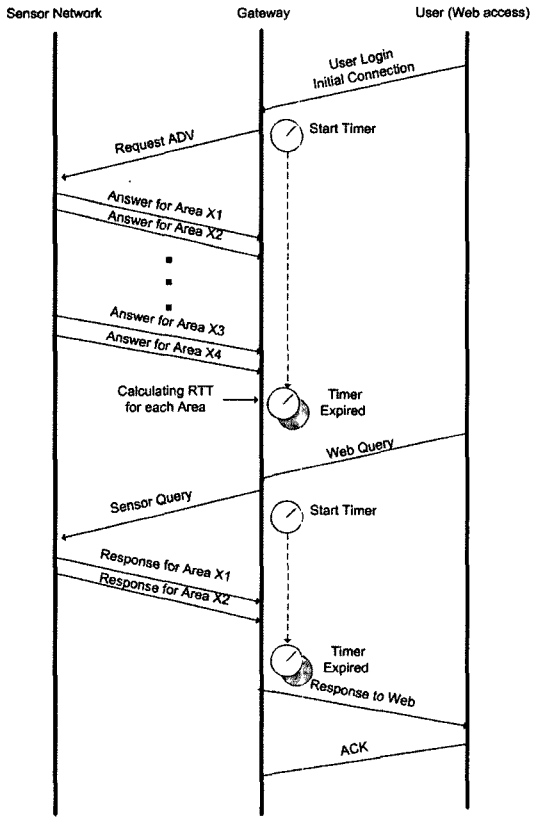

Fig. 3 The layered architecture of the sensor network management part of the sensor gateway

Data Analysis Process (DAP). The DAP plays a role of authenticating a user, anlyzing the query received from the web browser, making the web response from the sensor reply, and storing the query and results. As shown in Fig. 3, DAP consists of five key components; a data extracting module, a web query queue, a response table, a web reply module, and a DB module. Data delivered from the web server is converted into the web query through the data extracting module, stored into the web query queue, and then the web query is handed to the lower layer, SQML. The web query and the data aggregated at SDAL layer responding to the web query are stored into the DB, and they are used for making the web reply which is sent to the user's web browser.

Sensor Query Management Layer (SQML). The major role of SQML is to transform a web query into a proper sensor query or conversely transform a sensor query into a web query. This SQML layer has a flexible internal structure so that it can easily cope with a variety of sensor queries by modifying just part of the code for the query transform module. We use the modified SQTL [7] as a sensor query model. Table. 1 shows the sensor query structure (modified SQTL) used in the sensor

Table 1 Sensor query structure (modified SQTL)

Name	Meaning
Sensor query id	Use of identifying response about the sensor query
Question	Discerning which area…? or what is the value….?
Area	All area or selective area
Sensor Type	What kinds of sensor should answer
Contents	Higher than, lower than, max, min, or average
Interval (optional)	Notice with YY interval
Duration (optional)	During XX time

gateway. In addition, the Sensor Type field presents a type of interesting sensors in the current user's query. The sensor type should be resolved in gateway application according to deployed sensor types. All queries have a unique id and all the replies for the identical query also have a unique sensor query id. Our gateway can process several types of queries such as historical queries, snapshot queries, and long-running queries, through the interval and duration fields of the data format.

Sensor Data Aggregation Layer (SDAL). The sensor query transformed properly in SQML should be disseminated to all over the sensor network and the replies from the correspondent nodes with respect to the query should be aggregated. SDAL is responsible for the above role. In order to aggregate all the data in the sensor network, SDAL must know the information about sensors in the areas. At the beginning, the gateway does not have the information of the sensor network, and it thus floods the advertisement message to get the information of the network to all the areas of the sensor network, and then it sets the ADV (Advertisement) waiting timer.

The gateway has two kinds of timer; the ADV timer that is used for receiving and the replies from all or partial information about sensor nodes in the areas and the Area timer that is used for receiving the reply from individual area. On receiving the advertisement message from the gateway, each node reply with their basic information (kind of sensors, its area), their data are aggregated at each immediate node, and finally, the data are delivered to SDAL through the sink node.

On receiving the reply of any area, the gateway measures the round trip time for the area using the time stamp, and then it performs exponential averaging of this measured round trip time RTT (MRTT) by using the well known equation as follows [9];

$$SRTT(n+1) = \alpha \times SRTT(n) + (1-\alpha)MRTT(n) \quad (1)$$

Where $MRTT(n)$ is the round trip time measured for the nth transmitted query, and $SRTT(n)$ is the average round trip time for the first n transmissions. Then, by simply multiplying $SRTT(n+1)$ by a constant value, the timeout value of each Area timer is determined. Whenever the gateway sends a query to a specific area head, it set an Area timer. If the gateway does not receive the reply from a node representing the area until the Area timer is expired, it sends the query again.

On the other hand, the gateway waits for the replies from all or parts of the areas depending on the query until the ADV timer is expired. The timeout value of the ADV timer is updated by summing up the timeout values of the Area timers involved in the query so that the gateway can adaptively manage timeout value of the ADV timer. But, the initial timeout value of the ADV timer for sending the advertisement message should be determined without any information. If it is too small, the gateway cannot get the information about all of the areas, if the timeout value is too large, it increases the waiting time delay taken to gather all the necessary information from the sensor network. Fig. 4 shows the message flow sequence between the user and the sensor network.

3.2 Web Access to Web Gateway

In order for users using the Internet web browser to efficiently manage and query the sensor network, the sensor gateway should serve as a web server. To provide the functions of the web server described in [10-11], we port the HTTP 1.1 protocol [14] in our gateway. Also, not just the HTTP protocol but also the java script and the java applet are used together with our web server in the gateway for the dynamic and efficient data exchange.

Fig. 5 shows the data exchange between the web server in the gateway and the user's web browser. The Internet access part in the gateway consists of three components; a web server, a back socket, and a CGI. The web server exists for receiving user's page requests and for transferring the web pages to the user's web browser. The back socket is used to dynamically transfer the query and reply data between the gateway and the user's web browser. Since web replies are shown only as static page, the CGI is used to generate the dynamic web page for the data or results.

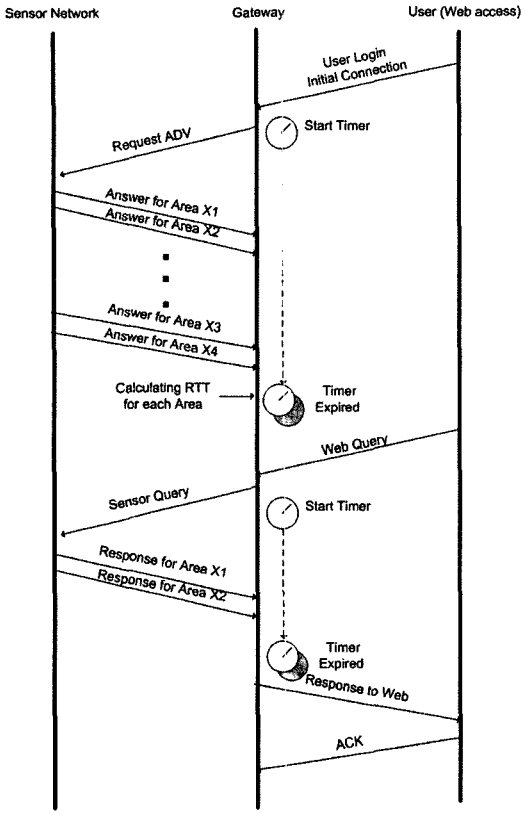

Fig. 4 Message flow sequence between the user and the sensor network

A user can connect to the gateway by just typing in the IP address or domain name of the gateway with the user's web browser. Then the web server in the gateway responds to the user's request by sending the login page for authentication to the user's web browser. After the authentication of the user, the web server sends a sensor management page to the user. Also, in this phase, the initial advertisement process is performed to gather the information of the sensor network, as explained in Subsection 3.A. The user can make a web query easily by just filling in the query form in the sensor management page, and send it to the gateway by pushing the sending button.

To efficiently process the user's query, some control data other than web pages should be exchanged between the user's web browser and the gateway. It requires a dynamic delivery and no intervention of the user. That is, the dynamic delivery of the control data should be invisible to the user. We embed a java applet code in the user's management page to achieve this requirement. The java applet code is executed on the user's computer. The applet opens a new socket with the port number 8888 different from the web port number 80, and the reading or writing process for exchanging control data performs its job through this socket. A java script code is also embedded in the web page to link the java applet to the HTML.

On receiving the web query from the user's web browser, the gateway processes it in its sensor network management part. After the web query is processed through the interaction with the sensor network, the result of the processing is sent to the

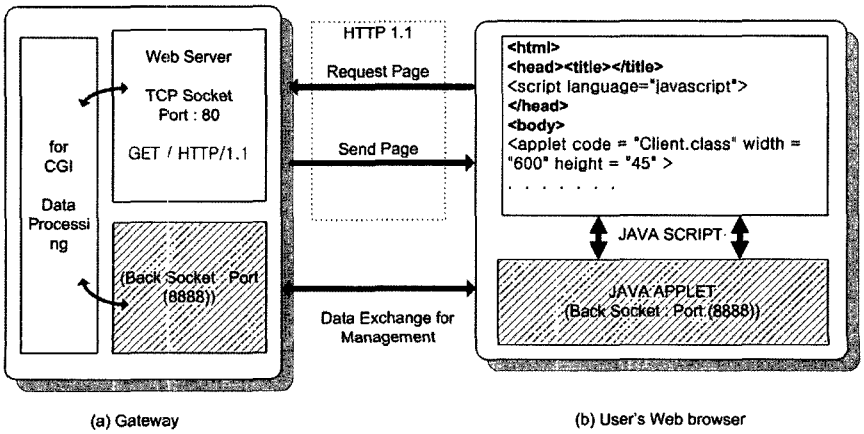

Fig. 5 Data exchange between the web server in the gateway and the user's web browser

user by using the back socket. The applet receives the result and hands over it to the java script. Then the java script performs an appropriate action. Eventually, the user can show the result of the web query with the web browser.

4. Implementation

Our sensor gateway has been programmed with C and C++ of the Microsoft Visual Studio. We have also used java script and java applet for the web pages. Fig. 6 shows the test-bed configuration consisting of three major parts; a sensor network emulator, a sensor gateway, and a user's web browser for web based management. The parts are all executed on different PCs, and the sensor network emulator and the sensor gateway are connected over the air with the wireless sensor nodes designed by us.

Since it is very difficult to implement a sensor network having a lot of sensor nodes and we focus on the operation of the sensor gateway, we have used the developed emulator instead of a real sensor network. In particular, for the useful data aggregation, a hierarchical sensor network is assumed. So, in emulator, a sensor network has a regional clustering based architecture and it generates random sensing data that is necessary for testing our gateway. However, we have used two real sensor nodes with a wireless interface in the test-bed; one corresponds to the sink node that is directly connected to the sensor gateway and the other corresponds to the nearest sensor node to the sink node. They communicate each other through the wireless medium. Since the codes of these nodes are executed in real environment, we can obtain some useful information such as the processing time and the transmission delay. Such information has been used in the emulator. Also, we can confirm the functionality of our sensor node by implementing the real sensor node. On the other hand, the sensor gateway is connected to the user's web browser through the Ethernet corresponding to the Internet.

4.1 Sensor Network Emulator

We have made the emulator to emulate a large scale sensor network. The program for the emulator is made up of C++. The emulator has three kinds of objects; an object for the area head, an object for the cluster head, and an object for the sensor node. The objects have abstracted an active large scale sensor network.

Initially, the emulator generates the cluster objects and the area objects, and the number of sensor nodes they have is selected randomly. The basic information about the sensor node object is also selected randomly. The emulator is connected with serial interface, RS-232C, to the real sensor node to communicate with the sink node through the wireless medium. That is, the sensor node connected to the emulator receives the query disseminated from the sink node connected to the gateway, and then it sends the query to the emulator program through the serial interface. The query is handed to the internal cluster head objects, and then the objects translate the query and generate the result for the query. The results from

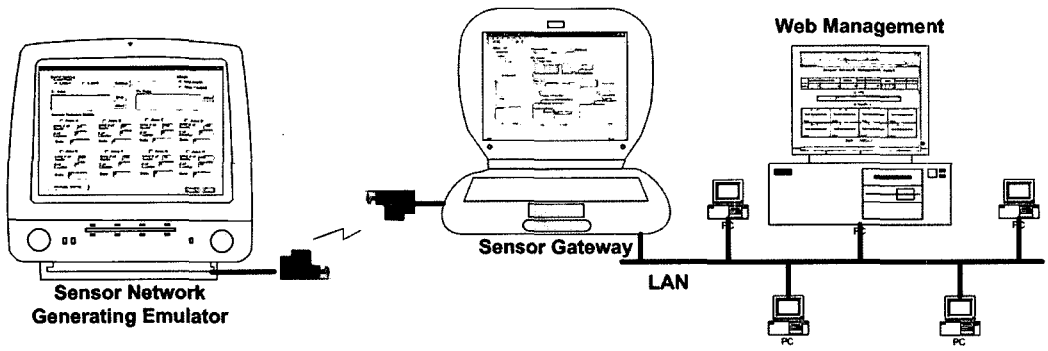

Fig. 6 Test-bed configuration for our sensor gateway

the cluster heads are aggregated in the correspondent area head objects. Each area head object generates a new result by processing the results from its cluster heads, and then it sends the new result after waiting for its own propagation time. Accordingly, the replies from the areas are arrived at the gateway at different times.

4.2 Querying and Managing Sensor Network through Web Browser

In this subsection, implementation details of the

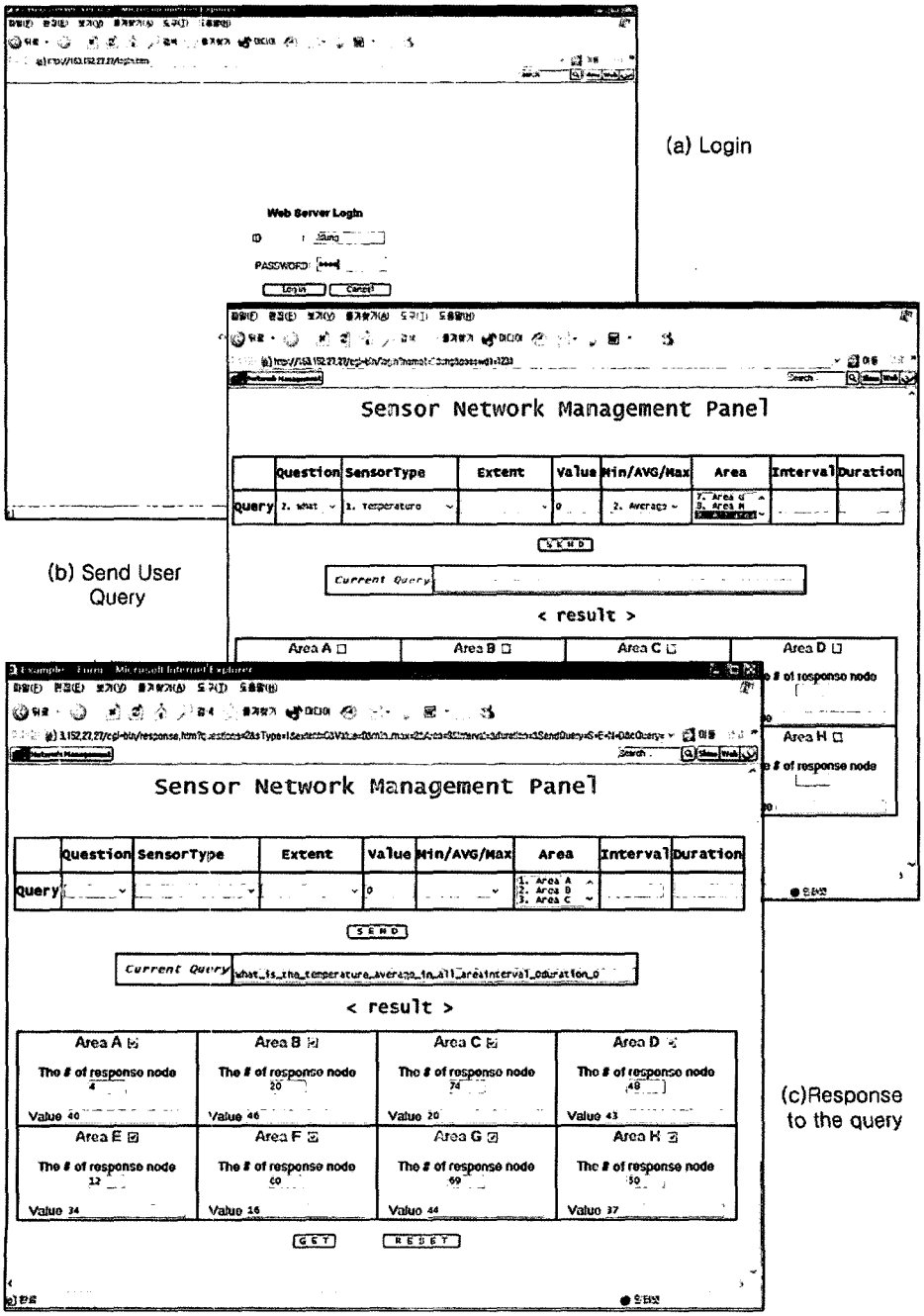

Fig. 7 The results showing each phase for processing the querying request

sensor gateway, querying a sensor network and managing data in a sensor network with respect to the user's web query, are demonstrated. Fig. 7 shows each phase for processing the querying request. First, after connecting to the gateway, the user receives the login page shown in Fig. 7(a), for authentication. If the login data of the user is correct, the gateway starts to send the advertisement message. On receiving all the replies for the advertisement message, the sensor gateway sends the sensor management page shown in Fig. 7(b) to the user's web browser.

The page has three fields consisting of a number of forms for input and output. First field is to make user's query, second field is to show the query sentence to identify the reply received currently from the gateway, and the last field shows the result for the user's query from the sensor network. Figure 7(b) shows how the user's query is generated and sent.

The sensor gateway analyzes the user's web query in the DAP, and after transforming the user's web query into the proper sensor query, it sends the query to the sink node.

On receiving the query, the sensor network emulator generates the reply of each area, and then it sends the reply with different propagation time. The gateway waits for the reply. If the reply does not arrive until the ADV timer is expired, the gateway requests the retransmission of the reply to the area. If all the expected replies arrive at the gateway, the gateway makes the result to be sent to the user's web browser, and then it sends the result of the query through the back socket.

At the user's web browser, the java applet code being executed on user's PC passes the result to the java script. Eventually, the user can show the result for the query of current sensor network shown in Fig. 7(c).

5. Conclusion

Due to wide popularization of the Internet and its facility in use, it is generally accepted that an unattended network can be efficiently managed and monitored over the Internet. In order to achieve the web-based query and sensor data management for wireless sensor networks, the sensor gateway connecting a sensor network to the fixed Internet is inevitable. In this paper, the design of a sensor gateway including web-based query server is presented and its implementation detail is illustrated. For managing sensor networks efficiently, the three functional modules for the sensor gateway are newly defined in the gateway: DAP (data analysis process), SQML (sensor query management layer) and SDAL (sensor data aggregation layer). The layered architecture of the presented gateway reduces dependency on a specific sensor network protocol. Thus, various types of routing protocols for sensor networks can cooperate with the presented gateway.

References

[1] I. F. Akylidiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks," IEEE Communication Magazine, 2002.

[2] C. Intanagonwiwat, R. Govindan, and D. Estrin "Directed diffusion: A scalable and robust communication paradigm for sensor networks," In Proceedings of 6th ACM/IEEE Mobicom Conference, 2000.

[3] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocols for Wireless Microsensor Networks," In Proc. of Hawaiian It'l Conf. on Systems Science, January 2000.P. Bonnet, J. Gehrke, and P. Seshadri, "Towards sensor database systems," In 2nd International Conference on Mobile Data Management, Hong Kong, January 2001.

[4] P. Bonnet, J. Gehrke, and P. Seshadri, "Querying the Physical World," Personal Communications, Volume: 7 Issue: 5, Oct. 2000.

[5] Y. Yao and J. E. Gehrke, "The Cougar Approach to InNetwork Query Processing in Sensor Networks," Sigmod Record, Volume 31, Number 3, September 2002.

[6] C. Jaikaeo, C. Srisathapornphat, C. Shen, "Querying and Tasking in Sensor Networks," SPIE's 14th Annual International Symposium on Aerospace/Defense Sensing, Simulation, and Control, 2000.

[7] C. Jaikaeo, C. Srisathapornphat, C. Shen, "Diagnosis of Sensor Networks," IEEE International Conference on Communications (ICC 2001), 2001.

[8] C. Shen, C. Srisathapornphat, C. Jaikaeo, "Sensor information networking architecture and applications," Personal Communications, IEEE, Volume: 8 Issue: 4 , Aug. 2001.

[9] B. McCombie, "Embedded Web servers now and in the future," Real-Time Magazine, no.1, Mar. 1998.
[10] I. Agranat, "Embedded Web Servers in Network Devices," Communication Systems Design, Mar. 1998.
[11] C. W. Tsai and R. S. Chang, "SNMP through WWW," International Journal of Network Management, Vol 8. 1998.
[12] M. J. Choi, et. al., "An Efficient and Lightweight Embedded Web Server for Web-based Network Element Management," IEEE/IFIP Network Operations and Management Symposium, Hawaii, April 2000.
[13] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, and B. Lee, T. RFC 2068, hypertext transfer protocol -http/1.1, 1997.
[14] W. Forum, "WAP Forum," 2000, http://www.wapforum.org

황 광 일

2002년 홍익대학교 전자전기전파공학부 졸업. 2004년 고려대학교 전자컴퓨터공학과 석사. 현재 고려대학교 전자컴퓨터공학과 박사과정. 관심분야는 AD HOC networks, RFID systems, Sensor networks, Ubiquitous networks

엄 두 섭

1987년 고려대학교 전자공학과 졸업. 1989년 고려대학교 전자공학과 석사. 1999년 일본 오사카대학 정보컴퓨터과학과 박사 1996년~1999년 ETRI 선임연구원. 1999년~2000년 원광대학교 전임강사. 현재 고려대학교 전자공학과 부교수. 관심분야는 AD HOC networks, RFID systems, Sensor networks, Ubiquitous networks, Internet QoS