

ZigBee 기반 센서 네트워크로의 유비쿼터스 접근을 위한 게이트웨이 아키텍처

Gateway Architecture for Ubiquitous Access to ZigBee-Based Sensor Networks

허웅*, 평치유*, 유강수**, 최재호*

전북대학교 전자공학부 영상정보신기술연구센터*, 전주대학교 교양학부**

Ung Heo(heoprinc@jbnu.ac.kr)*, Qiu Peng(qiupeng@jbnu.ac.kr)*,

Kang Soo You(gsyoun@jj.ac.kr)**, Jaeho Choi(wave@jbnu.ac.kr)*

요약

본 논문에서는 ZigBee/IEEE 802.15.4 기반 무선 센서 네트워크와 TCP/IP 기반 네트워크를 서로 연결할 수 있는 웹-센서 게이트웨이의 아키텍처를 제안한다. 제안한 게이트웨이의 주요 역할은 인터넷을 통하여 센서 노드에게 질의 또는 명령을 내리고 센서 데이터를 취득하도록 도와주는 기능이다. 다만, 인터넷과 센서 네트워크의 통신 프로토콜은 서로 불통함으로 양방향 프로토콜 변환이 필요하다. 본 논문에서는 이 변환 기능을 두 통신 프로토콜 APP 계층 위에 위치하도록 설계하여 인터넷 어플리케이션과 ZigBee 어플리케이션이 양 방향으로 변환 될 수 있도록 하였다. CGI 프로그램을 사용하여 변환 계층을 구현하였으며 인터넷의 사용자가 언제 어디서나 웹 페이지를 열어서 검색 요청이나 명령 등을 센서 노드에 전달할 수 할 수 있는 방식을 채택하였다. 이중 네트워크 연결을 위하여 TCP/IP 프로토콜을 센서 노드에 오버레이 하는 기법도 가능하나 구현의 복잡성, 에너지 효율, 통신 오버헤드 측면에서 제한적이다. 부족한 자원을 가진 센서 노드 환경에서는 본 논문에서 제안한 변환 기반 웹-센서 게이트웨이 방식이 이중 통신 프로토콜 연결 문제를 해결하는 비교적 효과적이며 우수한 방법임을 확인하였다.

■ **중심어** : | 웹-센서 게이트웨이 | TCP/IP 기반 네트워크 | 프로토콜 변환 | ZigBee/IEEE 802.15.4 |

Abstract

This paper investigates protocol architecture of a Web-ZigBee gateway for interconnecting TCP/IP-based networks and ZigBee/IEEE802.15.4-based wireless sensor networks. The Web-ZigBee gateway delivers data between the TCP/IP network and the ZigBee network. Since those two networks have different communication protocols, a protocol translation mechanism is needed. Herein, we propose a method to deliver query messages from the Internet to the sensor network and receive data from sensors. The protocol translation is performed in the translation layer that is placed above the two application layers, i.e., the Internet application layer and ZigBee application layer. Among various interfaces, we use CGI programming to take care of translation functions efficiently. The CGI manages query information from a client on the Internet and data from the ZigBee sensor network. Whereas the TCP/IP enabled sensor network overlays two heterogeneous communication protocols, overlaying layers increase the complexity and cost of implementing the sensor network. On the contrary, the sensors in our gateway-based system are not only light (because each communication protocol works independently without overlaying), but also efficient because the translation layer mostly alleviates header overloading.

■ **keyword** : | Web-Sensor Gateway | TCP/IP | Protocol Translation | ZigBee/IEEE 802.15.4. |

I. INTRODUCTION

The ZigBee protocol has all the benefits of the IEEE 802.15.4 protocol with added networking functionality [1]. The low cost allows the technology to be widely deployed in wireless control and monitoring applications. In addition, the low power allows a longer life with smaller batteries; hence, the ZigBee sensor network can provide greater reliability and range [2].

Most ZigBee sensor network applications aim at monitoring or detecting phenomena. Examples include office building environment control, wildlife habitat monitoring, and forest fire detection [3,4]. Such applications must provide a way for a monitoring entity to gain access to the data produced by the sensor network. By connecting the ZigBee sensor network to an existing network infrastructure, such as the global Internet, remote access to the sensor network can be achieved. Given that the TCP/IP protocol suite has become the *de facto* networking standard not only for the global Internet, but also for local networks, this suite is of particular interest when looking at methods for interconnecting sensor networks and TCP/IP networks.

Recently, more literature has been produced on the internet-ZigBee gateway. The basic functionalities of the gateway are query distribution, data aggregation, and message conversion. From the literature, it is widely accepted that web-based management should be used for managing and querying the ZigBee sensor network because the internet provides flexibility and convenient access.

Our research is also motivated by solving the data communication problem that occurs in ZigBee sensor networks and the Internet. In this paper, we propose a novel Web-ZigBee gateway architecture that translates the protocol between the two different

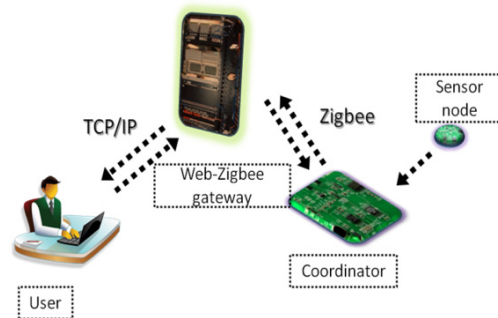


Fig. 1. Remote sensing connection structure

networks. That is, when the sensor node transmits data to the Internet, the Web-ZigBee gateway provides a protocol conversion to make both the ZigBee sensor network and the Internet understand each other, as shown in [Fig. 1].

The remainder of this paper is organized, as follows. Section II presents some related works. Section III presents our Web-ZigBee gateway architecture and ZigBee sensor network management. Section IV shows computer simulations. Section V shows the performance evaluation of the proposed scheme and Section VI offers our conclusion.

II. RELATED WORKS

There are several papers in the literature that are related to our work. In this section, three approaches for implementing web-sensor gateways are discussed.

1. IP-based approach

In this IP-based sensor network, each sensor is equipped with a global IP address for realizing the data communication between sensor nodes and clients on the Internet. In particular, because it is expected that the wireless sensor network (WSN) consists of

thousands to millions of tiny sensor nodes, all IP-sensor networks are infeasible without moving up to IPv6. With limited computational and communication capabilities, it is also impractical to overlay TCP/IP layers over the communication protocol layers of the sensor [5]. There is a similar method that uses the μ IP TCP/IP stack so that users can access the sensor node while the sensor nodes in the WSN communicate with each other over their own protocol [6]. In contrast to the all-IP-based sensor network, in this μ IP-based network, only some sensors are equipped with an IP stack. The concerns are implementation cost and sensor network life. This approach will be compared with our method in Section V.

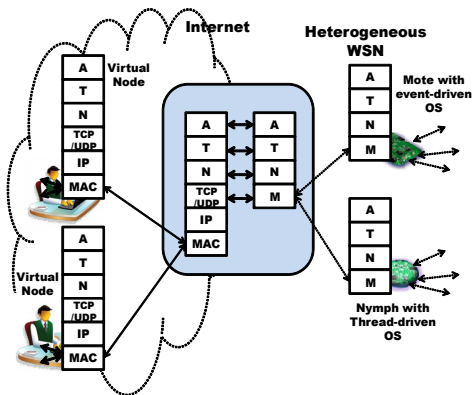


Fig. 2 The sensor network stack is preserved by the overlay gateway

2. IP host as a virtual sensor node approach

The essential features of the proposed overlay network structure for this approach are illustrated in [Fig. 2] The major components include the virtual nodes running over the Internet and the overlay gateway. The sensor overlay permits a rich and versatile environment for the interconnection of virtual nodes, which collectively form a virtual sensor network.

[Fig. 2] also shows the specific operation of an overlay gateway. Packets that arrive at the overlay gateway from the sensor network have MAC (M), network (N), transport (T), and application (A) layer headers specific to the sensor network. The overlay gateway then propagates the packet's network layer, application payload, and above headers (N,T,A) to the Internet, encapsulating them within TCP/UDP/IP packets. The overlay gateway effectively translates the sensor network's protocol stack – formerly layer 3 and above – to the application layer and above on the Internet. Virtual nodes, therefore, form a virtual sensor network that operates as an application-level overlay of the Internet. The overlay gateway itself is also a virtual node that participates in sensor network routing decisions [7].

III. The Proposed Web-ZigBee Gateway

In this section, we describe the architecture of the proposed Web-ZigBee gateway and the message exchange processes.

1. Our Web-ZigBee gateway architecture

The main task of the Web-ZigBee gateway is to provide an interface between the ZigBee sensor network and the client. The gateway consists of three parts: a TCP/IP protocol stack for communications between the client on the Internet and the Web server in the gateway; ZigBee protocol stacks for communications between the ZigBee AP in the gateway and the ZigBee sensor node in the sensor network; and the gateway translation layer.

The gateway translation layer is used for mapping and translating messages from the client on the Internet to a sensor node in the ZigBee sensor network. As shown in [Fig. 3], it is another layer

connecting the web server application layer and the ZigBee AP application layer. Clients anywhere on the internet can remotely access and control a sensor node using a dynamic web page.

The gateway translation layer is used for mapping and translating messages from the client on the Internet to a sensor node in the ZigBee sensor network. As shown in [Fig. 3], it is another layer connecting the web server application layer and the ZigBee AP application layer. Clients anywhere on the internet can remotely access and control a sensor node using a dynamic web page.

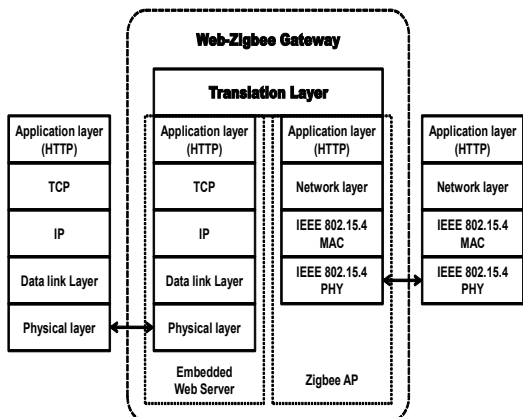


Fig. 3 Web-ZigBee gateway protocol architecture

2. From a client to the gateway

In the ZigBee sensor network, all of the sensor nodes send their information to a coordinator periodically. We assume there is only one sensor node in the sensor network, and a client requests the sensor data from this sensor node using the Web-ZigBee gateway. As shown in [Fig. 1], there are four main components in our system: the client, Web-ZigBee gateway, the ZigBee coordinator, and the ZigBee sensor node. More specifically, [Fig. 4] shows exchanges of control and data among the Internet client, Web-ZigBee gateway, and ZigBee

coordinator.

The message exchange process begins by initialization of the web server embedded in the Web-ZigBee gateway. The initialization process consists of creating a socket, calling to *bind ()* and *listen ()*, and announcing the server's willingness to *accept ()* incoming calls. For the client, a socket is also created and initialized. After that, the client attempts to establish a TCP connection to the server using *connect ()*, and it sends a request to the web server after three-way handshaking. The requested information includes the sensor ID and the type of data that the client wants.

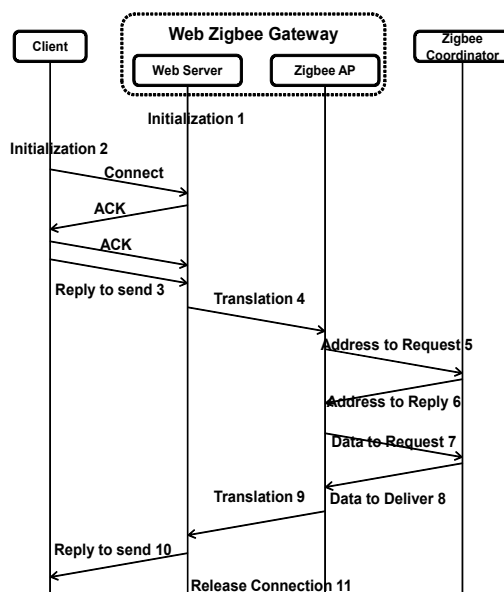


Fig. 4 Control and data exchange among the internet client, gateway, and ZigBee coordinator

3. In the Web-ZigBee gateway

Once the request from the client arrives at the web server in the gateway, the request has to be translated into a format that the ZigBee coordinator in the sensor network can understand. The packet

translation starts from the application layer of the web server part and ends with the application layer of the ZigBee AP part.

The client sends an HTTP request to the web server, and then the web server delivers it to the CGI program. The CGI parses the request so that it can extract the useful parameters, such as the sensor ID and command.

| | | |
|-------------------|------------------------|------------|
| NWKAddrOfInterest | RequestType | StartIndex |
| NWK address | Single device response | Ignore |

Fig. 5 Format of the IEEE_addr_req_Command Format

| |
|----------------------|
| NWK_Addr_of_interest |
| 16-bit NWK_address |

Fig. 6 Format of the Active_EP_req Command

3.1 Generating ZigBee AP frames

Once the web server receives the client request, the translation layer delivers client request information to the ZigBee AP to generate a payload that is then forwarded to the ZigBee coordinator.

The above C program for the web server motivates ZigBee AP to create data to search the ZigBee coordinator MAC address, ZigBee coordinator endpoint address, and request sensor device payload value. The three major function calls invoked in the application layer of the ZigBee AP are given below.

```
ZDP_IEEEAddrReq ()
/* search sensor node MAC address
ZDP_ActiveEPIFReq ()
/* search ZigBee coordinator endpoint
APSME-GET.request ()
/* obtain ZigBee coordinator payload
```

ZDP_IEEEAddrReq () is used to obtain the 64-bit

MAC address of the sensor node based on the known 16-bit Network address.

When *ZDP_IEEEAddrReq ()* is called, the *IEEE_addr_req* command is generated and sent to the sensor node. The format of this command is shown in [Fig. 5].

ZDP_ActiveEPIFReq () is used to obtain the list of endpoints in the sensor node device. Once this function is called, the *Active_EP_req* command is sent to the destination sensor node. The format of this command is shown in [Fig. 6].

The application layer of the ZigBee AP consists of three parts: the ZigBee Device Object (ZDO), Application Support Sublayer (APS), and Application Framework (AF), as shown in [Fig. 7]. Each ZDP command becomes a payload in the APS frame. Once the ZigBee AP gets a reply from the ZigBee coordinator, it calls the function *APSME-GET.request ()* to obtain the sensor data collected by the ZigBee coordinator. The sensor data is delivered by the APS frame.

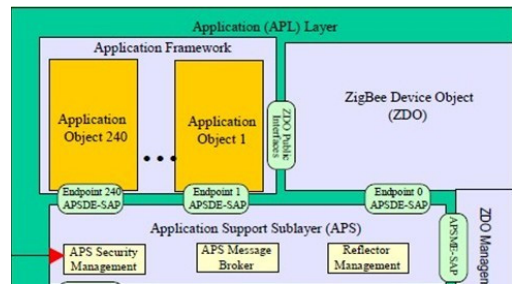


Fig. 7 ZigBee AP application layer

3.2 Data from the ZigBee AP to the web server

The ZigBee AP receives data from the ZigBee coordinator after sending the commands and delivers data translated by the CGI program in the translation layer to the web server. The translated data is

provided to the client through the web browser.

4. Between the gateway and ZigBee coordinator

In this section, the communication protocol between the ZigBee AP and the ZigBee coordinator is explained.

4.1 From the ZigBee AP to ZigBee coordinator

The protocol layers for between the ZigBee AP and the ZigBee coordinator are shown in [Fig. 8]. In the ZigBee sensor network, we assume that the ZigBee AP and ZigBee coordinator have the same 16-bit network address. These two devices can send and receive data from each other if they have the same cluster ID. In order to trace the sensor node and its attribution value, which the client requests, the ZigBee AP sends *IEEE_addr_req* and *Active_EP_req* commands to the ZigBee coordinator using the APS frames.

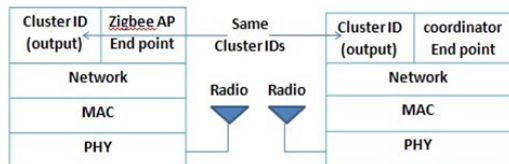


Fig. 8 Protocol layers for ZigBee AP and ZigBee coordinator

The cluster ID 0x001 means that this ASP frame is the *IEEE_addr_req* command, while 0x0005 means that it is the *Active_EP_req* command. The *IEEE_addr_req* command requests the MAC address of the coordinator, and the *Active_EP_req* command requests the endpoint address of the coordinator.

Upon obtaining the MAC address from *ZDP_IEEEAddrRsp()* (sent by the ZigBee coordinator as the response to *IEEE_addr_req*) and the endpoint address from *ZDP_ActiveEPIFRsp()*

(also sent by the coordinator as the response to *Active_EP_req*), the ZigBee AP sends *APSME-GET.request* to the ZigBee coordinator in order to get the sensor data from the sensor.

The *APSME-GET.request* is a command managed by the APS Management Service in the application layer of the ZigBee coordinator and is shown in [Fig. 9]. When the *APSME-GET.request* command is received at the ZigBee coordinator, the coordinator retrieves the requested sensor data from the memory space called the application support layer information base (AIB) and prepares the data to be sent to the ZigBee AP.

4.2 From the ZigBee coordinator to the ZigBee AP

Assuming that the sensor provides its data periodically to the coordinator, when the *APSME-GET.request* command arrives from the ZigBee AP, the ZigBee coordinator sends the requested data back to the AP using the data frame shown in [Fig. 10]. The figure shows the typical ZigBee data frame composed after being encapsulated by the header of each protocol layer.



Fig. 9 Frame format of *APSME-GET.request*

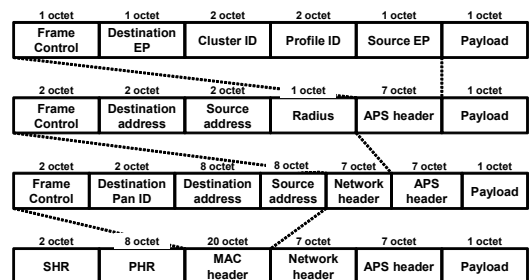


Fig. 10 Data frame composed by the ZigBee coordinator

Fig. 11 Data frame composed by the ZigBee coordinator

5. From the gateway to the client

The data frame sent by the ZigBee coordinator gets decapsulated at the ZigBee AP. The extracted sensor data is transferred to the translation layer so that it can be reformatted into the HTML format for the web server. The CGI program dynamically replies to the client on the Internet.

IV. Computer Simulations

In this section, we present the ZigBee remote control interface supporting a client's query, and display the sensor data in the client interface. As shown in the typical remote control interface in [Fig. 11], the client can choose the ZigBee device and check its attributes.

First, the client opens the remote control interface, chooses a ZigBee sensor, and then checks one of the attributes, such as temperature, voltage, or humidity. Then, the web-sensor gateway replies with the data as a response to the client query.

The following section of the paper provides the procedures exchanged among the client on the Internet, the web-sensor gateway, and the ZigBee coordinator. The implementation has been done in C/C++, HTML, and CGI programming.

1. From the client to the web server

The client submits a HTML Form like the one below:

```
<FORM ACTION="/cgi-bin/select_device.cgi" >
< P> please enter ZigBee device name then press
enter
<INPUT NAME="device A" SIZE="10">
<INPUT                               TYPE="SUBMIT"
VALUE="ENTER">
</FORM>
```

The client inputs the sensor name and then enters. The web server calls the CGI program and begins to convert the client request, as follows:

```
/* it's a MIME, come next client will send ASCII
printf ( "Contenttype: text/plain\n\n ");
n=0;
/* check whether environment variable exists
/* using POST method to send data from client to
server
if (getenv( "CONTENT-LENGTH "))
/* convert environment variable value type to int
n=atoi(getenv( "CONTENT-LENGTH "));
```

The client request is converted to an integer and stored in *n*.

2. From the web server to the ZigBee AP

Once the web server receives the client request, the request is translated by the CGI program so that the ZigBee AP can generate a command frame and send it to the ZigBee coordinator. The first function called by the ZigBee AP is as follows:

```
ZDP_IEEEAddrReq ()
```

This function requests the MAC address of the sensor node to the ZigBee coordinator. When this function is called, a command is generated and sent to the coordinator using the APS frame. The command is as follows:

```
IEEE_addr_req
```

Upon receiving this command, the ZigBee AP responds with the MAC address of the designated sensor node using the command call and actual command, as follows:

```
ZDP_IEEEAddrRsp ()
IEEE_addr_rsp
```

In the second step, the ZigBee AP generates the command call and actual command in order to obtain the endpoint address of the sensor node, as follows:

```
ZDP_ActiveEPIFReq ()
Active_EP_req
```

Similarly, the coordinator responds with the following command calls and actual command for delivering the endpoint address of the sensor node, as follows:

```
ZDP_ActiveEPIFRsp ()
Active_EP_rsp
```

After acquiring the MAC address and endpoint address of the sensor node, the ZigBee AP can request the sensor data using the command call, as follows:

```
APSME-GET.request ()
```

When the coordinator receives the data request from the ZigBee AP, it responds with its data, as follows:

```
/* Create and write file in text mode: */
if( stream = fopen( "data.txt", "w+" )) != NULL
) {
for ( i = 0; i < 25; i++ )
list[i] = (set)('device_data.asdu' - i);
/* Write 25 octes to stream */
numwritten = fwrite( list, sizeof( set ), 25, stream
);
printf( "Wrote %d items\n", numwritten ); }
else
printf( "Problem opening the file\n" );
```

The data.txt is used to store the sensor data, and it supports data updates using the sensor data *device_data.asdu*.

The sensor data in the *asdu* format is saved in the data.txt file and translated by the CGI program to be shown in the web browser, as follows:

```
/* Attempt to read DATAFILE*/
FILE *f = fopen(DATAFILE,"r");
int ch;
if(f==NULL){
printf( "%s%c%c",
"Content-Type:text/html;charset",13,10);
printf("<TITLE>Error</TITLE>");
printf("<P><EM>Error.can'topenfile</EM>");}
else {
printf( "%s%c%c", "Content-Type:text/plain",13,10
);
while((ch=getc(f))!=EOF)
putchar(ch); }
fclose(f)
```


Finally, the client can check the ZigBee sensor data shown in the web browser, as follows:

```
<FORM ACTION = "/cgi-bin/viewdata.cgi" >
<P><INPUT TYPE = "SUBMIT" VALUE =
"Temperature">
</FORM>
```

V. Performance Evaluation

In this section, we compare our web-sensor gateway based system with the TCP/IP enabled sensor network in terms of routing, overhead and energy efficiency, influence of packet losses, and security.

1. Routing

The routing scheme based on IPs is host-centric and does not fit well with the sensor network paradigm, where the main interest is the data generated by the sensors and the individual sensor is of minor importance. Most of the proposed communication protocols for sensor networks use data-centric routing [11][12].

In our proposed gateway-based method, routing between the client and gateway is independent from routing between the gateway and the sensor network. Routing and addressing done for the Internet have no influence on addressing and routing done for the sensor network.

2. Overhead and energy efficiency

The size of the TCP/IP packet header is between 28 and 40 bytes. When sending a few bytes of sensor data in a datagram, the header constitutes nearly 90 % of each packet. Energy efficiency is of prime importance for the sensor networks, and since the

radio transmission is often the most energy consuming activity in a sensor node, a header overhead of 90 % is unacceptable [13].

In the proposed scheme, because the web-sensor gateway separates the Internet from the ZigBee sensor network, the header overhead can be decreased by 25 %, compared to the TCP/IP enabled sensor network. This decrease results from the compact addressing fields used in the IEEE 802.15.4.

3. Influence of packet losses

TCP was designed for wired networks in which bit-errors are uncommon, but TCP is not appropriate for systems where packet drops are common. In the TCP/IP enabled network, TCP misinterprets the packet loss as congestion and lowers the sending rate, even though the network is not congested. The TCP/IP enabled system performs slower than the IEEE 802.15.4 based ZigBee sensor networks.

4. Security

From a security perspective, the web-sensor gateway architecture provides a good place to implement user and data authentication. Since all access to the sensor network goes through the gateway, it provides greater security than the TCP/IP enabled sensor network.

VI. Conclusions

A gateway architecture design for interconnecting the Internet with ZigBee based wireless sensor networks has been presented in this paper. The motive was to provide a client in the internet an ubiquitous means to access the sensor in the remote geographic area. The gateway architecture was investigated so that two heterogeneous network can

communicate each other. The presented gateway with a protocol translation mechanism works efficiently when the resource in the sensor is relatively limited. The gateway is particularly good choice over the overlaying method when the energy is the critical factor in the sensor network. While the overlaying method could provide an simpler implementation alternative, it requires higher computational overhead and energy consumption in comparison to our proposed method.

참 고 문 헌

- [1] www.digi.com/technology/wireless-ZigBee.jsp
- [2] en.wikipedia.org/wiki/ZigBee
- [3] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless sensor networks for habitat monitoring," Proc. of WSNA 2002, Atlanta, pp.88-97, 2002(9).
- [4] S. N. Simic and S. Sastry, "Distributed environmental monitoring using random sensor networks," Proc. of the 2nd International Workshop on Information Processing in Sensor Networks, pp.582-592, Palo Alto, pp.143-152, 2003.
- [5] Jin Zhang Daxin Liu, Yuezhu Xu, and Tong Wang, "Dis-symmetrical wireless sensor network Internet access technology based on optimum gateway," Proc. of ICCSIT 2009, Washington DC, pp.755-759, 2008(8).
- [6] A. Dunkels, J. Alonso, T. Voigt, H. Ritter, and J. H. Schiller, "Connecting wireless sensor nets with TCP/IP networks," Proc. of WWIC 2004, Frankfurt, pp.143-152, 2004(2).
- [7] H. Dai and R. Han. "Unifying micro sensor networks with the internet via overlay networking," Proc. of LCN'04, Tampa, pp.571-572, 2004(11).
- [8] A. Dunkels, J. Alonso, T. Voigt, H. Ritter, and J. H. Schiller, "Connecting wireless sensor nets with TCP/IP networks," Proc. of WWIC 2004, Frankfurt, pp.143-152, 2004(2).
- [9] A. Dunkels, "Full TCP/IP for 8-bit architectures," Proc. of MOBISYS '03, San Francisco, pp.85-98, 2003(5).
- [10] <http://dunkels.com/adam/uiip/>
- [11] J. S. Heidemann, F. Silva, C. Intanagonwiwat, R. Govindan, D. Estrin, and D. Ganesan, "Building efficient wireless sensor networks with low-level naming," Proc. of Symposium on Operating Systems Principles, Banff, pp.146-159, 2001.
- [12] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," Proc. of 6th Annual International Conference on Mobile Computing and Networking, Boston, pp.56-67, 2000.
- [13] V. Raghunathan, C. Schurgers, S. Park, and M. Srivastava, "Energy aware wireless microsensor networks," IEEE Signal Processing Magazine, Vol.19, No.2, pp.40-50, 2002(3).

저 자 소 개

허웅(Ung Heo)

정회원



- 2002년 2월 : 전북대학교 컴퓨터 공학과(공학사)
- 2004년 2월 : 전북대학교 컴퓨터 공학과(공학석사)
- 2004년 2월 ~ 현재 : 전북대학교 컴퓨터공학과(박사수료)

<관심분야> : 무선/센서 네트워크, 게이트웨이 아키텍처, 통신 네트워크 프로토콜 성능 모델링

평 치 유(Qiu Peng)

정회원



- 2007년 5월 : Hebei Normal University of Science and Technology(공학사)
- 2010년 2월 : 전북대학교 전자공학(공학석사)
- 2010년 9월 : 난징 NJIT 컴퓨터

공학과 연구원

<관심분야> : 웹-센서 게이트웨이 아키텍처

유 강 수(Kang Soo You)

정회원



- 1994년 2월 : 전북대학교 컴퓨터 공학과(공학석사)
- 2006년 8월 : 전북대학교 영상공학과(공학박사)
- 2006년 9월 ~ 현재 : 전주대학교 교양학부 교수

<관심분야> : 영상처리, 멀티미디어 통신

최 재 호(Jaeho Choi)

정회원



- 1985년 5월 : NCSU(B.S.E.E)
- 1988년 5월 : NCSU(M.S.E.E)
- 1993년 5월 : NCSU (Ph.D in Computer Eng.)
- 1994년 3월 ~ 현재 : 전북대학교 전자공학부 교수

<관심분야> 자가구성 네트워크, 네트워크 컨버전스