A Gateway Protocol Architecture for Zigbee Based Wireless Sensor Network Interconnecting TCP/IP Networks

Peng Qiu, Ung Heo, Jaeho Choi

Abstract

This paper investigates protocol architecture for a web-sensor gateway interconnecting internet and wireless sensor network, in which Zigbee sensors are connected over the IEEE802.15.4 communication protocol standard. The web-sensor gateway is to deliver data between TCP/IP and Zigbee/IEEE802.15.4 protocols, transparently. Since the gateway provides a means to remotely control and aggregate sensor data over the internet, it needs to be designed in the view point of users and in their convenience. In accordance, the common gateway interface technology satisfying users on the web browser to efficiently manage and query the sensors in the wireless sensor networks, ubiquitously, is also introduced. Finally, a simulation prototype for the web-sensor gateway is proposed and verified using OPNET simulation tool.

Keywords : Web-sensor gateway architecture, Zigbee/IEEE802.15.4, TCP/IP, Ubiquitous control, Data aggregation.

I. Introduction

Zigbee is a wireless technology developed as an open global standard to address the unique needs of low cost, low power, wireless sensor networks. The standard takes full advantage of the IEEE 802.15.4 physical radio specification. The Zigbee protocol carries all the benefits of the IEEE802.15.4 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 longer life with smaller batteries; hence the Zigbee sensor network can provide higher reliability and larger range [2].

As the Zigbee sensor network is getting much attention in the wireless communications society, research works on web-sensor gateway are also getting popular. Recently, more literatures on the internet-Zigbee gateway are getting produced. The basic functionalities of the gateway are query distribution, data aggregation, and message conversion [3] [4]. From the literatures, 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 convenience of access [5].

Our research is also motivated by solving the problem of data communication that happens in a Zigbee sensor network and internet. In this paper, we present a novel web-Zigbee gateway architecture that achieves the functions of protocol translation between two different networks. Most of all, we make use of an embedded CGI technology to set up a web server in the gateway so the users on the web browser can visit the Zigbee sensor network through web-sensor gateway [6].

The remainder of the paper is organized as follows. Section II presents some related works that are in the scope of our interests. Section III discusses our proposed web-sensor gateway architecture and web-based Zigbee sensor networks management. In Section IV, a simulation model structure is presented and the conclusions are made in Section V.

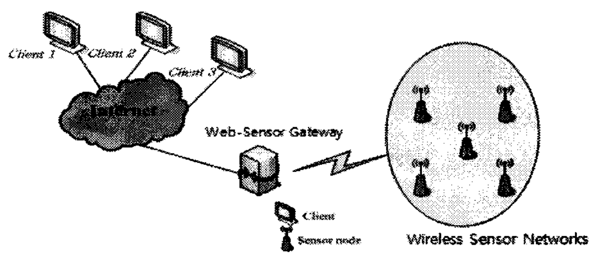

Fig. 1. Interconnecting WSN and internet systems

II. Related Works

There are several related literatures that are related to our work. In this section, three approaches for implementing web-sensor gateway are discussed. Actually, two of them are based on a protocol overlaying method:

*Dept. of Electronic & Information Engineering
Chonbuk National University
투고 일자 : 2008. 11. 24　수정완료일자 : 2009. 7. 28
게재확정일자 : 2009. 7. 29

2.1. Configure IP approach

This is one type of asymmetrical internet access technologies based on an optimum gateway. It uses a reactive style gateway discovery method to equip sensor nodes with global routing IP addresses for realizing the data communication between sensor nodes and main frames on the internet. In particular, all IP-sensor networks are infeasible because it is expected that the wireless sensor network (WSN) is consisted of thousands to millions of tiny sensor nodes, with limited computational and communication capabilities. It is also costly to add all of the nodes with their own IP address [7].

2.2. Internet over WSN approach

It implements a small μIP TCP/IP stack on the small sensor node so that users can access the sensor node while the sensor nodes in the WSN communicate with each other over their own protocol [8].

2.3. WSN over internet approach

Each IP host is regarded as a virtual sensor node, which routes packets the same way as a physical sensor node. The disadvantage of this approach is that an incidental stack in an IP host must be deployed [9].

Ⅲ. Proposed Web-Sensor Gateway

3.1. Connecting heterogeneous networks

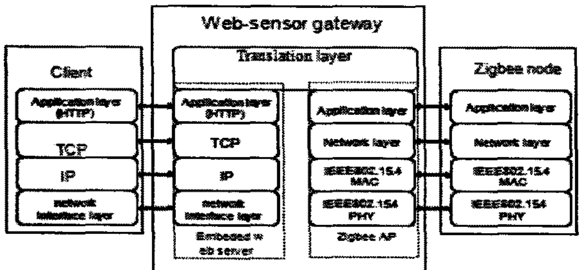

Fig. 2. Web-Sensor gateway protocol architecture

Fig.1 shows a schematic diagram illustrating the connection between the internet and the Zigbee sensor network via a web-sensor gateway. Initially, a client connects to the web server embedded in web-sensor gateway remotely and then sends queries through the internet. The gateway delivers the query to the Zigbee sensor network. The results for the query are returned back to the client's browser via the web-sensor gateway. When the client communicates with the web sever, the transportation of data should obey the TCP/IP protocol. On the other hand, the communications between the gateway and the Zigbee sensor node must undergo Zigbee/IEEE802.15.4 protocol standard. Because of the heterogeneous protocols the web-sensor gateway requires a protocol translation.

3.2. Web-sensor gateway architecture

One of the main tasks that the web-sensor gateway has is to provide an interface between the Zigbee sensor network and the client.

The gateway consists of three parts: Zigbee/IEEE 802.15.4 protocol stack for communications between the gateway and the Zigbee sensor node; embedded TCP/IP protocol stack for communications between the client somewhere in the internet and the gateway, and gateway translation layer. The gateway translation layer is for mapping and translating messages from Ethernet to Zigbee vice versa. As shown in Fig. 2, it is a bundle layer on top of the TCP/IP protocol stack and Zigbee protocol stack. The TCP/IP protocol stack provides communications layers for a web server that is embedded into the web-sensor gateway. The server applications are similar to the typical web server functions. For example, it provides a communication application such as the user browser via a HTTP protocol. The clients anywhere in the internet can remotely control a sensor node using a dynamic webpage.

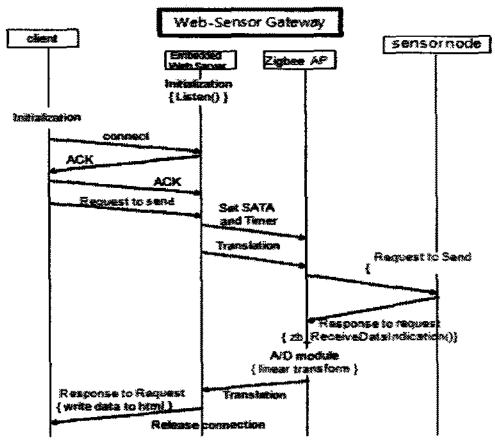

Fig. 3. Control and data exchanges among internet client, gateway and sensor node

The embedded web server that manages the dynamic web page can be implemented using the common gateway interface (CGI). For the network interface layer the Ethernet as its data link layer is used.

3.3. Control and data exchange processes

Figure 3 shows the control and data exchanges among the internet client, gateway, and Zigbee sensor node. The control and data exchange processes can be implemented using C/C++. It starts out by initializing the web server embedded in the gateway. The initialization process proceeds, creating a socket, calling to bind and to listen, and announces the server's willingness to accept incoming calls [9]. 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 the request to the web server after a three-way handshake. The embedded web server sends the enable signal to the Zigbee AP and sets the event timer. Two functions are called to setup the current event and timer as follows:

osal_set_event
(byte task_id, UINT16 event_flag)
osal_start_timer
(UINT16 event_id, UINT16 timeout_value)

The main processes among the client, embedded web server, Zigbee access point (AP), and Zigbee sensor node

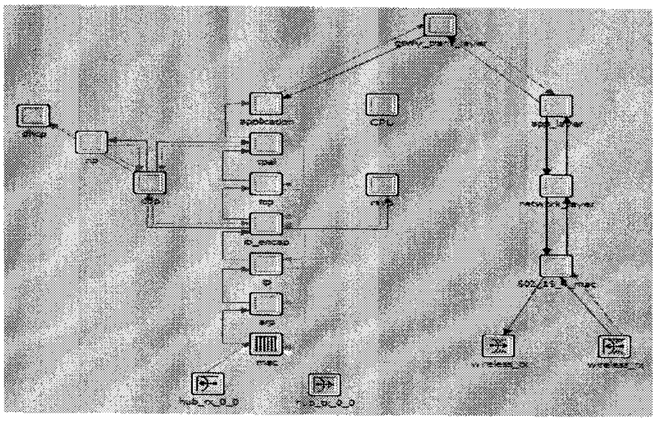

Fig. 4. Web-Sensor gateway node model

are setting event flags and event pollings. The typical flags are as follows:

```
#define ZB_CLIENT_EVENTS       0x00FF
#define TYPE_CHANGE_EVT        0x0010
#define MSG_RECEPTION_EVT      0x0020
```

They are set by the client, embedded web server, and Zigbee AP, respectively. The first event flag is checked by the embedded web server by using polls; the second one by the Zigbee AP; and the last one by the Zigbee sensor node. Whenever the corresponding event queue is not null, the high priority event is disposed first.

Now, one of the most important tasks performed in the gateway is the translation process. Since the embedded web-server communicates with the client using the TCP/IP protocol while the Zigbee AP communicates with the Zigbee sensor node using the Zigbee/IEEE802.15.4 protocol, the commands going to the Zigbee sensor node needs to be translated from TCP/IP into Zigbee while the replies coming back from the Zigbee sensor node to the web-server needs to be also translated in the similar manner.

The Zigbee AP in the gateway receives the translated version of client request in its application layer also identifies the destination Zigbee sensor node in the Zigbee sensor network. Then, the Zigbee AP at the gateway sends out the Request_to_Send Zigbee packet to that sensor node. The frame format for Zigbee application

Table 1. General APS frame format

Octets: 1	0/1	0/2	0/2	0/2	0/1	1	Variable
Frame control	Destination endpoint	Group address	Cluster identifier	Profile Identifier	Source endpoint	APS counter	Frame payload
	Addressing fields						
APS header							APS payload

support sublayer (APS) is shown in Table 1.

As shown in Table 1, the APS frame is divided into APS header and APS payload. The header carries the source, group, and destination addresses and also it may carry the cluster or profile identifier; and their lengths may vary from 0 to 2 bytes. The length of the payload may be varied.

Upon receiving the command from the Zigbee AP, the designated sensor node replies with the Response_to_Request Zigbee packet. As soon as the Zigbee AP in the gateway detects the arrival of the reply packet, it begins to process the packet along the layers in the reverse direction. At the physical layer of the Zigbee, the sensor data gets converted into binary form after passing through an A/D converter [11]. The MAC header gets translated at the MAC layer; and the Zigbee AP address is checked at the network layer whether the received packet is to be routed or not. The sensor data finally reaches the Zigbee application layer and gets loaded into the application support sub-layer.

The Zigbee application layer consists of APS, application framework (AF), and Zigbee devise object (ZDO). The responsibilities of the APS sub-layer includes request, confirm, response and indication primitives for data transfer.

At the gateway translation layer the sensor node data is mapped into the dynamic web browser of the embedded web-server at another end of the gateway.

So, clients can view the replied data in their browser. After viewing the response data the client may terminate the connection or perform another query.

Ⅳ. Simulation Model for our Gateway

4.1. Node model in OPNET

The network simulation tool OPNET is used to implement the proposed web-sensor gateway. Here, we present the OPNET simulation model for the gateway. In OPNET, the node domain provides node models that can

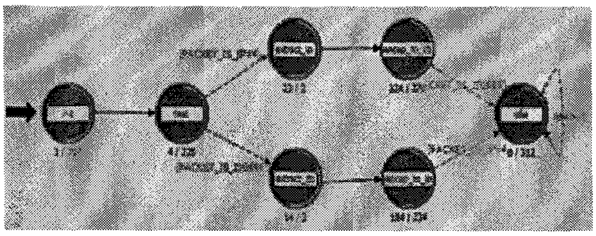

Fig. 5. The process state transition diagram

be deployed and interconnected at the network level for the modeling of communication. The node models are expressed as interconnected modules. Some modules offer capability that is substantially predefined and can only be configured through a set of built-in parameters. Modules are interconnected by either packet streams or statistic wires.

We design the node model architecture as illustrated in Fig. 4. The gateway translation layer module connects embedded server application layer and Zigbee AP application layer. In other words, it provides the interface between a wired network and a wireless network. Packets from the ipv4 format packets are converted to Zigbee format and vice versa.

In the figure, both red lines and blue lines are packet streams, which connect an output stream of a source module to the input stream of a destination module, allowing packets to be communicated and buffered between them. The yellow lines stand for statistic wire that allowing numerical data to be communicated. The other yellow lines with two arrows are called logical association for appropriate transmitter-receiver pairs only to specify that they should be kept together when attaching the node to a link [12].

4.2. Process model in OPNET

In OPNET, the process domain defines the logic flow and behavior for programmable module, i.e., processors or queues. The process is an instance of a process model and operates within one module. Communications between processes are supported by interrupts. The process editor makes use of a powerful state-transition diagram approach to support specification of any type of protocol, resource, application, algorithm, or queuing policy. States and transitions graphically define the progression of processes in response to events. Within each state, a general logic can be specified using a library of predefined functions and also can be specified using the full flexibility of the C language.

The packet translation process details are described further in a process model; and its state transition diagram is shown in Fig. 5. Whenever, the gateway translation layer is called to process the protocol translation, both translation process and testing process must be initialized and be registered at the state init. At the same time, there are three items to be specified and those are as follows:

- Translation direction
- Pointer to the associated test function
- Pointer to the translation function

After initialization and registration, the next state is to test. The testing processes determine the application protocol based on a signature they find. It can be a signature from the http application when the translation direction is from the client to the sensor node; or it can be a signature from the Zigbee application layer when the translation direction is from the sensor node to the client in the internet. This signature is a combination of field values that uniquely determines the format. The test state takes a descriptor block as an argument and the state transition take one of two paths. One takes the upper path after setting packet_is_ipv4 =1 and packet_is_zigbee = 0; and the other one is the lower path after setting packet_is_zigbee = 1 and packet_is_ipv4 =0.

In the extract_ip state followed by the encap_to_zb state, the request from the client gets extracted, and becomes unformatted before getting mapped to a Zigbee format. On the other hand, in the lower path, the extract_zb state followed by the encap_to_ip state is taken. The payload from the sensor node gets extracted and unformatted before getting mapped to IP application format. When the process gets to idle state, it stands for the end of state transition.

When the translation direction was from the client to the sensor node, the upper path in Fig. 5 has been taken and the state transition waits at the idle state until a response arrives at the Zigbee application layer. Then, the reverse order translation process begins by setting packet_from_zb = 1 and moves to the state extract_zb leaving the process blocked at state idle.

V. Conclusion

The contribution of this paper is focused on the architecture of web-sensor gateway, which enables mediating an access from internet to sensor networks. More specifically, we have presented the architecture of embedded web server, the architecture of the gateway translation layer, and the OPNET modeling of the web-sensor gateway and its components. By implementing the proposed model simulated and verified by the OPNET, we can successfully provide a web-sensor gateway that supports clients in the internet to obtain the essential sensor data from the sensor network with a dynamic and simple control of a web browser. In the near future, we expect that the gateway technology for integrating sensor networks and internet will contribute in many fields such as commercial building automation, home automation, personal and hospital care, and telecom applications.

Reference

[1] www.digi.com/technology/ wireless-Zigbee.jsp

[2] en.wikipedia.org/wiki/Zigbee

[3] A. Tanenbaum, Computer Networks (Fourth Edition), USA, pp.326-328, 2002.

[4] K. Hwang, J. In, N. Park, and D. Eom, "A design and Implementation of Wireless Sensor Gateway for Efficient Querying and Managing through World Wide Web," IEEE Trans. on Consumer Electronics, vol. 49, no.4, pp.1090-1097, Nov. 2003.

[5] Lee, L.T, Chui, S.T, Tsang, K.F, "A System Design for Residential Services Delivery Over Internet," Proc.of ICCE 2007, Las Vegas, USA, Jan. 2007.

[6] Peng Qiu, Ung Heo, Jaeho Choi, "Integrating Zigbee sensor networks with internet" 한국신호처리시스템학회 추계학술대회논문집, pp. 298-302, Nov. 2008.

[7] Jin Zhang, Daxin Liu, Yuezhu Xu and Tong Wang, "Dissymmetrical Wireless Sensor Network Internet Access Technology Based on Optimum Gateway," Proc.of ICCSIT 2008, Washington DC, USA, pp.755-759, Aug. 2008.

[8] Adam Dunkels, Juan Alonso, Thiemo Voigt, Hartmut Ritter, Jochen H.Schiller, "Connecting wireless sensornets with TCP/IP networks," Proc. of WWIC 2004, Frankfurt, Germany, pp.143-152, Feb. 2004.

[9] DAI H, HAN R. "Unifying micro sensor networks with the internet via overlay networking," Proc.of LCN'04, Tampa, Florida, USA, pp.571-572, Nov. 2004.

[10] A. Tanenbaum, Computer Networks (Fourth Edition), USA, pp.488-492, 2002.

[11] Shuang Cha , "The Research and Implementation of the Gateway of Zigbee Based Wireless Sensor Network," thesis, China, 2008.

[12] OPNET Modeler Concepts Reference Manual, USA, pp38-39, 2008.

Qiu Peng received his B.S. in computer science from Hebei Normal University of Science and Technology, China in 2007. Currently, he is a M.S. program student in electronic engineering at Chonbuk National University, Chonju, Korea. His current research interests include sensor networks, web-sensor gateway architecture and its design.

Ung Heo received his B.S. and M.S. degrees in computer engineering from the Chonbuk National University, Rep. of Korea in 2002 and 2004, respectively. Currently, he is a Ph.D. program student in computer engineering at Chonbuk National University. His research interests include wireless and mobile communications networks and wireless communications protocol performance modeling.

Jaeho Choi received his B.S.E.E., M.S.E.E., and Ph.D. in computer engineering from North Carolina State University, Raleigh, NC, USA in 1985, 1988, and 1993, respectively. Since 1994, he has been with Chonbuk National University, Chonju, Korea, as a faculty member. Now, he is full professor at the division of electronic engineering. His current research interests include wireless and mobile communications networks, sensor networks, and ad hoc networks.