International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:11, 2007

Implementation of IEEE 802.15.4 Packet Analyzer

Sung Jun Ban, Hyeonwoo Cho, ChangWoo Lee, and Sang Woo Kim

***Abstract*—**A packet analyzer is a tool for debugging sensor network systems and is convenient for developers. In this paper, we introduce a new packet analyzer based on an embedded system. The proposed packet analyzer is compatible with IEEE 802.15.4, which is suitable for the wireless communication standard for sensor networks, and is available for remote control by adopting a server-client scheme based on the Ethernet interface. To confirm the operations of the packet analyzer, we have developed two types of sensor nodes based on PIC4620 and ATmega128L microprocessors and tested the functions of the proposed packet analyzer by obtaining the packets from the sensor nodes.

***Keywords*—**Sensor network, embedded system, packet analyzer.

I. Introduction

UBIQUITOUS sensor network (USN), one of the most important components in the ubiquitous computing, has a wide range of applications, such as in ecosystem monitoring, intelligent alarms, healthcare, management of manufacturing process, and monitoring of cultivating environment [1]. Generally, the USN is composed of many sensor nodes and has complex connections such as multi-hops using wireless communication among the sensor nodes. Therefore, it is very difficult to debug the software errors of the USN. To solve this problem, a so-called packet analyzer is necessary.

The packet analyzer is a tool that has some functions to analyze wireless packets [2][3]. By using this tool, developers can easily obtain the information of the packet, such as structures, types, sizes, and data. Consequently, developers will find and correct errors rapidly and conveniently.

Microchip's ZENA and Texas Instruments (TI)'s packet sniffer are well-known packet analyzers. They consist of a hardware board and analyzer software. The hardware is used for obtaining packets and consists of an RF module, a microprocessor, and peripheral devices. The software runs on a Windows environment and graphically shows the result of packet analysis to the developers.

However, conventional packet analyzers do not support remote control because they require a universal serial bus (USB) interface between the hardware board and a personal computer for running the software. Therefore, developers who use these packet analyzers must visit the deployed sensor network area whenever in order to debug and test the system.

In this paper, we propose a new packet analyzer that has an Ethernet interface instead of a USB interface and is capable of remote access and control. This helps developers to access the packet analyzer remotely and monitor the information of packets.

In section II, we briefly explain the packet structure of IEEE 802.15.4. In section III and IV, we introduce an implemented system scheme and the functions of the proposed system, respectively. Finally, we discuss a topic for future work of the research.

The authors are with the Department of Electronic and Electrical Engineering, POSTECH, Pohang, South Korea (phone: +82-54-279-5018; e-mail: bansjkr, lighto, caprix, swkim@postech.ac.kr)

This research was supported by Ubiquitous Technology Research Center.

II. Structure of IEEE 802.15.4

IEEE 802.15.4 is a standard that defines wireless medium access control (MAC) and the physical layer (PHY) for low-rate wireless personal area networks (LR-WPANs). This standard allows transfer rates of 20 Kbps, 40 Kbps, and 250 Kbps, and uses two network topologies of star and peer-to-peer. In addition, it is suitable for low cost, low energy consumption, and low complexity among sensor devices [4][5].

The packet structure of IEEE 802.15.4 is separated into the PHY layer and its sub-layer named MAC layer. The PHY protocol data unit (PPDU) of the PHY layer is composed of several elements, including a preamble sequence, start of frame delimiter (SFD), PHY header, and PHY service data unit (PSDU). They are shown in Fig. 1.

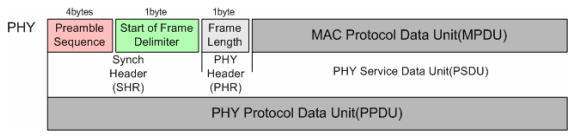

Fig. 1 Structure of the packet of IEEE 802.15.4

The preamble sequence is used for synchronizing messages from an RF transceiver. The SFD is used for indicating the starting point of the packet. The PHY header contains the length of the PSDU, which can have variable-length data.

The packet structure may be classified into certain types according to the MAC protocol data unit (MPDU) of the MAC layer which is referred to as the PSDU in Fig. 1. We introduce two typical packets, an acknowledgement (ACK) packet and a data packet in Fig. 2 and Fig. 3, respectively.

The MPDU consists of a MAC header (MHR), MAC payload, and MAC footer (MFR). They have specific information for the MAC protocol. The MHR is a field

involving the frame control, sequence number, and address information. The MAC payload contains data that has a variable length according to the frame type. The MFR represents frame checksum (FCS) for error correction.

As mentioned above, the structure of the packet has useful information for developers. However, it is not easy to obtain the information of a packet and identify its structure without a packet analyzer, because the received raw packet is simply a string of bits. Therefore, a packet analyzer is necessary for developing the sensor network system. In the next section, we will introduce the implemented packet analyzer.

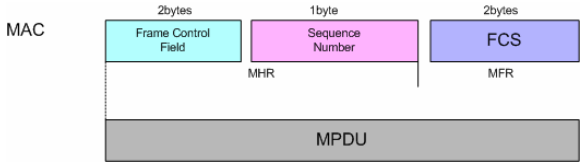

Fig. 2 Structure of the ACK packet of IEEE 802.15.4

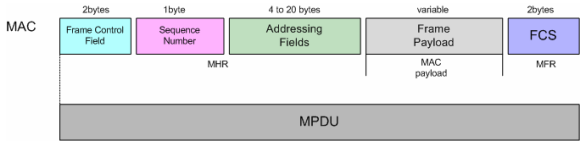

Fig. 3 Structure of the data packet of IEEE 802.15.4

III. Scheme of Implemented System

A. Server-Client Scheme

The scheme of the packet analyzer is represented in Fig. 4. The packet analyzer that is located in a sensor network area can receive packets from its neighborhood sensor nodes. Then, the packet analyzer displays the information and structure of the packet in real time.

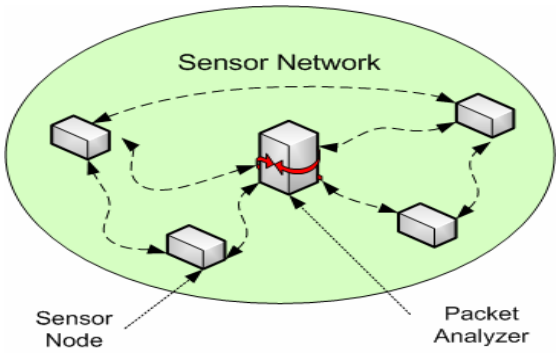

Fig. 4 The sensor network scheme with packet analyzer

Fig. 5 shows the conceptual scheme of the proposed packet analyzer. Developers can remotely access the packet analyzer via the Ethernet network because the hardware board of the proposed packet analyzer has an Ethernet interface. Therefore, they can remotely display the result of packet analysis on their PC and control the functions of the packet analyzer.

To develop the proposed system, we have adopted a server-client scheme as shown in Fig. 6.

The server program runs on the embedded Linux evaluation board based on Intel PXA 270. The evaluation board has peripherals, including an Ethernet controller and a CC2420 RF transceiver manufactured by TI. By using the Ethernet controller and the CC2420 RF transceiver, the server program can communicate with the client and receive packets from the sensor nodes.

The client program operates on a remote PC based on Windows. This program supports access to the server and graphically shows analyzed the packet information to the developers.

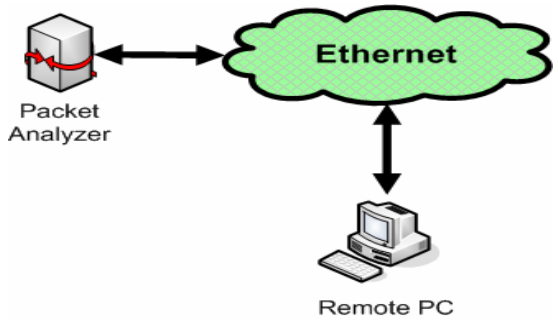

Fig. 5 Conceptual scheme of the proposed system

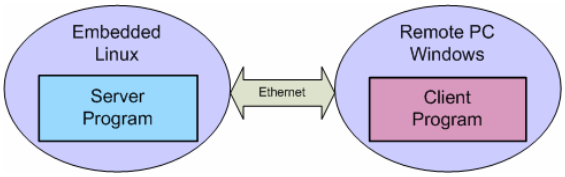

Fig. 6 Server and client relationship

B. Server Program

The server program can be divided into an application program part and device driver parts. The application program uses the functions provided by the drivers of each hardware device as shown in Fig. 7.

The Ethernet device driver is used for controlling the Ethernet device. We use the default Ethernet device driver provided by the evaluation board. Based on this device driver, we have coded a socket program to meet the specification of our system.

The CC2420 device driver provides several functions for handling the CC2420 transceiver. In this research, we have designed the device driver based on Linux kernel 2.6.x for satisfying many requirements of our system. The main functions of the device driver are shown in Table I.

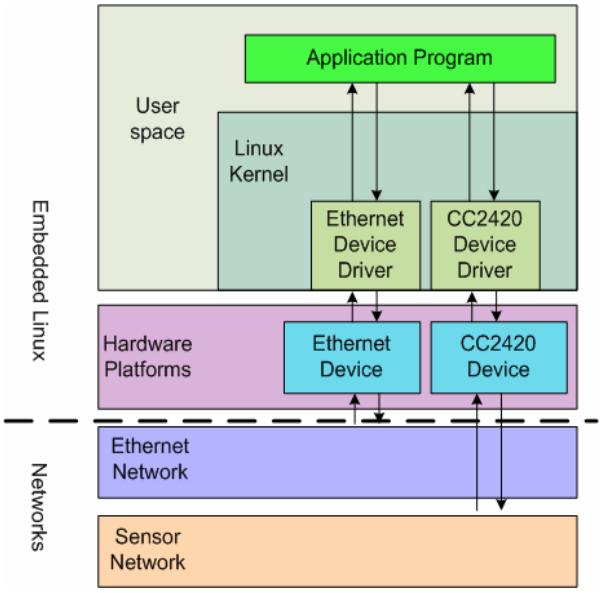

Fig. 7 Conceptual scheme of server program

TABLE I
FUNCTIONS OF CC2420 DEVICE DRIVER

Name	Function
Open	Open the device file of CC2420
Read	Read the data from CC2420
Write	Send the data to CC2420
CC2420_SET_INFO	Change the setting values of the packet analyzer
CC2420_GET_INFO	Get the setting value of the packet analyzer
CC2420_SET_CHANNEL	Change the channel value

C. Client Program

For the user's convenience, the client program provides a graphical user interface (GUI) and has been developed by using Microsoft Visual C++. Similar to conventional packet analyzers, the client program can graphically show the analyzed results of the packets to the user. For this purpose, the client program performs the functions for classifying and analyzing packets. In addition, the client can request the server to execute some commands and receive the response to a command from the server.

Fig. 8 represents the GUI of the client program implemented in this research. By using the client program, users can connect to the server program. The icons at the top of the client window perform useful functions such as connection to the server, auto channel search, start of analysis, stop of analysis, and disconnection. At the bottom of the window, the "setup" tab is used to set the parameters of the CC2420 transceiver and types of packet. The "transmit" tab is used to transmit a user-defined packet. The other tabs are expected to implement the functions for users' convenience later.

Fig. 8 GUI of the client program

IV. OPERATIONS OF IMPLEMENTED SYSTEM

A. Sensor Board

In this research, we have fabricated two types of sensor boards as shown in Fig. 9 in order to test the operations of the proposed packet analyzer. Both the boards are designed with a microcontroller based on ATmega128L and PIC 4620 and include a CC2420 RF module and several sensors.

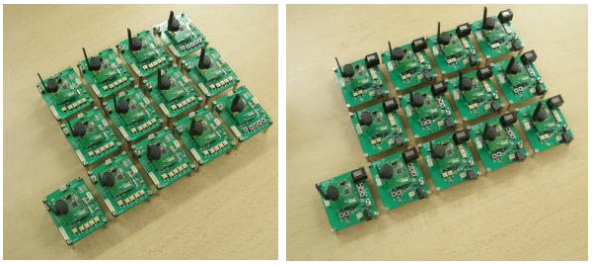

Fig. 9 Implemented sensor boards to test the operations of the proposed system

B. Operations of the Proposed Packet Analyzer

When a developer accesses the server program on the embedded system, the server program transmits its current setting values to the client program. The setting values are channel; RSSI offset value, group ID, source address, and destination address. The group ID, source address, and destination address are necessary to construct the packets of IEEE 802.15.4 in case the users need to operate the hardware as one of the sensor nodes. The channel represents a frequency band out of 16 channels defined in IEEE 802.15.4. The RSSI offset value is used for evaluating the RF signal strength. By using CC2420_GET_INFO as shown in Table I, these values can be acquired. Additionally, developers can change these

values by using the CC2420_SET_INFO as shown in Table I.

After the connection between the server and the client is established, the developers click the auto channel search icon to automatically find the current channel in which packets are activated. If the auto channel search function is executed, it tries to find the RF packets while switching RF channels in turn. It is a very useful function because conventional packet analyzers do not have this function and the channel must be set manually. This is cumbersome for the developers. To change the channel for auto channel search, we use CC2420_SET_CHANNEL as shown in Table I.

When the processes mentioned above are executed, the developers can analyze the packets in the sensor network field. After clicking the packet analyzer icon to begin the analysis, the client sends the command to the server. Immediately, the server captures the packets from the sensor nodes by using the "Read" function shown in Table I, and then the captured packets are transmitted to the client. Finally, the client creates the graphical result according to the packet structure of IEEE 802.15.4, as mentioned in section II. The analyzed packets are shown in Fig. 10. This result is acquired by obtaining packets from sensor nodes as shown in Fig. 9.

Fig. 10 Packet analysis result

C. Additional Operation of the Proposed System

To offer a more useful function to developers, we have implemented a function for the transmission of any user-defined packet. Through the interface of the second tab at the bottom of the window in Fig. 11, developers can execute this function.

If developers use this function, they can debug the sensor network more actively. We have confirmed this operation by using Microchip's ZENA since our system does not receive the packet while transmitting the user-defined packet. The result is shown in Fig. 12.

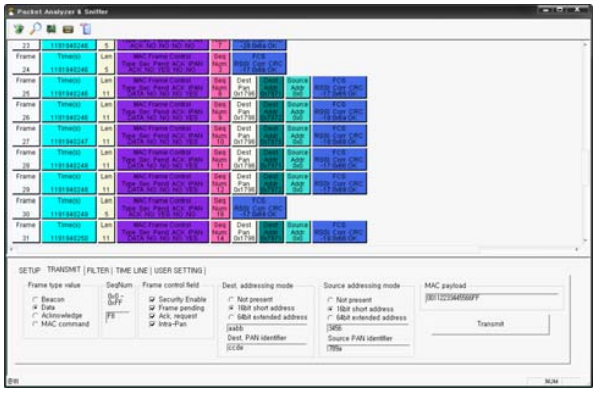

Fig. 11 GUI of transmit tab

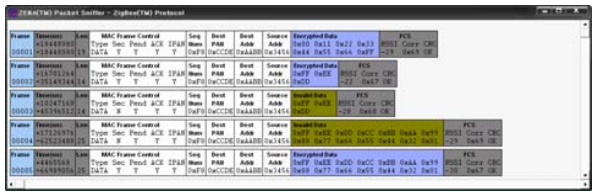

Fig. 12 User-defined packet transmission test by using ZENA

V. Conclusion

In this research, we have implemented a new packet analyzer with an additional function of transmitting user-defined packets to sensor nodes for debugging the sensor network system remotely. Our proposed system overcomes the disadvantages of conventional packet analyzers and provides a basis for advanced debugging methods.

In a future work, we will develop an intelligent algorithm that automatically detects the errors of the sensor network system.

References

[1] David Culler, Deborah Estrin, Mani Srivastava, "Guest Editors' Introduction: Overview of Sensor Networks," *Computer*, vol. 37, no. 8, pp. 41-49, Aug. 2004.
[2] Microchip, "ZENA Wireless Network Analyzer User's Guide", 2007.
[3] Texas Instruments (TI), "Application Note AN033".
[4] IEEE Standard for Information Technology-Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specification for Low-Rate Wireless Personal Area Networks (LR-WPANs), 2003.
[5] J. A. Gutierrez et al., "IEEE 802.15.4: A Developing standard for Low-Power, Low-Cost Wireless Personal Area Networks", in IEEE Networks, pp. 12-19, Sept. 2001.