

# 임베디드 리눅스 기반의 IEEE 802.15.4 패킷 분석기 설계 및 구현

## Design and Implementation of IEEE 802.15.4 Packet Analyzer Based on Embedded Linux

이창우, 조현우, 반성준, 김상우\*

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

**Abstract :** Ubiquitous sensor network (USN) is composed of many sensor nodes which are one of the simplest form of embedded system. In developing the sensor network system, a debugging tool is necessary to test and verify the system. Recently, a so-called packet analyzer has been developed for this purpose, and it supports IEEE 802.15.4 which is considered as the standard for the sensor network protocols. The major function of the packet analyzer is to take RF packets from sensor nodes and show the structure and the data of the packets graphically to the user. However, the conventional packet analyzers do not support remote control because they require a USB interface along with a personal computer. To make it available for remote control, we propose a new packet analyzer based on a server-client scheme in which a server program is implemented on embedded Linux and a client program is implemented on Windows for convenient use.

**Keywords :** embedded system, sensor network, packet analyzer, server-client

### I. 서론

유비쿼터스 컴퓨팅의 핵심 요소 중 하나인 Ubiquitous Sensor Network (USN)가 기존의 사용처인 군사지역 감시, 공공기관 관리, 생태계 감시 등에서 벗어나 농업, 어업, 상업, 건설업 등의 생산 자동화 및 관리에 확대 적용되고 있으며, 또한 의료, 복지, 교육, 방범 등 개인 생활에도 적용 가능성이 클 것으로 기대되고 있다[1,2]. 위와 같이 다양한 응용 서비스를 지원하는 센서 네트워크 시스템은 일반적으로 소형의 임베디드 시스템 형태인 센서 노드가 다수 사용되며, 각 센서 노드들이 서로간에 복잡한 무선 네트워크를 형성하기 때문에 센서 네트워크 개발 과정에서 발생하는 여러 가지 소프트웨어적 오류를 디버깅 하는 작업이 매우 힘들다. 그러므로 이를 해결하기 위한 도구가 센서 네트워크 시스템 개발 시에 필요하다.

이러한 필요성에 따라 개발된 패킷 분석기 (packet analyzer)는 무선 네트워크 상에서 각 노드들이 전송하는 패킷을 캡처하여 그 패킷의 구조와 패킷이 가지고 있는 데이터 등을 분석할 수 있는 도구이다[3,4]. 패킷 분석기를 사용할 경우 여러 노드간에 전달되는 패킷 정보를 분석할 수 있으므로 디버깅 과정의 가장 기본이 되는 오류 발견 작업을 비교적 쉽게 수행할 수 있다.

잘 알려져 있는 패킷 분석기로는 Microchip사의 ZENA [3]와 Texas Instruments (TI)사의 packet sniffer [4]가 있다. 두 패킷 분석기는 공통적으로 IEEE 802.15.4 [5] 표준을 만족하는 패킷을 분석할 수 있으며, Windows 환경의 PC에서 구동되는 패킷 분석 프로그램과 USB를 통해 해당 PC에 연결하여 사용되는 하드웨어로 구성되어 있다.

이러한 기존의 패킷 분석기는 실험실 환경에서 센서 노드를 테스트할 경우에는 무리 없이 사용 가능 하지만, 실제 환경에 센서 노드를 설치하고 테스트하는 경우에는 시스템 개발자가 직접 그 환경에 접근하여 사용해야 하는 불편함이 있다. 그 이유는 앞서 살펴본 바와 같이 패킷 분석기의 하드웨어가 USB 인터페이스를 가진 PC를 기반으로 하고 패킷 분석 프로그램이 원격 접속 방법을 지원하지 않기 때문이다.

본 논문은 기존 시스템의 문제점을 해결하고자, 원격 제어가 가능한 임베디드 리눅스 기반의 새로운 패킷 분석 시스템을 제안한다. 논문의 구성은 우선 IEEE 802.15.4 패킷 구조에 관해 간략히 살펴본 후, 구현한 시스템 구조 및 기능에 대해 소개하며, 결론을 통해 향후 방향을 논의하는 것으로 구성하였다.

### II. IEEE 802.15.4 패킷 구조

유비쿼터스 환경을 위한 무선 네트워킹 기술로 기대되는 IEEE 802.15.4 기술은 20Kbps, 40Kbps, 250Kbps의 전송 속도와 저렴한 가격, 긴 배터리 수명, 간단한 구조 및 연결성을 제공하여 10m 이내에서의 무선 연결을 필요로 하는 분야에 적합한 표준으로 개발되고 있다[5,6]. IEEE 802.15.4의 패킷 구조는 Physical (PHY) Layer와 Medium Access Control (MAC) Layer로 나누어 고려될 수 있으며, 이에 대한 구조를 그림 1에 나타내었다.

그림 1의 패킷 구조에서 PHY 계층의 PHY Protocol Data Unit (PPDU)를 살펴보면, Preamble Sequence는 RF 모듈에서 입력되는 메시지를 동기화하기 위해 사용되는 필드이며, Start of

\* 책임저자(Corresponding Author)

논문접수 : 2007. 9. 29., 채택확정 : 2007. 10. 26.

이창우, 조현우, 반성준, 김상우 : 포항공과대학교 전자전기공학과 (caprix@postech.ac.kr/lighto@postech.ac.kr/bansjkr@postech.ac.kr/swkim@postech.ac.kr)

※ 본 연구는 유비쿼터스 신기술 연구 센터(UTRC) 지원에 의해 연구되었음.

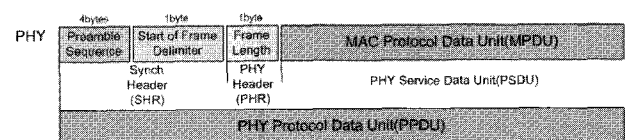


그림 1. IEEE 802.15.4의 패킷 구조.

Fig. 1. The structure of the packet of IEEE 802.15.4.

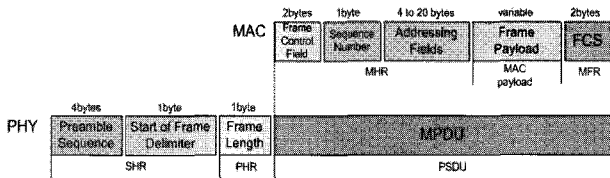


그림 2. IEEE 802.15.4의 DATA 패킷 구조.  
Fig. 2. The structure of the data packet of IEEE 802.15.4.

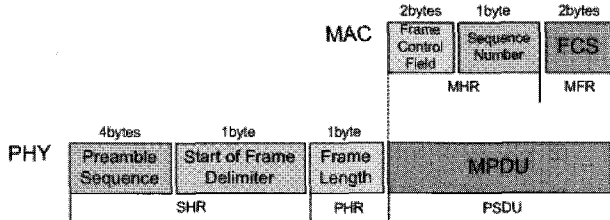


그림 3. IEEE 802.15.4의 ACK 패킷 구조.  
Fig. 3. The structure of the ACK packet of IEEE 802.15.4.

Frame Delimiter는 패킷의 시작을 나타내는 필드이다. PHY Header (PHR)은 PHY Service Data Unit (PSDU)의 길이를 나타내는 7비트와 1비트의 예약 비트로 이루어져 있고, 가변적인 길이를 가지는 PSDU는 임의의 데이터 전송을 위해 사용된다[5,6]. MAC Layer는 그림 1에 나타난 PHY Layer의 MAC Protocol Data Unit (MPDU)에 해당하며, MAC Layer를 어떤 형식으로 사용하는가에 따라 센서 네트워크 패킷 형태가 좌우된다. MPDU 부분은 크게 MAC Header (MHR), MAC payload, MAC Footer (MFR)로 나뉘어진다. MHR은 프레임 제어, 시퀀스 넘버 (Sequence number), 어드레스 정보 등을 포함하는 필드이며, MAC payload는 프레임 타입에 따른 특유의 정보를 포함하며 가변 길이를 가진다. MFR은 Frame Checksum을 포함하는 필드이다. 여기서는 IEEE 802.15.4를 만족하는 패킷 중에서 대표적인 DATA 패킷과 ACK 패킷에 대한 구조를 도식화하여 그림 2,3에 각각 나타내었다.

지금까지 살펴본 IEEE 802.15.4 패킷은 실제 데이터 전송에 있어 길게 나열된 일련의 비트로 표현된다. 그러므로 개발자 입장에서 일련의 비트만을 살펴보고 해당 부분의 의미를 곧바로 정확히 이해한다는 것은 어려운 일이다. 따라서 분석하고자 하는 패킷의 구조를 도식화하고 각 필드의 비트 값이 가지는 의미를 재구성하여 실시간으로 보여주는 것은 센서 네트워크 시스템 디버깅 작업시 손쉽게 오류를 발견하는데 도움이 된다. 다음 절에서는 본 연구에서 구현한 패킷 분석기에 대해 자세히 설명하고자 한다.

### III. 구현한 시스템의 구조

#### 1. 서버-클라이언트 구조

본 논문에서 구현한 패킷 분석기의 개념도는 그림 4에 나타나 있다. 이는 센서 네트워크 필드 내에 위치하고 있는 임베디드 보드가 패킷 분석기 역할을 담당하여 주변의 여러 센서 노드가 전송하는 패킷을 수신하며, 사용자는 센서 네트워크 필드 내에 위치하지 않고도 원격지의 컴퓨터에서 이더넷을 통해 임베디드 보드에 접속하여 패킷 분석 결과를 모니터링 할 수 있는 시스템이다. 이러한 원격지 패킷 모니터링 기

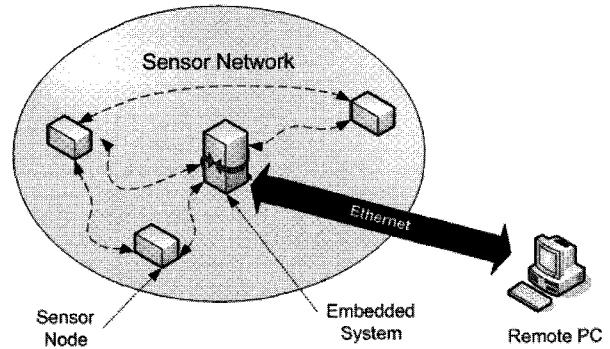


그림 4. 제안한 패킷 분석기 구조도.  
Fig. 4. The proposed packet analyzer scheme.

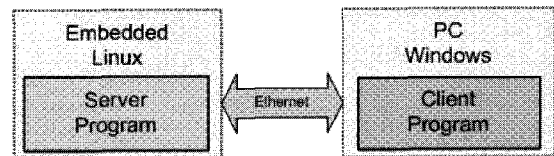


그림 5. 서버 클라이언트 관계.  
Fig. 5. Server and client relationship.

능은 센서 네트워크의 베이스 노드 (싱크 노드) [7]가 센서 네트워크의 패킷들이 멀티홉 (multi-hop)을 통해 최종적으로 전달되는 노드라는 점을 활용하여 구현될 수도 있을 것이다. 하지만, 베이스 노드로 전달된 패킷은 라우팅 알고리즘에 따라 시간 지연이 있어 실시간성을 보장할 수 없다[8]. 또한 패킷 분석기는 완성된 센서 네트워크가 아닌 개발 중인 센서 네트워크의 패킷을 분석하여 오류를 발견하는 것을 위해 주로 사용되기 때문에, 관찰하고자 하는 패킷이 정상적으로 베이스 노드까지 전달될 것이라 예상할 수 없다. 이러한 이유로 베이스 노드로의 접속을 통한 원격지 패킷 모니터링은 분명한 한계를 가진다.

본 연구에서 구현한 패킷 분석기는 그림 4의 기능을 수행할 수 있도록 하기 위하여 서버-클라이언트 (server-client) 구조를 이용한다. 서버는 센서 네트워크 필드 내에 위치한 임베디드 보드에 해당되며, 클라이언트는 원격지의 사용자가 이용하는 컴퓨터에 해당된다. 서버와 클라이언트에서는 각각 서버 프로그램과 클라이언트 프로그램이 구동된다.

서버는 Intel PXA270 기반의 32bit 임베디드 개발 보드와 함께 임베디드 리눅스를 운영체제로 사용하였다. 사용한 임베디드 개발 보드는 주변장치로 이더넷 컨트롤러와 IEEE 802.15.4를 지원하는 TI사의 CC2420 RF 트랜시버를 갖추었다. 서버 프로그램은 이더넷 컨트롤러를 이용하여 클라이언트와의 통신을 할 수 있으며, CC2420 RF 트랜시버를 이용하여 센서 노드로부터 전송되는 패킷을 수신하거나 자체적으로 패킷을 송신할 수 있도록 한다.

클라이언트는 원격지에서 사용자가 이용하는 컴퓨터로써, 본 논문에서는 Windows 기반의 PC로 가정하였다. 클라이언트 프로그램은 사용자가 이더넷을 통해 서버로 접속할 수 있도록 하는 기본적인 기능을 가지며, 서버와 연결된 후 서버로 명령을 전송하거나 모니터링 결과를 표시하는 GUI를 제공한다. 사용자는 이 프로그램을 통해 패킷 분석 결과를 도

식화된 형태로 모니터링 할 수 있고, 임의의 패킷을 센서 네트워크로 전송하는 작업도 가능하다. 이러한 서버-클라이언트 관계는 그림 5에 나타나 있다.

2. 서버 프로그램

서버 프로그램은 크게 디바이스 드라이버 부분과 응용프로그램 부분으로 나눌 수 있다. 디바이스 드라이버는 앞서 언급한 이더넷 컨트롤러와 CC2420 RF 트랜시버와 같은 하드웨어적 자원을 제어하고 활용 가능하도록 하는 함수를 제공한다. 응용 프로그램은 디바이스 드라이버가 제공하는 함수를 바탕으로 서버가 필요로 하는 기능을 구현한 프로그램이다. 이러한 구조를 가지는 서버 프로그램의 개념도는 그림 6에 나타나 있다.

본 연구에서는 리눅스 커널 2.6을 기반으로 하여 시스템을 개발하였다. 우선 디바이스 드라이버 부분부터 살펴보면, 이더넷 컨트롤러를 위한 디바이스 드라이버의 경우, 기존의 오픈 소스를 이용하여 별도로 구현하지 않고 사용하였다. 하지만 CC2420 RF 트랜시버의 경우, 후에 살펴볼 응용프로그램이 요구하는 모든 기능을 구현하는데 있어 완벽히 부합되는 디바이스 드라이버는 구현되어 있지 않기에, CC2420 RF 트랜시버의 디바이스 드라이버를 본 연구의 요구 조건에 맞추도록 직접 구현하였다. 개발된 CC2420 RF 트랜시버 디바이스 드라이버가 제공하는 핵심 함수는 표 1과 같다. CC2420 RF 트랜시버의 디바이스 드라이버는 문자 디바이스 드라이버로서, 응용 프로그램은 CC2420 RF 트랜시버를 Open(), Close(), Read(), Write()와 같은 파일 처리용 함수를 이용하여 일반 파일처럼 다루어 처리한다[9]. Open함수와 Close함수는 CC2420 RF 트랜시버 디바이스 파일을 각각 열고, 닫는 함수이다. Read함수는 CC2420 RF 트랜시버 디바이스 파일로부터 값을

읽어오는 함수이다. 즉, CC2420 RF 트랜시버가 수신한 RF 패킷을 Read함수를 사용하여 읽고, 그 값을 응용프로그램 상의 특정 배열에 저장할 수 있다. Write함수는 CC2420 RF 트랜시버 측으로 응용프로그램 상의 데이터를 전달하는 함수이다. 즉, CC2420 RF 트랜시버 디바이스 파일에 데이터를 쓰게 되면, 그 데이터가 CC2420 RF 트랜시버로 전송되고 CC2420 RF 트랜시버 내부에서는 해당 데이터를 변조하여 송신하게 된다. ioctl함수에서 미리 정의된 명령은 CC2420 RF 트랜시버를 사용하는데 필요한 여러가지 설정값들을 읽어들이어 확인하거나 변경하는 기능을 제공한다. 응용 프로그램에서 ioctl함수를 호출할 때, CC2420\_SET\_INFO를 사용하는 것과 동시에 하드웨어 설정 값을 지정할 경우, 해당하는 값으로 CC2420 RF 트랜시버의 설정 값을 변경할 수 있다. CC2420\_GET\_INFO를 사용할 경우 CC2420 RF 트랜시버의 설정값을 읽어 들여 지정한 변수에 저장하는 것이 가능하다. CC2420\_SET\_CHANNEL은 CC2420 RF 트랜시버의 여러 설정 값 중, 채널 값을 설정하는데 사용된다.

응용 프로그램 부분은 클라이언트의 명령을 수행하기 위한 루틴들로 이루어져 있다. 클라이언트 명령을 수행하는 루틴은 대부분 CC2420 RF 트랜시버를 이용하여야 하므로, 앞에서 설명한 CC2420 RF 트랜시버 디바이스 드라이버가 제공하는 함수를 기반으로 구현되었다. 서버가 클라이언트의 다양한 명령을 전달 받거나 처리 결과를 클라이언트로 반환하는 기능 구현에는 이더넷 기반의 소켓 프로그램을 사용하였다. 또한 보다 효율적인 명령 체계와 데이터 전송이 가능하도록 하기 위하여 사전에 정의한 프로토콜을 사용하였다. 즉, 서버와 클라이언트는 사전에 약속된 형식의 명령어들을 이용하며, 소켓 프로그램을 통해 명령어 및 데이터를 쉽게 주고 받을 수 있도록 그 형태를 구조체로 정의하여 사용하였다.

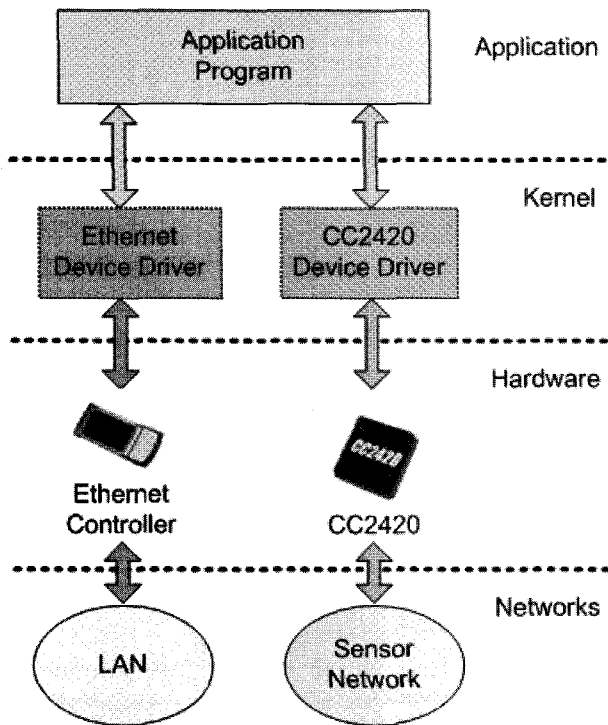


그림 6. 서버프로그램의 개념도.  
Fig. 6. Conceptual scheme of server program.

표 1. CC2420 RF 트랜시버 디바이스 드라이버 함수.

Table 1. Functions provided by CC2420 RF transceiver device driver.

함수 이름 및 인자		기능
	Open	CC2420의 device 파일을 연다.
	Close	CC2420의 device 파일을 닫는다.
	Read	CC2420으로부터 데이터를 읽어온다.
	Write	CC2420으로 데이터를 내보낸다.
ioctl	CC2420_SET_INFO	IEEE802.15.4가 정의하는 여러 가지 설정 값을 변경한다.
	CC2420_GET_INFO	현재 설정되어 있는 여러 가지 설정 값을 읽어온다.
	CC2420_SET_CHANNEL	RF 통신 채널을 설정한다. 11 (0x0B)부터 26 (0x1A)까지 설정 가능하다.

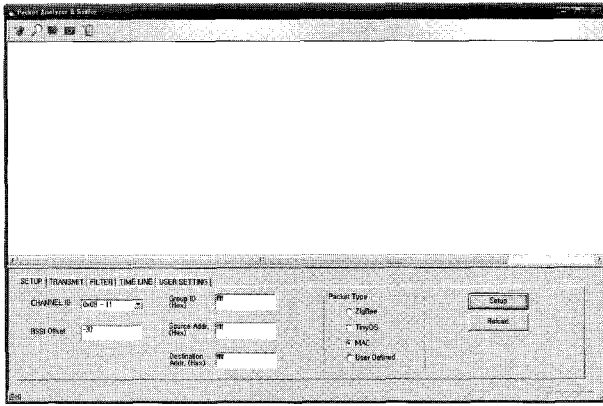


그림 7. 클라이언트 프로그램.  
Fig. 7. Client program.

3. 클라이언트 프로그램

클라이언트 프로그램은 서버 프로그램과는 달리 사용자의 직접적인 인터페이스이므로 사용자의 편의를 위해 Windows 환경에서 Microsoft Visual C++ 을 이용한 GUI 형태로 개발하였다. 클라이언트 프로그램은 서버로부터 전달 받은 패킷 분석 결과를 기존의 패킷 분석기와 같이 도식화하여 메인 화면에 나타낼 수 있다. 이를 위해 클라이언트 프로그램에는 내부적으로 패킷 구조를 파악하여 분류하는 과정이 포함되어 있다. 또한 클라이언트의 다양한 명령은 서버에 전달될 때, 사전에 약속된 프로토콜에 따라 전달되며, 서버는 해당 명령어에 따른 작업을 수행하고, 그 결과를 다시 클라이언트에게 반환한다.

그림 7은 본 연구에서 구현한 클라이언트 프로그램의 GUI를 보여준다. 사용자는 이 프로그램을 통해 쉽게 임베디드 보드의 서버 프로그램에 접속하고 원하는 작업을 수행할 수 있다.

클라이언트 프로그램의 GUI에 대해 살펴보면 상단에는 접속 시도, 자동 채널 검색, 패킷 분석 시작 및 중지, 접속 해제와 같이 사용 빈도가 높은 기능들의 아이콘이 있다. 하단의 탭을 살펴보면, 먼저 setup 탭에서는 패킷 분석기 역할을 하는 임베디드 보드의 CC2420 RF 트랜시버를 사용하기 위한 설정 값과 패킷 분석에 필요한 값들을 설정할 수 있다. Transmit 탭에서는 사용자가 정의한 임의의 패킷을 전송하는데 필요한 인터페이스를 제공해 주며, 나머지 탭은 패킷 분석을 위한 사용자 설정과 관련된 부분으로써 추후 사용자 편의를 위해 마련하였다.

IV. 구현한 시스템의 동작원리 및 기능

클라이언트가 임베디드 보드에 접속하게 되면 서버는 클라이언트에게 임베디드 보드가 현재 가지고 있는 설정 값들을 전송하며, 클라이언트는 이 값을 사용자에게 보여준다. 설정 값은 channel, RSSI offset value, group ID, source address, destination address이다. Group ID, source address, destination address는 IEEE 802.15.4의 MAC layer에서 정의하고 있는 패킷 구조에 따라 패킷을 구성하는데 필요한 값들이다. 채널은 표준에 정의되어 있는 16개의 주파수 대역을 말하며, RSSI offset은 RF 신호 세기 계산에서 사용되는 값이다. 서버에서

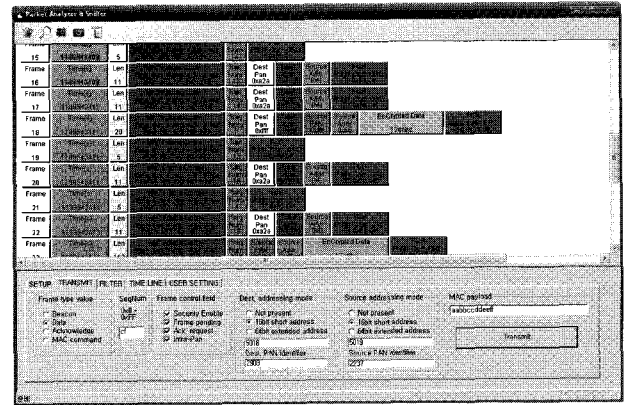


그림 8. 패킷 분석기의 모니터링 결과.  
Fig. 8. Monitoring result of packet analyzer.

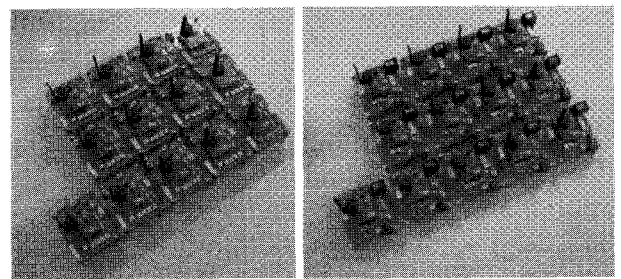


그림 9. ATmega128L과 PIC4620 기반의 센서 노드.  
Fig. 9. Sensor boards based on ATmega128L and PIC4620.

는 해당 값들을 취득하기 위해서 표 1에서 언급한 ioctl함수에서 CC2420\_GET\_INFO를 사용한다.

클라이언트가 서버에 접속한 후 사용자는 클라이언트를 통해 명령을 서버로 전송할 수 있다. 가장 대표적인 명령은 패킷 분석을 시작하는 명령이다. 해당 명령이 클라이언트로부터 요청되면, 서버는 센서 네트워크 상의 패킷을 캡처하여 이를 알맞은 형태로 변환한 후 클라이언트에게로 전송한다. 클라이언트는 서버로부터 받은 패킷을 클라이언트 프로그램의 메인 화면에 사용자가 쉽게 이해할 수 있는 구조로 도식화하여 나타낸다. 패킷 구조의 표현은 IEEE 802.15.4에 정의되어 있는 표준을 따른다. 이러한 패킷 분석기의 동작은 서버 프로그램에서 실행된 쓰레드가 CC2420 RF 트랜시버 드라이버가 제공하는 Read함수를 호출하는 방식으로 구현되었다. 패킷 분석 명령을 실행한 후 나타나는 패킷 분석 결과 화면은 그림 8에 나타나 있다.

본 연구에서는 개발한 패킷 분석기의 기능을 실험하기 위하여 자체적으로 제작한 센서 보드를 사용하였다. 해당 센서 보드는 그림 9에 나타나 있으며, 각각 Atmel사의 ATmega128L과 Microchip사의 PIC4620 마이크로프로세서를 사용하여 제작되었다. 두 종류의 센서 보드는 공통적으로 CC2420 RF 트랜시버를 사용하며, ATmega128L보드와 PIC4620보드에는 센서 네트워크를 구성할 수 있도록 최소한의 기능을 구현해 놓은 오픈 소스를 컴파일 하여 포팅하였다. 이와 같이 준비된 서로 다른 두 종류의 센서 네트워크 보드에 대해서, 개발한 패킷 분석기를 이용하여 패킷 분석을 시도한 결과 두 종류의 보드로부터 패킷을 모두 분석할 수 있음을 확인하였다. 그리

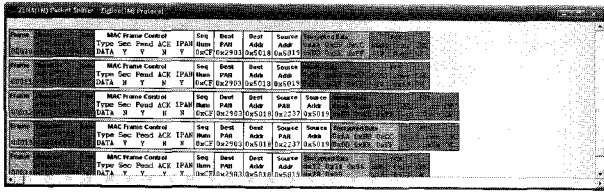


그림 10. ZENA를 통한 사용자 정의 패킷 전송 확인.  
Fig. 10. User-defined packet confirmation by using ZENA.

고 센서 노드가 보내는 패킷의 group ID, source address, destination address와 MAC payload에 실리는 데이터를 임의의 값으로 변경하여 실시한 실험에서 개발한 패킷 분석기는 임의로 변경한 값들을 정확히 표시하였다.

클라이언트의 다른 기능으로 사용자가 임의로 설정하기를 원하는 channel, RSSI offset, group ID, source address, destination address를 입력 받아 임베디드 보드에 해당 값들을 변경하는 것이 있다. 이 기능은 서버가 클라이언트로부터 위의 값들을 전달 받은 후 CC2420 RF 트랜시버 디바이스 드라이버 ioctl 함수에서 CC2420\_SET\_INFO 명령을 이용하여 값들을 설정하는 방법을 사용하여 구현하였다.

기존의 패킷 분석기는 패킷 분석 대상이 되는 센서 네트워크 시스템이 사용하는 채널을 사전에 알지 못할 경우 사용자가 채널을 시행 착오법을 통해 찾아야 하는 불편함이 있었다. 이를 해결하고자 본 논문에서 제안한 패킷 분석기는 자동으로 패킷이 존재하는 채널을 검색해주는 기능을 갖추고 있다. 자동 채널 검색 기능은 IEEE 802.15.4에서 정의한 16개의 채널을 순차적으로 검색하면서 각 채널에서 패킷이 감지되는지를 확인한다. 만약 특정 채널에서 패킷이 감지된다면 해당 채널 값을 자동으로 설정하며, 어떤 채널에서도 패킷이 감지되지 않았다면 채널 찾기가 실패했음을 사용자에게 알려준다. 이 기능은 ioctl함수에서 CC2420\_SET\_CHANNEL 명령을 사용하여 채널 값을 순차적으로 바꾸어 가면서 패킷을 검색하는 과정을 통해 구현할 수 있었다.

제안된 패킷 분석기가 가진 두드러진 기능 중 하나로, 사용자가 정의한 임의의 패킷을 센서 네트워크로 전송할 수 있도록 하는 것이 있다. 이 기능은 클라이언트가 제공하는 GUI 환경을 통해 사용자가 원하는 패킷 형태를 설정하고, 전송하고자 하는 데이터를 입력하면, 해당 패킷 형태와 데이터를 서버로 전송하고, 서버는 CC2420 RF 트랜시버 디바이스 드라이버가 제공하는 write함수를 이용하여 사용자가 정의한 패킷과 동일한 패킷을 센서 네트워크로 전송한다. 이 기능의 테스트는 본 연구에 사용한 개발 보드가 패킷 전송과 수신을 동시에 할 수 없는 이유로 인해 상용 패킷 분석기인 ZENA를 이용하여 사용자가 임의로 정의한 패킷이 정확하게 수신되는지 여부를 조사함으로써 이루어졌다. 이 실험에서는 MHR 및 MAC payload의 값을 임의로 변경시켜 실험을 행하였다. 실험 결과, 개발한 패킷 분석기를 통해 송신된 사용자 정의 패킷의 정보가 정확히 ZENA를 통해 나타남을 확인함으로써 사용자가 정의한 패킷이 정상적으로 전송되고 있음

을 확인할 수 있었고, 그 분석 결과는 그림 10에서 나타내었다. 이 기능은 센서 네트워크 시스템 구현 시 사전에 특정 명령을 데이터의 형태로 정의하고, 해당 명령을 담은 패킷을 사용자가 임의로 전송함으로써 센서 네트워크가 그 명령에 따라 반응하게 하는 방식을 통해 센서 네트워크를 테스트하거나 관리하는데 유용하게 사용될 수 있을 것이라고 고려된다.

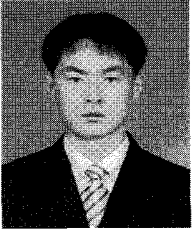
V. 결론

본 연구에서는 센서 네트워크의 디버깅 및 테스트를 위해 사용되는 패킷 분석기가 원격 지원이 가능하도록 하기 위하여 서버-클라이언트 구조의 새로운 패킷 분석기 프로그램을 구현하였다. 서버 프로그램은 32비트 Intel PXA270 마이크로 프로세서 기반의 임베디드 리눅스 환경에서 동작하며, CC2420 RF 트랜시버 디바이스 드라이버를 사용하여 다양한 기능을 수행한다. 클라이언트 프로그램은 원격지의 Windows환경의 PC에서 실행되는 프로그램으로서 사용자의 편의를 위해 GUI로 개발되었다.

개발된 패킷 분석기를 통해 센서 네트워크 필드에 직접 접근하지 않고도 원격지에서 패킷 분석 결과를 모니터링 하는 것이 가능함을 보였으며, 또한 서버가 클라이언트의 다양한 명령을 처리할 수 있도록 한 점은, 패킷 분석기가 패킷을 캡처하여 보여주는 본래의 기능뿐만 아니라, 센서 네트워크 관리에 도움을 줄 수 있는 추가적인 기능을 수행할 수 있는 도구로 발전할 수 있다는 사실을 시사한다. 향후 패킷 분석기가 자체적으로 시스템의 오류를 발견할 수 있는 알고리즘을 개발할 예정이며, 추가적으로 다양한 기능을 구현할 예정이다.

참고문헌

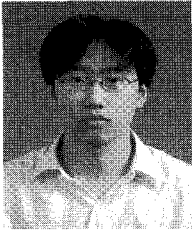
- [1] 김지은, 김세한, 정운철, 김내수, "USN 센서노드 기술동향," 전자통신동향분석, 제 22 권, 제 3 호, pp. 90-103, June 2007.
- [2] 동성수, 이종호, "센서 네트워크 응용을 위한 무선 통신 모듈 설계 및 구현," 제어·자동화·시스템공학 논문지, 제 12 권, 제 1 호, pp. 60-65, January 2006.
- [3] Microchip, "ZENA wireless network analyzer user's guide," 2007.
- [4] Texas Instruments (TI), "Application Note AN033".
- [5] IEEE Standard for Wireless Medium Access Control and Physical Layer Specification for Low-Rate Wireless Personal Area Networks, P802.15.4, 2003.
- [6] 전호인, "IEEE 802.15.4 WPAN 기술," 대한전자공학회, 전자공학회지, 제 32 권, 제 4 호, pp. 87-104, April 2005.
- [7] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "A survey on sensor networks," *Communication Magazine, IEEE*, vol. 40, Issue 8, pp. 102-114, August 2002.
- [8] 최재원, 이광휘, "무선 센서 네트워크에서 링크 에러율을 고려한 실시간 데이터 전달 기법," 대한전자공학회, 전자공학회논문지-TC, pp. 146-154, January 2007.
- [9] 유영창, 리눅스 디바이스 드라이버, 한빛 미디어, 2004.

**이창우**

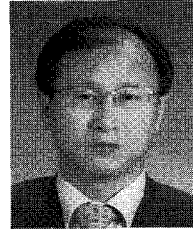
2002년 경북대학교 전자전기공학부(공학사). 2004년 포항공과대학교 전자전기공학과(공학석사). 2004년~현재 포항공과대학교 전자전기공학과 박사과정 재학중.

**조현우**

2006년 한양대학교 전자전기공학부(공학사). 2006년~현재 포항공과대학교 전자전기공학과 통합과정 재학중.

**반성준**

2006년 경북대학교 전자전기공학부(공학사). 2006년~현재 포항공과대학교 전자전기공학과 통합과정 재학중.

**김상우**

1983년 서울대학교 제어계측공학과(공학사). 1985년 동 대학원 제어계측공학과(공학석사). 1990년 동 대학원 제어계측공학과(공학박사). 1991년~현재 포항공과대학교 전자전기공학과 부교수. 관심분야는 최적제어, 최적화 알고리즘, 지능제어, 무선통신, 공정 자동화.