

확장성 향상을 위한 6LoWPAN 기반의 IP-USN 시스템 구현

오 승 교[°], Ardiansyah Musa Efendi^{*}, 최 덕 재^{**}

6LoWPAN Based IP-USN System Implementation for Improving Scalability

Seung-kyo Oh[°], Ardiansyah Musa Efendi^{*}, Deok-Jai Choi^{**}

요 약

IP기반의 IP-USN은 6LoWPAN이라는 표준 기술을 적용하여 IPv6 USN시스템을 구축함으로써 Ubiquitous Sensor Network의 센서 존재확인 및 지역적 한계의 단점을 극복한다. 하지만 USN의 노드는 관리, 비용 측면에서 주기적인 배터리 교체 없이 가능한 오랜 기간 동작해야 하는데 이를 위한 해결방안이 미비하다. 본 논문에서는 저전력으로 동작하는 최적의 IP-USN을 위해 TI사의 저전력 SoC솔루션인 CC2530과 Contiki OS를 기반으로 노드를 구성하고 IP 네트워크와 센서네트워크를 연결하는 게이트웨이를 설계 및 구현하여 6LoWPAN을 지원하는 IP-USN을 구현한다. 또한 구현된 시스템의 에너지 소비량을 측정 및 분석하여 독립전원 공급을 위한 방안을 모색한다.

Key Words : 6LoWPAN, ZigBee, IP-USN, Wireless Sensor Network, Energy consumption

ABSTRACT

IP-based IP-USN overcomes that disadvantages of verify the existence and geographic limits of the sensor by applying a standard technique called 6LoWPAN. However, in terms of the management and cost, The USN node should work for a long period of time without periodic batter replacement. but this solution is insufficient. In this paper, we configure the node using TI's CC2530 that low-power Soc solution and Contiki OS for optimal low-power operation IP-USN and implement the gateway to support connecting IP networks and Sensor networks. In addition, a system implemented to measure and analyze the energy consumption of an independent power supply to look for ways.

I. 서 론

21세기로 넘어오면서 인터넷이 급격히 성장함에 따라 정보의 수집과 활용 및 통신이 인간-인간 관계에서 인간-사물, 사물-사물 관계까지 넓은 범위로

확장되었다. 이러한 변화는 USN이라는 새로운 개념을 탄생시켰다. USN은 Ubiquitous Sensor Network의 약자로, 필요한 모든 곳에 전자태그와 센서를 부착하여 사물의 정보나 주변 환경 정보 등을 수집한 후 실시간으로 그 정보들을 관리할 수

※ 본 연구는 교육부와 한국연구재단의 지역혁신인력양성사업으로 수행된 연구 결과임 (No.2011-04-대-03-018)

♦ 주저자 겸 교신저자 : 전남대학교 전자컴퓨터공학부 차세대네트워크 연구실, silencetg@gmail.com, 정희원

* 전남대학교 전자컴퓨터공학부 차세대네트워크 연구실, me@ardisragen.net

** 전남대학교 전자컴퓨터공학부 차세대네트워크 연구실, dchoi@jnu.ac.kr, 종신회원

논문번호 : KICS2013-05-202, 접수일자 : 2013년 5월 2일, 최종논문접수일자 : 2013년 8월 16일

있도록 구성된 네트워크를 의미한다. USN을 구성하는 센서 노드들은 센싱을 하기 위하여 자체전력을 가져야하고, 센싱을 통하여 수집한 정보들을 저전력으로 전송해야 했는데, 바로 이 점이 USN의 핵심적인 문제라고 할 수 있다. 이에 따라 ZigBee protocol 기반 USN이 주로 개발되었다. ZigBee는 IEEE 802.15.4에서 표준화가 진행된 저전력 및 저속전송 통신을 지향하며 USN 구축에 적합한 솔루션 중 하나이다.

그러나 ZigBee기반 USN은 센싱정보를 수집하여 활용하는데 있어 센서 존재 확인이 어려운점, 통신거리의 제한으로 인한 지역적인 한계 등의 문제점이 있다. 이에 따라, 이런 문제점들을 보완할 수 있는 IP기반의 USN이 등장하게 되었다. IP-USN이라고 부르며, USN이 대표적으로 ZigBee 기술을 적용했듯 IP-USN에서는 '6LoWPAN'이라는 기술을 적용하여 개발되고 있다.

본 논문에서는 센서노드에 6LoWPAN을 탑재하여 IPv6기반의 USN 시스템을 구현하고 차후 자체 전력 공급을 위한 에너지 하베스팅 기술을 적용하기 위해 구현된 시스템의 소비전력을 분석한다. 이 시스템은 6LoWPAN이 탑재된 센서노드와 이를 IPv6인터넷 망과 연결할 수 있도록 하는 게이트웨이, 외부에서 IP를 통해 네트워크에 접근 가능한 안드로이드 기반의 어플리케이션으로 구성된다. 사용자는 스마트폰의 어플리케이션을 통해 본 시스템에 접근하여 각각의 센서노드를 제어할 수 있다. 이는 현재 주목받고 있는 센서네트워크를 이용한 홈 자동화 시스템, 빌딩 자동화 시스템, 유헬스 시스템 등 IP기반의 센서 네트워크 연구에 활용 될 수 있으며 TI사의 저전력 SoC솔루션인 CC2530을 기반으로 노드를 구성하고 저전력으로 동작하는 최적의 IP-USN 시스템의 필요 전력을 측정 및 분석하여 에너지 하베스팅 기술을 적용한 독립전원 공급이 가능함을 보임으로써 전원관리 문제를 해결 할 수 있다.

본 논문의 구성은 다음과 같다. 2장은 무선센서네트워크의 플랫폼 및 표준을 살펴보고 기존 연구 사례들을 살펴본다. 3장에서는 6LoWPAN 네트워크와 인터넷의 연결을 위한 게이트웨이에 대해 설명한다. 4장에서는 시스템의 구현 방법 및 동작 예시를 보인다. 마지막으로 5장에서는 본 논문의 결론과 향후 과제를 다룬다.

II. 관련 기술 개발 현황

2.1. 무선센서네트워크 플랫폼 분석

무선센서네트워크를 이용한 응용 서비스로는 홈 자동화 시스템, 빌딩 자동화 시스템, 군사적 감시 및 경찰 시스템, 개인건강보호, 산업제어, 전등 및 상업제어, 컴퓨터 주변장치, 가전제품 등이 있다. 이러한 서비스를 구현하기 위해서 센서 노드간에 Contiki OS와 같은 센서노드용 OS를 사용하고 ZigBee나 6LoWPAN과 같은 표준 프로토콜이 구현된 스택을 사용하고 있다.

지난 몇 년간 센서 노드 하드웨어의 발달과 더불어 많은 센서 네트워크용 초소형 운영체제가 개발되어왔다. 현재 많이 알려져 있는 센서 네트워크용 OS로는 TinyOS^[1], Contiki^[2], T-kernel^[3], MANTIS^[4], SOS^[5] 등이 있으며, 국내 기술로 개발된 것으로는 Nano-Qplus^[6]가 있다.

이 중 Contiki OS는 크기가 작으며 오픈소스로 개발된 멀티태스킹 OS로써 동적 모듈 재 할당 기능을 제공한다. Contiki는 기본적으로 TinyOS, SOS와 같이 event-driven 모델을 따르고 있다. 하지만 Protothread라는 기술을 통해 stack reservation 없이 멀티쓰레드 같은 코드 간의 동기화를 제한적으로 제공한다. 네트워크 스택으로는 uIP와 Rime을 지원하는데 uIP는 TCP/IP 스택을 축소하여 임베디드 환경에서 동작할 수 있도록 만든 것으로 인터넷을 통해 통신을 할 수 있도록 해준다. Rime은 저전력 통신을 위해 구현된 멀티 홉 데이터 전송을 지원하며, 컴파일된 코드가 600byte 미만인 간단한 프로토콜이다. 또한 현재는 IPv6, 6LoWPAN을 위한 스택과 RPL 라우팅 프로토콜을 지원하고 있다^[7].

저전력 무선 네트워크의 핵심 기술로 높은 주목을 받고 있는 ZigBee 기술은 IEEE 802.15.4의 PHY/MAC 기술을 바탕으로 'ZigBee Alliance'에서 추가적인 기능을 부여하여 더욱 향상된 기술을 지원하고 있다^[8]. ZigBee의 특징은 가격을 최대한 낮추고 전력 소모를 최소화하는 대신 소량의 정보를 통신할 수 있다는 것이다. 하지만 IP를 고려하지 않는 설계로 인한 문제점이 제기되었고, 저전력 무선 네트워크 기술을 개발하는 기업을 중심으로 이미 익숙하고 널리 활용되어 왔던 IP 프로토콜을 이용하여 IEEE 802.15.4기반의 ZigBee 대응 기술을 개발하고자 2005년 3월 62차 IETF 회의에서는 6LoWPAN WG을 창설하였다.

6LoWPAN 워킹그룹은 LoWPAN 상에서 IPv6

패킷 전송 방안을 정의하는 것을 목표로 하고 있다. 6LoWPAN은 ZigBee와 마찬가지로 IEEE 802.15.4를 PHY/MAC으로 하는 저전력 WPAN에 IPv6를 탑재하여 기존의 IP 네트워크와 연결하는 기술로 IPv6의 adaptation layer에 해당된다. IPv6를 이용하게 되면 IP 네트워크를 통해 통신하므로 지역적인 한계를 극복할 수 있고 기존의 IP기반 서비스들과 연계할 수 있으므로 시간과 장소에 구애 없이 다양한 분야의 서비스가 가능하다. 앞서 언급되었던 ZigBee 기반 USN과 6LoWPAN 기반 USN(IP-USN)을 간략하게 비교하면 표 1 과 같다.

Service binding method	provide	Existing TCP/UDP use
Integrated Network Management	Difficult	Easy
service discovery	Local	Global
Mobility	Difficulty support	Easy support

6LoWPAN은 IP를 사용함으로써 기존의 구축된 통신 및 응용서비스 인프라를 그대로 이용할 수 있어서 비용이 절감될 뿐만 아니라 잘 알려지고 검증된 IP 기술들을 사용할 수 있어 신뢰성과 안정성도 도모할 수 있다.

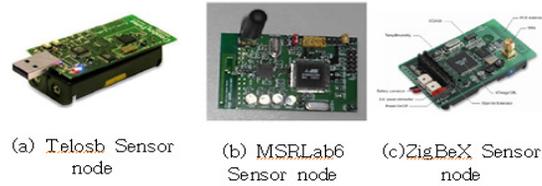


그림 1. 센서네트워크 테스트베드 사례
Fig. 1. Sensor network testbed example

본 논문에서는 저전력 RF통신에 최적화된 CC2530칩과 무선센서네트워크를 위한 초소형 OS인 Contiki OS를 사용하여 센서노드에 6LoWPAN을 탑재하고 IPv6기반 IP-USN시스템을 구현한다. 이 시스템은 센서노드간의 무선통신과 IPv6기반의 인터넷망을 연결하기 위한 게이트웨이 그리고 각 센서노드를 제어하기 위한 어플리케이션으로 구성된다. 서비스를 제공하는 사용자는 안드로이드 기반의 어플리케이션을 통해 시스템에 접근하여 각 센서노드를 제어할 수 있다. 이는 이더넷/Wi-Fi 연결이 가능한 모든 시스템에서 각 노드에 개별적으로 접근할 수 있는 사물인터넷의 기초로서 향후 저전력, 고신뢰의 무선센서네트워크 연구에 활용 할 수 있을 것으로 기대한다.

먼저 그림 1 (a)은 Telosb와 Tiny OS를 응용한 테스트베드 이다. 이 연구에서는 Berkeley WEBS(Wireless Embedded System)그룹에서 연구한 IPv6를 지원하는TinyOS를 기반으로 대규모의 무선센서네트워크를 구성하였다. 또한 아파치를 이용한 Web서버와 MySQL을 이용한 데이터베이스 그리고 PHP를 이용한 웹페이지를 통해 무선센서네트워크를 모니터링 하고 관리할 수 있는 6LoWPAN기반의 무선센서네트워크를 구현하였다⁹⁾.

2.2. 테스트베드 구현 사례

무선센서 네트워크를 위한 플랫폼 및 표준 프로토콜의 응용을 위해서 테스트베드를 구축하고 이를 응용한 연구가 활발하게 수행되고 있다.

Hue, Hongwei 등이 구현한 IPv6기반의 테스트베드에 이용된 MSRLab6의 센서노드(MSR6680)는 그림 1 (b)와 같다. IPv6을 응용하기 위한 테스트베드로서 센서노드의 H/W는 ATmega128L과 CC2420으로 구성되어 있다. 저용량의 센서노드에 IPv6을 지원하기 위해서 Tiny IPv6 프로토콜 스택을 개발하였다. Tiny IPv6는 코드사이즈를 줄이기 위해서 uIP와 같은 방법을 사용하여 8비트 MCU에서도 동작이 가능하도록 하였다. IPv6의 패킷을 IEEE 802.15.4의 프레임에 캡슐화하여 센서노드들간에 데이터를 주고 받으며 동작하고, 자체 라우팅 프로토콜을 설계하여 센서네트워크에 응용한 시스템이다.

표 1. ZigBee기반 USN과 6LoWPAN기반 USN의 비교
Table 1. ZigBee based USN and 6LoWPAN based USN comparison

	ZigBee based USN	6LoWPAN based USN
PHY/MAC layer	IEEE 802.15.4 standard	
Standard Organization	ZigBee Alliance	IEFT 6LoWPAN WG
Network Address	16bit or 64bit (PAN)	IPv6 (Globally unique)

ATmel사의 ATmega128L과 TI사의 CC2420을 이용하여 구글맵을 통한 위치추적 서비스를 제공하는 6LoWPAN 테스트베드를 구현한 연구의 센서노드는 그림 1 (c)와 같다. 이 연구에서는 센서노드에 6LoWPAN을 탑재하여 IP기반 응용서비스인 구글맵을 이용한 LTSGM(Location Tracking Service Based on Google Map)시스템을 제안하고 테스트

베드를 구축하였다. 이 시스템은 웹브라우저가 탑재된 단말기를 통해서 시스템에 접근하여 위치에 기반한 센서노드의 측정값과 모바일노드의 위치추적 서비스를 제공받을 수 있는 시스템이다¹⁰⁾.

기존에 구현된 센서노드는 TI사의 CC2420트랜시버를 사용 하였으며 CC2420을 기반으로한 SoC 타입의 CC2430과 본 연구에서 사용된 SoC 타입의 CC2530을 비교하면 표 2 와 같다.

CC2530은 ZLL을 위한 골든 유닛 인증을 받은 솔루션으로 다양한 용량의 Flash Memory를 제공하고 최대 256kB를 지원함으로써 용도와 상황에 따라 대용량의 어플리케이션을 탑재할 수 있어 확장이 용이하다. 또한 기존연구에 사용되던 CC2420을 기반으로 한 CC2430보다 Radio성능과 에너지 소모면에 있어서 보다 높은 성능을 가지고 있다.

기존 연구의 구현사례를 보면 무선센서네트워크가 단순한 지역적인 센서정보를 수집하기 위한 목적뿐만 아니라 인터넷망과 연동하여 각각의 센서노드를 직접 제어하거나 모니터링 할 수 있는 방향으로 이를 응용한 다양한 서비스를 제공하기 위한 방향으로 많은 연구가 진행되고 있음을 알 수 있다. 이는 향후 IPv6가 대중화 되면 무선센서네트워크 또한 하나의 네트워크로 통합 될 것이며, 또한 대규모의 무선센서네트워크의 활용과 제어를 통해 현재 IP 기반으로 이루어지는 서비스들에 응용될 것이다.

표 2. CC2430과 CC2530의 비교
Table 2. CC2430 and CC2530 comparison

	CC2430	CC2530
Features		
Type	SoC	SoC
Program memory	32/64/128 kB Flash	32/64/128/256 kB Flash
RAM	8K(<4L during PM2/3)	8K in all PMs
Clock loss detection	No	Yes
Operating Temperature	-40 to -85	-40 to +125
Programmable output power ranging from	-25 to 0 dBm	-28 to 4dBm
Radio Performance		
Sensitivity (dBm)	-92	-97

Max Tx Power (dBm)	0	+4.5
Link Budget	92	101.5
Low Power		
Operating Voltage	2.0 - 3.6 V	2.0 - 3.6 V
Rx Current	27mA	24mA
Tx Current (0 dBm)	27mA	29mA
Tx Current (+4.5 dBm)	NA	34mA
CPU activity current (32 MHz)	10.5mA	6.5mA
Xtal startup time	0.5ms	0.3ms

III. 시스템 설계 및 구현

3.1. 시스템 설계

IP-USN에서 무선센서네트워크의 각각의 센서노드는 정보를 전송하는데 그치지 않고 고유의 IP주소를 보유하고 있어 사용자가 각 노드에 직접적으로 접근하여 모니터링 또는 제어를 할 수 있어야한다.

특히 대규모의 센서네트워크를 구축할 때 각각의 노드가 주소를 갖는 것은 유지, 보수, 관리, 제어 측면에서 매우 중요하다. 본 논문에서는 대규모 센서네트워크를 구성하고 응용서비스를 제공하기 위한 6LoWPAN기반의 무선센서네트워크 시스템을 구현한다. 또한 구현된 시스템을 바탕으로 센서노드의 소비전력을 측정 및 분석하고 에너지 하베스팅을 이용하여 자체전력 공급을 위한 방안을 모색한다.

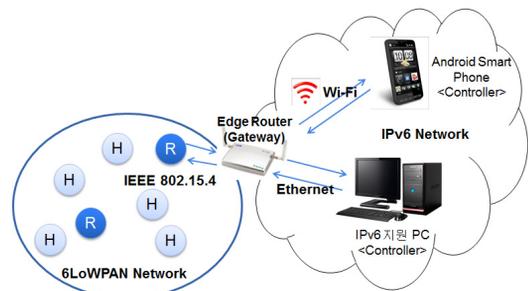


그림 2. IP-USN 시스템 아키텍처
Fig. 2. IP-USN system architecture

그림 2 는 구현하는 시스템의 전체적인 모습이 다. 본 시스템은 각 센서노드를 모니터링 또는 제어 할 수 있는 IPv6인터넷 망에 연결된 PC또는 스마트폰 그리고 802.15.4 통신을 지원하는 6LoWPAN 기반의 센서노드 마지막으로 IPv6인터넷 망과 무선 센서네트워크 망을 연결해주는 게이트웨이로 구성되어 있다.

센서노드는 802.15.4기반의 저전력 통신을 지원하는 CC2530 모듈을 사용하며 이 노드는 디지털 주변광 센서인 ISL 29023에 연결되어 있다. 센서는 정상작동 시 70 μ A의 전력을 소모하며 전원이 꺼졌을 시 0.3 μ A로 소비 전력이 줄어든다. 센서노드는 게이트웨이로부터 브로드캐스팅된 RPL(IPv6 Routing Protocol Low power and Lossy Networks)의 DIO(DoDAG Information Object) 메시지를 받아 DAG(Directed Acyclic Graph)를 형성함으로써 무선센서네트워크를 구성한다. 네트워크를 구성 후 일정간격으로 데이터를 전송하고 ping 테스트를 통해 네트워크의 구성을 확인한다.

게이트웨이는 IPv6인터넷 망과 6LoWPAN기반의 무선센서네트워크 망을 연결하는 역할을 하며 자신을 RPL Root노드로 설정한 후 통신 영역 안에 있는 노드들에게 DIO메세지를 브로드캐스팅함으로써 무선센서네트워크를 생성한다. 또한 802.15.4기반의 네트워크와 IPv6기반의 인터넷망 사이에서 데이터를 각각의 포맷에 맞게 변환하고 전달하는 역할을 수행한다.

센서노드를 모니터링하고 제어하는 PC 또는 스마트폰 어플리케이션은 IPv6기반의 인터넷망에서 게이트웨이를 통해 각각의 노드에 직접 접근하여 데이터를 보내거나 받음으로써 무선센서네트워크의 노드들을 제어 하는 역할을 한다.

3.2. 센서노드 구현

무선센서네트워크를 구성하는 센서노드는 기본적으로 데이터를 수집하고 무선 통신을 이용하여 코디네이터 노드로 전송하는 역할을 한다. 그림 3 은 본 테스트베드에서 사용된 노드이며 하드웨어는 TI CC2530과 SmartRF05 Evaluation Board를 이용하였다. TI CC2530은 IEEE 802.15.4를 위한 true system-on-chip(SoC) 솔루션으로 2.4GHz RF 트랜시버, 8051 MCU, 최대 256KB 플래시 메모리, 8KB RAM등 강력한 성능을 제공하며, 저렴한 가격으로 무선센서네트워크를 구성 할 수 있는 칩 중 하나이다^[11].

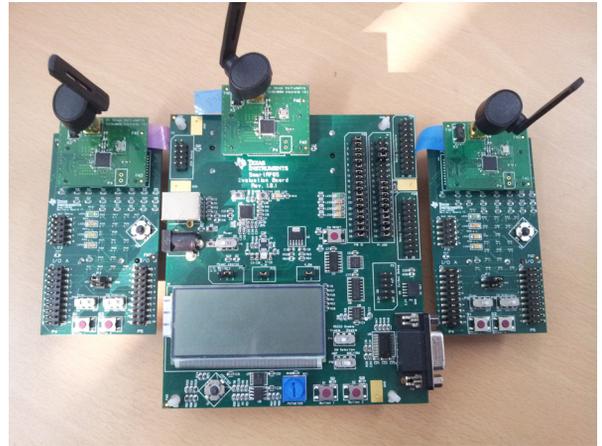


그림 3. 센서노드의 H/W 모습
Fig. 3. Sensor node H/W

본 연구의 테스트 베드에서는 센서네트워크용 OS인 Contiki OS를 사용하며 Ping 테스트와 센서 노드의 LED 제어 명령을 전송함으로써 무선센서네트워크의 형성을 테스트 한다.

Contiki OS는 2012년 7월 2.6 버전^[12]을 공개했고 2.6버전부터 CC2530과 6LoWPAN 대한 플랫폼을 제공하고 있으며 리눅스 기반으로 되어있다. 오픈소스인 Contiki OS는 소스코드 또는 시뮬레이터(Cooja)를 포함한 테스트 환경이 갖추어진 VMware 이미지 형태로 제공되어진다. 소스코드를 다운받아 IAR 컴파일러로 포팅 후 인스톨하거나 가상머신 이미지파일을 받아 VMware를 통해 구동 후 컴파일 할 수 있다. 본 테스트베드 구현에서는 VMware를 이용하여 리눅스 환경에서 컴파일 하였다. 본 테스트베드에서 사용되는 하드웨어 플랫폼에 Contiki를 인스톨하기 위해서는 SDCC(Small Device C Compiler)를 필요로하고 추가적으로 gcc, flex,bicon, libboost-graph-dev, Python, srecord를 리눅스 환경에서 설치해야한다. 컴파일 환경 구성 후 컴파일하게 되면 hex파일을 얻을 수 있다. 이 파일을 TI에서 제공하는 SmartRF Flash Programmer를 통해 하드웨어로 다운로드하여 Contiki와 CC2530을 기반으로 한 센서노드를 구현하였다.

센서노드는 IPv6 Address AutoConfiguration을 통해 prefix FE80을 사용하는 local ipv6 address를 생성하고 이후 RPL root로 동작하는 coordinator노드 또는 게이트웨이로부터 Global ipv6주소의 prefix를 받아 Global ipv6 address를 생성한다. 본 테스트베드의 센서노드는 무선센서네트워크의 형성을 테스트하기 위해 부모노드와 연결 후 일정시간 간격으로 데이터를 전송하고 ping테스트를 수행한

다. 또한 제어 어플리케이션으로부터 받은 명령에 따라 노드의 LED를 On/Off하는 간단한 형태이다. 센서노드에서 수행되는 태스크는 6LoWPAN 스택 태스크와 어플리케이션 태스크가 있다.

6LoWPAN 스택 태스크에서는 6LoWPAN 패킷 포맷을 바탕으로 무선센서네트워크 노드간 데이터를 주고받을 수 있는 구조를 가지고 있다. 전송할 수 있는 패킷의 최대 크기는 127byte이고 여기서 PHY/MAC이 사용하는 최대 정보를 고려하면, 80byte안에 IPv6/TCP 또는 IPv6/UDP 헤더 및 센서 응용을 위한 데이터가 표현되어야 한다. 이를 위해 6LoWPAN에서는 dispatch와 HC1, HC2를 구성하여 헤더를 압축한다 [13][14]. 어플리케이션 데이터는 본 테스트베드에서 구동 되는 각 센서노드를 식별할 수 있는 노드의 식별 문자가 포함된다.

3.3. 게이트웨이 구현

인터넷망과 무선센서네트워크의 연동을 위해서는 서로 다른 망을 연결해주고 패킷을 변환시켜주는 게이트웨이를 필요로 한다. 이러한 기능을 하는 게이트웨이는 현재 많은 연구가 활발히 진행 중이며 본 연구의 시스템에서는 직접 설계하고 구현한 게이트웨이를 사용하였다.

본 연구에서 제안하는 게이트웨이는 IPv6/NDP 매커니즘 그리고 RPL을 지원하며 이더넷/Wi-Fi 와 6LoWPAN 네트워크 사이에서 데이터 변환 서비스를 제공한다.

그림 4, 그림 5는 게이트웨이의 설계와 구현에 사용된 컴포넌트와 아키텍처다. 기존에 연구된 게이트웨이와 다르게 이더넷, Wi-Fi망의 연결 그리고 라우터와 브릿지 모드를 모두 제공 할 수 있도록 설계하였다. 이를 구현하기 위해 802.15.4, Ethernet,USB 무선 인터페이스를 지원하는 각각의 컴포넌트들을 조립하였다. 보드는 RapsberrPi(RPi)를 사용하였다. RPi는 ARM 프로세서 기반의 소형 컴퓨터이며 100Mbit 이더넷, 2 USB, 그리고 GPIO 모듈을 탑재하고 있다. 802.15.4 인터페이스는 Nooliberry보드를 RPi 보드에 추가적으로 부착하여 구현하였고, Wi-Fi 인터페이스는 Edimax EW-7811Un 802.11n USB 어댑터를 보드에 연결하여 구현하였다.



그림 4. 게이트웨이의 H/W 컴포넌트
Fig. 4. Gateway H/W component

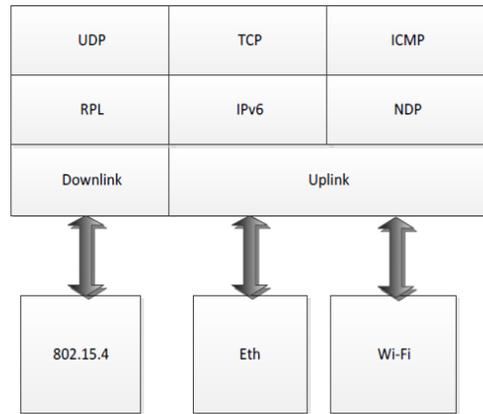


그림 5. 게이트웨이 아키텍처 다이어그램
Fig. 5. Gateway Architecture Diagram

본 연구에서 제안하는 게이트웨이는 라우터 모드와 브릿지 모드 두 가지 형태로 동작 할 수 있다. 이 두 가지 형태의 모드는 그림 6, 그림 7과 같다. 라우터 모드에서는 두 개의 IPv6 서브넷을 연결하는 완전한 IPv6 라우터로 동작한다. 무선센서네트워크 서브넷은 RPL 프로토콜에 의해 관리되고 이더넷/Wi-Fi 서브넷은 IPv6 NDP에 의해 관리된다. 라우터 모드에서는 무선센서네트워크와 IPv6기반의 네트워크를 각각 서브넷으로 분리하기 때문에 무선센서네트워크의 노드들을 분별 할 수 있다. 브릿지 모드는 게이트웨이가 스위칭 기능을 제공하여 IPv6기반의 네트워크와 RPL기반의 무선센서네트워크를 하나의 서브넷으로 연결한다. 802.15.4 인터페이스를 대상으로 들어오는 패킷 또는 이더넷/Wi-Fi에서 들어오는 멀티캐스팅 패킷 모두 무선센서네트워크 세그먼트로 전달되고 반대로 이더넷/Wi-Fi 인터페이스를 대상으로 들어오는 패킷 또는 802.15.4 인터페이스에서 들어오는 멀티 캐스팅 패킷 모두 이더넷/Wi-Fi 세그먼트로 전달된다. 게이트웨이는 이더넷

/Wi-Fi 측면에서 NDP proxy 서버로 동작하고 NDP 파라미터를 이용하여 무선센서네트워크의 환경을 구성한다. 브릿지 모드는 현존하는 NDP기반의 IPv6 네트워크와 무선센서네트워크가 완벽하게 하나로 통합 될 수 있도록 하며 여러 무선센서네트워크를 하나의 가상 IPv6 서브넷으로 만든다.^[15]

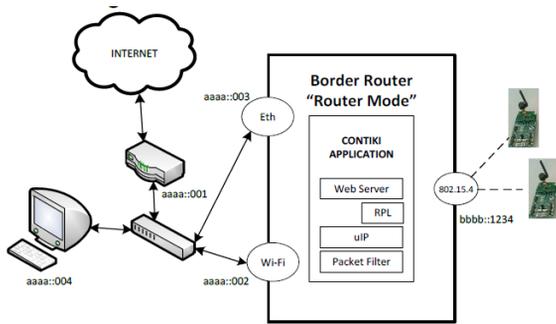


그림 6. 라우터 모드
Fig. 6. Router Mode

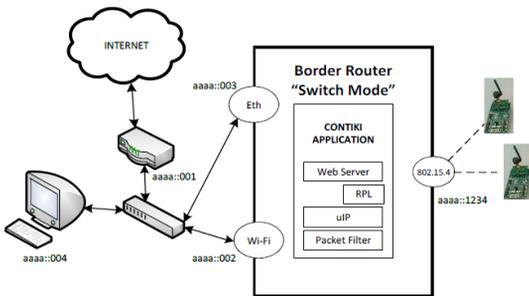


그림 7. 스위치 모드
Fig. 7. Switch Mode

3.4. 어플리케이션 구현

어플리케이션은 IP를 이용하여 사용자가 직접 센서 노드에 접근하여 각각의 노드를 제어 할 수 있도록 한다. 본 테스트베드에서 사용 되는 어플리케이션은 간단한 조작으로 센서노드의 LED를 On/Off 하는 기능을 제공하여 무선센서 네트워크와 인터넷/Wi-Fi의 상호연결을 테스트 한다. 이 어플리케이션은 안드로이드 기반의 스마트폰에서 동작하며, 스마트폰은 언제, 어디서나 인터넷과 Wi-Fi를 연결할 수 있다는 장점을 가진다.

본 시스템의 어플리케이션은 IPv6를 구현하기 위해서 안드로이드의 API인 Inet6Address를 사용하여 구현하였다. 어플리케이션의 처음화면은 그림 8 (a)와 같다. 어플리케이션이 활성화되면 IPv6 Address Autoconfiguration에 의해 IPv6주소를 생성하게 된다. 그림 8 (b)는 센서 노드를 선택 할 수 있는 화면을 보여주고 있다. 각각의 버튼 Device1, Device2

에는 센서노드의 IP주소가 지정 되어있고 버튼을 선택하게 되면 자동적으로 도착지 주소가 해당 주소의 센서노드로 할당된다. 이후 그림 8 (c)(d)와 같이 각 센서노드에 On/Off 명령 메시지를 보냄으로써 무선센서네트워크의 각 노드를 제어 할 수 있다.



그림 8. 어플리케이션 화면 구성
Fig. 8. Application View

IV. 구현 결과 및 분석

앞에서 기술한 바와 같이 본 연구의 시스템은 Contiki OS와 CC2530을 기반으로 6LoWPAN을 지원하는 무선센서네트워크를 구현 하였다. 구현된 모듈은 PC에 Serial to USB로 연결한 후 하이퍼터미널을 사용하여 각 모듈의 동작을 확인하였다. 또한 6LoWPAN의 패킷을 확인하기 위하여 TI CC2531모듈과 PacketSniffer를 통해 AirPacket를 캡처하여 노드간 무선으로 송수신되는 패킷의 정보를 얻을 수 있었다.

본 연구에서 설계, 구현한 테스트 베드를 확인하기 위한 실험은 크게 두 가지로 분류된다. 하나는 무선센서네트워크 내부의 통신이고, 다른 하나는 외부 IPv6인터넷 망과 무선센서네트워크와의 통신이다. 무선센서네트워크 내부 통신을 확인하기 위해서는 PAN Coordinator와 센서노드인 Client노드의 통신을 확인하고, 추가적으로 무선통신의 약점인 제한적인 통신 거리와 이동성을 보완하기 위하여 RPL을 사용하는 라우터를 통한 통신을 확인하였다. 또한 무선센서네트워크와 IPv6 인터넷망과의 연동을 확인하기 위하여 게이트웨이에서 노드로의 ping테스트를 하였고, IPv6인터넷망의 Controller로부터 노드로의 접근과 제어를 테스트 하였다.

무선센서네트워크의 Coordinator는 기존 네트워크의 서버와 같은 역할을 하며 하나의 무선센서네트워크에 반드시 하나만 존재한다. 최초 동작시 RPL을 수행하게되고 RPL의 Root 노드가 없을 경우 자신을 Root로 설정하고 DODAG(Destination Oriented Directed Acyclic Graph)를 형성한다. 무선센서네트워크의 Local 주소는 fe80::/64와 같이 prefix인 fe80에 자신의 MAC주소를 사용하여 주소를 설정한다. 또한 MAC주소와 prefix를 이용하여 외부에서 식별이 가능한 유일한 IPv6 Global 주소를 생성한다. 본 테스트베드에서는 Global주소를 위한 prefix로 aaaa를 사용하였다. 주소 생성 후 주변 노드들에게 알리기 위한 DIO메시지를 브로드캐스트하고 정해진 포트를 통해 응답을 기다린다. 브로드캐스트된 DIO 메시지를 받은 센서노드는 응답(ACK) 메시지를 보내주게 되며 Coordinator를 부모 노드로 등록하여 DAG를 형성한다.

위와 같은 절차가 끝난 후, Coordinator와 센서노드의 연결이 이루어지고 uip_udp_packet_send()함수를 통해 UDP통신을 할 수 있다.

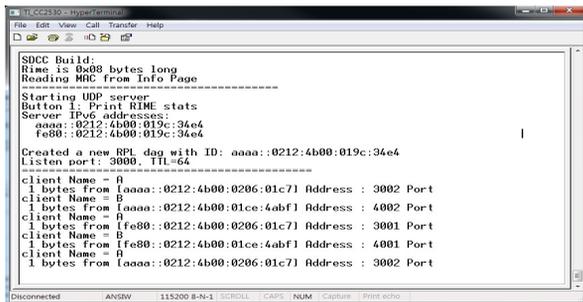


그림 9. Coordinator와 센서노드의 통신테스트 화면 (Coordinator)
Fig. 9. Coordinator and Sensor Node Communication View (Coordinator)

그림 9는 두 개의 센서노드와 통신하는 coordinator노드의 동작화면이다. 3000번 포트를 통해 통신하고 있으며 A와 B, 두 개의 센서노드와 Local주소, Global주소를 통해 송신된 패킷을 정상적으로 수신하고 있는 것을 확인할 수 있다.

센서노드는 Coordinator노드의 주소로 패킷을 송신하고 Coordinator노드로부터 response 메시지를 받게 되는데 이는 센서노드의 동작 화면인 그림 10을 통해 확인할 수 있다.

802.15.4를 통해 무선 통신을 하는 노드들의 패킷확인을 위해 CC2531모듈을 PC에 USB로 연결한 후 TI에서 제공하는 PacketSniffer 프로그램을 이용

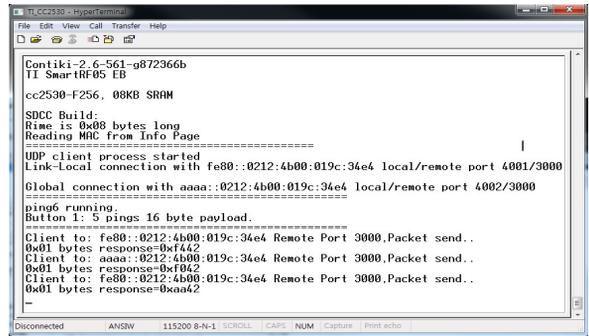


그림 10. Coordinator와 센서노드의 통신테스트 화면 (Sensor Node)
Fig. 10. Coordinator and Sensor Node Communication View (Sensor Node)

하여 AirPacket 캡처를 수행하여, 그림 11과 같이 Coordinator가 FFFF주소 즉, 브로드캐스팅하는 DIO 메시지와 각 노드사이에 송수신되는 패킷의 정보를 확인하였다.

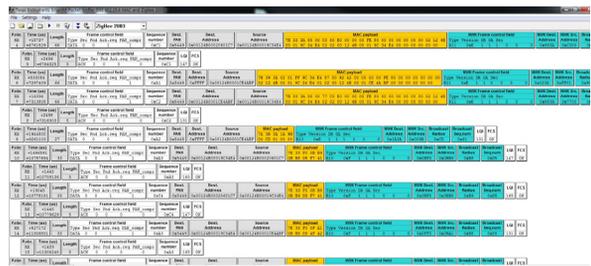


그림 11. 에어패킷 캡처화면
Fig. 11. AirPacket capture view

단일 목적지에 관한 경로들을 구성하는 DODAG(Destination Oriented DAG)를 구성하기 위한 패킷은 DIS(DODAG Information Solicitation), DIO, DAO(DODAG Advertisement Object)의 세 가지 형태로 나누어진다. DIS는 DODAG 정보를 찾기위해 DIO를 요청하는 메시지이고, DIO는 DODAG 정보를 범위내의 노드에게 알려주는 메시지이다. 그리고 DAO 메시지는 센서노드가 Coordinator노드에게 연결되는 경로를 생성하기 위한 메시지이다.

Dest. Address	Source Address	MAC payload
0xFFFF	0x00124B0001CE4AE7	7A 3B 3A 1A 9B 00 CD 59 00 00

그림 12. DIS 메시지
Fig. 12. DIS Message

그림 12와 같이 DIS메시지는 MAC payload에

위와 같은 환경을 구현 및 테스트 한 후, 본 테스트베드에서는 안드로이드 기반의 스마트폰 어플리케이션을 이용해 IP주소로 각 노드에 접근하고 패킷을 송신하여 LED를 제어하는 실험을 하였다. 실험 결과 (그림 19)과 같이 무선센서네트워크 내의 센서노드가 IPv6인터넷망의 Controller로부터 LED를 On/Off 하는 제어 메시지를 수신함을 확인할 수 있다.

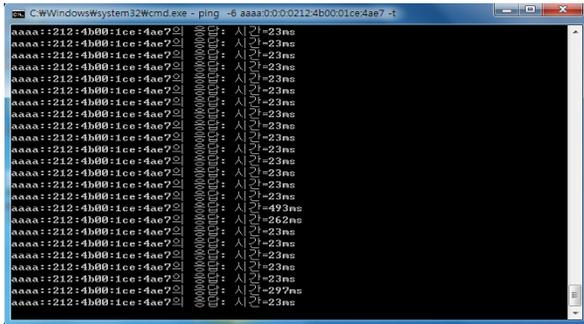


그림 17. Gateway에서 센서노드로 ping6명령 전달 화면
Fig. 17. Gateway to SensorNode Ping6 test view

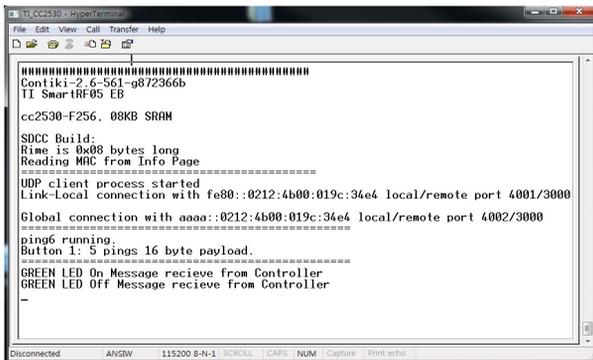


그림 18. Controller에서 센서노드로의 On/Off 명령 전달 결과 화면
Fig. 18. Controller to SensorNode On/Off Control view

실험결과 본 연구에서 구현된 시스템에서 IPv6 인터넷망과 6LoWPAN기반의 무선센서네트워크 망이 서로 연동될 수 있도록 정상적으로 구축되었음을 알 수 있다.

앞서 서론에서 언급했던 것과 같이 무선센서네트워크의 핵심적인 문제는 전력공급이다. 각 노드는 소비전력을 최소화 하여 동작하고 자체 전원으로 오랜 기간 동작해야만 한다. 따라서 전력 관리가 매우 중요하다. 본 시스템의 게이트웨이는 USB를 통해 항상 연결되어 있기 때문에 배터리를 필요로 하지 않는다. 그러나 센서노드의 경우 배터리를 사용한다. 본 연구에서 디바이스의 에너지 소비량 측정

결과 10Ω의 저항으로 직렬 연결하였을 때 I는 30mA보다 낮은 수치를 보였다. 하지만 6LoWPAN 장치의 에너지 소비량은 입력전압과 독립적이다. 전류가 결정되면 평균 전류 소비는 일반 식 (1)을 사용하여 계산 할 수 있다.

$$I_{avg} = \sum_{i=0}^n \sum_{i=0}^n \left(\frac{I_i}{P_i}\right) * I_i + \left(1 - \sum_{i=0}^n \left(\frac{T_i}{P_i}\right)\right) * I_{sleep} \quad (1)$$

- Ti = 장치가 평균 전류 Ii를 소비하는 시간
- Pi = 평균 소비를 측정된 총 시간
- Isleep = Sleep모드에서의 소비량
- Iavg = Pi 시간동안의 평균 소비 전류

Ii, Isleep, Ti를 알면 동작 시간을 기준으로 Iavg를 알 수 있다. 또한 평균 소비전력을 기준으로 식 (2)를 통해 6LoWPAN장치의 예상 수명을 계산 할 수 있다.

$$\frac{Battery\ Capacity [mAh]}{Average\ Current [mA]} = Lifetime [h] \quad (2)$$

6LoWPAN 노드에 사용되는 배터리 용량은 배터리 타입에 따라 다르다. 본 연구에서는 두 개의 AA 듀라셀 디럭스 건전지를 사용 하였다. 이 배터리의 사양은 표 3과 같다. 에너지 소비와 6LoWPAN 노드의 입력 전압은 주로 응용프로그램과 사용하는 센서에 따라 달라진다. 소규모의 작은 네트워크의 노드는 292.5ms 동안 평균 34.6mA를 소비한다. 전류계를 사용하여 현재 시스템의 Sleep 상태 전류 측정 결과 4.8μA 였고 측정된 값을 기준으로 일반 식 (1)에 대입하여 장치의 전체 평균 소비전력을 구할 수 있다. 5초(=5000ms)의 패킷 간격을 갖는 다고 할 경우 소비 전력은 다음과 같다.

$$\left(\frac{292.5}{5000} * 34.6\right) + \left(1 - \frac{292.5}{5000}\right) * 0.0048mA = 2.212mA$$

계산된 소비전력을 바탕으로 계산한 현재 센서노드의 예상 수명은 다음과 같다.

$$\frac{2850mAh}{2.212mA} = 1288hrs = 53days$$

표 3. AA 듀라셀 디럭스 배터리 사양
Table 3. Deluxe Duracell AA Battery Specifications

Max Charge Voltage	1.5 V
Nominal Voltage	1.2 V
Nominal Capacity	2850 mAh
Standard Charge	270mA/16 h
Fast Charge	2700mA/ 1.1 h

만약 센서노드가 데이터 폴링을 사용하지 않는 소규모의 어플리케이션을 사용하고 매 5초 간격으로 패킷을 전송하도록 설정이 되어 있을 경우 보드는 두 개의 AA 듀라셀 디럭스 건전지를 사용하여 53일 동안 동작 할 수 있다. 따라서 본 연구에서 구현되어진 시스템은 기존에 연구 되었던 시스템¹⁶보다 오랜 시간 동작하지만 아직도 주기적인 배터리 교체가 필요없는 자체 전력 공급을 위한 대안이 필요로 한다.

그 대안으로 최근 몇 년 동안 에너지 하베스팅은 주로 센서 네트워크에 무선 센서장치의 에너지 자원을 보충하기 위해 적용되어왔다^{16,17}. 기존의 연구에서 에너지 하베스팅을 이용하여 6LoWPAN 기반 온도 모니터링 노드에 이 기술을 적용하여 시스템을 에너지 독립적으로 구현한 사례가 있으나^{16,18} 노드가 데이터를 송신하는 동안 노드에 필요한 세 부적인 에너지 소비량을 검토하지 않았다.

본 연구에서는 구현된 시스템의 전체 평균 소비 전력을 오실로스코프를 통해 측정하였고 그 결과는 표 4와 같다.

무선센서 네트워크 시스템이 자체적으로 전력을 공급하는 전원 독립적인 시스템이 되기 위해서는 다음과 같은 조건을 만족해야 한다. t 시간동안 각 가의 노드가 에너지 하베스팅으로 얻는 전체 에너지의 양은 $P_s(t)$, 같은 시간동안 에너지 소비량은 $P_c(t)$ 일 때 여분의 에너지 $E(t)$ 는 식 (3)을 통해 추정 할 수 있다.

표 4. 6LoWPAN 기반 센서노드 소비 전력
Table 4. 6LoWPAN based sensor node power consumption

Event	Description	Duration [ms]	Current [mA]
1	Waking Up	45	0.68
2	Processing data packet	25.6	30.2
3	Transmit packet and receive ACK	16.6	78.6

4	Request and receive ACK	21	98.6
5	Post processing packet	18	29.4
6	Request Data (Single Poll)	29	94.3
7	Prepare to Sleep	6	26.4
8	Set up radio	4.5	24.2
9	Start CSMA-CA	5.2	90.8
10	Switch from RX to TX	3	64.4
11	Switch from TX to RX	2.9	62.3
12	Prepare for deep Sleep	2.1	24.2

$$E(t) = \int_0^t (P_s(t) - P_c(t))dt \quad (3)$$

$$E(t) > 0, \forall t > 0 \quad (4)$$

여분의 에너지가 식 (4)를 만족하는 경우 노드는 에너지 독립적인 것으로 간주된다.

6LoWPAN 노드는 에너지 하베스팅을 통해 에너지를 공급함으로써 배터리의 수명을 극대화 시켜주고 주기적으로 배터리를 교체 할 필요없이 장시간 동작 할 수 있다. 동일 시간을 기준으로 가장 높은 효율을 제공하는 기술은 태양광을 이용한 에너지 하베스팅 기술이다. 태양광 발전을 통해 얻을 수 있는 전력량이 무선센서 네트워크의 센서 노드가 사용하는 전력을 충족한다면 배터리 교체 없이 무한한 수명을 갖는 무선센서네트워크를 구축¹⁹할 수 있을 것이다.

V. 결 론

본 논문에서는 6LoWPAN기반의 무선센서네트워크와 IPv6 기반의 인터넷망을 연결하는 게이트웨이를 제안하고, IP-USN 테스트베드를 구현하였다. 또한 구현된 시스템을 바탕으로 소비전력을 분석하여 전원관리 문제에 대한 해결 방안을 모색 하였다. 본 시스템은 저전력, 저손실, 저가격을 만족하는 TI CC2530을 사용하였기 때문에 광범위한 무선센서네트워크를 구성하는데 효율적일뿐만 아니라, 오픈소스인 Contiki OS를 사용함으로써 무선센서네트워크를 구성하고 확장함에 있어서 보다 효과적인 환경을 제공한다. 본 논문에서 구현한 테스트베드는 이를 실생활에 적용하거나 연구에 활용할 수 있는 기본 플랫폼이 될 것 이라고 기대한다. 또한 추후에는 본 논문에서 제안하는 에너지 하베스팅을 이용한 자체적으로 전원을 공급하는 무선센서네트워크를 구축하고 시스템 구조를 확장하여 다양한 프로토콜과

서비스를 연구할 테스트베드로 활용할 예정이며 무선 센서네트워크의 주요 이슈인 소모전력 최소화, 보안, 효율적인 라우팅에 관한 연구를 수행할 계획이다.

References

- [1] TinyOS, *TinyOS main webpage with information and tutorials*, Retrieved Oct., 21, 2012, from <http://www.tinyos.net>.
- [2] A. Dunken, N. Finne, J. Eriksson, and T. Voigt. "Run-time dynamic linking for reprogramming wireless sensor networks," in *Proc. Embedded Networked Sensor Syst. (SenSys '06)*, pp. 15-28, Boulder, U.S.A., Nov. 2006.
- [3] L. Gu and J. A. Stankovic, "t-kernel: providing reliable OS support for wireless sensor networks," in *Proc. Embedded Networked Sensor Syst. (SenSys '06)*, pp. 1-14, Boulder, U.S.A., Nov. 2006.
- [4] S. Bhatti, J. Carlson, H. Dai, J. Deng, J. Rose, A. Sheth, B. Shucker, C. Gruenwald, A. Torgerson, and R. Han, "Mantis OS: an embedded multithreaded operating system for wireless micro sensor platforms," *ACM Mobile Networks Applicat. (MONET)*, vol. 10, no. 4, pp. 563-579, Aug. 2005.
- [5] C.-C. Han, R. Kumar, R. Shea, E. Kohler, and M. Srivastava, "A dynamic operating system for sensor nodes," in *Proc. Mobile Syst., Applicat., Services (MobiSys '05)*, pp. 163-176, Seattle, U.S.A., June 2005.
- [6] NanoQplus, *NanoQplus*, Retrieved Dec., 8, 2012, from <https://sites.google.com/site/nanoqplusos>.
- [7] Wikipedia, *Contiki OS*, Retrieved Aug., 13, 2012, from <http://en.wikipedia.org/wiki/Contiki>.
- [8] ZigBee Alliance, *ZigBee Alliance*, Retrieved Aug., 26, 2012, from <http://www.zigbee.org>.
- [9] A. P. Escola, "Development of a wireless sensor network with 6LoWPAN support," M. S. Thesis, Dept. Sci. Telecommun. Eng. Manage., University Politecnica De Catalunya, Spain, July 2009.
- [10] K. Kim, J. Seo, D.-J. Hwang, and H. Choo, "Implementation of 6LoWPAN testbed: location tracking service based on Google map," *J. Korean Soc. Internet Inform. (KSII)*, vol. 10, no. 5, pp. 13-26, Oct. 2009.
- [11] Texas Instruments, *CC2530*, Retrieved Sep., 3, 2012, from <http://www.ti.com/product/cc2530>.
- [12] Contiki, *Contiki OS The Open Source OS for the Internet of Things*, Retrieved Oct., 15, 2012, from <http://www.contiki-os.org>.
- [13] E. S. Kim and Y. W. Kim, "6LoWPAN based IP-USN Standardization," *Electron. Telecommun. Trends*, vol. 22, no. 6, pp. 24-32, Dec. 2007.
- [14] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, *Transmission of IPv6 packets over IEEE 802.15.4 networks*, IETF RFC 4944, Sep. 2007.
- [15] A. M. Efendi, A. F. P. Negara, S. Oh, and D. Choi, "A Design of 6LoWPAN RPL border router with multi-uplink interface: ethernet and Wi-Fi," in *Proc. Int. Conf. Internet Services Technol. Inform. Eng. (ISTIE '13)*, pp. 55-60, Bogor, Indonesia, May 2013.
- [16] V. C. Lee, "Energy harvesting for wireless sensor networks," M. S. Thesis, Dept. Elect. Eng. Comput. Sci., Univ. California, Berkeley, U.S.A., Spring 2009.
- [17] R. J. M. Vullers, R. V. Schaijk, H. J. Visser, J. Penders, and C. V. Hoof, "Energy harvesting for autonomous wireless sensor networks," *IEEE Solid-state Circuits Mag.*, vol. 2, no. 2, pp. 29-38, Spring 2010.
- [18] D. S. Tudose, A. Voinescu, M. Petreanu, A. Bucur, D. Loghin, A. Bostan, and N. Tapus, "Home automation design using 6LoWPAN wireless sensor networks," in *Proc. 2011 Int. Conf. Distributed Computing Sensor Syst. Workshops (DCOSS)*, pp. 1-6, Barcelona, Spain, June 2011.
- [19] A. M. Efendi, S. Oh, A. F. P. Negara, and D. Choi, "Battery-less 6LoWPAN-based wireless home automation by use of energy harvesting," *Int. J. Distributed Sensor Networks*, vol. 2013, Article ID 924576, 8 pages, June 2013.

오 승 교 (Seung-kyo Oh)



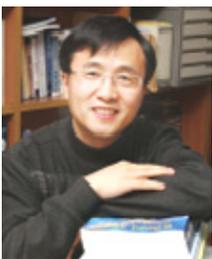
2012년 2월 전남대학교 전자
컴퓨터공학부 졸업
2012년~현재 전남대학교 전자
컴퓨터공학부 석사
<관심분야> IPv6, 네트워크 관
리, 무선통신, Sensor
Network

Ardiansyah Musa Efendi



2012년~현재 전남대학교 전
자컴퓨터공학부 석사
<관심분야> 네트워크, Sensor
Network

최 덕 재 (Deok-Jai Choi)



1982년 서울대학교 컴퓨터공학
과졸업
1984년 KAIST 전산학과 석사
1995년 University of Miss
ouri-Kansas City Computer
Science and Telecommunication
Program 박사

1996년~현재 전남대학교 전자컴퓨터공학과 교수
<관심분야> 상황인식, Pervasive Computing, Sensor
Network, Future Internet, IPv6