

센서 네트워크를 위한 초경량 TCP/IP_{v6} 프로토콜의 설계 및 구현

김신재¹, 김영균¹, 이완직¹, 허석렬¹, 신범주^{1*}
¹부산대학교 바이오메디컬공학과

Design and Implementation of Tiny TCP/IP_{v6} Protocol for Wireless Sensor Networks

Shin-Jae Kim¹, Young-Gyun Kim¹, Wan-Jik Lee¹, Seok Yeol Heo¹
and Bum Joo Shin^{1*}

¹Department of BioMedical Engineering, Pusan National University

요 약 유비쿼터스 세상을 실현하기 위한 핵심 기술로 최근 무선 센서 네트워크에 관한 연구가 활발하게 이루어지고 있다. 그러나 유비쿼터스 환경에서 요구되는 가장 근본적인 요구사항 중의 하나가 센서 네트워크와 인터넷의 상호 동작임에도 불구하고, 대다수의 연구 결과들은 센서 네트워크 자체의 기술에만 집중되어 있다. 본 논문에서는 센서 네트워크 노드를 IPv6 인터넷에 연결할 수 있게 하는 초경량 TCP/IP_{v6} 프로파일을 설계하였다. 설계된 프로파일의 동작을 확인하기 위하여 TinyOS 상에 구현하고, 성능을 시험하였다. 성능 시험 결과는 본 논문의 초경량 TCP/IP_{v6}의 전송률이 TinyOS의 컴포넌트를 이용한 전송률에 비해 크게 저하되지 않음을 나타내었다.

Abstract As a core technology realizing ubiquitous world, recent researches are being concentrated to wireless sensor network. However, most research results were focused to the sensor network technology itself, even though interworking between the sensor network and Internet is also one of primitive requirements for ubiquitous world. In this paper, we design the tiny TCP/IP_{v6} profile which makes it possible to inter-connect the sensor network device to IPv6 based Internet. To confirm operation of the designed profile, we experimentally implemented and evaluated minimum TCP/IP_{v6} based on TinyOS. The evaluation result shows that throughput of our tiny TCP/IP_{v6} is almost same as that of TinyOS component.

Key Words : USN, WSN, IPv6, TCP/IP

1. 서론

IPv6 기반의 BcN(Broadband Convergence Network)과 무선 센서 네트워크를 연동하기 위한 방법의 하나로 센서 네트워크 노드에 IPv6 프로토콜을 탑재하는 기술이 주목받고 있다. TCP/IP는 통신장애에 대한 강건함과 공개된 표준이라는 장점을 가지고 있다. 특히 IPv6의 128 비트 주소 지원, 주소 자동생성과 같은 특성은 센서 네트워크의 요구사항에 잘 부합되기 때문에 IPv6는 센서 네트워크를 위한 통신 프로토콜로서 아주 적합하다. 또한 센서 네트워크 노드에 TCP/IP_{v6}를 탑재함으로써 인터넷

환경과 쉽게 연동이 가능하며, 인터넷 환경에서 널리 사용되는 응용들을 센서 네트워크에서 쉽게 사용할 수 있다[1-3].

그러나 TCP/IP_{v6}는 안정적으로 전원이 공급되는 환경에서 고성능 연산 능력을 갖는 시스템을 대상으로 개발되었기 때문에 수정을 하지 않고 센서 네트워크에 그대로 적용하기에는 센서 노드의 낮은 성능과 적은 메모리 용량, 낮은 I/O 대역 그리고 전원 공급 제약 등과 같은 많은 문제점을 가지고 있다. 이를 해결하기 위해서는 호스트 기반의 TCP/IP_{v6} 프로토콜을 센서 네트워크 환경에 맞게 수정하고 최적화하는 작업이 필요하다[4,5].

*교신저자 : 신범주(bjshin@pusan.ac.kr)

접수일 09년 05월 25일

수정일 09년 07월 08일

게재확정일 09년 08월 19일

본 논문에서는 센서 네트워크 환경에 적합한 초경량급 IPv6, ICMPv6, TCP 프로토콜 기능을 설계하고, 이를 센서 네트워크의 대표적인 운영체제인 TinyOS 환경에서 구현하였다. 구현된 프로토콜들은 센서 노드에 직접 탑재하여 프로토콜의 성능 측정을 수행하였다.

본 논문의 구성은 다음과 같다. 2장에서는 기존 연구와 본 논문의 연구에 대해 비교한다. 3장에서는 본 논문의 프로토콜이 수행될 센서 네트워크 및 센서 노드 환경과 초경량 TCP/IPv6 기능 프로파일에 대해 정리한다. 4장에서는 초경량 TCP/IPv6 프로토콜 기능 설계와 구현한 내용을 기술한다. 마지막으로 5장에서 결론과 향후 계획을 다룬다.

2. 관련 연구

2.1 6LoWPAN (IPv6 over LoWPAN)

IETF의 6LoWPAN 워킹 그룹은 센서 네트워크의 표준 PHY/MAC 계층인 IEEE 802.15.4 상에서 효과적인 IPv6 패킷 전송 방안을 정의하는 것을 목표로 한다. 본 워킹 그룹은 MTU 크기가 작은 IEEE 802.15.4 상에서 IPv6 패킷을 효율적으로 전송하기 위해 IPv6 헤더 압축/해제, 패킷 단편화 및 재조립 기능에 대해 연구를 수행하였다 [1,2].

전력 소모, 메모리 사용 등이 극히 제한된 센서 네트워크 특성에 따라 IPv6 관련 프로토콜들의 기능 최적화 및 경량화 연구도 매우 필요한 분야이기 때문에 현재 6LoWPAN에서 이와 관련된 연구를 진행하고 있다.

2.2 경량급 TCP/IP 프로토콜 스택

TCP/IP 프로토콜에 관한 대표적 경량급 프로토콜로 uIP, lwIP, TinyIPv6가 있다. uIP와 lwIP는 Swedish Institute of Computer Science에서 개발하였으며, IPv4를 지원한다. uIP는 8 비트 마이크로프로세서에서 동작할 수 있도록 개발된 반면, lwIP는 임베디드 시스템을 위한 범용적인 용도로 설계되었다[6]. uIP는 자원 사용량의 최소화애 초점을 맞춘 반면, lwIP는 프로토콜의 성능을 고려하여 자원의 희생을 감수한다는 차이가 있다[7].

TinyIPv6는 일본의 Non-PC Network Appliance Committee에서 일반적인 AV 기기나 가전제품들의 인터넷 접속을 목표로 개발한 초경량 프로토콜로 IPv6, ICMPv6, UDP가 지원된다[8]. 그러나 이 프로토콜은 IPv6의 이웃 노드 탐색(Neighbor Discovery), 주소 자동설정 등을 위한 브로드캐스팅 패킷 전송이 빈번하게 발생

하기 때문에 저전력 소모가 필수적인 센서 네트워크의 환경에는 적합하지 않다.

3. 기능 프로파일 설계

3.1 동작 환경

센서 네트워크는 동작하는 환경이나 동작 목적에 따라서 특성이 많이 다르기 때문에 센서 네트워크를 위한 프로파일을 설계 하는데 있어서 대상이 되는 모델이나 환경을 구체적으로 명시할 필요가 있다[9,10]. 본 논문에서는 센서 네트워크와 센서 노드에 대하여 아래와 같은 환경을 기반으로 하는 프로파일을 설계하였다.

3.1.1 센서 네트워크 환경

PHY/MAC은 IEEE 802.15.4 표준을 따른다.

802.15.4에서 정의한 Multi-Hop 전송을 지원하며, 이를 위한 계층 2 멀티 홉 라우팅 기능이 지원된다. 각 노드의 MAC 주소는 EUI-64 방식의 48 비트 long 주소 또는 중복 없는 16 비트 short 주소를 사용한다.

하나의 PAN ID로 정의되는 PAN에는 오직 하나의 IPv6 게이트웨이만 존재한다.

3.1.2 센서 노드의 최소 사양

센서 네트워크를 구성하는 각각의 센서 노드들은 다음과 같은 최소 사양을 만족하는 것으로 한다.

8 비트 RISC 방식 CPU

4 KByte 이상의 RAM

50 KByte 이상의 프로그램 메모리

IEEE 802.15.4의 단일 통신 인터페이스

3.2 초경량 TCP/IPv6 기능 프로파일

본 논문에서 설계한 센서 노드를 위한 초경량 TCP/IPv6 기능 프로파일은 센서노드의 제한적 자원을 고려하여 TCP/IPv6 프로토콜 규격 가운데 지원하지 않더라도 기본적인 TCP/IPv6 통신에 영향을 미치지 않는 기능만을 명시한다. 따라서 본 절에서 언급되지 않은 사항들은 일반 외부 호스트와 종단간 프로토콜 호환성을 보장하기 위해 모두 필요한 기능들이다. 이와 관련한 상세한 내용은 [11,12]에서 참조할 수 있다.

3.2.1 센서 네트워크 환경에 대한 IPv6 최적화

1) 기본사항

IPsec을 위한 기능은 모두 지원하지 않는다. IPsec은 기본적으로 충분한 스택키를 가지면서 대역폭과 비용에 구애 받지 않는 환경을 대상으로 설계되었기 때문에 센서네트워크에는 부적합하다.

라우팅 테이블 관리를 위한 기능은 지원하지 않는다. 일반 센서 노드는 라우터로 동작하지 않고 단말 노드로만 동작하기 때문에 디폴트 라우터 리스트 정도의 간단한 정보만 관리하면 된다.

다중 인터페이스 지원을 위한 기능은 지원하지 않는다. 센서 노드는 IEEE 802.15.4를 지원하는 단일 통신 인터페이스를 가지기 때문에 다중 인터페이스 기능은 필요하지 않다.

Mobile IPv6 관련 기능은 지원하지 않는다. MIPv6의 올바른 동작을 위해서는 키 분배 및 관리, IPsec과 같은 많은 오버헤더를 필요로 하기 때문에 현재의 센서 네트워크 환경에서는 고려하기 어렵다.

Path MTU 탐색 기능은 지원하지 않는다. Path MTU 탐색 기능은 IPv6 노드가 1280 옥텟 이상의 패킷을 전송하기 위해서 필요한 기능이며, 항상 1280 옥텟 이하로만 보낼 경우에는 구현할 필요가 없다. 센서 네트워크에서 1280 옥텟 보다 큰 사이즈의 패킷을 송수신 하는 것은 비현실적이다.

2) IPv6 주소 처리

하나의 노드는 하나의 link local 주소와 하나의 Global Unicast 주소만 지원한다.

Global unicast 주소는 stateless address auto-configuration 방식에 의해 생성한다.

All-Nodes-Multicast, All-Routers-Multicast 이외의 멀티캐스트 주소는 사용하지 않는다.

Anycast 주소관련 기능은 지원하지 않는다.

3.2.2 센서 네트워크 환경에 대한 ICMPv6 최적화

1) Error Reporting 기능

센서 노드의 동작과 연결 여부를 점검하기 위하여 Destination Unreachable 메시지 생성만을 지원한다.

2) Query 기능

echo request에 대한 응답을 지원하기 위한 echo 서버 기능 이외에는 지원하지 않는다.

3.2.3 센서 네트워크 환경에 대한 ND 최적화

1) Router Discovery 관련 기능

센서 노드들은 IPv6 라우터로 동작하지 않기 때문

에 RS 메시지 수신 및 RA 송신에 따른 처리가 필요하지 않다.

2) Redirect 관련 기능

일반 센서들은 IPv6 라우터 기능을 수행하지 않기 때문에 Redirect 패킷 생성 및 수신 처리를 위한 기능은 지원하지 않는다.

3) Address Autoconfiguration 관련 기능

RA 송신 기능은 수행하지 않는다.

EUI-64방식의 48비트 물리 주소를 사용하기 때문에 Stateless address auto-configuration 동작 시에 DAD 기능은 수행하지 않는다.

Default Router 리스트 및 Prefix 리스트 관리를 수행하지 않는다.

4) Neighbor Discovery 관련 기능

EUI-64 주소 및 중복되지 않는 16비트 주소를 IPv6 주소에서 획득할 수 있으므로 NS와 NA 송신 및 수신 기능은 수행하지 않는다.

센서노드는 address resolution을 위한 NS 송신을 하지 않기 때문에 Neighbor Cache Entry 관련 기능은 필요하지 않다.

3.2.4 센서 네트워크 환경에 대한 TCP 최적화

1) TCP Options 관련

외부 인터넷의 호스트와 연동을 위한 MSS(Maximum Segment Size) 협상의 다른 옵션은 지원하지 않는다.

2) PUSH

지원하지 않는다. 다만 데이터를 무한 버퍼링하지 않도록 하며, 마지막 세그먼트에는 PSH 비트를 1로 설정한다.

3) Urgent Data Delivery

지원하지 않는다.

4) Window Management

흐름제어를 위한 기본적인 윈도우 관리는 수행한다.

Zero-Window에 대한 Probing 기능은 지원한다.

송신 SWS(Silly Window Syndrome) Avoidance: Nagle 알고리즘 방식에 의해 지원한다.

수신 SWS Avoidance는 지원하지 않는다.

5) Connection Management

Clock 기반의 ISN(Initial Sequence Number) 설정: 지원한다.

Simultaneous OPEN: 지원하지 않는다.

비정상적인(멀티캐스트 또는 브로드캐스트) 원격지 주소에 대한 Local-OPEN 거부 및 비정상적인 원격지로 부터의 SYN Segment 거부: 지원한다.

RST 세그먼트 내의 데이터 수신 : 지원하지 않는다.
연결 해제 시에 2*MSL 기간 동안 TIME-WAIT 상태 보존: 지원하지 않는다.

6) Error Control

Checksum: 송수신에 대한 Checksum 기능을 모두 지원한다.

RTO(Retransmission Time Out) 계산: 지원하지 않으며, 이미 정해진 상수 값을 대신 사용한다.

Out-of-order 순서 번호를 가진 세그먼트 수신 : 지원하지 않는다.

Fast Recovery 기능(동일한 순서 번호를 가진 3개의 ACK 수신시에 재전송 수행) : 지원하지 않는다.

7) Congestion Control

Jacobson's Slow Start : 지원하지 않는다.

Jacobson's Congestion Avoidance : 지원하지 않는다.

8) Connection Failure

임계치 값, R1 이상의 재전송 발생 시에는 Dead-Gateway-Diagnosis를 위해 이를 IP 계층에 통보: 지원하지 않는다.

임계치 값, R2 (> R1) 이상의 재전송 발생 시에 Connection 해제: 지원하지 않는다.

9) Keep Alive

지원하지 않는다.

4. 초경량 TCP/IPV6 구현

본 장에서는 앞 장에서 기술한 초경량 TCP/IPV6 프로토콜 프로파일 규격을 이용한 프로토콜 스택을 TinyOS 상에서 수행될 수 있도록 설계 및 구현한 내용에 대해 기술한다.

4.1 구현 환경

IPv6와 같은 프로토콜 스택이 소형, 저가의 하드웨어

플랫폼에서 효율적으로 동작하기 위해서는 플랫폼의 프로세서, 메모리 용량 등의 하드웨어 사양을 고려한 구현이 필요하며, 또한 프로토콜의 구현 방식은 운영체제와 밀접한 관계를 가지므로 프로토콜 구현 구조, 구현 알고리즘 등도 운영체제가 제공하는 방식대로 구성되어야 한다. 본 논문에서는 TelosB 플랫폼과 TinyOS 운영체제와 컴포넌트 기반의 nesC를 구현 환경으로 채택하였다 [13],[14]. 이러한 Telos 계열 플랫폼은 TinyOS의 대표적인 하드웨어 플랫폼이라고 불릴만큼 TinyOS에 적합한 구조로 설계되어 있다. TelosB는 IEEE 802.15.4를 지원하며, 250 Kbps의 전송율과 10 KB RAM을 갖는 TI사의 MSP430 프로세서를 채용한 보드이다.

4.2 프로토콜 설계

4.2.1 프로토콜 계층

본 논문의 초경량 TCP/IPV6 프로토콜의 계층 구성은 표 1과 같다.

[표 1] 초경량 TCP/IPV6 프로토콜 계층 구성

계층	지원 프로토콜
Transport	TCP
IP	IPv6, ICMPv6
L2 Routing	Flooding/TinyAODV
Data-Link/PHY	TinyOS AM (IEEE 802.15.4)

표 1에서 보는 바와 같이 초경량 TCP/IPV6 프로토콜은 IPv6 기반 프로토콜로서 TCP를 지원하고 있으며, L2 라우팅 계층은 센서 네트워크에서 멀티 홉 지원을 위해 제공되는 기능이 이식된 멀티 홉 라우팅 프로토콜들로 구성된다. 센서 네트워크에서 멀티 홉 라우팅은 아직 표준으로 정해진 것이 없기 때문에 특정 라우팅 프로토콜을 지정하지 않았다. 또한 센서 네트워크의 응용이나 네트워크 환경에 따라 멀티 홉 라우팅 프로토콜을 선택해서 사용하는 것이 더욱 효율적이라고 판단했기 때문이기도 하다.

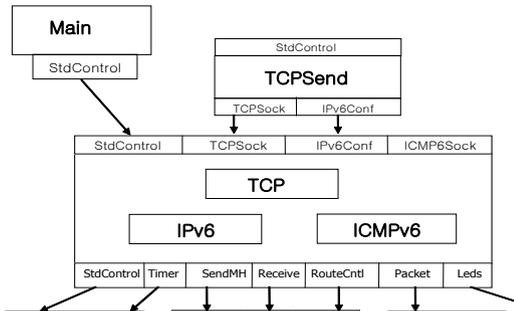
예를 들어 센서의 개수와 최대 홉 거리가 크지 않으면 간단한 플러딩 프로토콜을 사용하는 것도 좋은 선택이 될 수도 있다. TinyAODV는 기존의 AODV(Ad-hoc On Demand Vector) 프로토콜을 간소화 시켜 TinyOS에서 동작 시킬 수 있도록 구현한 On-Demand 방식의 멀티 홉 라우팅 프로토콜이다.

따라서 본 논문의 프로토콜에서는 두 가지의 멀티 홉 라우팅 프로토콜을 지원하며 이들 중 하나를 선택해서 사용할 수 있도록 하였다.

링크 계층에서는 기본적으로 IEEE 802.15.4 MAC 방식을 사용한다. 그렇지만 TinyOS MAC 프레임 구성은 IEEE 802.15.4 프레임과 조금 다른 TinyOS 특유의 AM(Active Message) 기반의 링크 프레임 방식을 사용한다. TinyOS의 AM 방식은 IEEE 802.15.4 MAC 프로토콜 프레임에 TinyOS에서 사용하는 2 바이트 형태의 센서 노드 주소, TinyOS에서 정의된 여러 형태의 통신용 메시지 Type, 센서 노드들의 그룹 ID 등을 나타내는 필드를 추가해서 만든 TinyOS 전용 링크 계층 프로토콜 프레임으로 볼 수 있다.

4.3 구현 및 시험

앞에서 기술한 바와 같이 초경량 TCP/IPv6 프로토콜은 TinyOS를 기반으로 하여 nesC 프로그래밍 방식에 의해 구현되었다. 따라서 구현된 프로토콜 모듈도 하나의 컴포넌트로 구성되며, 이 프로토콜을 이용하는 상위 컴포넌트에게 인터페이스를 제공한다. 또한 구현된 프로토콜 모듈도 자신이 필요로 하는 인터페이스를 제공하는 다른 컴포넌트들을 사용하게 된다. 그림 1의 Configuration 예는 본 논문의 프로토콜 상에서 TCPSend 응용 프로그램을 동작시켰을 때의 컴포넌트들의 구성도이다. TCPSend는 소켓에서 send()를 이용해서 데이터를 전송하는 것과 동일한 기능을 하는 간단한 프로그램이다. 음영으로 표시된 컴포넌트는 본 논문에서 직접 구현하거나 일부 수정한 컴포넌트이며 음영으로 표시되지 않은 컴포넌트는 TinyOS 자체에서 제공하는 컴포넌트이다.



[그림 1] 초경량 TCP/IPv6 컴포넌트 Configuration 구성

그림 1을 보면, 실제 초경량 TCP/IPv6 구성 프로토콜들(TCP, IPv6, ICMPv6)이 하나의 컴포넌트로 구성됨을 알 수 있다. 이렇게 통합 구현한 이유는 계층 간의 호출 등에 의해 발생하는 오버헤드를 줄이고, 프로토콜 동작에 소요되는 메모리 사용도 많이 줄일 수 있기 때문이다. 멀티 홉 라우팅을 담당하는 L2 라우터 기능은 응용의 목적과 센서 네트워크 환경에 따라 옵션으로 선택할 수 있고

록 별도의 컴포넌트로 구성하였다.

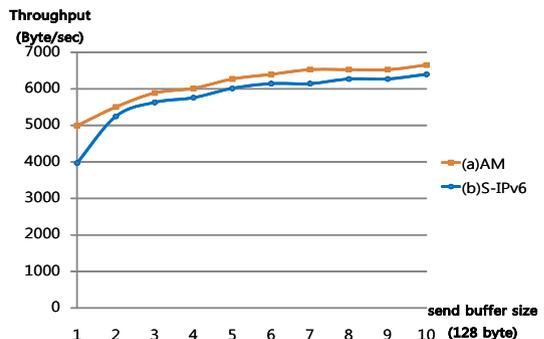
그림 2는 구현된 초경량 TCP/IPv6 프로토콜을 센서 노드에 탑재하여 테스트한 환경이다. 두 개의 센서 노드를 사용하여 원 홉으로 서버/클라이언트 형태로 구성하였으며, 그 중 하나의 노드를 PC와 연결하여 결과를 확인하였다.



[그림 2] 초경량TCP/IPv6 프로토콜 테스트 환경

본 논문에서 구현한 초경량 TCP/IPv6의 성능을 테스트하기 위하여 TinyOS의 기본 송수신 컴포넌트와 전송률을 비교 측정하였다. TinyOS의 기본 송수신 컴포넌트는 TinyOS의 MAC(TOS_Msg) 상위에 직접 응용 컴포넌트(필요에 따라서 L2Route경유)가 동작하는 구조를 가지며, 본 논문의 초경량 TCP/IPv6 프로토콜은 TinyOS의 MAC 상위에 탑재된다.

TinyOS의 기본 송수신은 TinyOS에서 사용되는 메시지 형식인 AM(Active Message)를 사용하였으며 초경량 TCP/IPv6 프로토콜은 TCPSend라는 별도의 응용 프로그램을 사용하였다. 그림 3은 AM을 사용한 TinyOS의 기본 송수신과 본 논문의 초소형 TCP/IPv6 프로토콜의 전송률 측정 결과를 보여 준다.



[그림 3] 초경량 TCP/IPv6 전송률 성능 측정

그림 3에서 (a)는 초경량 TCP/IPv6를 탑재하지 않고 TinyOS에서 제공하는 송수신 컴포넌트를 사용했을 때의 전송률을 나타내며 그림 3에서 (b)는 센서 노드에 초경량 TCP/IPv6 프로토콜을 탑재하여 전송률을 측정된 결과를 보여주고 있다. 각각의 경우 모두 소켓버퍼의 크기를 달리하여 전송률을 측정하였다.

그림 3에서 볼 수 있듯이 본 논문에서 설계구현한 초경량 TCP/IPv6 프로토콜을 탑재한 노드의 전송률이 버퍼의 크기가 1에서 2까지 급격히 증가하다가 3에서 10까지 조금씩 증가하는 모습을 확인할 수 있다. 그림 3에서 초경량 TCP/IPv6를 사용한 방법이 TinyOS에서 제공하는 컴포넌트를 이용해서 데이터를 전송할 때의 방법보다 전송률 측면에서 약 85-90% 정도를 보이며 약 10-15% 정도의 성능 하락을 보인다. 이는 본 논문의 초경량 TCP/IPv6 프로토콜의 동작 과정이 센서 노드에 많은 영향을 주지 않음을 알 수 있다.

5. 결론 및 향후 계획

본 논문에서는 센서 네트워크와 IPv6 기반의 인터넷 연동을 위한 핵심적인 기술 분야인 저 전력, 저사양의 센서 노드에서 실행 가능한 최소 사양의 TCP/IPv6 프로토콜 프로파일을 구성하기 위한 기능 분석을 수행하였고, 이를 기반으로 센서 노드에 적합한 초경량 TCP/IPv6 기능 프로파일을 설계하였다. 설계된 프로파일의 동작을 확인하기 위하여 프Telosb 플랫폼과 TinyOS 운영체제 환경에서 시험적으로 구현하였다. 시험적인 성능 테스트 결과 TinyOS에서 제공하는 컴포넌트를 사용해서 전송률을 측정할 것과 초경량 TCP/IPv6를 탑재한 후 측정된 전송률 사이에 큰 차이가 없음을 확인하였다.

지금까지 구현된 내용은 TCP 연결기능과 흐름제어 기능을 중심으로 구현하였고 이에 대한 시험적인 성능 테스트를 수행 하였다. 현재 구현된 플랫폼에는 에러제어 기능은 포함되지 않았는데 향후 추가적인 구현 작업을 수행 할 것이며, IPv6 기능 중 센서 네트워크 환경에 적용하기 어렵거나 또는 적용 유무를 판단하기 힘든 기능에 대해서는 앞으로 지속적인 연구와 분석을 수행할 것이다.

참고문헌

- [1] G. Montenegro et al., "Transission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, IETF, 2007. 9.
- [2] N. Kushalnagar et al., "TPv6 over Low-Power Wireless Personal Area Networks (6LowPANs) : Overview, Assumptions, Problem Statement, and Goals", RFC 4919, IETF, 2007. 8.
- [3] 백상헌 외, "유비쿼터스 센서 네트워크를 위한 무선 센서 네트워크와 인터넷 연동기법", Telecommunication Review, 제15권, 제 2호, pp. 337-350. 2005. 6.
- [4] 이완직 외, "Tiny TCP/IPv6 기능 프로파일 연구," 한국전자통신연구원 연구보고서, 2006. 11.
- [5] 이완직 외, "초소형 TCP/IPv6 기능 프로파일 연구," 한국전자통신연구원 연구보고서, 2007. 10.
- [6] "The lwIP TCP/IP Stack," <http://www.sics.se/~adam/lwip>.
- [7] "The uIP Embedded TCP/IP Stack," <http://www.sics.se/~adam/uipl/>.
- [8] Yuko Izuhara, "Specification of TinyIPv6 Protocol Stack for Remote Control and Implementation on FPGA," IPSJ, 2002. 4.
- [9] 정기진 외, "USN용 초소형 TCP/IPv6 프로토콜:S-IPv6", 한국정보기술학회 하계학술대회논문집, 2008. 5.
- [10] 정기진 외, "센서 네트워크를 위한 초소형 UDP/IPv6 프로토콜 설계 및 구현", 한국산업정보학회 논문지, 제13권, 제4호, pp. 73-82, 2008. 12.
- [11] 김용운 외, "저전력 무선 센서 네트워크에서의 호스트를 위한 IPv6 구현 가이드라인", TTA 표준 TTAK.KO-10.0281, 한국정보통신기술협회, 2008. 12.
- [12] 김용운 외, "저전력 무선 센서 네트워크에서의 호스트를 위한 TCP/UDP 구현 가이드라인", TTA 표준 TTAK.KO-10. 0283, 한국정보통신기술협회, 2008. 12.
- [13] Stephen Dawson-Haggerty, at. el., "TinyOS 2.1 tutorial", <http://docs.tinyos.net/index.php/Ipsn2009-tutorial>, 2009. 4.
- [14] Matt Welsh and David Gay, "nesC: A component-oriented language for networked embedded systems," <http://www.eecs.harvard.edu/~mdw/talks/nesc-msr.pdf>.

김 신 재(Shin-Jae Kim)

[정회원]



- 2004년 2월 : 밀양대학교 컴퓨터 공학과(공학사)
- 2007년 9월 ~ 현재 : 부산대학교 바이오메디컬공학과 석사과정

<관심분야>

센서네트워크, 라우팅 프로토콜

허 석 렬(Seok Yeol Heo)

[정회원]



- 1986년 2월 : 경북대학교 전자공학과 학사
- 1991년 2월 : 경북대학교 컴퓨터 공학과 석사
- 2009년 2월 : 경북대학교 컴퓨터 공학과 박사
- 1992년 3월 ~ 2006년 2월 : 밀양대학교 컴퓨터공학부 교수
- 2006년 3월 ~ 현재 : 부산대학교 바이오메디컬공학과 교수

<관심분야>

RFID/USN, 컴퓨터 네트워크, u-Health

김 영 균(Young-Gyun Kim)

[정회원]



- 2008년 8월 : 밀양대학교 컴퓨터 공학부(공학사)
- 2008년 9월 ~ 현재 : 부산대학교 바이오메디컬공학과 석사과정

<관심분야>

USN, 센서네트워크 응용

신 범 주(Bum Joo Shin)

[정회원]



- 1991년 2월 : 경북대학교 대학원 컴퓨터공학과 (공학석사)
- 1998년 8월 : 경북대학교 대학원 컴퓨터공학과(공학박사)
- 1987년 3월 ~ 2002년 2월 : 한국전자통신연구원 책임연구원
- 2002년 4월 ~ 2006년 3월 : 밀양대학교 컴퓨터공학부 교수
- 2006년 3월 ~ 현재 : 부산대학교 바이오메디컬공학과 부교수

<관심분야>

센서시스템, 메디컬 소프트웨어 응용

이 완 직(Wan-Jik Lee)

[정회원]



- 1992년 2월 : 경북대학교 통계학과 학사
- 1994년 2월 : 경북대학교 컴퓨터 공학과 석사
- 2007년 8월 : 경북대학교 컴퓨터 공학과 박사
- 1997년 3월 ~ 2006년 2월 : 밀양대학교 정보통신공학부 교수
- 2006년 3월 ~ 현재 : 부산대학교 바이오메디컬공학과 교수

<관심분야>

통신프로토콜, 프로토콜 구현, 네트워크 보안