

센서 네트워크를 위한 초소형 UDP/IPv6 프로토콜 설계 및 구현[†]

(Design and Implementation of tiny UDP/IPv6
Protocols for Sensor Networks)

정기진*, 이준섭**, 김용운**, 손영호***, 이완직*, 허석렬*
(Ki-Jin Jung, Jun Seob Lee, Yong-Woon Kim, Young-Ho Sohn,
Wan-Jik Lee, Seok-Yeol Heo)

요약 센서 정보의 효율적인 활용과 관리를 위해서는 인터넷을 통한 센서 정보의 수집과 관리가 절대적으로 필요하다. 이를 위해서는 센서 네트워크와 인터넷을 연동하는 기술이 절대적으로 필요하다. 센서 네트워크와 인터넷을 연동하는 방법은 게이트웨이를 통한 변환 방법이 주로 연구되었지만 최근에는 센서 노드에 IP기반 인터넷 프로토콜들을 직접 탑재하는 기술이 크게 주목받고 있다. 특히 IPv6는 풍부한 주소공간과 주소 자동생성과 같은 특성이 센서 네트워크와 잘 부합되기 때문에 센서 네트워크용 통신 프로토콜로 아주 적합하다. 본 논문에서는 센서 네트워크 환경에 적합한 초소형 UDP/IPv6 프로토콜 기능을 설계하고, 이를 TinyOS 기반의 nesC로 구현하였다. 구현된 프로그램은 TOSSIM과 TinyViz를 이용하여 시뮬레이션 결과를 확인하였고, 센서 노드(Mote)에 직접 탑재하여 시험적인 성능 테스트를 수행하였다.

핵심주제어 : USN, WSN, 센서 네트워크, IPv6, TCP/IP

Abstract Collecting and managing the sensor information through the Internet is critical to effectively manage and utilize the sensor information. The interworking technique of the sensor network and internet is required to realize the desirable condition for managing and utilizing the sensor information. Among many interworking techniques, the translation technique using the gateway had been widely studied. However, the technique which mounts IP-based Internet protocols directly on the sensor node is getting more attention recently. Particularly, IPv6 is suitable for the communication protocol for the sensor network, because its features, such as abundant address space or address auto-configuration, are well matched with the sensor network. In this paper, we design the tiny UDP/IPv6 protocol functions which are suitable for the sensor network environment, and we implemented the functions with TinyOS based nesC. After we examined the simulation results of by using TOSSIM and TinyViz, we carried out the experimental performance evaluation for the program by mounting the tiny UDP/IPv6 on the sensor node (Mote).

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

† 본 연구는 한국전자통신연구원 위탁연구(5010-2008-0010) 지원으로 수행되었음

* 부산대학교 바이오정보전자공학과

** 한국전자통신연구원 u-인프라표준연구팀

*** 영남대학교 전자정보공학과

1. 서론

센서를 통해 환경 정보를 수집하고 센서 네트워크를 통하여 실시간 정보를 수집/관리하는 기술은 USN의 핵심 기술이다. 센서 노드의 정보를 처리하기 위해서는 센서 정보를 수집하고 관리하는 서버를 센서가 설치되어 있는 로컬 사이트에 두고 독립적으로 운영할 수도 있지만, 센서 정보의 효율적인 활용과 관리를 위해서는 인터넷을 통한 센서 정보의 수집과 관리가 절대적으로 필요하다. 이를 위해서는 센서 네트워크가 독자적인 네트워크로서가 아니고 인터넷의 액세스 네트워크로서 접속되어야 한다. 따라서 싱크노드는 센서 데이터의 수집 창구였던 전통적인 기능 외에 게이트웨이로서의 역할이 중요하게 부각되고 있다[1-2]. 이때 만일 센서 노드에 IP가 아닌 별도의 주소체계를 가지고 있다면, 게이트웨이는 데이터 패킷에 대하여 주소 변환을 통해 인터넷과 연동 시켜야 하며, 이러한 주소 변환 방법 때문에 표준화된 일관된 기술을 만들기 힘들다. 대표적인 센서 네트워크 기술인 Zigbee는 처음부터 센서 네트워크와 인터넷 연동을 고려하지 않았기 때문에 이러한 문제점을 안고 있다[3]. 이런 문제점을 인식하고 이를 해결하기 위한 방안을 Zigbee 얼라이언스에서도 연구 중에 있지만 아직까지 기초적인 수준에 머물러 있다.

다른 방안으로서 IP 기반의 인터넷 프로토콜들을 센서 노드에 직접 탑재하는 것을 고려해 볼 수 있다. 하지만 매우 제한적인 자원을 가진 센서 노드의 특성을 고려해 볼 때, 센서 노드에 IP 기반의 인터넷 프로토콜을 직접 적용하는 것은 과도한 시스템 자원 소모 때문에 부적절한 선택이라고 생각되어 왔다. 하지만 최근에는 센서 H/W 기술의 발전과 IPv6 기반의 BCN과의 연동 효율성 및 확장성 관점에서 센서 노드에 IPv6 기반 프로토콜을 탑재하는 센서 네트워크 기술이 크게 주목 받기 시작하고 있다.

센서 네트워크와 인터넷 연동을 위한 프로토콜로서 IPv6 기반 센서 네트워크 기술을 주목하는 이유는 차세대 인터넷 프로토콜인 IPv6는 풍부한 주소 공간과 주소 자동생성과 같은 특성이 센서 네트워크와 잘 부합되어 센서 네트워크를 위한 통신 프로토콜로서 아주 적합하다고 할 수 있다. 또

한, 센서 노드에 IP를 탑재함으로써 인터넷 환경과 쉽게 연동이 가능하며, 기존 IP 기반의 인터넷 프로토콜을 센서 노드에 맞게 수정 탑재함으로써 인터넷 환경에서 널리 사용되는 응용을 센서 네트워크에 쉽게 적용할 수 있는 장점이 있다[4].

IPv6 기반의 인터넷 프로토콜들을 센서 네트워크에 구현하기 위해서는 최적화 및 경량화 작업이 필수적이다. 또한 각 프로토콜 규격을 기능별로 초소형화 하여 센서 네트워크에서 수행되는 응용의 형태와 수행하는 역할(라우터 역할 여부, 주소설정 방식 등)에 따라 이들 프로토콜 기능을 조합할 수 있도록 프로토콜 프로파일을 모듈 기반으로 설계하는 것이 필요하다.

본 논문에서는 센서 네트워크 환경에 적합한 초소형 IPv6, ICMPv6, UDP 프로토콜 기능을 설계하고, 이를 센서 네트워크의 대표적인 운영체제인 TinyOS 환경에서 nesC(network embedded system C)로 구현하였다. 구현된 프로토콜들은 TinyOS 시뮬레이션 도구인 TOSSIM과 TinyViz를 이용하여, 동작 상태를 확인하였으며, 센서 노드(Mote)에 직접 탑재하여 프로토콜 동작에 대한 시험적인 성능 측정을 수행하였다.

본 논문의 2장에서는 관련 연구 내용에 대해 소개하고, 3장에서는 센서 네트워크와 센서 노드에 관한 가정과 초소형 UDP/IPv6 기능 프로파일에 대해 설명한다. 4장에서는 초소형 UDP/IPv6 프로토콜 기능 설계와 구현에 대한 내용을 기술한다. 마지막으로 5장에서 결론과 향후 연구 과제를 다루었다.

2. 관련연구

본 장에서는 센서 네트워크 환경에서 IPv6 패킷 전송에 관한 표준화를 진행하고 있는 6LoWPAN 워킹 그룹의 연구 내용과 초소형 TCP/IP 프로토콜 구현에 관한 연구 내용을 소개한다.

2.1 6LoWPAN (IPv6 over LoWPAN)

IETF의 6LoWPAN 워킹 그룹은 센서 네트워크의 표준 PHY/MAC인 IEEE 802.15.4 상에서 효과

적인 IPv6 패킷 전송 방안을 정의하는 것을 목표로 하고 있다. 이에 따라 6LoWPAN 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[5].

하지만 전력 소모, 메모리 사용 등이 극히 제한적인 센서 네트워크 특성에 따라 IPv6 관련 프로토콜들의 기능 최적화 및 경량화 연구도 매우 필요한 분야지만, 현재 6LoWPAN에서는 이와 관련된 연구는 진행하지 않고 있다.

2.2 경량급 TCP/IP 프로토콜 스택

현재까지 TCP/IP 프로토콜의 경량화에 관한 연구가 많이 수행되었지만, 본 절에서는 그 중 대표적인 경량급 프로토콜인 uIP, lwIP, TinyIPv6에 대해서 살펴본다.

uIP

uIP는 Swedish Institute of Computer Science (SICS)에서 소형 8비트 마이크로프로세서에서 동작할 수 있도록 개발한 TCP/IP 프로토콜 스택이다[6]. ARP, IPv4, ICMP, TCP 기능이 포함되어 있으며, uIP는 통신하는 상대가 완전한 TCP/IP 기능을 가질 필요가 없다. 즉 상대 기기가 초소형 TCP/IP 프로토콜 스택이라도 호환성 있게 동작할 수 있다는 장점을 가진다. uIP는 수 Kbyte 이내의 코드 크기로 구현되며 수백 바이트 이내의 메모리에서도 동작할 수 있다.

lwIP

lwIP 역시 SICS에서 개발되었으며, 임베디드 시스템을 위한 TCP/IPv4 프로토콜의 대부분의 기능을 가지며 범용적인 용도로 설계되었다[7]. lwIP 역시 상대 호스트와의 완벽한 TCP/IP 프로토콜의 호환성을 제공한다. 하지만 구현적인 측면에서 uIP는 자원 사용량의 최소화에 초점을 맞추어 프로토콜 성능을 희생시킨 측면이 강한 반면, lwIP는 약간의 자원 사용은 감수하더라도 어느 수준 이상의 프로토콜 성능을 고려하였다는 차이가 있다.

TinyIPv6

TinyIPv6는 일본의 Non-PC Network Appliance Committee에서 2002년에 개발된 초소형 IPv6 기반 프로토콜 스택이다[8]. TinyIPv6는 홈네트워크 환경에서 인터넷으로 연결된 홈 가전들의 원격 조정을 위해 개발되었으며, 프로토콜 스택은 IPv6, ICMPv6, UDP로 구성된다. 이 프로토콜 스택은 Low Cost Network Appliances (LCNA)라고 불리는 PC가 아닌 일반적인 AV 기기나 가전제품들의 인터넷 접속 등의 네트워크 기능 개발을 목표로 설계되었다.

앞에서 언급한 바와 같이 현재까지 많은 종류의 경량급 TCP/IP 프로토콜 스택들이 연구되었지만, 대부분의 프로토콜 스택들이 uIP, lwIP와 같이 IPv4 기반이며, 임베디드 시스템과 같이 전력이 안정적으로 공급되는 시스템을 대상으로 설계된 프로토콜들이다. TinyIPv6 스택은 IPv6를 기반으로 경량화된 프로토콜 스택이지만, 이 프로토콜 스택 역시 임베디드 시스템을 대상으로 설계되었다. 따라서 TinyIPv6는 IPv6의 이웃 노드 탐색 (Neighbor Discovery), 주소 자동설정 등을 위한 브로드캐스팅 패킷 전송이 빈번하게 발생하기 때문에 저전력 소모가 필수적인 센서 네트워크의 환경에는 적합하지 않다.

3. 초소형 UDP/IPv6 기능 프로파일 설계

3.1 가정

센서 네트워크는 일반 네트워크와 달리 동작하는 환경이나 목적에 따라서 동작이 많이 다르기 때문에 센서 네트워크를 위한 프로파일을 설계하는데 있어서 대상이 되는 모델이나 환경을 구체적으로 명시할 필요가 있다. 따라서 본 논문에서는 센서 네트워크와 센서 노드에 대하여 아래와 같이 구체적인 환경을 명시하고 이를 기반으로 프로파일을 설계 하였다.

센서 네트워크에 관한 가정

- PHY/MAC은 IEEE 802.15.4 표준을 따른다.

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

센서 노드에 관한 가정

- 8 또는 16 비트 이상의 RISC 방식 CPU
- 4 KByte 이상의 RAM
- 50 KByte 이상의 프로그램 메모리
- IEEE 802.15.4의 단일 통신 인터페이스

3.2 초소형 UDP/IPv6 기능 프로파일

3.2.1 센서 네트워크 환경에 대한 IPv6 최적화

본 절에서 기술한 내용을 제외한 나머지 기능은 RFC 표준을 따르는 것을 원칙으로 한다.

보안관련

- IPsec을 위한 기능은 모두 지원하지 않는다. IEEE 802.15.4는 기본적으로 AES(Advanced Encryption Standard)를 이용한 링크 계층의 보안을 제공하지만 상위계층 보안 기법은 다루고 있지 않다. 상위계층의 보안은 응용의 요구를 신중하게 고려해야 하며 과도한 계산이나 대역폭이 필요한 보안 솔루션은 센서 노드에게 적합하지 않다. IPsec은 기본적으로 충분한 스택 크기를 가지면서 대역폭과 비용에 구애 받지 않는 환경을 대상으로 설계되었기 때문에 센서 네트워크에는 부적합하다.

라우팅 테이블 관리 관련

- 라우팅 테이블 관리를 위한 기능은 지원하지 않는다. 8 또는 16 비트 RISC 방식 CPU 센서 네트워크에서 싱크노드와 같이 특별한 기능을 갖는 노드를 제외한 일반 센서 노드는 라우터로서 동작을 하지 않고 단말 노드로만 동작하기 때문에 디폴트 라우터 리스트 정도의 간단한 정보만 관리하면 된다. 따라서 일반 센서 노드에서는

FIB(Forwarding Information Base)와 관련된 라우팅 테이블 관리 기능은 필요하지 않다.

다중 인터페이스 관련

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

모바일(Mobile) IPv6 관련

- 모바일 IPv6 관련 기능은 지원하지 않는다. 차량이나 이동체에 부착된 센서와 같이 센서 노드가 이동성을 갖는 경우도 있지만 일반적으로는 센서 노드의 이동성은 고려하지 않는다. 따라서 MIPv6 기능은 센서 노드에 필요하지 않다. 만일 센서 노드가 이동성을 갖는 경우에도 MIPv6의 올바른 동작을 위한 여러 가지 구성 요소들이나, 키 분배 및 관리, IPsec과 같은 많은 오버헤드를 필요로 하기 때문에 현재의 센서 네트워크 환경에서는 고려하기 어렵다.

Path MTU 탐색 기능 관련

- Path MTU 탐색 기능은 지원하지 않는다. IPv6에서는 전송 경로의 모든 링크에 대해서 1280 옥텟 이상의 MTU를 요구한다. Path MTU 탐색 기능은 IPv6 노드가 1280 옥텟 이상의 패킷을 전송하기 위해서 필요한 기능이며, 항상 1280 옥텟 이하로만 보내겠다면 구현할 필요가 없다. 센서 네트워크 환경에서 1280 옥텟 보다 큰 사이즈의 패킷을 송수신 하는 일은 비현실적이다. 따라서 센서 네트워크의 MTU 크기는 IPv6와 호환성을 유지하기 위해서는 1280 옥텟으로 설정되어야 하며, 이 경우에는 Path MTU 탐색 기능이 필요하지 않다.

IPv6 주소 관련

- 하나의 노드는 하나의 링크 로컬 주소와 하나의 글로벌 유니캐스트 주소만 지원한다. 일반 IPv6 호스트는 다중 인터페이스와 다중 주소체계를 지원하지만 센서 노드는 다중 인터페이스를 지원하지 않기 때문에 센서 노드에 대해서는 링크 로컬과 글로벌 주소를 하나씩 할당하는 것이 바

람직하다. 글로벌 유니캐스트 주소는 비상태형(stateless) 주소 자동 설정 방식에 의해 생성한다. DHCPv6와 같은 상태형(stateful) 주소 생성을 위한 기능은 제거한다. All-Nodes-Multicast, All-Routers-Multicast 이외의 멀티캐스트 주소는 사용하지 않으며, 애니캐스트 주소에 관한 기능은 지원하지 않는다.

패킷 관련

- Flow label 및 트래픽 클래스와 같은 QoS에 관한 패킷 처리는 수행하지 않으며, 모든 IPv6 확장 헤더의 처리와 생성을 지원하지 않는다.
 - 홑 바이 홑 옵션: Jumbo Payload 옵션은 센서 네트워크에 필요 없으며, 라우터 경보(router alert) 옵션은 IPv6 라우터 기능을 수행하지 않기 때문에 지원할 필요가 없다.
 - 목적지 옵션: 모바일 IPv6를 지원하지 않으므로 Home Address Option을 지원할 필요가 없다.
 - 라우팅 헤더: 라우터 기능을 수행하지 않으므로 Type 1 옵션 지원이 필요 없으며, 모바일 IPv6를 지원하지 않으므로 Type 2 옵션도 지원할 필요가 없다.
 - 단편 헤더: 단편에 관한 처리는 센서 네트워크의 IPv6 게이트웨이에서 처리하기 때문에 센서 노드에서 지원할 필요가 없다.
 - AH/ESP 헤더: IPsec 기능을 수행하지 않으므로 지원하지 않는다.
 - 모바일 헤더: 모바일 IPv6 기능을 수행하지 않으므로 지원하지 않는다.
 - 수신 패킷 내의 확장 헤더가 포함된 경우, 모든 확장 헤더는 무시한다. 다만 패킷 파싱을 위해 헤더 형식(type)과 길이(length)는 검사한다.

3.2.2 센서 네트워크 환경에 대한 ICMPv6 최적화

ICMPv6 Error Reporting 관련

- Destination Unreachable 메시지 생성 이외의 모든 Error Reporting 관련 기능은 지원하지 않는다.

ICMPv6 Query 기능

- echo 서버 기능 이외에는 지원하지 않는다.

3.2.3 센서 네트워크 환경에 대한 ND 최적화

라우터 Discovery 관련

- 센서 노드들은 IPv6 라우터로 동작하지 않기 때문에 RS 메시지 수신 및 RA 송신에 따른 처리가 필요하지 않다. 센서 노드는 주소 생성과 링크 상태에 대한 정보에 열기 위해서 다음과 같은 이벤트 발생시 IPv6 라우터(게이트웨이)에게 RS 메시지를 전송한다.
 - 인터페이스가 초기화되었을 경우
 - 센서 노드가 링크에 접속되었을 경우
 - 센서 노드가 링크에 재접속되었을 경우

자동 주소 설정 관련

- 비상태형 주소 자동 설정 방식 동작 시에 DAD(Duplication Address Detection) 기능은 지원하지 않는다.
 - EUI-64 방식의 48 비트 주소는 RFC 2464에 따라 주소를 생성 한다.
 - 16 비트 주소를 사용할 경우에는 RFC 4944에 따라 주소를 생성 한다.
- 디폴트 라우터 리스트 및 프리픽스 리스트 관리를 수행하지 않는다.
 - PAN 내에 단일 IPv6 게이트웨이만 존재하며, 센서 노드는 하나의 글로벌 주소만 사용하므로 하나의 디폴트 라우터와 하나의 프리픽스만 등록한다.
 - RA 수신에 의해 새로운 라우터/프리픽스가 탐지되었을 경우에는 기존 라우터/프리픽스를 삭제하고 새로운 항목을 등록한다.
 - 타이머에 의한 오버헤드를 줄이기 위해 등록된 라우터/프리픽스에 대한 타이머 동작을 수행하지 않고, Lifetime이 0인 RA 패킷을 수신하였을 경우에만 삭제한다.

Neighbor Discovery 관련

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

변환(Address Resolution)을 위한 NS 송신과 NA 수신 기능을 수행하지 않는다.

NUD (Neighbor Unreachability Detection) 관련

- NUD 기능은 지원하지 않는다. IPv6에서 NUD 기능은 노드와 이웃 노드 혹은 노드와 IPv6 게이트웨이 사이의 연결성(reachability)을 결정하는데 사용된다. 이를 위해서 노드는 NCE (Neighbor Cache Entry)를 유지하면서 유니캐스트 전송에 의한 갱신과 재전송 타이머에 의한 주기적인 탐색 기능을 수행한다. 갱신에 실패한 이웃 노드에 대해서는 주소 변환을 다시 수행하고, IPv6 게이트웨이에 대해서는 새로운 디폴트 라우터 탐색을 시도한다. 만일, 3.1절과 같이 센서 노드가 자신의 MAC 주소로 EUI-64 또는 중복되지 않는 16비트 주소를 사용한다면 주소 변환을 수행할 필요가 없기 때문에 센서 노드에 대한 NUD는 필요하지 않다. 또한 계층 2 멀티홉 라우팅 프로토콜을 이용해서 IPv6 게이트웨이에 대한 탐지, 실패, 복구, 교체를 지원할 수 있기 때문에 IPv6 게이트웨이에 대한 NUD 역시 필요하지 않다.

NCE 관련

- 하나의 센서 네트워크는 하나의 IPv6 서브넷에 대응되기 때문에 여러 개의 프리픽스를 관리할 필요가 없다. 따라서 센서 노드는 단일 프리픽스만을 가지며 RA 메시지에 따른 프리픽스 주소 관리가 필요하다. 하지만 타이머 동작에 따른 오버헤드가 크기 때문에 프리픽스 주소에 대한 타이머 관리하는 하지 않는다. 센서 노드는 주소 변환을 위한 NS 송신을 하지 않기 때문에 NS 메시지에 따른 NCE 관리 기능은 필요하지 않다.

Redirect 패킷 생성 및 처리 관련

- Redirect 패킷 생성 및 수신 처리를 위한 기능은 지원하지 않는다. 일반 센서들은 IPv6 라우터 기능을 수행하지 않으며, 하나의 PAN 내에는 하나의 IPv6 게이트웨이만 존재하기 때문에 Redirect 기능은 필요하지 않다.

타이머 관련

- 타이머를 관리하는 것은 센서 노드에게 많은 부담이 되기 때문에 가능한 타이머와 관련된 기능을 최소화하는 것이 필요하다. 이를 위해서 다음과 같은 타이머 속성을 갖는다.

- 라우터 Lifetime(디폴트라우터 지정), Reachable Time(NUD)은 0이 아닐 경우 무한대로 동작하도록 한다. 만일 연결오류와 같은 라우팅 정보(예를 들어 RERR)가 수신되면 0으로 설정한다.
- 프리픽스에 대한 Valid Lifetime과 Preferred Lifetime은 무한대로 설정한다. Valid Lifetime과 Preferred Lifetime은 무한대 설정이 가능하나(0xffffffff), 다른 타이머 값은 동적으로 변한다. 동적으로 변하는 타이머 값을 내부적으로 상수 또는 무한대로 설정한다면 수신 패킷에 대한 프로세싱이 필요 없으며 이에 따른 필드 압축(예를 들면 Reachable Time 필드(32비트)를 1비트)도 가능하다.

3.2.4 센서 네트워크 환경에 대한 UDP 최적화

UDP 프로토콜 자체가 상위의 응용 프로세스들을 위한 포트 주소 설정 이외의 별다른 기능이 없기 때문에 아래와 같은 기능들만 제거하였다.

다중인터페이스

- UDP 기능 중 다중 인터페이스 지원을 위한 기능은 수행하지 않는다.

IP 옵션

- IP 옵션의 내용을 UDP 프로토콜을 통해 응용에게 전달하는 기능은 수행하지 않는다.

4. 초소형 UDP/IPv6 설계 및 구현

여기에서는 앞에서 기술한 초소형 UDP/IPv6 프로토콜 프로파일 내용을 테스트하기 위해 프로파일 규격을 적용한 UDP/IPv6 프로토콜 스택 구현에 대해서 설명한다.

4.1 구현환경

초소형 UDP/IPv6과 같은 프로토콜 스택이 소

형, 저가의 하드웨어 플랫폼에서 효율적으로 동작하기 위해서는 플랫폼의 프로세서, 메모리 용량 등의 하드웨어 사양을 고려한 구현이 필요하다. 또한 프로토콜의 구현 방식은 운영체제와 밀접한 관계를 가지므로 프로토콜 구현 구조, 구현 알고리즘 등도 운영체제가 제공하는 방식대로 구성되어야 한다. 본 논문에서는 Telosb 플랫폼과 TinyOS 운영체제를 구현 환경으로 선정하였다. Telos계열 플랫폼은 TinyOS의 대표적인 하드웨어 플랫폼이라고 불릴 만큼 TinyOS에 적합한 구조로 설계되어 있다.

4.2 초소형 UDP/IPv6 프로토콜 설계

본 절에서는 3.2절의 센서 노드를 위한 초소형 UDP/IPv6 프로토콜 프로파일 규격을 바탕으로 TinyOS의 nesC 방식에 맞추어 설계한 초소형 UDP/IPv6 프로토콜 계층 구성과 지원 프로토콜 기능에 대해 기술한다.

4.2.1 초소형 UDP/IPv6 프로토콜 계층 분석

초소형 UDP/IPv6 프로토콜의 계층 구성은 다음의 <표 1>과 같다.

<표 1> 초소형 UDP/IPv6 프로토콜 계층 구성

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

본 논문의 초소형 UDP/IPv6 프로토콜 스택은 <표 1>에서 보는 바와 같이 IPv6 기반 프로토콜로서 UDP를 지원 하고 있으며, L2 Routing 계층은 센서 네트워크에서 멀티 홉 지원을 위해 포팅한 멀티 홉 라우팅 프로토콜들로 구성된다.

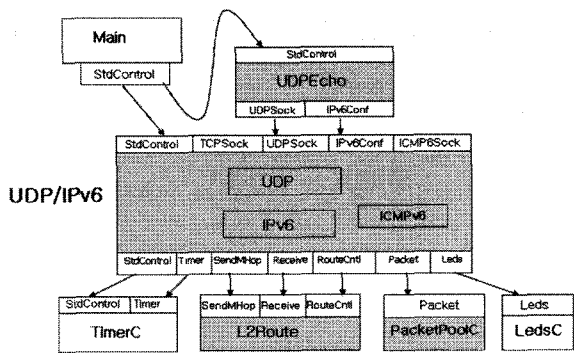
센서 네트워크에서 멀티 홉 라우팅은 아직 표준으로 정해진 것이 없기 때문에 특정 라우팅 프로토콜을 지정하지 않았다. 또한 센서 네트워크의 응용이나 네트워크 환경에 따라 멀티 홉 라우팅 프로토콜을 선택해서 사용하는 것이 더욱 효율적이

라고 판단했기 때문이기도 하다. 예를 들어 센서의 개수와 최대 홉 거리가 크지 않으면 간단한 Flooding 프로토콜을 사용하는 것도 좋은 선택이 될 수도 있다. TinyAODV는 기존의 AODV (Ad-hoc On Demand Vector) 프로토콜을 간소화시켜 TinyOS에서 동작 시킬 수 있도록 구현한 On-Demand 방식의 멀티 홉 라우팅 프로토콜이다. 따라서 초소형 UDP/IPv6 프로토콜에서는 두 가지의 멀티 홉 라우팅 프로토콜을 지원하며 이 둘 중 하나를 선택해서 사용할 수 있도록 하였다.

링크계층에서는 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 2바이트 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 초소형 UDP/IPv6 프로토콜 구현 및 테스트

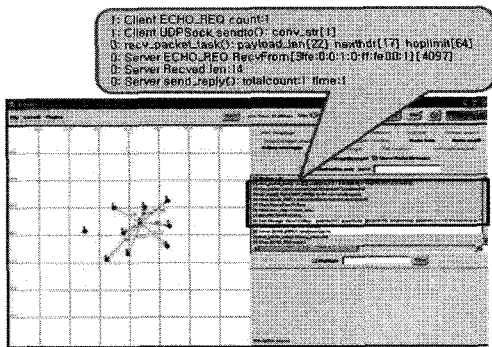
4.2절에서 기술한바와 같이 초소형 UDP/IPv6 프로토콜은 TinyOS의 nesC 프로그래밍 방식에 의해 구현되었다[8]. 따라서 초소형 UDP/IPv6도 하나의 컴포넌트로 구성되어 초소형 UDP/IPv6을 이용하는 상위 컴포넌트에게 인터페이스를 제공하고, 초소형 UDP/IPv6도 자신이 필요로 하는 인터페이스를 제공하는 다른 컴포넌트들을 사용하게 된다. (그림 1)의 Configuration 예는 본 논문의 초소형 UDP/IPv6 프로토콜 상에서 UDPEcho 응용 프로그램을 동작시켰을 때의 컴포넌트들의 구성도이다. UDPEcho는 ICMP를 사용하는 Ping과 동일한 기능을 UDP 프로토콜을 이용하여 작성한 간단한 프로그램이다. 음영으로 표시된 컴포넌트는 본 논문에서 직접 구현하거나 일부 수정한 컴포넌트이며 음영으로 표시되지 않은 컴포넌트는 TinyOS 자체에서 제공하는 컴포넌트이다.



(그림 1) UDP/IP6 컴포넌트 Configuration 구성

(그림 1)을 보면, 실제 초소형 UDP/IP6 구성 프로토콜들(UDP, IPv6, ICMPv6)이 하나의 UDP/IP6 컴포넌트로 구성됨을 알 수 있다. 이렇게 통합 구현한 이유는 계층 간의 호출 등에 의해 오버 헤드를 줄이고, 프로토콜 동작을 위한 메모리 사용도 많이 줄일 수 있기 때문이다. 멀티 홉 라우팅을 담당하는 L2 Route 기능은 응용의 목적과 센서 네트워크 환경에 따라 옵션으로 선택할 수 있도록 별도의 컴포넌트로 구성 하였다.

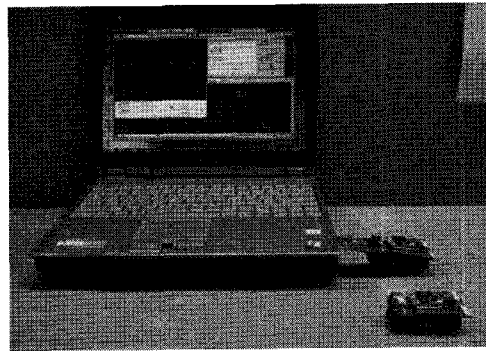
(그림 1)의 구성을 바탕으로 프로토콜을 구현한 후, TinyOS에서 제공하는 TOSSIM과 TinyViz를 사용하여 초소형 UDP/IP6의 UDP 프로토콜이 정상 동작하는지 시뮬레이션을 통하여 확인하였다.



(그림 2) UDP 프로토콜 시뮬레이션 테스트

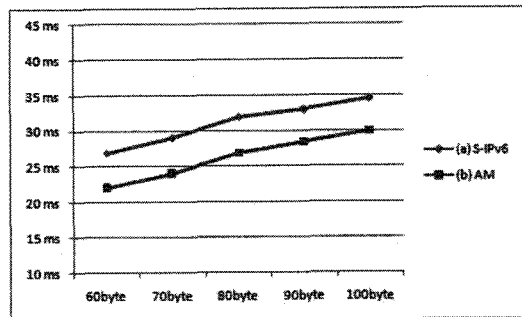
(그림 2)는 클라이언트에서 UDP 프로토콜의 sendto() 함수를 이용하여 echo_request 메시지를 전달하고, 서버 노드는 RecvFrom() 함수를 이용하여 전달받은 echo_request 메시지에 대한 echo_reply 메시지를 전달하고 있음을 보여준다.

(그림 3)은 위와 같이 구현된 UDP/IP6 프로토콜을 센서 노드에 탑재하여 테스트한 환경이다. 두 개의 센서 노드를 사용하여 원 홉으로 서버/클라이언트 형태로 구성 하였으며, 그중 하나의 노드를 PC와 연결하여 결과를 확인 하였다.



(그림 3) UDP/IP6 프로토콜 테스트 환경

(그림 4) (a)는 (그림 3)의 환경을 토대로 센서 노드에 UDP/IP6 프로토콜을 탑재하여 RTT (Round Trip Time) 시간을 측정 한 결과이고, (그림 4) (b)는 UDP/IP6을 탑재하지 않고 TinyOS에서 기본적으로 제공하는 송수신 컴포넌트를 사용하여 RTT 시간을 측정 한 값이다.



(그림 4) RTT Time 측정 (a)IPv6 탑재,(b)일반 송수신

(그림 4)에서 알 수 있듯이 본 논문에서 설계 · 구현한 UDP/IP6을 탑재한 패킷의 송수신 RTT 지연과 TinyOS에서 제공하는 컴포넌트를 이용한 송수신 RTT 지연이 크게 차이나지 않음을 확인할 수 있었다.

5. 결론 및 향후과제

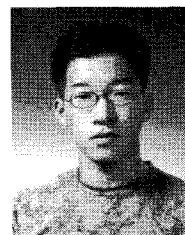
본 논문에서는 센서 네트워크 환경에 적합한 최소 사양의 IPv6, ICMPv6, UDP 프로토콜 프로파일을 설계하였으며, 설계된 프로파일 규격을 TinyOS 운영체제 상에서 Telosb 계열의 센서 노드에서 구현하였다. 또한 구현된 프로토콜의 시현적인 성능 측정을 수행하여, 본 논문의 UDP/IPv6 프로토콜을 활용한 패킷 송수신 RTT 지연 시간과 TinyOS 기반의 MAC 레벨의 패킷 송수신 RTT 지연 시간이 큰 차이가 없음을 확인하였다.

하지만 현재의 구현에서는 TCP 프로토콜이 포함되지 않았으며, 6LoWPAN에서 정의한 IPv6 헤더 압축, 패킷 단편화 및 재조립 기능들이 생략된 상태이다. 따라서 향후 이러한 프로토콜 기능 설계와 구현을 진행하고, 구현된 프로토콜들의 성능 개선을 위한 연구를 계속 수행할 예정이다.

참 고 문 헌

- [1] 정기진 외, "USN용 초소형 TCP/IPv6 프로토콜 : S-IPv6," 한국정보기술학회, 2008. 5.
- [2] 백상현 외, "유비쿼터스 센서 네트워크를 위한 무선 센서 네트워크와 인터넷 연동기법," *Telecommunication Review*, 제15권 제2호, 2005. 4.
- [3] ZigBee Alliance, "ZigBee Specification v1.0," 2005. 6.
- [4] 박승민, "센서 네트워크 플랫폼 및 운영체제 기술동향," *전자통신동향분석*, 제21권 제1호, 2006. 2.
- [5] G. Montenegro and et al., "Transmission of IPv6 Packets over IEEE 802.15.4 Networks," RFC 4944, 2007. 9.
- [6] "The uIP Embedded TCP/IP Stack," <http://www.sics.se/~adam/uip/>.
- [7] "The lwIP TCP/IP Stack," <http://www.sics.se/~adam/lwip/>.
- [8] "TACA Project: Documentation," <http://www.taca.jp/docs.html/>.
- [9] <http://www.tinyos.net>, TinyOS web page.

- [10] 강정훈, "nesC Programming" 센서네트워크 워크샵 발표자료, 한국통신학회, 2006. 10.
- [11] IEEE Computer Society, "IEEE Std. 802.15.4-2003," 2003. 10.
- [12] Crawford, M., "Transmission of IPv6 Packets over Ethernet Networks," RFC 2464, December 1998.
- [13] Narten T and et al., "Neighbor Discovery for IP version 6 (IPv6)," RFC2461, December 1998.
- [14] S. Deering and et al., "Internet Protocol, Version 6 (IPv6) Specification," RFC 2460, December 1998.
- [15] S. Thomson and T. Narten, "IPv6 Stateless Address Autoconfiguration," RFC 2462, December 1998.
- [16] A. Conta and et al., "Internet Control Message Protocol : (ICMPv6) for Internet Protocol Version 6 (IPv6) Specification," RFC 2463, December 1998.



정 기 진 (Ki-Jin Jung)

- 2007년 2월 : 부산대학교 정보통신공학과 (공학학사)
- 2007년 3월 ~ 현재 : 부산대학교 바이오정보전자공학과 석사과정

▪ 관심분야: RFID/USN, 통신프로토콜



이 준 섭 (Jun Seob Lee)

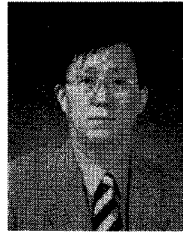
- 1997년 고려대학교 전산학과 학사
- 1999년 고려대학교 공학석사
- 1999년 3월 ~ 현재 한국전자통신연구원 선임연구원

▪ 2001년 3월 ~ 현재 충남대학교 박사과정
 ▪ 관심분야 : RFID, 센서 네트워크, 객체 식별자 (OID)



김 용 운 (Yong-Woon Kim)

- 1990년 동아대학교 전자공학과 학사
- 1995년 포항공과대학교 공학석사
- 1995년 2월 ~ 2001년 4월 한국 전자통신연구원 선임연구원
- 2002년 2월 ~ 2004년 4월 (주)이니텍 CTO
- 2004년 5월 ~ 현재 한국전자통신연구원 선임연구원
- 관심분야 : RFID/USN, 컴퓨터 네트워크, 기후변화 대응 정보통신기술



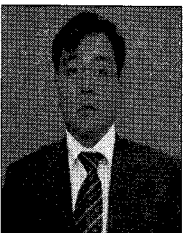
허 석 렬 (Seok-Yeol Heo)

- 1986년 경북대학교 전자공학과 학사
- 1991년 경북대학교 컴퓨터공학과 석사
- 2008년 경북대학교 컴퓨터공학과 박사
- 1992년 3월 ~ 2006년 2월 밀양대학교 컴퓨터공학부 교수
- 2006 3월 ~ 현재 부산대학교 바이오정보전자공학과 교수
- 관심분야 : RFID/USN, 네트워크 보안, 컴퓨터 네트워크, u-Health



손 영 호 (Young-Ho Sohn)

- 1986년 2월 : 경북대학교 전자공학과 (공학사)
- 2002년 5월 : ECE Dept. Texas A&M Univ. (공학박사)
- 2002년 9월 ~ 2005년 2월 영남대학교 전자정보공학부 객원교수
- 2005년 3월 ~ 현재 영남대학교 전자정보공학부 조교수
- 관심분야 : 센서시스템/네트워크, 광섬유통신망



이 완 직 (Wan-Jik Lee)

- 1992년 경북대학교 통계학과 학사
- 1994년 경북대학교 컴퓨터공학과 석사
- 2007년 경북대학교 컴퓨터공학과 박사
- 1997년 3월 ~ 2006년 2월 밀양대학교 정보통신공학부 교수
- 2006 3월 ~ 현재 부산대학교 바이오정보전자공학과 교수
- 관심분야 : 통신 프로토콜, 프로토콜 구현, 네트워크 보안