IEEE 802.15.4 MAC Protocol 검증을 위한 PHY 계층 시뮬레이터 구현

이호응 박현주
한밭대학교 정보통신공학과
hoeung@naver.com, phj@hanbat.ac.kr

Implementation of PHY Layer Simulator for IEEE 802.15.4 MAC Protocol Verification

Hoeung Lee. Hyunju Park
Department of Information Communication, HANBAT National University

요 약

무선 센서 네트워크는 수많은 노드들 간의 상호작용에 의해 서로 통신을 한다. 이러한 센서 네트워크에 대한 프로토콜 구현은 정확한 동작을 확인하기 위해 많은 노드들을 배치하여 실제 동작을 확인하며 개발을 해야 하는 번거로움을 갖고 있다. 본 논문에서는 무선 센서 네트워크에 적합한 프로토콜인 IEEE 802.15.4의 PHY 계층에 대한 시뮬레이터를 구현하여 여러 노드들이 있는 상황을 시뮬레이션 할 수 있는 환경을 갖춰서 상위 MAC 프로토콜의 동작을 검증할 수 있는 기반 환경을 구현 하였다.

1. 서 론

IEEE 802.15.4 표준은 저전력, 저속 통신 및 저가격을 만족시킬 수 있는 센서 네트워크를 위한 프로토콜로서 PHY 계층과 MAC 계층에 대해 정의하고 있다.

이러한 프로토콜은 특정한 상황을 모니터링 하거나 저속 및 저전력을 요구하는 산업제어, 가정 자동화, PC 주변기기 제어, 장난감, 헬스케어와 같은 응용에 활용될 수 있다. 수많은 노드들이 분산 배치되어 각 노드들 간의 상호 작용에 의해 서로 통신을 하게 되는 이러한 환경에서 IEEE 802.15.4 프로토콜에 대한 구현이 정확히 동작하는지 검증하기 위해서는 많은 노드들을 실제로 배치하여 동작을 확인하며 개발해야 하는 어려움이 있다. 특히 IEEE 802.15.4 표준에서 제시하고 있는 여러 가지 동기 신호에 맞게 동작 하고 있는지 검증하기 위한 방안도 필요하다.

본 논문에서는 IEEE 802.15.4 표준의 PHY 계층에 대한 시뮬레이터를 구현하여 상위 MAC 계층 구현을 검증할 수 있는 기반 환경을 제시한다. 또한 시뮬레이터 상에서 구현된 MAC 계층을 실제 CPU에 이식할 때 쉽게 이식 가능하도록 하드웨어 추상 계층을 제공한다.

본 논문의 2장에서 IEEE 802.15.4의 PHY 계층에 대한 특징을 설명하고, 3장에서는 PHY 계층을 시뮬레이션 하여 구현된 PHY Simulator 구현 방법에 대해 제시한다. 4장에서는 PHY Simulator 상에서 상위 계층을 구현한 실행 결과를 보여준다. 5장에서는 결론을 맺는다.

2. IEEE 802.15.4의 PHY 계층

IEEE 802.15.4 에서는 '그림 1'과 같이 상위 MAC 계층에서 접근할 수 있는 2개의 SAP(Service Access Point)를 제공한다. PD-SAP(PHY Data-SAP)는 PHY 계층을 통하여 데이터를 송수신할 수 있도록 하는 서비스이고 PLME-SAP는 PHY 계층을 제어하는 기능을 담당하는 서비스로서 채널 설정, 채널상의 에너지 검출, Clear Channel 확인, 송수신상태 등을 설정 및 확인할 수 있도록 한다.

상위 계층인 MAC 계층은 오직 PHY 계층에서 제공하는 2 종류의 SAP(PD-SAP, PLME-SAP)를 통해 데이터를 송수신할 수 있고 송수신기의 상태를 설정 및 확인할 수 있다. 결국 PHY 계층에서 제공하는 2 종류의 SAP에 대한 서비스를 시뮬레이션 하여 MAC 계층에게 제공한다면 실제 하드웨어 환경에서 구현될 MAC 계층 S/W를 시뮬레이터 환경에서 실행하여 검증할 수 있다. 또한, 하드웨어 계층에 대한 일관된 인터페이스를 제공한다면 시뮬레이터 환경에서 구현된 MAC 계층 소스코드를 실제 하드웨어에 이식할 때 쉽게 이식 가능하도록 할 수 있다. 3장에서는 PHY 계층에 대한 시뮬레이터 환경을 구현한 후, MAC 계층이 하위 계층을 접근하기 위한 일관된 인터페이스를 제공하여 실제 하드웨어로의 이식이 원활하게 이루어지도록 하는 방법을 제시한다.

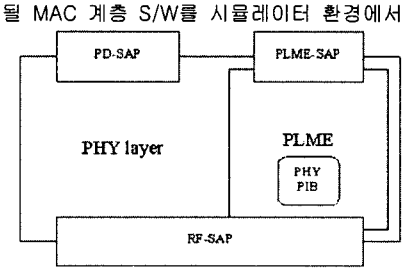

그림 1 PHY 참조 모델

3. PHY Simulator

이 논문에서 목표로 하는 것은 수많은 디바이스가 있는 상황에서 MAC 계층에 대한 구현이 논리적으로 정확하게 동작 하고 동기 신호에 맞는 정확한 타이밍으로 동작 하는지 검증할 수 있는 환경을 제공하기 위한 것이다.

따라서 PHY 계층으로부터 제기될 수 있는 패킷 에러 상황이나 노드들 간의 거리에 대한 신호 감쇠효과, Energy Detection, Link Quality등은 고려하지 않았다.

PHY 계층의 Primitive들 중에서 CCA(Clear Channel Assessment) Primitive는 현재 노드의 채널에 흐르는 캐리어가 있는지 없는지를 나타내며 ED(Energy Detection) Primitive는 현재 채널에 흐르는 캐리어의 신호 세기를 나타낸다. 이러한 Primitive들은 MAC 계층에서 데이터 송수신을 하기 전에 행해지는 채널 충돌 회피전략인 CSMA-CA 메커니즘등 에서 활용된다. 이 두 개의 PHY 계층 Primitive를 간단히 시뮬레이션 하기 위해 MAC 계층이 CCA Primitive를 요청했을 때 시뮬레이

터는 현재 채널 상에 흐르는 데이터가 있을 경우 '참'을, 그렇지 않을 경우 '거짓'을 반환하여 구현하였다. 마찬가지로 ED(Energy Detection)에 대한 MAC 계층의 요구에 대해서도 단지 채널 상에 흐르는 데이터가 있을 경우와 그렇지 않은 경우로 이분화 하여 결과를 나타내 주는 방식을 택하였다.

이러한 제한 사항은 MAC 계층의 논리적인 동작과 정확한 타이밍을 검증하기 위해 필수적인 요소는 아니므로 동작을 검증하는 데에는 문제가 없다고 판단하였다.

3.1 개념도

'그림 2'는 IEEE 802.15.4 PHY 계층에 대한 시뮬레이터인 PHY Simulator의 개념도를 나타내고 있다.

개념도 상에서 가장 위에 있는 'Device'는 각각 하나의 디바이스를 나타낸다. 디바이스에 붙어있는 'PS_Client'는 PHY 계층에 대한 일관된 인터페이스를 디바이스에게 제공하여 PHY 계층을 사용할 수 있도록 한다. 중간에 있는 PS_Data_Structure는 시뮬레이터 상에서 관리해야 할 자료들을 담고 있다. 마지막으로 PS_Daemon은 PS_Data_Structure와 시뮬레이터에 대한 전체적인 동작을 관리한다.

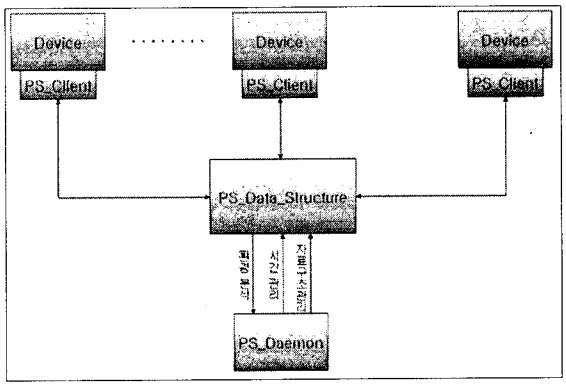

그림 2 PHY Simulator 개념도

이 시뮬레이터의 특징은 아래와 같다.

- 여러 개의 디바이스가 동시에 참여할 수 있다.
- IEEE 802.15.4의 0 ~ 26 Channel을 지원한다.
- Symbol 단위의 시간 정확도로 디바이스를 동기화할 수 있다.
- 시뮬레이션 속도를 조절하여 동작 상황을 파악할 수 있다.

3.2 자료구조

'그림 2'에서 PS_Data_Structure는 시뮬레이터를 위한 자료구조를 나타내며 '그림 3'에 이것에 대한 자세한 구조를 나타내었다. 자료구조는 총 3개의 영역으로 나뉘고 그 특징은 '그림 3'과 같다.

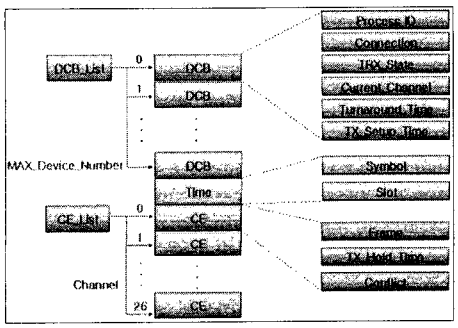

그림 3 PHY Simulator의 자료구조

· DCB List (Device Context Block List)

DCB는 각 디바이스에 대한 상태 정보를 나타내고 DCB에 대한 묶음을 DCB_List 자료구조를 통해 관리한다. '표 1'은 DCB에 들어가는 항목에 대해 나타내고 있다.

항목	설명
Process_ID	디바이스 식별자.
Connection	ON : 디바이스가 존재함. OFF : 디바이스가 존재하지 않음.
TRX_State	RX_ON : Receiver가 켜 있음. TX_ON : Transmitter가 켜 있음. TRX_OFF : Transceiver가 꺼 있음.
Current_Channel	디바이스의 현재 채널.
Turnaround_Time	Transceiver의 상태 전환 시간. Symbol 단위
TX_Setup_Time	전송하기 위해 준비하는 시간. Symbol 단위

표 1 DCB 내용

· T (Time)

현재 시뮬레이터 환경에서의 시간을 나타낸다. 모든 디바이스가 이 항목에서 나타낸 값을 기준으로 시간을 동기화 한다. '표 2'는 T에 들어가는 항목에 대해 나타내었다.

항목	설명
Symbol	현재 시간을 나타냄. Symbol 단위.

표 2 T의 내용

· CE List (Channel Environment)

각 채널별로 현재 전송되고 있는 Frame을 나타낸다.

항목	설명
Frame	현재 전송되는 Frame.
TX_Hold_Time	Frame의 전송이 종료되는 시간. Symbol 단위.
Conflict	전송 중간에 충돌이 발생하면 1로 설정된다. 송신 Device는 이 항목을 보고 Signal을 보낼지 결정한다.

표 3 CE의 내용

3.3 알고리즘

시뮬레이터에 참여하는 모든 디바이스는 PS_Data_Structure 안에 있는 T.Symbol(T 자료구조의 Symbol 항목) 값을 기준으로 시간을 동기화 한다. T.Symbol의 값이 증가하면 시뮬레이터의 시간은 증가하게 되며 PS_Daemon에 의해 관리된다. 사용자는 PS_Daemon의 실행을 제어하여 시간을 정지시키거나 개시할 수 있으며 시간 증가에 대한 속도를 제어할 수 있다. 디바이스들 간에 Frame 송수신을 하거나 채널상의 캐리어 검출을 할 때 또는 채널 상에서 동시 다발적인 디바이스들 간의 Frame 전송으로 충돌이 발생하는 모든 상황들 역시 T.Symbol 값을 기준으로 한다. 아래에 시뮬레이터의 핵심 알고리즘인 데이터 송수신과 CCA Primitive에 대한 시뮬레이터 구현 방안에 대해 설명한다.

· 데이터 송수신

'그림 4'는 특정 채널에 대해 디바이스가 Frame을 송신하는 요청을 보낸 후의 시뮬레이터 처리 과정을 나타낸다. 이 그림에서 '현재 시간'이 Frame 전송을 디바이스가 시뮬레이터에게 요구하는 시간이고 '전송 종료 시간'이 Frame 전송이 종료될 시간을 나타낸다. 시뮬레이터는 '현재 시간'에 Frame 전송 요청을 디바이스로부터 받으면 요청된 Frame의 전송이 종료될

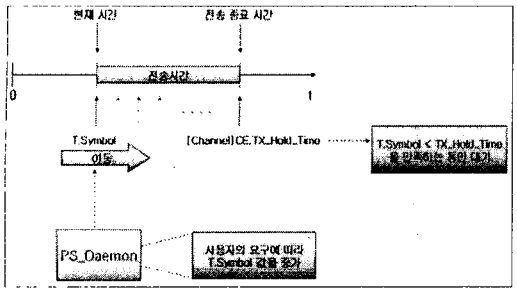

그림 4 송수신 절차

시간을 Symbol 단위로 계산하여 CE_List 자료구조의 TX_Hold_Time에 설정한다. 이어서 시뮬레이터는 현재 시간을 나타내는 T.Symbol값을 TX_Hold_Time과 계속 비교하여 전송 종료 시간에 도달할 때 까지 전송을 중지하고 대기한다.

한편, PS_Daemon은 일정 시간을 주기로 T.Symbol의 값을 증가하여 시간이 흐를 수 있도록 한다. 전송 종료 시간에 도달하면 시뮬레이터는 각 디바이스의 DCB(Device Context Block)를 조사하여 '표 5'의 조건을 만족하는 디바이스에게 Frame을 전송한다.

TRX_State	==	RX_ON	AND
Connection	==	ON	AND
Channel	==	송신 Device의 채널	

표 5 Device가 Frame을 수신할 조건

· CCA(Clear Channel Assessment)

채널 상에 캐리어가 검출되는지의 여부를 판단하는 CCA 과정을 '그림 5'에 나타내었다. '그림 5'의 상황에서 Device_2는 Device_1에 의해 개시된 Frame 전송 중에 캐리어를 검출하는 시도를 나타내고 있다. Device_2가 현재 채널 상에 캐리어의 검출을 판단하기 위해 현재 시간 T.Symbol과 Device_2가 설정되어 있는 Channel에서 진행되고 있는 Frame 전송 종료 시간인 CE.TX_Hold_Time을 비교할 수 있다. 만약 현재 시간이 Frame의 전송 종료 시간을 나타내는 CE.TX_Hold_Time보다 작다면 현재 채널에는 Frame이 전송되고 있다는 것으로 판단할 수 있게 된다. 결국 T.Symbol < CE.TX_Hold_time을 만족한다면 현재 채널에 캐리어가 검출됐다는 결과를 반환할 것이다.

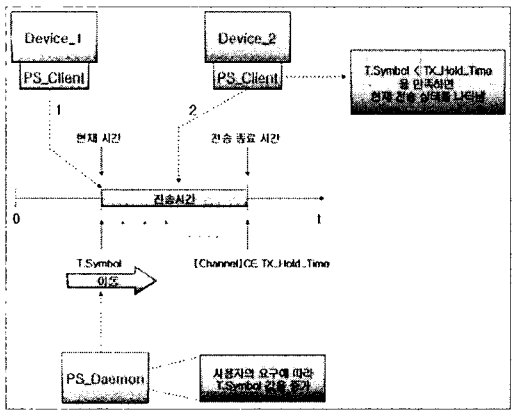

그림 5 CCA 절차

4. 실행 결과

PHY Simulator는 리눅스 운영체제 환경에서 C언어로 구현되었다. 각 디바이스는 리눅스에서 하나의 프로세스와 대응되며 PS_Data_Structure는 리눅스의 IPC(Inter Process Communication)중에서 Shared Memory로 구현하였다. 또한 각 디바이스들 간의 Frame 전송은 리눅스의 Signal과 Shared Memory를 혼합하여 구현되었다.

'그림 6'은 3개의 디바이스를 시뮬레이터 상에 참여시킨 후 시뮬레이터의 정상적인 동작을 검증하는 과정이다. 그림의 상단 왼쪽은 Device_1을 실행시킨 그림이다. 오른쪽 그림은 Device_3를 실행시킨 그림으로써 Device_1 으로부터 전송된 Frame의 수신을 화면에 출력한다. 화면의 아랫부분은 PS_Daemon의 실행 결과를 보인다.

PS_Daemon의 터미널에서 보이는 현재 화면은 시뮬레이터에 참여한 디바이스의 상태 정보인 DCB의 모든 내용을 화면에 출력한다. 이 밖에도 각 채널에 전송중인 Frame의 상태를 출력하는 명령 및 시간의 흐름을 조절하는 명령이 있다.

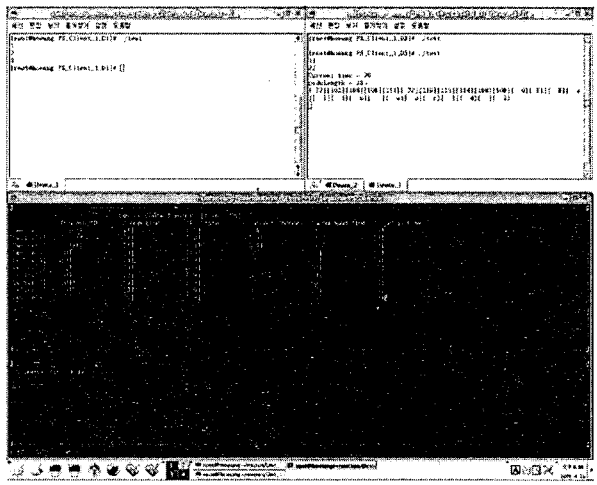

그림 6 시뮬레이터 실행 화면

5. 결론

본 논문에서는 무선 센서 네트워크를 위한 통신 프로토콜인 IEEE 802.15.4의 MAC 계층 구현을 쉽게 검증하기 위한 방법으로 PHY 계층에 대한 시뮬레이터 구현 방법을 제시하였다. 시뮬레이터 환경을 구현하기 위한 자료구조, 알고리즘을 살펴보았으며 실제적인 구현을 통해서 시뮬레이터의 정상적인 동작을 확인하였다. 이 시뮬레이터는 IEEE 802.15.4 프로토콜의 최소 시간 단위인 Symbol 단위로 시간을 동기화 하여 시뮬레이션 하기 때문에 MAC Protocol 구현에 대한 정확한 검증을 할 수 있었다. 또한 시뮬레이터에 접근하기 위한 일관된 인터페이스들은 시뮬레이터 환경에서 검증된 MAC Protocol 구현을 실제 하드웨어 시스템에 쉽게 이식가능하게 할 것이다. 향후 프로토콜 스택뿐만 아니라 상위 어플리케이션까지 통합하여 시뮬레이션 할 수 있는 알고리즘을 연구할 것이다.

6. 참고문헌

[1] IEEE 802.15.4 "Part 15.4:Wireless Medium Access Control(MAC) and Physical Layer(PHY) Specifications for Low-Rate Wireless Personal Area Networks(LR-WPAN)." 2003.

[2] The Network Simulator, http://www.isi.edu/nsnam/ns/

[3] P. Levis, N. Lee, M. Welsh, and D. Culler. "TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications." Proceedings of the First ACM Conference on Embedded Networked Sensor Systems, 126-137, November. 2003.