

# 센서 네트워크에서 데이터 전송 지연을 최소화하는 MAC 프로토콜의 설계

## An MAC Protocol Design in Minimizing of Data Transmission Delay for Wireless Sensor Networks

김만석\*      김상수\*\*      고헌신\*\*\*      조기환\*\*\*\*      이문호\*\*\*\*\*  
Man-seok Kim      Sang-soo Kim      Kwang-shin Koh      Gi-hwan Cho      Moon-ho Lee

### 요 약

배터리를 기반으로 동작하는 센서 노드들로 구성된 센서 네트워크(Sensor Network)는 에너지의 효율적인 사용이 가장 큰 이슈이다. 센서 네트워크의 MAC 프로토콜도 에너지 효율성을 고려하여 설계되고 있으며, 일반적인 접근방법은 listen/sleep 모드를 주기적으로 반복하는 것이다. 그러나, 데이터 전송과정에서 수신 노드의 sleep으로 인하여 즉시 다음 이웃 노드로 데이터를 전송할 수 없는 데이터 전송지연이 필연적으로 발생한다. 본 논문은 에너지 효율성을 고려하면서 데이터 전송지연을 최소화하는 DT-MAC(Data Transmission Centric MAC)을 제안한다. DT-MAC에서 데이터를 수신한 수신 노드는 데이터 수신과 연관된 sleep 모드 중 잔여 sleep 기간을 데이터 전송이 가능한 pseudo\_listen 기간으로 전환한다. pseudo\_listen 기간에서는 데이터 전송이 가능하므로 sleep 모드로 인한 데이터 전송지연을 가능한 최소화 할 수 있다. 따라서 DT-MAC은 에너지 효율성을 제공하면서 재난 및 화재 경보와 같은 실시간 센싱 데이터를 요구하는 센서 네트워크의 응용에 적합한 MAC 프로토콜이다.

### Abstract

The effective power consumption is the primary issue in a sensor network which consists of the sensor nodes with limited battery power. So, most of the MAC protocols in sensor networks have been designed with the consideration of energy efficiency. Generally, these protocols make use of the listen and sleep mode periodically. However, this approach inevitably causes a long transmission delay on the data forwarding path, which is mainly resulted from the sleep time of the receiver node. This paper deals with a design of DT-MAC (Data Transmission centric MAC) protocol, which minimizes the data transmission delay while it forces each node to consume its energy efficiently. Thus, a node received a packet converts its remained sleep time to the pseudo\_listen time, in which the node is able to transmit a packet. With benefit of the pseudo\_listen period, the data transmission delay along with the data forwarding path will be shortened as much as it possible. Therefore, DT-MAC protocol is very suitable to the various applications which require a real time sensing data such as disaster and fire alarm.

☞ Keyword : sensor network, MAC protocol, energy savings, transmission delay, 센서 네트워크, MAC 프로토콜, 에너지 절약, 전송지연

## 1. 서 론

무선 센서 네트워크는 일단의 컴퓨팅 능력과 무선통신 능력을 갖춘 센서 노드를 생활 현장에 배치하여 구성하는 자율적인 네트워크이다. 노드들은 현장에서 획득한 정보를 무선 네트워크를 이용하여 송신하거나 관제센터 등으로부터 수신한 정보에 따라서 반응하는 역할을 수행한다. 무선 센서 네트워크 기술은 산업 전반에서 일상생활에 이르기까지 수많은 분야에 활용되어진다. 특히 군사, 교통, 환경 감시, 의료, 홈 네트워크 등

\* 정 회 원 : 전북대학교 정보보호공학과 수료(박사)  
mseok@bbank.co.kr

\*\* 준 회 원 : 한국산업인력공단, kiss4rang@hanmail.net

\*\*\* 정 회 원 : 성화대학 공학계열 조교수, kskoh@sunghwa.ac.kr

\*\*\*\* 정 회 원 : 전북대학교 전자정보공학부 부교수  
ghcho@chonbuk.ac.kr

\*\*\*\*\* 정 회 원 : 전북대학교 전자정보공학부 교수  
moonho@chonbuk.ac.kr

[2006/07/27 투고 - 2007/09/01 심사 - 2007/12/14 완료]

에서 유용하게 적용 되고 있으며 유비쿼터스 센서 네트워크(USN: Ubiquitous Sensor Network)의 핵심 기술로 인식되고 있다[1].

무선 센서 네트워크의 센서 노드들은 일반적으로 무선 인터페이스를 통하여 구성되고, 다중 홉을 이용하여 데이터 전송을 한다는 점에서 무선 Ad hoc 네트워크[2]를 갖는다. 그러나, 센서 네트워크에서 각 센서 노드들은 극히 제한된 양의 배터리를 기반으로 동작하므로 센서 노드들의 생명주기(lifetime)를 연장하는 것이 가장 큰 쟁점 사항 중에 하나이다. 따라서 무선 센서 네트워크에서는 무선 Ad hoc 네트워크의 MAC(Medium Access Control) 프로토콜과 다른 특성을 고려한 MAC 프로토콜이 요구된다.

무선 센서 네트워크의 대표적인 MAC 프로토콜인 S-MAC(Sensor-MAC)[3]은 채널 접근의 공정성, 충돌 문제 및 전송지연 문제보다도 에너지 효율성을 극대화하기 위해 제안된 프로토콜이다. S-MAC은 센서 노드들이 주기적으로 listen과 sleep를 반복하고, sleep 모드 동안 전원을 차단함으로써 전력 소모를 줄이는 방법을 사용한다. 센싱된 데이터가 발생하여 데이터를 싱크 노드(sink node)까지 전송해야 하는 경우에 센싱 노드는 listen 기간에 한정하여 데이터를 이웃 노드에게 데이터를 전송하고 sleep모드로 전환한다. 데이터를 수신한 이웃 노드는 다음 이웃 노드에게 데이터를 전송해야하는데, 이 노드 역시 sleep 모드로 전환되어 다음 주기의 listen 모드에서 데이터를 전송한다. 즉, 에너지를 절약하게 위해서 적용한 sleep에 의하여 필연적으로 데이터 전송지연이 발생한다.

본 논문에서는 에너지의 효율적인 이용이 가능하면서 기존 S-MAC 프로토콜의 데이터 전송지연 문제를 완화하기 위하여 데이터 전송지향적인 DT-MAC(Data Transmission Centric MAC) 프로토콜을 제안한다. DT-MAC 프로토콜에서 데이터를 수신한 수신 노드는 데이터 수신과 연관된 sleep 모드 중 잔여 sleep 기간을 데이터 전송이 가능한

pseudo\_listen 기간으로 전환한다. 임의 노드는 pseudo\_listen 기간에 데이터 전송이 가능하므로 데이터 전송과정상에 위치한 중간노드의 sleep 모드로 인한 데이터 전송지연을 가능한 최소화할 수 있다. 데이터의 전송지연을 고려하여 설계된 DT-MAC은 에너지 효율성을 제공하면서 재난 및 화재 경보와 같은 실시간 센싱 데이터를 요구하는 센서 네트워크의 응용에 적합한 MAC 프로토콜이다.

본 논문의 구성은 다음과 같다. 2장에서는 S-MAC을 중심으로 기존 MAC 프로토콜에 대해서 기술한다. 3장에서는 본 논문에서 제안하는 DT-MAC에 대해서 상세히 기술하고, 4장에서는 DT-MAC 프로토콜과 기존 MAC 프로토콜과의 성능 분석 결과를 기술하고, 5장에서 결론을 내린다.

## 2. 관련 연구

센서 네트워크 환경에서 에너지 소비 최소화를 목적으로 설계된 대표적인 MAC 프로토콜로 S-MAC 프로토콜[3]을 들 수 있다. S-MAC은 IEEE 802.11[4]과 같이 경쟁을 기반으로 다수의 노드들이 채널을 획득하는 MAC 프로토콜이다. 일반적으로 MAC프로토콜은 무선 네트워크의 채널을 확보하고 hidden terminal 문제를 해결하기 위해 IEEE 802.11에서 적용된 RTS/CTS 제어 메시지를 사용한다. 송신 노드는 데이터 전송 전에 이웃 노드에게 RTS 메시지를 전송하고, 수신 노드는 RTS 메시지에 대한 응답으로 CTS 메시지를 보낸다. 송신 노드와 수신 노드를 제외한 그 주변 노드들은 데이터 전송이 진행되는 동안 대기 상태를 유지한다.

S-MAC의 기본 개념은 PAMAS[5]로부터 파생되었다. 제어 메시지와 데이터 메시지를 위해 2개의 채널을 사용하는 PAMAS와는 다르게 S-MAC은 오직 하나의 공유 채널을 사용하는 차이점이 있다. 센서 노드는 임의 주기로 listen/sleep 모드를

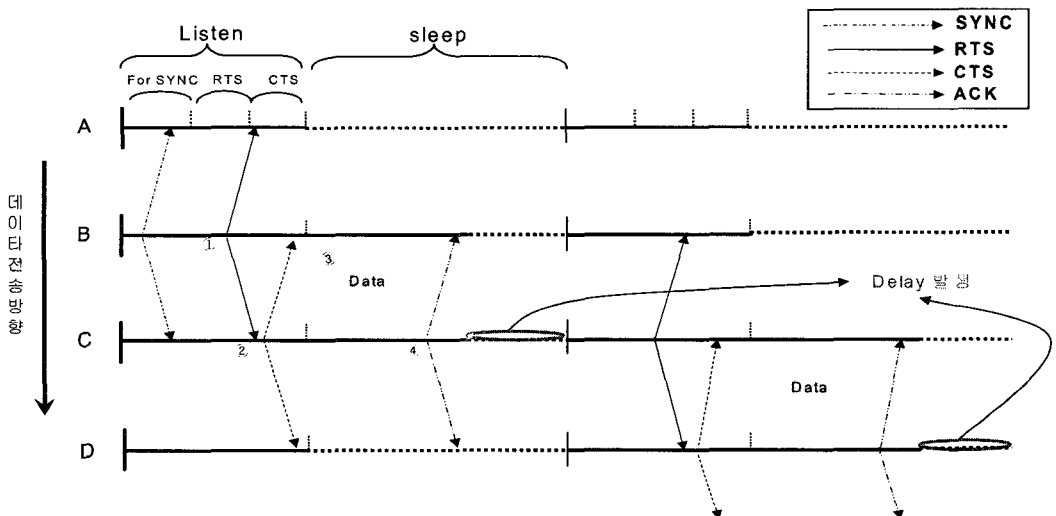
반복한다. listen 주기의 시작 시점에 노드들은 선택적으로 SYNC 메시지를 전송함으로 통신에 참여하는 노드들의 상태 주기를 일치시키는 동기화 작업을 수행한다. 송신노드는 listen 기간 동안에 RTS/CTS를 이용하여 수신노드와 데이터 전송에 사용할 채널을 확보한다. 이때 송신노드는 RTS에 해당 데이터를 전송하기 위해 요구되는 시간을 포함시킴으로써 송수신 노드의 sleep 모드 시작 시점을 동기화한다. 센서 네트워크의 모든 노드들이 SYNC로 동기화한 시점에 sleep 모드로 전환한다. 그러나 송수신 노드는 데이터 전송이 예상되는 시간만큼 listen 상태를 유지하여 데이터를 전달하고 ACK를 기점으로 나머지 기간은 sleep 모드로 전환 한다.

[그림 1]은 노드 B에서 생성된 데이터를 노드 C, D를 경유하여 목적지 노드까지 데이터를 전송하는 과정에서 S-MAC의 동작과정을 보인다. SYNC 메시지에 의한 시간 동기화 이후에 송신노드 B는 수신노드 C에게 RTS 메시지를 전송하고 ① 자신이 목적지인 RTS 메시지를 수신한 노드 C는 그에 대한 응답으로 CTS 메시지를 송신노드 B에게 전송한다②. 잠시 후 노드 A, D는 sleep 모드로 전환하는 반면에 송수신 노드인 노드 B,

C는 RTS에 의해서 동기화된 예상되는 데이터 전송 시간만큼 listen 모드를 유지한다. CTS 메시지를 수신한 송신노드 B는 수신노드 C에게 데이터를 전송하고③, 데이터 수신을 완료한 수신 노드 C는 수신 완료로 의미하는 ACK 메시지를 송신노드 B에게 전송한다④. 이 시점에 노드 B, C는 비로소 sleep 모드로 전환하게 된다.

데이터를 수신 완료한 노드 C는 ①~④ 과정을 반복하여 수신한 데이터를 노드 D로 전송해야 한다. 그러나 노드 D가 sleep 모드로 설정된 관계로 다음 listen 모드로 전환될 때까지 데이터를 전송할 수 없다. 즉 노드 C의 총 sleep 기간 중에 데이터 전송 기간을 제외한 잔여 sleep 기간 동안 (비록 노드 D로 데이터를 전송할 수 있는 충분한 시간임에도 불구하고) 전송지연이 발생한다. 노드 C는 다음 listen 주기까지 기다린 후에 비로소 노드 D로 데이터를 전송하게 된다. 따라서 sleep/listen으로 구성된 한 주기 동안에 다음 홉으로 오직 한번의 데이터 전송이 진행된다. 즉, 에너지 효율성을 고려한 sleep 모드로 인해 데이터 전송지연이 발생한다.

이러한 기존 S-MAC에서의 데이터 전송지연 문제를 개선하기 위해 adaptive listen 기법을 사용



(그림 2) S-MAC의 동작과정

한 adaptive S-MAC[7]이 제안되었다. RTS/CTS 메시지의 NAV(Network Allocation Vector)값을 근거로 자신이 목적지인 데이터를 수신 완료한 노드는 바로 sleep 모드로 전환하지 않고 어느 일정 시간 동안 깨어 전송할 데이터의 유무를 판단하여 만일 전송할 데이터가 존재하면 바로 전송함으로써 어느 정도의 데이터 전송지연을 해결하고 있다. 데이터 전송지연을 해결하기 위해 제안된 또 다른 방법으로 DSMAC(Dynamic Sensor-MAC) [6]이 있다. S-MAC에서의 listen/sleep 주기가 미리 정의된 고정된 값에 의해 반복되는 것에 비해 DSMAC에서는 데이터 트래픽을 고려하여 동적으로 duty cycle(listen 기간과 sleep 기간의 비율)를 변화시킴으로써 데이터 전송지연을 줄이는 방법을 이용하고 있다.

adaptive S-MAC의 경우 단지 이웃 노드와의 데이터 전송 때에만 국지적으로 데이터 전송지연 문제를 해결하고 다수의 노드를 경유하여 데이터를 전송할 경우에는 기존 S-MAC에서 sleep 모드로 인해 제기되는 같은 현상의 데이터 전송지연이 발생한다. DSMAC에서는 데이터 트래픽이 많이 발생한 노드의 경우 어느 정도 지연이 해결되었지만 싱크 노드까지의 데이터 전송을 고려하면 같은 데이터 전송지연이 발생한다. adaptive S-MAC이나 DSMAC의 경우 단대단의 데이터 전송지연은 일부 해결되었지만, 다수의 노드를 경유하는 멀티 홉 (multi-hop) 데이터 전송 환경을 고려하면 원천적으로 전송지연을 해결하지 못하고 있다.

### 3. DT-MAC 프로토콜

DT-MAC은 기본 동작 방식이나 동기화 방법등에서 S-MAC과 거의 유사하지만 데이터 송수신시 기존 S-MAC에서 sleep 모드로 인해 발생하는 데이터 전송지연을 보완하기 위해 제안하는 프로토콜이다. 폭포수(Water Fall) 형태의 데이터 전송을 위해 RTS 및 CTS 메시지를 재정의 하였으며,

FW(Forward Wake-up) 및 FDS(Future Data-Send) 프레임을 새로 추가하였다. 또한, sleep 모드 기간 중에 임의의 노드가 데이터 전송을 위해 active 모드로 전환하는 pseudo\_listen 모드를 정의하였다.

DT-MAC 프로토콜이 폭포수 형태의 데이터 전송을 지원하기 위해서는 다음과 같은 전제 조건이 성립해야한다. 센서 네트워크는 센서 노드와 이를 관리하는 싱크 노드로 구성되어 있다. 싱크 노드는 각 센서 노드의 위치 정보를 수집하여 이들 정보를 센서 노드에게 알려 준다. 이를 통해 각 센서 노드는 다음 경로의 노드 정보를 알고 있다고 가정한다. 또한 센서 네트워크의 토폴로지는 프로토콜이 동작하는 동안에 변화하지 않는다고 가정한다.

#### 3.1 접근 방법

DT-MAC 프로토콜은 hidden terminal 문제를 해결하기 위해 IEEE 802.11의 프레임 구조를 기반으로 IEEE 802.11의 RTS/CTS 메시지와 같은 역할을 수행하는 D-RTS/D-CTS 메시지를 사용한다. 즉, 노드는 데이터를 전송하기 위해 D-RTS 메시지를 이웃 노드에게 전송하고, 자신이 목적지인 노드는 D-CTS 메시지를 전송한다.

프로토콜의 동작은 4단계 핸드셰이크인 D-RTS/D-CTS/DATA/ACK 기반에서 동작하며, 원활한 데이터 전송을 위해 추가적으로 FW 및 FDS 메시지를 정의하였다. FW 메시지는 데이터 전달 경로상의 노드들이 임의의 시점에 sleep 모드에서 깨어날 수 있도록 알리는 메시지며, FDS 메시지는 실제 데이터 메시지의 충돌을 피하기 위해 실제 데이터 전송 전에 보내는 더미 메시지이다. FW 메시지는 D-CTS 메시지를 수신한 노드에 의해서 처음 전송되는 메시지가 전달될 경로에 존재하는 노드에게 전송되는 메시지이다. FW 메시지는 데이터 전달 경로상의 노드에게 연속적으로 전달되며, 이 메시지를 수신한 노드는 자신이 데이터 전달에 관련 되는 시점에 sleep 모드에서 깨

어나, 데이터 송수신을 수행한다. 이때, 데이터 전송에 참여한 각 노드들이 깨어나서 데이터를 전송하는 모형을 형상화하면 마치 계단모양을 유지하면서 연속적인 데이터 전송이 가능함을 알 수 있다.

데이터 전송지연을 최소화하기 위하여 임의의 중간노드는 전송하고자 하는 데이터가 발생하는 경우에, sleep 기간 중에서 데이터 전송에 사용된 시간을 뺀 남아있는 sleep 기간을 pseudo\_listen 모드로 전환한다. sleep모드에서 깨어나 데이터 전송이 가능한 구간으로 전환된 기간을 pseudo\_listen 기간으로 정의하여 D-RTS/D-CTS/DAT/ACK 메시지 전송이 이루어진다. 이 pseudo\_listen 모드는 처음 센싱된 데이터를 전송하기 위해 수행하는 D-RTS/D-CTS/FDS/FW/DATA/ACK 절차와 처리 절차가 구별된다. 목적지 노드로부터 D-CTS 메시지를 수신한 노드는 캐리어 센싱 후 데이터를 전송한다. 같은 시기에 임의의 D-CTS 메시지를 수신한 노드는 임의의 시간 후에 노드들을 깨우기 위해 FW 메시지를 송신하며, FW 메시지를 수신한 노드는 곧 바로 이웃 노드에게 FW 메시지를 전송한다. 이때 임의의 D-CTS 메시지를 송신한 노드는 동시에 데이터 메시지와 FW 메시지를 수신할 수 있다. FW 메시지를 수신한 pseudo\_listen 모드의 노드는 자신이 어느 시점에 sleep에서 깨어야 할지 알 수 있으며 이점이 sleep과의 차이이다. 따라서 충돌에 의해 데이터가 손실될 수 있다. 이러한 데이터의 충돌을 회피하기 위해서 D-RTS 메시지를 송신한 노드는 실제 데이터를 전송하기 전에 의미 없는 FDS 메시지를 전송한 후 실제 데이터를 전송한다.

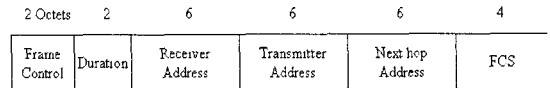
### 3.2 메시지 포맷

DT-MAC 프로토콜은 원활한 데이터 흐름을 제공하고 데이터 전송지연을 최소화하기 위해 기존의 RTS/CTS 메시지에 다음 홉의 경로 지정에 위한 필드 추가와 새로운 메시지 타입인 FW/FDS를

정의한다.

#### • D-RTS/D-CTS

DT-MAC에서는 IEEE 802.11의 메시지 구조에 새로운 필드인 next hop address 필드를 추가한다. D-RTS 메시지의 경우에 duration 필드 값을 이용하여 NAV를 설정한다. D-RTS 메시지를 수신한 노드는 그에 대한 응답으로 D-CTS를 보낸다. D-CTS 송신 노드의 주변 노드들은 이 메시지를 오버히어링(Over Hearing)하며, 오버히어링한 이웃 노드 중에서 next hop address 필드 값과 자신의 주소와 일치하는 노드는 다음 전송 경로의 노드에게 FW 메시지를 송신한다. [그림 2]는 D-CTS 프레임 구조를 도식화하고 있다.



[그림 2] D-RTS/D-CTS 프레임 구조

- duration 필드 : 데이터 전송에 필요한 시간을 알 수 있다. 이 필드 값에 의해 NAV 값을 설정한다.
- next hop address 필드 : D-CTS 메시지를 오버히어링한 노드 중에서 FW 메시지를 송신할 노드를 지정하기 위해서 정의한 필드이다.

#### • FW/FDS

S-MAC에서 실질적인 데이터 전송은 sleep 기간에 수행되며, 다음 홉으로의 연속적인 데이터 전송을 위해서 각 노드들은 자신의 데이터 송수신 시점에 sleep 모드에서 깨어나야 한다. 센서노드에서 싱크 노드까지의 데이터 전달 경로에 존재하는 노드들에게 전송된다. FW 메시지를 수신한 노드들은 duration, current count 필드 값을 기준으로 자신이 언제 깨어날지를 계산하여 sleep 모드에서 pseudo\_listen 모드로 전환한다.

CTS 메시지를 수신한 노드들은 FW 메시지를 전송하고 CTS 송신 노드는 FW 메시지를 오버헤어린다. 이때 D-RTS를 송신한 노드에서 데이터를 전송하거나 전송중이면 데이터와 FW 메시지 간 충돌이 발생할 수 있다. 실제 데이터를 전송할 때 수신 노드에서 FW 메시지와 발생할 수 있는 충돌을 예방하기 위해 무의미한 FDS 메시지를 전송한다. FDS 메시지는 D-CTS 메시지 수신 후에 보내고, 그 후에 실제 데이터 패킷을 송신한다.[그림 3]은 새로 추가된 FW 프레임 구조를 도식화하고 있다.

2 Octets	2	6	6	2	2	4
Frame Control	Duration	Receiver Address	Transmitter Address	Total Count	Current Count	FCS

[그림 3] FW/FDS 프레임 구조

- total count 필드 : 데이터 전달 경로 상의 노드들 중 어느 홉까지 FW 메시지를 전달할지를 결정하는 필드이다. 이 값은 n번째 홉에서 listen 기간동안 FW 메시지를 수신할 수 있도록 해야 하며, 동일한 sleep 기간 내에서 각 홉을 경유하여 전달될 수 있는 최대값을 계산해야 한다.
- current count 필드 : FW 메시지가 현재 몇 번째 홉까지 전달되었는지를 알 수 있는 필드이다. 이 값을 기준으로 데이터 전달 경로상의 노드들은 홉 위치에 따라 각기 다른 시간에 깨어난다.

### 3.3 전송모드 설정

DT-MAC은 이웃 노드와 동기화를 유지하면서 데이터 전송지연을 줄이기 위해서 데이터 전송이 가능한 pseudo\_listen 기간을 정한다. 송신 노드에서 데이터를 전송할 때 데이터 송신과 관련된 각 프레임에서는 필요에 따라 sleep 모드의 어느 일정 시간을 pseudo\_listen 기간으로 전환한다. sleep

모드에서 pseudo\_listen 모드로의 전환은 센싱 데이터가 발생한 소스 노드에서 싱크 노드로 데이터를 전달할 때 전달 경로 상에 존재하는 노드 중에서 FW 메시지를 수신한 노드에서 발생한다. FW 메시지를 수신한 노드는 자신이 어느 시점에 sleep에서 깨어야 할지 알 수 있으며, 그때 깨어나서 데이터 전송에 관련한 후 일정 시간이 경과한 후 다시 sleep 모드로 전환한다. 전송모드의 변화과정을 분석, 설정하기 위한 절차와 방법은 다음과 같다. 먼저 DT-MAC의 특성상 다음 식(1)이 성립한다.

$$\text{프레임 전송 기간} = \text{listen 기간} + \text{sleep 기간} \quad (1)$$

데이터 전송모드에서 소요되는 프레임, listen, sleep 각각의 기간은 [그림 4]에서 도식하고 있다. 이웃 노드와의 동기화를 위해 생성된 SYNC 메시지는 몇 초 후에 sleep 할지를 가리키는 정보를 포함하고 있으며 데이터 전송을 알리는 RTS 메시지는 데이터 전송시 경과되는 시간 정보를 포함한다. 이들 메시지 정보에 의해서 listen과 sleep 기간은 다음 식(2)을 도출할 수 있다.

$$\text{listen} + \text{sleep} = T_{sync} + SYNC_{sleep\_time} + X \quad (2)$$

식(2)에서  $T_{sync}$ 은 listen 주기 시작에서 SYNC 메시지 생성까지 걸린 시간이고,  $SYNC_{sleep\_time}$ 은 SYNC 메시징내의 sleep\_time 필드 값이며, X는 데이터 전송이 없을 때의 실제 sleep 모드를 유지하는 시간이다. 식(2)에 의해 식(3)이 성립한다.

$$X = (\text{listen} + \text{sleep}) - (T_{sync} + SYNC_{sleep\_time}) \quad (3)$$

만일, 노드에 RTS 또는 CTS 메시지가 송수신 되었다면, 다음 식(4)도 성립한다.

$$\text{listen} + \text{sleep} = T_{rts} + \text{duration\_time} + Y \quad (4)$$

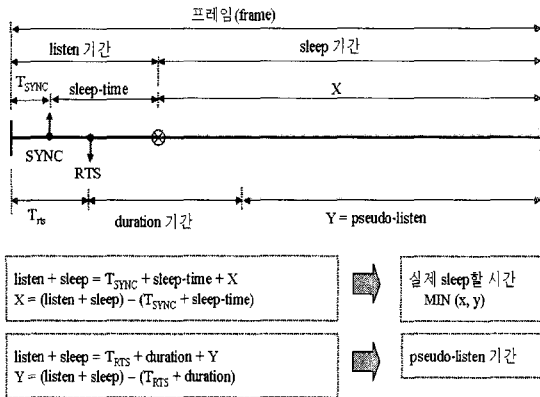
$$Y = (\text{listen} + \text{sleep}) - (T_{rts} + \text{duration\_time}) \quad (5)$$

식(5)에서  $T_{rts}$ 은 처음 listen에서 RTS 생성까지 걸린 시간을 의미한다. 식(3)과 (5)를 통해 구한 X, Y값에 의해 실제 sleep 모드를 유지하는 시간  $T_{sleep}$ 를 구한다.

$$T_{sleep} = \text{MIN}(X, Y) \quad (6)$$

식(6)에 의해 구해진 값으로 특정 sleep 기간을 pseudo\_listen 기간으로 전환한다.

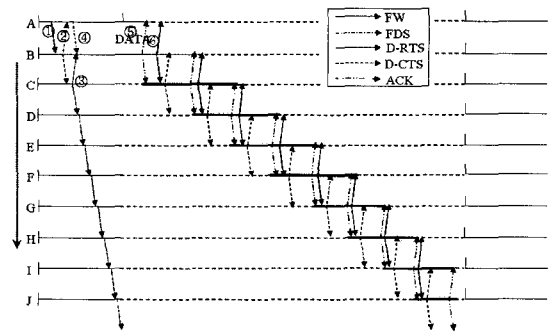
[그림 4]는 데이터 전달시 노드에서 발생할 수 있는 경우와 그에 따른 모드 변환을 위한 계산식의 한 예를 도식하고 있다.



(그림 4) 전송모드 설정 방법

### 3.4 데이터 전송 모형

본 논문에서 제안한 DT-MAC은 listen과 sleep 모드의 적용과 더불어 pseudo\_listen 모드를 추가함으로써 데이터 전송 경로 상에 위치한 노드들 사이에 데이터 전송 과정은 waterfall 구조를 갖는다. [그림 5]는 데이터 전송 과정의 개념적 형상을 예제로 보이고 있다.



(그림 5) DT-MAC의 데이터 전송모형

맨 처음 송신 노드 A는 수신 노드 B에게 D-RTS 메시지를 전송하고(①), 응답으로 수신 노드 B는 송신 노드 A에게 D-CTS 메시지를 전송한다(②). 그리고 송신 노드 B로부터 CTS 메시지를 오버헤어링한 노드 C는 노드 D에게 곧 데이터 전송이 진행될 것을 알리는 FW 메시지를 전송한다(③). 한편 이 시점에 송신 노드 A는 수신 노드 B에서 FW 메시지와 충돌을 고려하여 무의미한 FDS 메시지를 전송한다(④). FDS 메시지를 전송한 송신 노드 A는 일정 시간이 경과한 후에 실제 데이터를 수신 노드 B에게 전송한다(⑤). 마지막으로 데이터 수신을 완료한 노드 B는 노드 A에게 ACK 메시지를 전송한다(⑥). 송신 노드 A에서 수신 노드 B로 데이터를 전송할 때 앞의 ①부터 ⑥까지의 과정을 수행한다.

소스노드에서 싱크노드까지 데이터를 전송할 때, DT-MAC은 이미 다음 홉 경로는 설정되어 알고 있다고 가정한다. [그림 5]에서 노드 B의 D-CTS 메시지를 오버헤어링한 이웃 노드들은 D-CTS 메시지내의 next hop address 필드 값을 확인한다. next hop address 필드 값이 자신의 주소와 일치하지 않는 노드는 기본적으로 D-CTS 메시지를 수신 할 때와 동일한 같은 과정을 수행하고, 자신의 주소와 일치하는 노드는 곧바로 다음 홉의 노드에게 FW 메시지를 전송한다. FW 메시지의 최대 전송 횟수는 listen 및 sleep 시간을 고려하여 산정하며, FW 메시지 필드내의 total count,

current count 필드 값을 기반으로 FW 메시지의 다음 홉으로의 전송 횟수를 제어한다.

노드 A에서 노드 B로 전송된 데이터는 adaptive S-MAC처럼 수행한다. 자신이 전송할 데이터가 있는지 확인 한 후, 만일 전송할 데이터가 있으면 D-RTS/D-CTS/DATA/ACK 4 way 핸드셰이크 과정에 따라 다음 홉인 노드 C로 전송한다. 이 때 C를 포함한 FW 메시지를 수신한 노드들은 자신이 전송에 관여할 시점에 sleep 모드에서 pseudo\_listen 모드로 전환한다. 전환 시점은 FW 메시지내의 current count 필드값과 데이터 전송에 소요된 시간을 근거로 산출한다. 각 노드는 pseudo\_listen 기간에 데이터를 전송한 후 다시 sleep 모드로 전환한다. 노드 B에서 노드 C로의 데이터 전송의 경우와 동일하게 전송지연이 없이 노드 C에서 노드 D로의 데이터 전송도 sleep 지연 없는 데이터 전송이 가능하다. FW 메시지를 수신한 노드에 대해서는 sleep 지연 없는 데이터 전송이 가능하다.

#### 4. 성능 분석

본 절에서는 adaptive listening을 적용하지 않은 S-MAC과 적용한 S-MAC, 그리고 본 논문에서 제안한 DT-MAC을 중심으로 각 프로토콜의 전송지연을 분석한다. 경쟁 기반 MAC 프로토콜을 사용하는 멀티 홉 네트워크에서 고려되는 전송지연으로 carrier sense delay, backoff delay, transmission delay, propagation delay, processing delay, queueing delay 등을 고려할 수 있다. 센서 노드를 위해 설계된 S-MAC에서는 추가적으로 각 노드의 sleep 주기에 의해 발생하는 지연이 존재하며, 이를 sleep 지연이라고 정의한다. 편의를 위해 본 논문에서는 carrier sense delay, transmission delay 그리고 sleep delay에 대해서만 고려한다.

소스 노드부터 싱크 노드까지 N 홉을 경유한다고 가정할 때, carrier sense delay는 각 홉에서

임의적이며 n 홉에서의 값은  $t_{cs,n}$ 으로 나타낸다. 패킷 크기가 고정되어 있다면 transmission delay도 고정적인 값을 가지며  $t_{tx}$ 로 표기한다.

먼저 sleep이 없는 MAC 프로토콜에 대해서 살펴보자. 한 노드가 패킷을 수신하면 그 노드는 즉시 carrier sense을 시작하고 다음 홉으로 패킷을 전송하려고 한다. 이 때 n 홉에서 평균 delay는  $t_{cs,n} + t_{tx}$ 라 정의할 수 있으며, 전체 지연 시간은

$$D(N) = \sum_{n=1}^N (t_{cs,n} + t_{tx}) \text{이다.} \quad (1)$$

따라서, sleep 없는 MAC 프로토콜의 N 홉에서의 평균 지연은

$$E[D(N)] = N(t_{cs} + t_{tx}) \text{이다.} \quad (2)$$

식(2)은 sleep 없는 MAC 프로토콜에서 멀티 홉 지연은 홉 수의 증가와 함께 선형으로 증가함을 알 수 있다. 선의 기울기는 carrier sense 시간에 패킷 전송 시간을 더한 평균값이다.

다음으로 각 홉에서 sleep delay가 있는 S-MAC에서 n번째에서의 sleep delay를  $t_{s,n}$ 로 나타낸다. listen과 sleep의 한 쌍인 프레임의 크기는  $T_f$ 로 정의하며 sleep 기간에 의해 변경될 수 있다.

이 때 n 홉에서의 delay는

$$D_n = t_{s,n} + t_{cs,n} + t_{tx} \text{이다.} \quad (3)$$

adaptive listening을 적용하지 않은 S-MAC에서 carrier sense는 각 프레임이 시작할 때 시작한다. 어느 한 노드가 메시지를 수신한 후, 해당 메시지를 이웃 노드에게 전송하기 위해서는 이웃 노드-다음 홉 노드-가 깨어 있을 때까지 기다려야한다. 이 절차는 다음과 같은 식으로 표현할 수 있다.

$$T_f = t_{cs,n-1} + t_{tx} + t_{s,n} \quad (4)$$

따라서 n 홉에서의 sleep delay는 아래와 같다.

$$t_{s,n} = T_f - (t_{cs,n-1} + t_{tx}) \quad (5)$$



식 (5)와 (3)에 의해

$$D_n = T_f + t_{cs,n} - t_{cs,n-1} \text{ 이 성립한다. (6)}$$

그리고, 한 프레임내에서 패킷이 어느 임의의 시간에 소스 노드로부터 생성될 수 있기 때문에 첫 번째 홉에서는 예외적이다. 즉, 첫 번째 홉에서의 sleep delay  $t_{s,1}$ 은  $(0, T_f)$ 사이의 임의의 값을 가지며,  $(0, T_f)$ 상에서 균일하게 분포한다고 가정하면 이것의 평균값은  $T_f/2$ 이다. 식(6)과 연계하면  $N$  홉에서의 전체 delay는 다음과 같다.

$$\begin{aligned} D(N) &= D_1 + \sum_{n=2}^N D_n \\ &= t_{s,1} + t_{cs,1} + t_{tx} + \sum_{n=2}^N (T_f + t_{cs,n} - t_{cs,n-1}) \\ &= t_{s,1} + (N-1)T_f + t_{cs,N} + t_{tx} \end{aligned} \quad (7)$$

따라서, adaptive listen을 적용하지 않은 S-MAC에서  $N$  홉에 걸친 평균 지연은

$$\begin{aligned} E[D(N)] &= E[t_{s,1} + (N-1)T_f + t_{cs,N} + t_{tx}] \\ &= T_f/2 + (N-1)T_f + t_{cs} + t_{tx} \\ &= NT_f + T_f/2 + t_{cs} + t_{tx} \text{ 이다. (8)} \end{aligned}$$

식(8)을 통해서 각 노드의 sleep 스케줄을 따르는 S-MAC에서 멀티 홉 지연 또한 홉 수의 증가와 함께 선형으로 증가함을 알 수 있으며 선의 기울기는 프레임 길이인  $T_f$ 이다. 이 값은 sleep 없는 평균 지연 값인  $(t_{cs} + t_{tx})$ 보다 크므로 홉 수가 증가 할수록 지연 차이 또한 커진다.

다음으로 adaptive listen을 적용한 S-MAC을 살펴보자. [그림 6]은 멀티 홉 네트워크의 일부분으로 3 개의 홉을  $n$ 에서  $(n+2)$ 로 표기한다.

먼저 노드  $i$ 는 노드  $j$ 가 깨어났을 때 데이터 전송을 위해 carrier sensing을 시작한다. adaptive listen을 적용하지 않은 S-MAC의 노드  $n$ 에서의 지연은 식(3)과 같다.

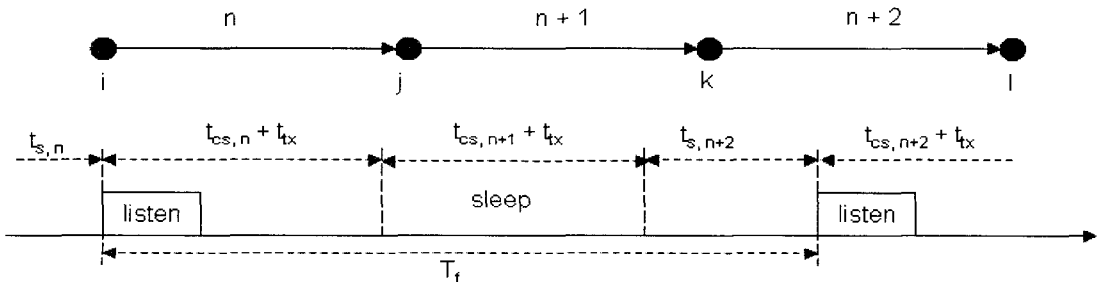
노드  $i$ 와  $j$ 가 RTS/CTS 메시지를 교환하는 동안, 다음 홉 노드인  $k$ 는  $j$ 의 CTS 메시지의 오버헤어링을 통해 노드  $i$ 에서  $j$ 로의 전송이 완료된 후 곧바로 노드  $k$ 는 깨어날 수 있으므로, 노드  $j$ 는 노드  $k$ 로의 데이터 전송을 위해 carrier sense을 시작한다. 이와 같은 방법에 의해서  $(n+1)$  홉에서의 지연은 다음과 같다.

$$D_n = t_{cs,n+1} + t_{tx} \quad (9)$$

이전  $n$  홉에서의 지연과 비교해보면  $(n+1)$  홉에서는 sleep 지연이 없다.

또한 노드 1은 노드  $j$ 로부터 2 홉 거리에 있기 때문에  $j$ 의 CTS 메시지를 들을 수 없다. 그러므로 노드 1은 노드  $j$ 에서  $k$ 로의 데이터 전송이 완료되었을 때 깨어날 수가 없으므로 노드  $k$ 에서 1로의 데이터 전송을 위해서는 노드 1의 listen 모드까지 기다려야 한다.  $(n+2)$  홉에서의 지연은 식 (3)과 같으며 adaptive S-MAC에서 sleep delay는 각 홉마다 발생한다. 따라서  $N$  홉 상의 지연은 다음과 같다.

$$\begin{aligned} D(N) &= t_{s,1} + t_{cs,1} + t_{tx} + t_{cs,2} + t_{tx} + t_{s,3} + \dots \\ &\quad + t_{cs,N-1} + t_{tx} + t_{cs,N} + t_{tx} \end{aligned} \quad (10)$$



[그림 7] adaptive listen적용 때의 sleep 지연

[그림 6]에 의해

$$T_f = t_{cs, n} + t_{tx} + t_{cs, n+1} + t_{tx} + t_{s, n+2} \quad (11)$$

식(10), (11)을 조합하면

$$D(N) = t_{s, 1} + (N/2-1)T_f + t_{cs, N-1} + t_{cs, N} + 2t_{tx} \quad (12)$$

따라서 N 홉의 adaptive listen을 적용한 S-MAC에서의 평균 지연은 다음과 같다.

$$\begin{aligned} E[D(N)] &= T_f/2 + (N/2 - 1)T_f + 2t_{cs} + 2t_{tx} \\ &= NT_f/2 + 2t_{cs} + 2t_{tx} - T_f/2 \text{ 이다.} \end{aligned} \quad (13)$$

adaptive S-MAC의 평균 지연도 홉 수의 증가에 따라 선형으로 증가한다. 단지 선의 기울기는  $T_f/2$ 으로 adaptive listen을 적용 안했을 때 보다 지연이 반으로 줄었다.

마지막으로 본 논문에서 제안한 DT-MAC의 성능을 분석한다. [그림 6]과 같이 adaptive S-MAC은 노드 i, j, k로의 데이터 전송에는 sleep delay가 없지만 노드 k에서 l로 데이터를 전송하고자 할 때에는 sleep delay가 존재한다. DT-MAC은 노드 l과 같이 i로부터 몇 홉 떨어진 노드에서 발생되는 sleep delay를 줄여준다.

먼저 노드 i에서 k까지의 데이터 전송 방법은 adaptive listen과 같으므로 발생하는 지연도 마찬가지로 식(3)과 같다. 노드 j의 CTS 패킷을 오버헤어링한 노드 k는 즉시 다음 홉의 노드 l에게 FW 메시지를 전송한다. FW 메시지를 수신한 노드 l은 노드 k처럼 자신이 sleep 모드의 어느 시점에 깨어야할지 알 수 있다. N 홉에서 (N - 2)개의 FW 메시지를 수신할 만큼 listen 모드가 충분히 크다고 가정하면 지연은 다음과 같다.

$$\begin{aligned} D(N) &= t_{s, 1} + t_{cs, 1} + t_{tx} + t_{cs, 2} + t_{tx} + \dots \\ &\quad + t_{cs, N-1} + t_{tx} + t_{cs, N} + t_{tx} \end{aligned} \quad (14)$$

또, 프레임과 관련하여

$$\begin{aligned} T_f &= t_{cs, n} + t_{tx} + t_{cs, n+1} + t_{tx} + \dots \\ &\quad + t_{cs, N-1} + t_{tx} + t_{cs, N} + t_{tx} \end{aligned} \quad (15)$$

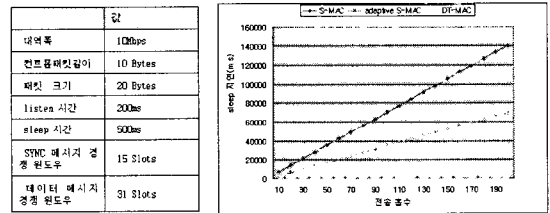
식(14), (15)에 의해서

$$D(N) = t_{s, 1} + (N - 1)T_f \quad (16)$$

N홉의 DT-MAC에서의 평균 지연은

$$\begin{aligned} E[D(N)] &= T_f/2 + (N - 1)T_f \\ &= NT_f - T_f/2 \text{ 이다.} \end{aligned} \quad (17)$$

식(17)에 의하면, sleep에 의한 지연은 발생하지 않고 단지 경쟁에 의한 지연 및 전송지연만 존재한다. 이 같은 경우는 처음 가정처럼 N 홉에서 (N - 2)개의 FW 메시지를 수신할 만큼 listen 모드가 충분히 크다고 가정한 경우이다.



[그림 8] 프로토콜별 sleep 측정 결과 값

S-MAC, adaptive S-MAC, DT-MAC에서 sleep 지연을 비교, 분석하기 위하여, 위에서 산출한 (8), (13), (17) 식에 [그림 7]의 왼쪽 표에서 제시한 파라미터를 적용한 결과가 [그림 7]의 왼쪽 그래프에서 보이고 있다. S-MAC은 센서 네트워크를 구성하는 전송 홉수가 증가할수록 sleep 지연이 비례적으로 증가함을 알 수 있다. adaptive S-MAC은 S-MAC에 비해 sleep 지연이 상대적으로 적지만, sleep 지연 역시 증가함을 알 수 있다. 본 실험의 측정에서 DT-MAC은 한 주기의 sleep 시간동안 FW 메시지가 실험에서 제시된 전송 홉에 모두 전달될 수 있기 때문에 sleep 지연이 발생하지 않는다.

## 5. 결론

저전력을 요구하는 센서 노드들의 특성상 센서 네트워크의 MAC 프로토콜은 에너지 효율적인 관점에서 설계된 프로토콜이 S-MAC이다. S-MAC은 주기적으로 sleep 모드로 전환함으로써 에너지 낭비를 줄였지만, 그에 대한 트레이드오프로 sleep에 의한 데이터 전송지연이 발생한다. 또한 adaptive S-MAC은 S-MAC에 비해 낮은 sleep지연을 보장하지만 여전히 sleep 기간에 따른 긴 데이터 전송지연을 유발한다. 본 논문에서는 이러한 sleep 기간에 의한 데이터 전송지연을 완화하기 위해 DT-MAC을 제안하였고, 비교, 분석 결과 두 가지 다른 S-MAC기법들에 비해 작은 전송지연을 보장하고 있다.

향후 연구과제로는 실제 센서 노드에 DT-MAC을 구현하여 실험 환경을 구축하고 분석하는 것이다. 효율적인 비교 분석을 위해, 실험환경 내에서 센서 노드들의 밀집도를 변경하면서 전달 경로 상의 홉 수에 따른 전송 속도를 분석할 필요가 있다.

## 참고 문헌

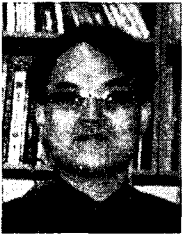
- [1] 백상현, 장민, 장덕현, 조기덕, 최양희, 권태경, “유비쿼터스 센서 네트워크를 위한 무선 센서 네트워크와 인터넷 연동기법,” *Telecommunication Review* 제15권 2호, pp. 337-350, 2005. 4.
- [2] U. Jonsson, F. Aliksson, T. Larsson, P. Johansson and G. Maguire Jr., “MIPMANET Mobile IP for Mobile Ad Hoc Networks,” *IEEE MobiHoc* 2000.
- [3] W. Ye, J. Heidemann and D. Estrin, “An Energy-efficient MAC Protocol for Wireless Sensor Networks,” *INFOCOM* 2002, Vol. 3, pp. 1567-1576, June 2002.
- [4] LAN MAN Standards Committee of the IEEE Computer Society, *Wireless LAN medium access control(MAC) and physical layer(PHY) specification*, IEEE, New York, NY, USA, IEEE Std 802.11-1997 edition, No. 8, pp. 1125-1131, Aug. 1997.
- [5] S. Singh and C. Raghavendra, “PAMAS: Power aware multi-access protocol with signaling for ad hoc networks,” *ACM Computer Communication Review*, Vol. 28, No. 3, pp. 5-26, July 1998.
- [6] P. Lin, C. Qiao and X. Wang, “Medium access control with a dynamic duty cycle for sensor networks,” *IEEE Wireless Communications and Networking Conference*, Vol. 3, pp. 1534-1539, Mar. 2004.
- [7] W. Ye, J. Heidemann and D. Estrin, “Medium Access Control With Coordinated Adaptive Sleeping for Wireless Sensor Networks,” *IEEE/ACM Transactions on Networking*, Vol. 12, No. 3, pp. 493-506, June 2004.

## ● 저자 소개 ●



### 김 만 석(Man-seok Kim)

1988년 한국방송통신대학 전자계산학과 졸업(학사)  
2002년 전북대학교 정보과학대학원 컴퓨터과학과 졸업(석사)  
2005년 전북대학교 대학원 정보보호공학과 수료(박사)  
1982년 전북은행 전산실  
1990년 호주 FNS  
2002년~현재 전북은행 전자금융팀  
관심분야 : IT감사, 정보보안, 컴퓨터통신, 무선인터넷  
E-mail : mseok@jbbank.co.kr



### 김 상 수(Sang-soo Kim)

1995년 2월 목포대학교 전산통계학과 졸업(학사)  
2006년 2월 전북대학교 컴퓨터정보학과 졸업(석사)  
1995년 (주)교보문고 전산실  
2000년 (주)가인정보기술  
2006년~현재 한국산업인력공단 출제실  
관심분야 : 이동컴퓨팅, 컴퓨터통신, 무선 네트워크, 센서 네트워크  
E-mail : kiss4rang@hanmail.net



### 고 광 신(Kwang-shin Koh)

1993년 목포대학교 전산통계학과 졸업(학사)  
1995년 목포대학교 전산통계학과 졸업(석사)  
2005년 전북대학교 전산통계학과 졸업(박사)  
1995년~1997년 광은경영경제연구소 연구원  
1998년~현재 성화대학 공학계열 조교수  
관심분야 : 이동컴퓨팅, 무선 네트워크, 컴퓨터통신  
E-mail : kskoh@sunghwa.ac.kr



### 조 기 환(Gih-wan Cho)

1985년 전남대학교 계산통계학과 졸업(학사)  
1987년 서울대학교 대학원 계산통계학과 졸업(석사)  
1996년 영국 Newcastle대학교 대학원 전산학과 졸업(박사)  
1987년~1997년 한국전자통신연구원 선임연구원  
1997~1999년 목포대학교 컴퓨터과학과 전임강사  
1999년~현재 전북대학교 전자정보공학부 부교수  
관심분야 : 이동컴퓨팅, 컴퓨터통신, 무선 네트워크 보안, 센서 네트워크, 분산처리 시스템  
E-mail : ghcho@dcs.chonbuk.ac.kr



### 이 문 호(Moon-ho Lee)

1967년 전북대학교 전기공학과 졸업(학사)  
1976년 전북대학교 대학원 전기공학과 졸업(석사)  
1990년 일본 Tokyo University 정보통신공학과 졸업(박사)  
1981년~현재 전북대학교 전자정보공학부 교수  
관심분야 : 이동통신, 정보이론, UWB  
E-mail : moonho@chonbuk.ac.kr