

무선 센서 네트워크에서 효율적인 멀티캐스트 라우팅 프로토콜

장동섭*, 조인휘*

*한양대학교 전자컴퓨터통신공학과

e-mail : dongseob.jang@gmail.com, iwjoe@hanyang.ac.kr

An Effective Multicast Routing Protocol in Wireless Sensor Network

Dong-Seob Jang*, In-Whee Joe*

*Department of Electronics & Computer Engineering , Hanyang University

요 약

초기 센서 네트워크의 라우팅 프로토콜은 데이터를 수집하는 업링크 위주의 알고리즘이었다. 그러나 최근에는 특정 노드를 제어하기 위해 토폴로지 정보를 관리하고, 이를 바탕으로 하는 다운링크 알고리즘이 연구되고 있다. 또한 동시에 다수의 노드에게 질의를 전송하는 멀티캐스트 알고리즘의 필요성이 대두되고 있으나 현재까지 센서 네트워크에 적용할 만한 멀티캐스트 라우팅 프로토콜은 거의 전무한 실정이다.

본 논문에서는 비교적 리소스 제한이 적은 게이트웨이나 싱크에서만 관리하던 라우팅 테이블을 메모리가 작은 일반 센서 노드에서도 관리할 수 있게 충분한 작은 비트맵형 라우팅 테이블의 구성을 제안하고, 이를 이용하여 효율적인 유니캐스트 및 멀티캐스트 라우팅 프로토콜을 제안한다. 제안된 프로토콜은 멀티캐스트 상황에서 불필요한 포워딩을 제거함으로써 네트워크 내의 패킷 발생량을 최소화시켜 주는 매우 효율적인 방식이다. 또한 라우팅 테이블의 유지 및 관리가 수월하여 센서 노드에서의 데이터 처리 부담을 줄였다.

1. 서론

초기의 무선 센서 네트워크의 응용은 군용 감시 시스템에서 시작되어 주로 환경 정보 및 상황을 수집하는 시스템이었다. 최근에는 교통 제어를 위한 시스템 및 홈 네트워크 또는 지능형 빌딩을 위한 센서 네트워크와 같은 질의를 통한 제어 분야로 그 활용 범위가 점차로 확대되고 있다. 싱크로의 데이터 수집만을 고려하여 업링크로만 구성되던 무선 센서 네트워크의 라우팅이 이제는 토폴로지 정보를 바탕으로 질의를 통하여 센서 노드를 제어하는 다운링크 프로토콜까지 고려되어야 한다. 그러나 최근까지 센서 네트워크를 위해 제안된 몇 가지의 다운링크 라우팅 프로토콜들에는 완벽하게 양방향으로 데이터를 전송할 수 있는 알고리즘이 제시되지 못한 실정이다. 그나마도 유니캐스트만을 고려하고 있어서, 동시에 다수의 노드를 컨트롤하기 위한 멀티캐스트 기능을 지원하는 라우팅 알고리즘의 개발이 시급한 실정이다¹.

센서 네트워크의 라우팅 프로토콜 요구조건은 다른 통신 네트워크에서 요구되는 것과 동일하다. 그러나 센서네트워크에서는 노드들의 전원, 메모리, RF 반경들이 부족한 경우가 많으며 다른 네트워크와 동일하게 동작을 시킬 경우 이동성의 제약 및 전원과 메모리의 사용량이 증가하는 문제가 발생하여 노드와 네트워크 전체에 심각한 부하를 주게 되므로, 기존의

알고리즘 및 라우팅 데이터 관리정책의 문제점을 근본적으로 해결해야 한다.

본 논문에서는 무선 센서 네트워크에서 사용되는 몇 가지 다운링크 방법에 대해 알아보고, 이들이 멀티캐스트에 적합하지 않음에 대해 설명한다. 그리고 하나의 노드를 한비트에 저장하는 비트맵형 라우팅 테이블을 제안하여 최소한의 메모리 사용과 효율적인 노드의 이동성을 지원하는 알고리즘을 제안한다. 제안하는 알고리즘은 하나 또는 다수의 노드로 데이터 전송 시에 불필요한 플러딩이 전혀 발생하지 않아 유니캐스트 및 멀티캐스트 전송의 효율을 극대화하였다.

2. 관련 연구

현재 사용되는 다운링크 알고리즘으로는 브로드캐스트와 Source Routing 이 일반적이며 좀 더 발전된 형태로는 Connect Source Routing 이 있고, 본 논문과 라우팅 테이블의 구성방법이 유사한 어드레스맵 알고리즘이 있다.

우선 브로드캐스트는 가장 기본적인 통신 알고리즘으로 네트워크 전체로 패킷이 퍼져 나가는 플러딩 방식으로 네트워크 부하가 매우 커서 멀티캐스트뿐만 아니라 유니캐스트에서도 좋지 않은 효율을 보인다.

다음으로 Source Routing[2] 알고리즘은 싱크가 전체 토폴로지 정보를 가지고, 다운링크시 목적지까지의 전달 경로를 데이터에 포함시켜 전송한다. 그러나 전송 경로를 패킷에 포함 시켜야 하므로 목적 노드까지의 홉수가 늘어나면 패킷의 길이가 커져서 전송중 패

본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT 연구센터 지원사업의 연구결과로 수행되었음(IITA-2008(C1090-0801-0047))

킷 손상이 증가하거나, 최대 패킷 길이의 한계에 직면하게 된다.

이를 해결하기 위한 방법으로 제안된 Connect Source Routing[3]은 데이터가 해당 노드에 도착하기 전까지 거쳐간 경로를 유지함으로써 싱크와의 지속적인 통신을 통해 목적지를 찾아가게 된다. 이 알고리즘은 경로 정보를 나누어서 패킷에 담게 되어 홑 수의 제한은 극복했지만 통신량은 Source Routing 에 비해 증가하게 되었다. 그러나 Source Routing 이나 Connect Source Routing 은 유니캐스트를 위한 경로를 패킷에 실는 것 만으로도 벽차기 때문에 소규모 네트워크에 적합하며, 다수의 노드의 경로를 패킷에 담아야 하는 멀티캐스트에 적용시키기란 만만치 않다.

끝으로 어드레스맵[4] 을 이용한 다운링크 알고리즘은 본 논문에서 제안하는 비트맵형 라우팅 테이블과 유사한 방법이지만, 불필요하게 많은 노드 수를 고려하다 보니 라우팅 테이블의 주소 표현에서 노드의 중복이 발생하여 불필요한 플러딩 발생을 야기했다. 어드레스맵에서의 중복 노드가 늘어 날수록 불필요한 통신 량이 늘어나 멀티캐스트에는 적합하지 않다.

3. 제안하는 알고리즘

센서 네트워크에서 사용되는 센서 노드의 리소스는 매우 제한적이어서 현재까지는 게이트웨이나 베이스노드와 같이 비교적 리소스가 풍부한 노드에서 토폴로지 정보를 관리하는 것이 일반적이었다.

본 논문에서는 일반 센서 노드에서 라우팅 테이블을 관리할 수 있도록 노드:비트를 1:1 로 대응시켜 메모리 사용을 최소화 하는 비트맵형 라우팅 테이블 알고리즘을 제안한다. 그리고 제안하는 알고리즘을 이용하여 유니캐스트에서 어드레스맵 라우팅 프로토콜이 가지는 플러딩 문제점을 완전히 해결하였고, 나아가 멀티캐스트 에서도 탁월한 성능을 보여준다.

3.1 비트맵형 라우팅 테이블 표현

현재 센서 네트워크에서 사용되는 노드 식별자는 대부분이 16 비트로 구성되는 NodeID 로써 최대 65536 개의 ID 표현이 가능하다. 그러나 트리 토폴로지 에서 실제로 원활하게 동작할 수 있는 네트워크 구성 노드의 수는 수 백개 정도로 제한된다. 따라서 본 논문에서는 128 바이트의 메모리 공간으로 1024 개의 노드를 관리할 수 있는 라우팅 테이블을 기준으로 하며, 이 크기는 필요에 따라 늘리거나 줄일 수 있다.

본 논문에서 제안하는 라우팅 테이블 구성은 그림(1)처럼 비트맵형으로 나타내며 uint8_t Address[128]로 저장공간을 확보했다. 그림(1)은 NodeID 가 10 인 노드가 자신의 NodeID 를 테이블에 맵핑한 라우팅 테이블이다. 만약, NodeID 가 6 이라면 Address[0]가 0x20, NodeId 가 125 이면 Address[15]가 0x10 으로 설정된다. 식(1)은 Address 계산을 일반화 시킨 식이다.

0 번 노드는 싱크로 사용되므로 테이블 전체의 비트를 0 으로 함으로써 테이블을 초기화 한다.

그림(2)는 업링크 단계에서 형성된 토폴로지를 나타

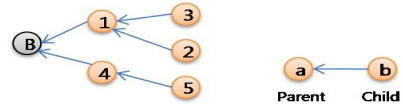
	7	6	5	4	3	2	1	0
Address[127]	0	0	0	0	0	0	0	0
Address[126]	0	0	0	0	0	0	0	0
...
Address[2]	0	0	0	0	0	0	0	0
Address[1]	0	0	0	0	0	0	1	0
Address[0]	0	0	0	0	0	0	0	0

(그림 1) 라우팅 테이블 (NodeID=10)

$$\text{Address} = 1 \ll (\text{NodeID} - 1) \dots\dots\dots \text{식}(1)$$

(단, 1 <= NodeID <= 1023)

3.2 라우팅 테이블 형성 및 관리



(그림 2) 네트워크 토폴로지 예

낸다. 1~5 번의 각 노드는 센서를 통해 수집한 데이터를 그림(2)와 같이 멀티 홑 링크를 따라서 싱크로 전달하며, 업링크를 위해 TinyOS-2.x 의 CTP (Collection Tree Protocol) 프로토콜을 사용한다. CTP 프로토콜은 MCFA (Minimum Cost Forwarding Algorithm) 을 향상시킨 프로토콜로 싱크까지의 최소비용을 계산하여 부모 노드를 선택하는 알고리즘으로 주변의 환경 변화 및 노드의 이동으로 인해 노드간 링크의 비용이 바뀌면, 부모 노드의 링크에 변화가 생기고 이에 따라 토폴로지가 변화하게 된다.

NodeID=B	Address[0]	0	0	0	0	0	0	0	0
NodeID=1	Address[0]	0	0	0	0	0	0	0	1
NodeID=2	Address[0]	0	0	0	0	0	0	1	0
NodeID=3	Address[0]	0	0	0	0	0	1	0	0
NodeID=4	Address[0]	0	0	0	0	1	0	0	0
NodeID=5	Address[0]	0	0	0	1	0	0	0	0

(그림 3) 각 노드의 초기 라우팅 테이블 (Address[127] ~ Address[1] 생략)

그림(2)의 토폴로지에 따른 라우팅 테이블의 형성 과정은 다음과 같다.

- 1) 초기의 각 노드는 그림(3)과 같이 자신의 NodeID 를 맵핑한 라우팅 테이블을 가진다.
- 2) 각 노드는 센싱 데이터 발생시 부모 노드로 전송하며, 자식 노드로부터 데이터를 수신하면 그 노드의 주소에 해당하는 비트를 '1'로 설정하고 다시 부모 노드로 포워딩한다.

NodeID=B	Address[0]	0	0	0	1	1	1	1	1
NodeID=1	Address[0]	0	0	0	0	0	1	1	1
NodeID=4	Address[0]	0	0	0	1	1	0	0	0

(그림 4) 토폴로지에 따른 라우팅 테이블 구성

위 과정을 통해서 형성된 각 노드의 비트맵 테이블을 그림(4)에 나타냈다. 결과적으로 최상위의 노드는 모든 하위 노드들의 주소를 '1'로 설정한 라우팅 테이블이 형성하고, 각 노드들은 자신의 주소와 자신의 모든 하위 노드들의 주소가 저장된 라우팅 테이블을

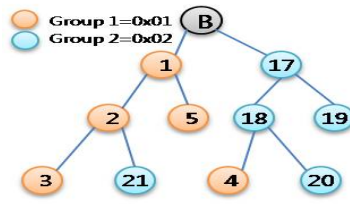
형성한다.

센서 네트워크의 특성상 토폴로지의 변화가 잦기 때문에 주기적인 라우팅 테이블의 업데이트가 필요하다. 토폴로지의 변화에 따라 추가, 소멸, 변경된 링크를 비트맵 테이블의 해당 주소를 '1' or '0'으로 변경하는 것으로 테이블을 관리 할 수도 있으나 토폴로지의 변화가 클 경우 이 처리는 복잡해진다. 따라서 본 논문에서는 실시간 토폴로지 구성되는 임시 테이블을 생성하여 주기적으로 업데이트한다.

3.3 Downlink : Unicast 및 Multicast

다운링크시에 유니캐스트는 위에서 설명한 비트맵형 라우팅 테이블을 이용한다. 싱크가 제어 명령을 수신하면, 상위 노드로부터 명령을 수신한 각 노드는 목적지가 자신이면 명령을 수행하고, 그렇지 않으면 라우팅 테이블과 비교해서 목적 노드가 테이블내에 존재하면 즉, 자신의 하위 노드이면 포워딩한다. 그림(5)에서 3번 노드에게 명령을 전송할 경우,

- 1) 베이스노드로부터 데이터를 수신한 1번과 17번 노드는 우선 자신에게 온 데이터인지를 확인한다. 1번과 17번 모두 자신의 데이터가 아니므로 각각 라우팅 테이블에서 3번 노드를 확인한다. 17번 노드는 3번 Address가 '0'이므로 더 이상 포워딩 하지 않으며, 1번 노드는 해당 Address가 '1'이므로 하위 노드로 포워딩한다.
- 2) 1번으로부터 데이터를 수신한 2번과 5번 노드는 (1)과 같은 과정을 통해 2번 노드만이 수신한 데이터를 포워딩한다.
- 3) 2번으로부터 데이터를 수신한 3번과 21번은 (1)과 같은 과정을 통해서 3번은 수신한 데이터의 목적지이므로 명령을 수행하게 된다.



ID	Routing Table	group
base	0x ... 001F 001F	0x03
1	0x ... 0010 0017	0x03
2	0x ... 0010 0006	0x03
3	0x ... 0000 0004	0x00
4	0x ... 0000 0008	0x00
5	0x ... 0000 0010	0x00
17	0x ... 000F 0008	0x03
18	0x ... 000A 0008	0x03
19	0x ... 0004 0000	0x00
20	0x ... 0008 0000	0x00
21	0x ... 0010 0000	0x00

(그림 5-1) Topology (그림 5-2) Routing table
(그림 5) Downlink and Multicast

그림(5-1)과 같이 두 개의 그룹에 총 11개의 노드로 토폴로지가 구성되어 있을 경우 각 노드가 가지는 라우팅 테이블은 위에서 설명했듯이 그림(5-2)와 같이 구성된다. 그림(5-2)의 그룹은 그룹 테이블로써 멀티캐스트를 위해 사용되며, 형성 과정은 위에서 살펴본 비트맵형 라우팅 테이블의 형성 과정과 같은 알고리즘을 사용한다. 단, 하위 노드의 GroupID만 맵핑하고, 자신의 GroupID는 테이블에 맵핑하지 않는다. 본 논문에서는 한 바이트의 공간을 사용하여 네트워크내의 노드들을 최대 8개의 그룹으로 분류할 수 있도록 하였다. 라우팅 테이블의 크기와 마찬가지로 그룹 테이블도 필요에 따라 조정이 가능하다.

본 논문에서 멀티캐스트는 다음의 세가지 경우를 고려한다.

첫째, 그룹에 상관없이 다수의 노드에게 멀티캐스트하는 방법을 제안한다. 유니캐스트 다운링크 과정과 비슷하지만 다수의 목적노드 ID와 라우팅 테이블을 비교하여 자신의 주소를 제외하고 하나라도 맵핑되어 있으면 포워딩한다. 그림(5)에서 2, 3, 18, 19번 노드로 명령을 전송할 경우,

- 1) 1번 노드의 테이블에서 2와 3을 찾을 수 있고, 17번 노드의 테이블에서 18과 19를 찾을 수 있으므로 1과 17노드는 둘 다 포워딩한다.
- 2) 1번으로부터 데이터를 수신한 2번은 테이블에서 3번을 찾을 수 있으므로 포워딩하고 자신이 목적지 중 하나이므로 명령을 수행한다.
- 3) 17번으로부터 데이터를 수신한 18번과 19번은 둘 다 목적지 이므로 명령을 수행하며, 테이블내에서 18번은 2, 3, 19를, 19번은 2, 3, 18을 찾을 수 없으므로 포워딩하지 않는다.

둘째, 특정한 그룹에 속한 노드에게 멀티캐스트하는 방법을 제안한다. NodeID에 상관없이 해당 그룹이면 명령을 수행한다. 그리고 그룹 테이블을 검색하여 해당 그룹에 속하는 노드가 테이블내에 맵핑되어 있으면 하위 노드로 포워딩한다. 그림(5-1)의 토폴로지는 같은 그룹의 노드 일지라도 서로 다른 서브 트리가 될 수 있음을 보여주고 있지만, 제안하는 알고리즘으로 잘못된 플러딩없이 제어 하고자 하는 그룹의 노드를 정확히 찾아가게 된다. 그림(5)에서 Group1에 속하는 노드에게 명령을 전송할 경우,

- 1) 1번 노드와 17번 노드 모두 그룹 테이블이 0x03으로 목적 그룹 0x01이 그룹 테이블에 맵핑되어 있으므로 하위 노드로 포워딩하며, 1번 노드는 목적 그룹에 속하므로 명령을 수행하고, 17번 노드는 목적 그룹에 속하지 않으므로 명령은 수행하지 않는다.
- 2) (1)의 과정과 동일하게 2, 3, 5번 노드는 데이터를 수신해 명령을 수행하고, 3번과 5번 노드의 그룹 테이블은 0x00이므로 더 이상의 데이터 송신은 일어나지 않는다. 그리고 18번 노드의 그룹 테이블은 0x03으로 4번 노드에게 데이터가 전달되고, 4번 노드는 이를 받아 명령을 수행할 수 있게 된다.

셋째, 연속된 NodeID로 지정한 범위의 노드에게 멀티캐스트하는 방법을 제안한다. 시작 ID와 마지막 ID를 명시해서 데이터를 전송함으로써 시작 주소부터 마지막 주소까지에 해당하는 노드들이 데이터를 수신하여 명령을 수행하게 된다. 그림(6)은 2~5번 노드로 멀티캐스트할 때 1번 노드에서의 라우팅 테이블과 멀티캐스트데이터의 테이블 비교를 보여준다.

위에서 살펴본 바와 같이 본 논문에서 제안하는 알고리즘은 최소한의 메모리를 사용하여 비트맵형 테이블을 구성하고, 유니캐스트 및 멀티캐스트에서 불필요한 플러딩을 제거함으로써 패킷 발생량을 최소화하여 네트워크 부하를 줄였으며, 네트워크 혼잡이 줄어들어 패킷 충돌도 줄일 수 있다. 그리고 라우팅 테이블의 업

데이트 및 검색이 빠르고 간단해서 센서 노드에서의 데이터의 처리 지연을 줄였다.

NodeID 1 0000 0000 0001 0000 0000 0000 0000 0000	5 3 2
Multicast(2~5) 0000 0000 0000 0000 0000 0000 0000 0000	4 1

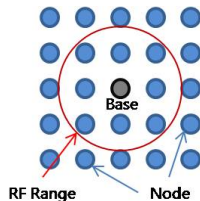
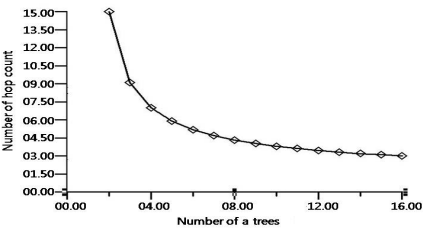
(그림 6) 멀티캐스트를 위한 라우팅 테이블 비교

4. 구현 및 결과

본 알고리즘의 테스트 플랫폼은 전자부품연구원 (KETI)에서 개발한 K-Mote 를 50 개 사용하였다. K-Mote 는 MSP430-1611 MCU 를 사용하며 RF 모듈로 2.4GHz 대역의 CC2420 칩을 사용한다. 기본적인 업링크 알고리즘은 TinyOS-2.x 에서 제공하는 CTP 를 사용하였으며, 각 센서 노드의 센싱 주기는 1 초이며 업링크는 5 번의 센싱 후 즉, 매 5 초 마다 이루어 진다. 라우팅 테이블의 업데이트는 10 초 주기로 하였다.

어드레스맵[4] 라우팅 프로토콜에서 나타난 것처럼 식(2)에서 h 는 총 노드 수에 대한 완전한 a 진수의 트리 구조를 가질 때의 평균적인 홉수를 나타내며, 그림(7)은 65536 개의 노드에 대한 완전한 a 진수 트리 구성에 따른 평균적인 홉수를 나타낸다.

$$h = \log_a \left(\sum_{k=0}^h a^k \right) - 1 \dots\dots\dots \text{식(2)}$$



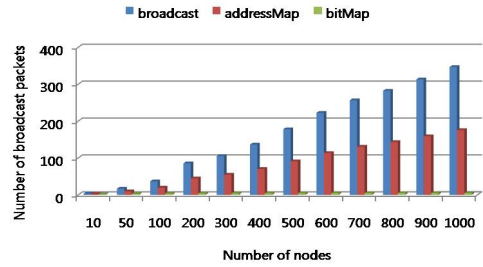
(그림7)hop count of tree's structure (그림 8) arrangement

일반적으로 그림(8)과 같은 노드 배치를 가질 때 구성되는 토폴로지는 2 ~ 10 진 트리로 구성된다. 이 때의 유니캐스트와 멀티캐스트에 대한 평균 패킷 발생량을 그림(9)와 그림(10)에 나타냈다. 네트워크에서 토폴로지에 따른 NodeID 의 구성은 매우 다양하다. 따라서 그림(9)와 (10)은 패킷 발생량에 있어서 최상의 NodeID 배치와 최악의 NodeID 배치를 모두 고려하여 그 평균량을 나타낸 것이다. 브로드캐스트방식의 기본 패킷 발생량은 전체 노드 수와 같지만, 그림(9)와 (10)에서는 터미널 노드들의 패킷 송신이 발생하지 않도록 한 개선된 브로드캐스트방식에서의 패킷 발생량이다.

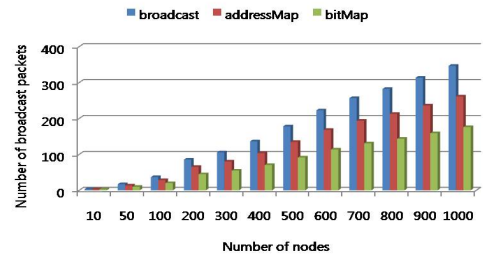
그림(9)에서와 같이 유니캐스트의 패킷 발생량에서 본 논문의 알고리즘은 홉 수만큼의 패킷 발생량을 보이고 있다. 이는 플러딩이나 어드레스맵 방식과 비교했을 때 매우 효과적인 알고리즘임을 증명하고 있다. 멀티캐스트의 경우 다수의 노드가 전체 노드 일수도 있음을 감안한다면, 본 논문의 알고리즘은 그림(10)에서 보여지는 것처럼 멀티캐스트에서의 패킷 발생량도 최소화 할 수 있다.

테스트 결과는 유니캐스트 에서 평균 90%, 멀티캐스트에서 평균 80%의 성공률을 보였다. 이는 패킷 손

실에 의한 것으로 노드간 재전송을 구현하지 않았기 때문이다.



(그림 9) Broadcast packets of Unicast



(그림 10) Broadcast packets of Multicast

5. 결론

센서 네트워크에서의 패킷량은 에너지 소모를 줄이는 문제와 함께 고려해야 할 사항 중 하나이다. 그러나 대부분의 기존 다운링크알고리즘은 이 문제를 크게 언급하지는 않았다. 본 논문에서는 비트맵형 라우팅 테이블 구성하여 플러딩으로 인한 패킷 발생을 완전 제거 함으로써 효과적으로 목적 노드를 제어할 수 있는 알고리즘을 제안하였다.

결국 본 논문의 알고리즘은 적은 메모리를 사용하여 리소스가 제한적인 일반 센서 노드에서 라우팅 테이블을 관리하고, 이를 이용하여 그림(9)과 (10)에 나타난 것처럼 최악의 노드 배치를 고려하더라도 유니캐스트및 멀티캐스트에서 패킷 발생량을 최소화하여 무선 센서 네트워크 환경에 효율적인 다운링크라우팅 방법이라 할 수 있다.

참고문헌

- [1] Kemal Akkaya, Mohamed Younis, "A Survey on routing protocols for wireless sensor networks" Ad Hoc Networks 3 (2005) 325-349
- [2] David B. Johnson, David A. Maltz and Yih-Chun Hu, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks", Internet-Draft, draft-ietf-manet-dsr-09.txt. Work in progress, April 2003.
- [3] 안태원, 조인휘, "무선 센서 네트워크를 위한 다운링크 라우팅 알고리즘" 한국통신학회 추계 종합 학술발표회 논문 초록집 p.133, November 2005.
- [4] 안태원, 조인휘, "무선 센서 네트워크에서의 효율적인 양방향 라우팅" 한국정보과학회 한국 컴퓨터 종합 학술대회 논문집(A), p.137~141, April 2006.