

무선센서네트워크 환경에서 계층적 라우팅 프로토콜의 성능개선을 위한 Short-cut 라우팅 알고리즘

강문경* · 진교홍*

*창원대학교

E-mail : ilovemk7942@nate.com, khjin@chanwon.ac.kr

요 약

무선센서네트워크 환경에서 활용될 수 있는 대표적인 계층적 라우팅 프로토콜로는 Zigbee(지그비) 계층구조 라우팅과 IETF의 6LoWPAN WG에 드래프트로 제출되었던 HiLow 프로토콜 등이 있다. 이들 계층구조 라우팅 프로토콜은 깊이 우선(Depth-first) 방식과 폭 우선(Width-first) 방식을 각각 사용하여 자동 주소 할당을 수행한 후 이를 이용해 계층적 경로 배정이 되도록 한다. 이에 따라 망상의 노드들이 물리적으로 가까운 위치에 존재하더라도 경로는 계층적으로 제공되어 데이터 전달에 많은 홉 수가 소요된다. 본 논문에서는 이러한 경로 배정 시 발생하는 비효율성을 개선하고 싱크(Sink) 주위 노드의 에너지 소모율을 줄일 수 있는 경로 배정 알고리즘을 제안한다.

키워드

WSN, Hierarchical routing, Zigbee, HiLow, SCRO

1. 서 론

무선센서네트워크(WSN, Wireless Sensor Network)는 유비쿼터스 컴퓨팅 분야의 핵심 인프라 기술 중의 하나로서 사물 및 환경 정보의 수집이 요구되는 지역에 설치된 다수의 센서노드들이 스스로 네트워크를 구축하고 싱크노드(Sink Node)에게 정보를 전달하는 네트워크이다.

WSN은 Application-Specific한 특징을 보이며, 다음과 같은 사항이 요구된다. 먼저 센서노드는 배터리 사용이 필수적이므로 최대한 에너지 소모를 줄여야 하며, 많은 센서노드가 배치될 수 있도록 낮은 가격을 유지하여야 한다. 또한 센서노드들의 개별 관리가 어려우므로 자동 주소 할당 기능을 통해 주소 지정과 경로배정이 가능해야 한다[1, 2].

저속 WPAN의 표준인 IEEE 802.15.4는 제한된 출력과 성능을 갖는 단거리 무선 통신으로 폭넓은 응용에 사용될 수 있도록 설계되었다[3]. 이는 WSN 환경을 구축하기에 가장 적합한 기술로 평가되고 있다. 현재 WSN은 IEEE 802.15.4의 PHY와 MAC 계층 위에 Zigbee Alliance[4]에서 설계한 Network 계층과 Transport 계층을 올려 센서 네트워크 응용을 구현하고 있다. 한편, IETF의 WG인 6LoWPAN에서는 Adaptation 계층을 추가하여 기존 인터넷 프로토콜인 TCP, IPv6 프로토콜을 그대로 유지하면서 WSN을 구현하는 방안을 연구하고 있다[5].

자동주소 할당과 이를 통한 경로배정이 가능한 계층구조(Hierarchical) 라우팅 프로토콜에는 Zigbee(지그비) 네트워크의 깊이 우선(Depth-first) 계층구조 라우팅과 6LoWPAN의 폭 우선(Width-first) 계층구조 라우팅 HiLow[3]가 있다. 이들 라우팅 기법은 센서노드에 16bit Short Address를 할당하여 헤더 사이즈를 축소함으로써 노드의 오버헤드를 줄인다. 라우팅 테이블을 사용하지 않고 Address만을 이용하여 경로를 설정한다. 그러나 계층적 어드레싱(Addressing) 과정에서 결정된 경로 때문에 직접 전송이 가능한 가까운 거리의 노드이더라도 계층적 경로에 따라 정보전달이 이루어져 많은 홉 전달이 필요하다. 또한 노드의 고장이나 연결 에러가 발생하게 되면 계층적 어드레싱을 다시 수행해야 하므로 관리 메시지가 많이 발생될 수 있다[6].

본 논문의 구성은 다음과 같다. II장에서는 Zigbee 네트워크의 계층구조 라우팅과 6LoWPAN의 계층구조 라우팅 프로토콜인 HiLow에 대해 간단히 살펴보고, III장에서는 제안된 알고리즘을 소개하며, IV장에서는 제안된 기법의 시뮬레이션 및 성능을 분석하고 평가하였다. 마지막으로 V장에서는 결론 및 향후 연구 계획을 기술하였다.

II. 관련 연구

2.1. Zigbee 계층구조 라우팅

Zigbee 계층구조 라우팅은 어드레싱 과정과 경로배정 과정으로 구성된다. 어드레싱 과정에서 망상의 모든 노드는 계층구조로 연결될 다른 노드로부터 16bit Short Address를 할당받는다. 노드의 연결은 IEEE 802.15.4의 Association Request Command와 Association Response Command를 설정한 메시지를 이용한다. 식 (1)을 통해서 깊이가 d 인 노드들의 주소 Offset 값인 $C_{skip}(d)$ 를 구한 후, 식 (2)에 대입하여 할당할 16bit Short Address를 계산한다[4].

$$C_{skip}(d) = \begin{cases} 1 + C_m \times (L_m - d - 1), & \text{if } = 1 \\ \frac{1 + C_m - R_m - C_m \times R_m^{L_m - d - 1}}{1 \times R_m}, & \text{otherwise} \end{cases} \quad (1)$$

$$A_n = A_{parent} + 1 + C_{skip}(d) \times (n - 1) \quad (2)$$

- C_m : 최대자식의 개수
- L_m : 네트워크트리의 최대 깊이
- R_m : 자식으로 가질 수 있는 최대 Zigbee 라우터 개수
- d : 현재노드의 깊이
- $C_{skip}(d)$: d 의 노드에게 참여하고 있는 자식들의 짧은 주소 차이
- A_n : n 번째 자식노드의 주소
- A_{parent} : 현재노드의 부모노드 주소
- n : n 번째 자식노드

경로배정 과정은 노드 간의 연결 정보를 Neighbor Table에 등록해 놓고, 이 정보들을 이용하여 소스노드에서 목적지노드까지 메시지를 전송할 수 있도록 다음 노드를 지정한다.

$$A < D < A + C_{skip}(d - 1) \quad (3)$$

식 (3)은 현재노드가 가질 수 있는 자식노드의 주소 범위를 나타낸다. 식 (3)의 조건을 만족하면 목적지 노드가 현재노드의 자식노드 중에 하나라는 것을 알 수 있다. 현재노드는 식 (3)을 자식들에게 재귀적으로 적용시켜 자신의 자식들 중 센서노드를 자식으로 가질 것으로 유추되는 노드에게 메시지를 전달한다. 식 (3)을 만족하지 않으면 부모노드에게 전달한다.

2.2. HiLow

HiLow도 어드레싱 과정과 경로배정 과정으로 구성된다. 어드레싱 과정은 Zigbee 계층구조 라우팅과 비슷하지만 식 (4)와 같이 Zigbee와는 다른 어드레싱 공식을 사용하여 16bit Short Address를 할당한다[3].

$$AC_n = MC \times A_p + n \quad (4)$$

- AC_n : 현재 할당된 자식의 수가 $n-1$ 개 일 때 다음 할당되는 자식노드의 주소
- A_p : 주소를 할당하는 부모노드의 주소
- MC : 최대 자식노드의 수(Max Children)

HiLow의 경로배정 과정은 다음과 같다.

Find_NextHop() 함수는 다음 홉으로 설정할 노드를 찾는데 이용하고, k 노드의 부모 또는 자식 중에 깊이 D 를 갖는 노드는 Find_Ancester_D(D, k) 함수를 이용해서 찾는다. Find_NextHop() 함수는 현재 노드가 목적지노드의 부모노드인 경우와 자식노드인 경우, 그리고 그 외의 경우로 나뉘어서 다음 홉을 지정한다.

Zigbee 계층구조 라우팅과 다르게 HiLow는 깊이에 대한 변수가 고려되지 않기 때문에 네트워크의 크기가 제한되지 않는다. 이와 같이 어드레싱 공식이 더 간단하다는 특성이 있지만, 결과적으로 두 계층구조 라우팅 알고리즘의 성능이 비슷하기 때문에 어플리케이션에 맞게 선택하여 사용된다.

III. 제안된 알고리즘

3.1. 이웃 리스트 (Neighbor List, NL)

노드 자신과 통신 가능한 거리 POS (Personal Operating Space) 내에 위치한 노드들을 Neighbor Node라 하고, Neighbor Node들의 정보목록을 Neighbor List(이하 NL)라고 정의한다. 제안된 알고리즘은 2.1절에서 언급되었던 Neighbor Table에 부모노드와 자식노드뿐만 아니라 이웃노드도 인식할 수 있도록 Relationship Field에 Neighbor를 추가한다. 어드레싱 과정에서, 각 노드는 다른 노드들이 교환하는 메시지들을 Overhearing하여 이웃노드의 정보를 NL에 추가한다.

3.2. 리다이렉트 응답 (Redirect_ACK)

제안된 알고리즘은 노드가 계층적 경로뿐만 아니라 이웃노드의 정보도 이용하여 소스노드에서 목적지노드까지의 최단경로를 찾으려 한다. 하지만 목적지노드의 정보를 가지는 센서노드가 다수인 경우, 목적지노드에 다른 경로를 지나온 메시지가 중복으로 도착한다. 이를 방지하기 위해 Redirected Acknowledge(이하 Redirect_ACK)라는 새로운 프레임 타입을 도입한다. 프레임 형식은 Acknowledge의 메시지 형식에 소스 주소(Source Address)와 시퀀스 넘버 (Sequence Number)가 추가된 형태이다.

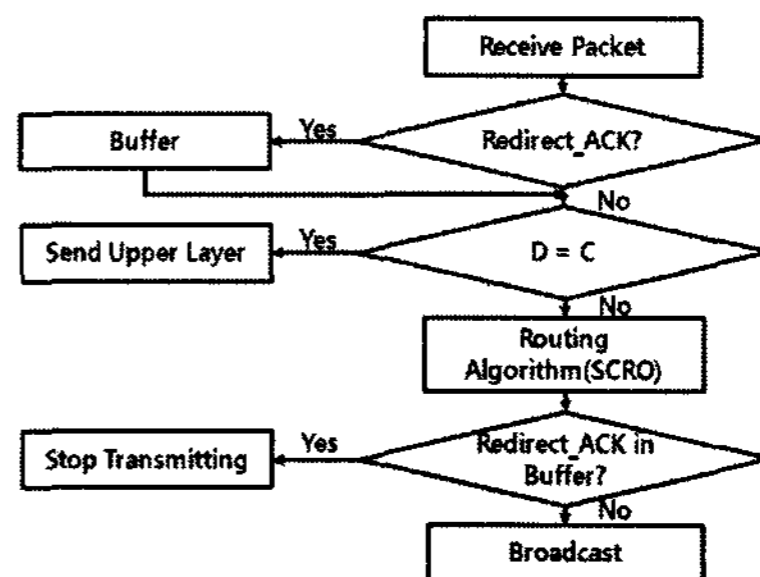


그림 1. 수신 패킷 처리 과정

그림 1은 노드가 Overhearing하여 얻은 메시지의 처리 과정이다. 먼저, 메시지를 수신하면 프레임 형식이 Redirect_ACK인지 확인한다. Redirect_ACK인 경우 버퍼에 저장 해두고 다음 항목으로 넘어간다. Redirect_ACK가 아닌 경우는 바로 다음 항목으로 넘어간다.

현재노드가 목적지노드라면, 현재노드의 상위 계층으로 메시지의 내용을 보내고 적절한 처리를 한다. 현재노드가 목적지노드가 아니라면, 본 논문에서 제안된 라우팅 알고리즘 SCRO를 이용하여 가장 적당한 다음 홉 노드를 찾는다.

메시지 전송 전에 반드시 보내려는 메시지에 대한 Redirect_ACK가 버퍼에 존재하는지 검토하고, 없으면 메시지를 브로드캐스트 한다. 보내려는 메시지에 대한 Redirect_ACK가 버퍼에 존재한다면 전송을 중지하고, 보내려는 메시지를 삭제한다.

3.3. 라우팅 알고리즘

라우팅 알고리즘 SCRO(Short-cut Routing Algorithm Improving Hierarchical Routing Protocol)은 3.1절의 NL과 3.2절의 Redirect_ACK를 이용하여 계층구조 라우팅의 문제점인 Sub-optimal 경로설정을 개선하여 Short-cut Routing이 가능하도록 한다.

```

function Find_NextHop()
if (D is belong to current node's NL)
send Redirect_ACK
next_hop = D
else
next_hop = Find_NearestNode()
return next_hop
end function

function Find_NearestNode()
loop(i=0;i<DC-Dc;i++)
F = Find_Ancestors_D(DC-i, C)
if (F is belong to current node's NL)
nearest_node = F
end if
end loop
loop(j=0;j<DD-Dc;j++)
F = Find_Ancestors_D(DD-j, D)
if (F is belong to current node's NL)
nearest_node = F
break
end if
end loop
return nearest_node
end function

function Find_Ancestors_D(D, k)
find an ancestor of a descendant of node k
depth D
end function

F : Family node address
NL : Neighbor List
DC : Depth of current node
DD : Depth of Destination node
Dc : Depth of center node
i, j : Random variable
    
```

그림 2. 제안된 라우팅 알고리즘 SCRO

그림 2은 SCRO의 Pseudo Code이다. SCRO 라우팅 알고리즘은 Find_NextHop() 함수, Find_Ancestors_D(D,k) 함수와 새로운 Find_NearestNode() 함수로 구성된다. Find_NextHop() 함수는 목적지노드가 현재노드의 NL에 존재하면 목적지노드가 다음 홉이 되고, 존재하지 않으면 Find_NearestNode() 함수를 이용하여 다음 홉을 찾게 된다. Find_NearestNode() 함수는 루프를 사용하여

현재노드의 NL에 계층 경로상의 노드들이 존재하는지 찾고, 그 중 가장 목적지노드에 가까운 노드를 선택하여 다음 홉으로 지정한다. 우선순위가 가장 낮은 노드는 부모노드이며, 이는 HiLow의 경로배정 알고리즘의 결과와 동일하다.

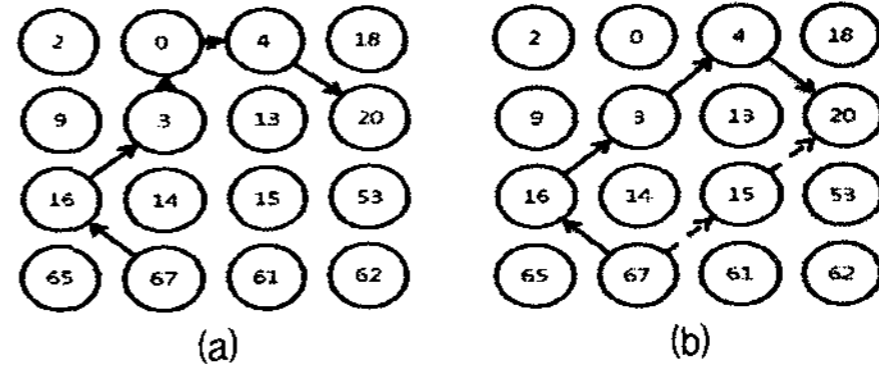


그림 3. 알고리즘 동작 예
(a)HiLow, (b)제안된 알고리즘(실선), Redirect_ACK를 이용한 제안된 알고리즘(점선)

그림 3의 (a)는 계층적 경로를 나타내고, 그림 3 (b)의 실선은 이웃노드만을 이용한 SCRO이며 점선은 Redirect_ACK도 이용한 SCRO의 예이다. 그림 3에서 보는바와 같이 Redirect_ACK 기능을 포함한 SCRO가 최단거리의 경로배정을 보여준다.

노드 67이 노드 20에게 메시지를 전송하려면 노드 67의 이웃노드들은 메시지를 Overhearing하여 그림 1와 같이 메시지를 처리한다. 노드 15는 자신의 NL에 노드 20이 존재하므로 다른 노드들에게 Redirect_ACK를 브로드캐스트 하고, 노드 20에게 메시지를 직접 전송한다.

IV. 시뮬레이션 및 성능 분석

컴퓨터 시뮬레이션으로 제안된 라우팅 알고리즘의 성능을 기존 계층 구조 라우팅과 비교 분석 하였다. 시뮬레이션은 C언어를 이용하였으며 시뮬레이션의 모델은 그림 4와 같다.

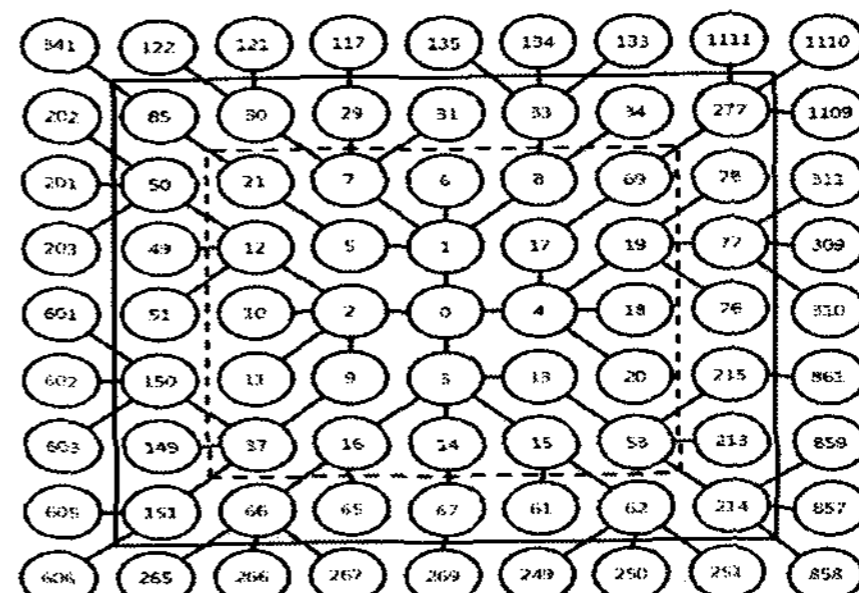


그림 4. (5x5),(7x7),(9x9)의 시뮬레이션 모델과 계층구조 (노드 0이 코디네이터이다.)

5x5 네트워크는 점선으로, 7x7 네트워크는 실선으로 각 네트워크의 경계를 표시하였다. 노드들의 위치는 행과 열의 그리드 형태를 유지하고, 각 네트워크마다 왼쪽-상단을 (0, 0)으로 가정하였다.

POS 범위는 대각선에 위치한 노드까지로 가정하였으며, 시뮬레이션 모델의 노드 주소는 HiLow의 동적 주소 할당 메커니즘을 따른다. 할당받은 16bit Short Address에 대한 계층구조를 노드 사이의 실선으로 나타내었다.

Zigbee 네트워크의 변수 L_m 과 C_m 이 5이고, R_m 이 4인 Zigbee 네트워크는 MC가 4인 HiLow 네트워크와 동일한 계층구조를 가지기 때문에 계층구조 라우팅에 따라 소스노드에서 목적지노드까지의 홉 수 계산결과가 같다. 시뮬레이션 시나리오는 노드들이 소스노드와 목적지노드가 되는 모든 경우에 대한 홉 수와 각 노드별 메시지 전송 횟수를 알아본다.

표 1. 실험 결과 (홉 수 비교)

	Coordinator Position (x,y)	Ideal Routing	Zigbee & HiLow	SCRO
5	Center (2,2)	2.266	3.277	2.266
	Corner (0,0)	2.266	3.693	2.342
5	Edge (0,2)	2.266	3.930	2.460
7	Center (3,3)	3.219	4.555	3.318
	Corner (0,0)	3.219	5.126	3.573
7	Edge (0,3)	3.219	5.574	3.910
9	Center (4,4)	4.163	5.866	4.461
	Corner (0,0)	4.163	5.548	4.876
9	Edge (0,4)	4.163	7.271	5.454

표 1은 네트워크의 크기와 코디네이터의 위치에 따른 Ideal Routing, Zigbee & HiLow 그리고 SCRO에 대한 평균 홉 수를 나타낸다. 테이블의 Ideal Routing 항목은 실제 목적지노드까지의 최단 홉 수이다. Zigbee & HiLow는 Ideal Routing과 약 1홉 정도의 차이가 나는데 비해 SCRO는 비슷한 홉 수를 가진다. 5x5에서 9x9로 네트워크 크기를 확장함에 따라 Ideal Routing 홉 수와의 차이도 더 커진다는 결과를 확인할 수 있다.

또한 코디네이터의 위치에 따라 다른 결과를 확인할 수도 있다. 특히, 코디네이터가 네트워크의 중앙에 위치할 때, 제안된 라우팅 알고리즘의 성능이 가장 좋다.

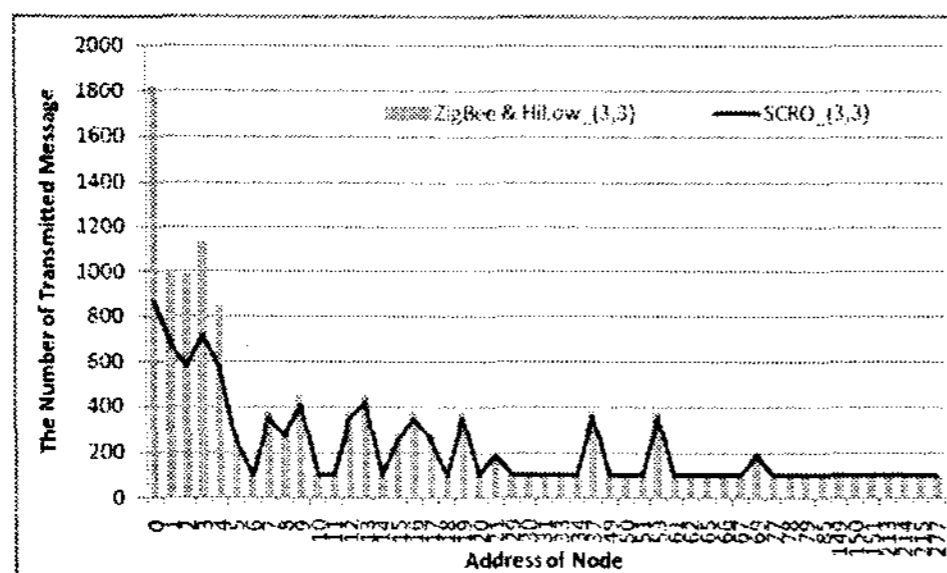


그림 7. 각 노드별 전송 횟수 비교 (7x7 모델, 코디네이터는 0이고 위치는 (3, 3)이다.)

그림 7은 네트워크의 크기가 7x7이고,

코디네이터의 위치가 중앙인 (3, 3)의 경우로 코디네이터의 실제 위치는 그림 4에서 확인할 수 있다. 각 노드별 메시지 전송 횟수를 Zigbee & HiLow의 계층구조 라우팅과 제안된 라우팅 알고리즘 SCRO를 비교하고 있다.

Zigbee & HiLow의 계층구조 라우팅은 각 노드별 메시지 전송 횟수가 노드 0과 그 주위노드에 집중되어 있는 양상이다. 제안된 기법은 특정 노드에서 Zigbee & HiLow의 계층구조 라우팅보다 적은 메시지 전송 횟수를 가진다. 특히 노드 0에서 55%의 감소율을 보여주고 있다. 이것은 SCRO는 코디네이터와 그 주위노드들의 에너지 소모가 효율적이라는 것을 보여준다.

V. 결론 및 앞으로의 연구

제안된 라우팅 알고리즘 SCRO는 계층구조 라우팅을 기반으로 하여 적은 메모리와 낮은 에너지 소모율을 가진다. NL의 노드관계에 Neighbor라는 개념을 추가하고 라우팅의 중복을 방지하는 새로운 메시지 프레임 Redirect_ACK의 사용으로 계층구조 라우팅의 고정된 경로가 최단경로를 지원하도록 보완하였다.

시뮬레이션을 통해 SCRO를 검증한 결과 시뮬레이션 시나리오 상에서 생성되어 전송되는 망상의 전체 메시지 횟수, 노드별 메시지 전송 횟수, 소스노드에서 목적지노드까지의 홉 수가 Zigbee & HiLow의 계층구조 라우팅 알고리즘에 비해 우수한 성능을 보였다.

참고문헌

- [1] 박준성, 임채성 외 5명, "IPv6 기반의 유비쿼터스 센서 네트워크(USN) 기술," 한국정보처리학회, VOL. 13, NO. 04, pp. 79-87, July. 2006.
- [2] 임채성, Waleed Mansoor 외 4명, "IPv6 기반 센서 네트워크(6LoWPAN)을 위한 라우팅 프로토콜 기술," 전자공학회지, 제33권, 제8호, pp.854-863, Aug. 2006.
- [3] K. Kim, Ed., J. Park, S. Daniel Pard, Ed. and J. Lee, "Hierarchical Routing over 6LoWPAN," draft-daniel-6lowpan-HiLow-hierarchical-routing-00.txt(work in progress), Jun. 2007.
- [4] IEEE Computer Society, "IEEE Std. 802.15.4-2003," October. 2003.
- [5] N. Kushalnagar, G. Montenegro and C. Schumacher, "6LoWPAN : Overview, Assumptions, Problem Statement and Goals," IETF(RFC4919), Feb. 2007.
- [6] 유승화, 김희정, 김선호, 남동일, 유제택, 정명호, "RFID 및 USN에 IPv6 적용방안 및 활용 분야에 관한 연구," 한국전산원, Dec. 2005.