

논문 2007-44TC-10-4

# Zigbee 네트워크 계층 구조에서의 효율적인 주소 할당 방법

## (Effective address assignment method in hierarchical structure of Zigbee network)

김재현\*, 허수정\*, 박용완\*\*, 강원석\*\*\*, 이동하\*\*\*

(Jaehyun Kim, Soojung Hur, Yongwan Park, Wonsek Kang, and Dongha Lee)

### 요약

IEEE 802.15.4를 기반으로 하는 Zigbee 센서 네트워크에서는 전송되는 패킷에는 2byte의 지역 주소가 존재한다. Zigbee 센서 네트워크 내에서 이 2byte의 주소를 가지고 각 센서의 주소를 식별한다. 이러한 Zigbee 센서 네트워크는 많은 센서들을 하나의 네트워크로 구성할 수 있어야 ubiquitous 시대를 뒷받침 할 수 있는 솔루션으로 자리매김할 수 있다. 그러나 Zigbee 계층 구조에서 표준에 제정되어 있는 알고리즘인 Cskip 알고리즘을 이용한 주소 할당 방식에는 많은 수의 센서를 배치하는 것에 문제가 있다. Cskip 알고리즘을 이용하여 주소를 할당하기 위해서는 라우터나 코디네이터가 가질 수 있는 자식노드의 최대 개수, 자식노드가 라우터로 될 수 있는 최대 개수, 네트워크의 최대 깊이를 알아야 한다. 이러한 네트워크는 유연성이 없고 공간적인 측면으로 봤을 때 주소 할당에 있어서도 많은 낭비를 초래하게 된다. 본 논문에서는 이러한 주소 공간의 낭비를 최소화 할 수 있는 주소할당 알고리즘을 제안한다.

### Abstract

Zigbee sensor network base on IEEE802.15.4 has local address of 2 byte on transmit packet data which is pick up the address for each sensor node. Sensor network is requested low power, low cost, many nodes at hues physical area. There for Zigbee is very good solution supporting for next Ubiquitous generation but the Zigbee sensor network has address allocation problem of each sensor node. Is established standard from Zigbee Alliance, to the address allocation method uses Cskip algorithm. The Cskip algorithm use the hazard which allocates an address must know Hop of the maximum modification and child node number. There is to address allocation and from theoretically it will be able to compose a personal 65536 sensor nodes only actual with concept of space, only 500 degree will be able to compose expansion of the low Zigbee network. We proposed an address allocation method using coordinate value for Zigbee sensor network.

**Keywords :** Wireless Sensor Network, Sensor Routing Protocol, Address Allocation, Zigbee Sensor Network

### I. 서론

현재 센서 네트워크는 u-IT839정책의 중심에 서 있는 기술로써 앞으로 ubiquitous 시대를 이루기 위한

핵심적인 기술이 될 것이다. 이러한 센서 네트워크는 인간이 미치는 모든 공간에서뿐만 아니라 인간의 손이 닿지 않는 공간에 대해서도 센서를 이용하여 데이터를 수집하고, 그 수집한 데이터를 가지고 사람들이 살아가는데 있어서 더욱 많은 정보를 줄 것이다. 예를 들어 산에 산불 감시 센서를 많이 설치할 하여 특정 지역에 약간의 불이 날 경우 즉시 알람이 울려서 그 지역에 불을 꺼서 산불을 예방하거나, 어느 지역의 바람이나 온도, 습도 등 날씨에 관련된 정보를 센서를 이용하여 수집하여 기상청에 보내 주어서 더욱 정확한 날씨 정보를 알 수 있도록 하는 시스템을 만들 수 있다. 그러기 위해서는 우선에 많은 센서들을 사용해야 하기 때문에 센서

\* 학생회원, \*\* 정회원-교신저자, 영남대학교 정보통신 공학과

(Dept. of Information and Communications Engineering, Yeungnam Univ)

\*\*\* 정회원, 대구경북과학기술원 (Daegu Gyeongbuk Institute of Science & Technology)

※ 본 연구는 과학기술부 및 대구경북과학기술연구원의 연구개발사업의 일환으로 수행하였음

접수일자: 2007년8월27일, 수정완료일: 2007년10월18일

의 가격 문제가 해결되어야 한다. 그리고 많은 센서를 설치하기 때문에 한번 설치하면 교체하는데 대한 비용도 고려해야 하므로 네트워크의 수명도 길어야 한다. 또한 센서들이 설치되는 장소가 산이나 바다와 같이 아주 악조건의 환경일 수 있기 때문에 센서를 보호해 주기 위한 주변 장치도 고려해야 한다. 이러한 문제점들을 감안하였을 때 우선적으로 네트워크가 어떤 일을 할지를 확실히 정의한다. 그리고 이에 따라 메인으로 쓸 칩을 결정해야 한다. 예를 들어 어느 한 지역의 온도를 측정하여 그에 따른 어떤 처리를 해 주는 시스템을 만들고자 한다. 그러면 센서 노드에서 할 일은 일정한 시간 간격으로 온도를 센싱하여 그 값을 Zigbee 통신을 이용하여 부모 노드에게 전달해 주면 된다. 이것은 적은 메모리와 간단한 처리로 수행할 수 있기 때문에 메인 칩을 설정할 때 고사양의 칩을 설정하지 않아도 된다. 만약 많은 기능을 담고 있는 칩을 사용하여 이 시스템을 설계한다면 칩의 가격이 올라가게 되고, 소비 전력도 높아져서 배터리에도 영향을 끼치게 된다. 그래서 앞으로 센서 네트워크에서 중요하게 생각해야 하는 것 중 하나가 각각의 시스템마다 노드의 작업량이 얼마나 되는지를 정확히 파악하여 칩을 선택하는 것이다. 그리고 메인 칩과 Zigbee 통신 모듈을 하나의 칩으로 만들어 주는 SOC(System On Chip) 기술과 접목할 필요가 있다. 현재 SOC 솔루션으로는 텍사스인스트루먼트(TI)에서 개발한 위치 추적 엔진이 있다. 그리고 Zigbee 통신 모듈인 CC2420과 8051 마이크로 컨트롤러를 하나의 칩으로 만든 CC2431이 있다. 또한 라디오 펄스에서 만든 8051 마이크로 컨트롤러와 Zigbee 통신 모듈을 하나의 칩으로 만든 MANGO가 있다. 이렇게 SOC 기술과 Zigbee 통신을 접목시키면 소비 전력이 줄어서 네트워크의 수명을 연장할 수 있다. 그리고 센서의 가격과 크기를 줄일 수 있어 효과적인 센서 네트워크를 구축할 수 있다.

현재 무선 센서 네트워크에서 Zigbee 통신이 많이 고려되어지고 있다. 그 이유는 저 전력이며 많은 센서 노드들이 하나의 네트워크를 구성할 수 있기 때문이다. 그러나 Zigbee 센서 네트워크에서 주소를 할당하는 방식으로 Cskip 알고리즘이 있다. 분산 주소 할당방식 중의 하나인 Cskip 알고리즘을 쓰기 위해 3가지의 고정된 파라미터를 알고 있어야 한다. 이 3가지 파라미터는 라우터나 코디네이터가 가질 수 있는 자식노드의 최대 개수, 자식노드가 라우터로 될 수 있는 최대 개수, 네트워크의 최대 깊이이다. 여기에서 라우터나 코디네이터가

가질 수 있는 자식노드의 최대 개수가 2개이고, 자식노드가 라우터로 될 수 있는 최대 개수가 2개라고 하면 네트워크의 최대 깊이는 16이 된다. 이렇게 논리적으로 형성된 네트워크에 실제적으로 센서를 배치하게 되면 낭비되는 주소 공간이 많은 것을 알 수 있다. 본 논문에서는 많은 센서 노드들을 하나의 네트워크로 구성할 수 있도록 하는 주소할당 알고리즘을 제안한다. II장에서는 Zigbee 네트워크 계층에서 어떤 일을 하는지에 대해 알아보고 III장에서는 Zigbee의 계층 구조 라우팅에서의 문제점에 대해 분석하고 이 문제점을 해결할 수 있는 알고리즘을 제안하고 IV장에서는 제안한 알고리즘과 기존의 Zigbee의 라우팅 알고리즘의 성능을 비교하고, V장에서는 결론 및 향후 연구 과제에 대해 기술하도록 하겠다.

## II. Zigbee의 네트워크 계층과 문제점

### 1. Zigbee의 네트워크 계층

Zigbee 표준에서 제정되어 있는 라우팅 알고리즘으로써 IEEE 802.15.4 MAC이 beacon-enable mode일 경우 트리 라우팅 기법을 이용하고, non-beacon mode일 경우에는 매쉬 라우팅 기법을 사용하도록 권장하고 있다. Zigbee 표준에서 정의 하고 있는 Zigbee 네트워크 계층의 업무는 다음과 같다.<sup>[3]</sup>

- ① 상위 계층에서 내려온 데이터에 상황에 맞는 네트워크 식별자를 부착하여 하부 계층으로 전달한다.
- ② 하부 계층으로부터 올라온 데이터의 페이로드 부분을 분리하여 상위 계층으로 전달한다.
- ③ 멀리 떨어진 노드까지 데이터가 전달될 수 있도록 멀티 홉 라우팅 경로를 설정한다.
- ④ Zigbee 디바이스가 시작될 때 통신 초기화를 수행한다.
- ⑤ 새로운 네트워크를 구성할 수 있는 능력을 지원한다.
- ⑥ 존재하는 Zigbee 네트워크에 새로 가입하거나 이미 가입한 네트워크를 떠날 수 있는 능력을 지원한다.
- ⑦ 만약 자신이 Zigbee 코디네이터와 라우터로써 동작된다면 자신에게 가입하는 일반 디바이스의 16bit 주소를 새로 설정할 수 있어야 한다.
- ⑧ 주변에 존재하는 이웃 노드들의 검색 및 관리를 할 수 있어야 한다.

기본적으로 트리 라우팅 기법은 분산 주소 할당 기법에 의해 동작한다. 이 분산 주소 할당의 하나인 Cskip 알고리즘은 트리 라우팅이 가능하도록 의도적인 16bit 주소를 설정한다. 아래에서 어떻게 Zigbee 센서 네트워크에서 주소 할당이 이루어지는지 보여준다. 우선 Cskip 알고리즘을 사용하려면 다음 정보가 각 노드들에게 정의 되어야한다.

- ①  $C_m$  : 코디네이터나 라우터가 가질 수 있는 최대 자식 개수
- ②  $L_m$  : 트리 토폴로지의 최대 깊이
- ③  $R_m$  : 자식으로써 갖게 되는 최대 라우터의 개수
- ④  $d$  : 지금 노드가 속한 깊이

이렇게 정의된 값들을 이용하여 각 노드에 대해 논리적인 주소를 할당하는데 다음 식 (1)과 (2)에 의해 주소를 할당한다.

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

$$A_n = A_{paren} + Cskip(d) \times R_m + n \quad (1 \leq n \leq C_m) \quad (2)$$

- $R_m$  : 자식으로 가질 수 있는 Zigbee 라우터 개수
- $R_n$  :  $n$ 번째 라우터의 번호  $0 \leq R_n \leq R_m - 1$
- $C_m$  : 한 노드가 가질 수 있는 최대 자식의 개수
- $L_m$  : 네트워크 트리의 최대 깊이
- $A_n$  :  $n$ 번째 노드의 네트워크 주소
- $A_{paren}$  : 부모노드의 주소
- $d$  : 현재 노드의 깊이

깊이가 결정되어 있고, 최대 자식의 개수와 노드가 가질 수 있는 최대 라우터의 개수가 정해져 있을 때 식 (1)과 (2)를 이용하여 주소를 분산적으로 할당 할 수 있다.

이렇게 주소를 할당했을 때 어느 한 라우터가 자식 노드로부터 통신하고자 하는 목적지 주소를 받았을 때 식 (3)이 성립을 하면 자신의 다른 자식들에게 이 데이터를 전송하고 성립하지 않으면 이 값을 부모 노드에게 전송한다. 이 데이터를 받은 부모 노드는 이 같은 작업을 반복함으로써 트리 라우팅이 이루어진다.

$$A_r < D < A_r + Cskip(d-1) \quad (3)$$

- $A_r$  : 라우터 자신의 주소
- $D$  : 목적지 주소

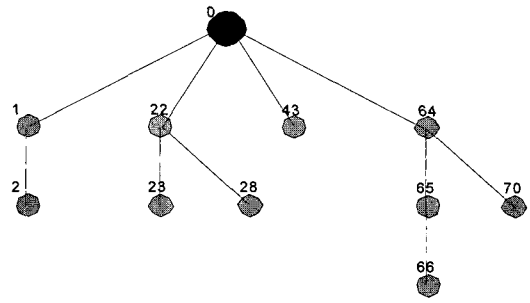


그림 1.  $C_m = R_m = 4$ ,  $L_m = 3$  일 때의 주소 할당  
Fig. 1. Address allocation when  $C_m = R_m = 4$  and  $L_m = 3$ .

표 1.  $C_m = R_m = 4$ ,  $L_m = 3$  일 때의 Cskip(d)  
Table 1. Cskip(d) when  $C_m = R_m = 4$  and  $L_m = 3$ .

D	Cskip(d)
0	21
1	5
2	1
3	0

그림 1에서는 최대 자식 노드의 개수가 4이고, 노드가 가질 수 있는 최대 라우터의 개수가 4이고, 네트워크의 최대 깊이가 3일 때의 주소가 할당된 예를 보여 주고 있다. 이 예제에서 볼 때 현재의 깊이  $d$ 가 0일 때 Cskip(0)의 값이 식 (1)에 의해서 21이 된다(Table 1 참조). 코디네이터는 0번으로 주소가 할당되어 자식 노드의 주소를 할당할 때 첫 번째 자식 노드의 주소를 자신의 주소에 1을 더해서 주소 값을 1로 할당을 한다. 두 번째 자식의 주소 값을 Cskip(0)값인 21에 자신의 주소인 1을 더해서 22를 할당하는 방식으로 첫 번째 깊이의 주소 값을 할당한다. 두 번째 깊이의 주소 값을 할당할 때 Cskip(1)값을 이용하여 첫 번째 깊이에 있는 노드의 주소를 할당할 때와 같은 방식으로 할당한다. 다음 그림 2에서는  $C_m$  값과  $R_m$ 의 값이 틀릴 때 예를

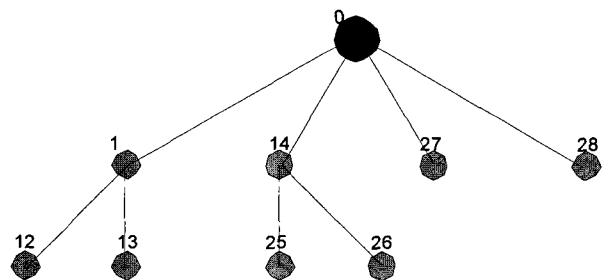


그림 2.  $C_m = 4$ ,  $R_m = 2$ ,  $L_m = 3$  일 때의 주소 할당  
Fig. 2. Address allocation when  $C_m = 4$ ,  $R_m = 2$  and  $L_m = 3$ .

표 2.  $C_m = 4, R_m = 2, L_m = 3$  일 때의  $Cskip(d)$   
 Table 2.  $Cskip(d)$  when  $C_m = 4, R_m = 2$  and  $L_m = 3$ .

d	$Cskip(d)$
0	13
1	5
2	1
3	0

보여준다.

그림 1과 2에서 보듯이 최대 라우터의 개수가 줄어들어  
 으으로써  $Cskip(0)$ 의 값이 줄어드는 것을 알 수 있다.

### 2. Zigbee 주소할당의 문제점

센서 네트워크에서는 수많은 센서가 하나의 네트워  
 크에서 동작하는 경우가 생길 수 있다. 그러나 이론적  
 으로 봤을 때 최대 자식노드로 가질 수 있는 개수가 2  
 개이고, 최대 라우터의 개수가 2개라고 가정하였을 때  
 그 네트워크의 최대 깊이는 15를 넘을 수가 없다. 그 이  
 유는 자식노드가 2인 계층 구조를 이룬다면 깊이가 1일  
 때 노드의 개수는 3개, 깊이가 2일 때는 7개, 이렇게 2  
 의 승수로 노드 수가 증가하기 때문에 깊이가 15일 때  
 총 노드의 개수가 65536개가 되어 16비트로 구분 할 수  
 있는 개수를 다 채워버리게 된다. 이 말은 Zigbee 센서  
 노드의 통신 거리가 30m라고 한다면 코디네이터에서부  
 터 15번째 깊이에 있는 노드와의 거리가 450m 정도밖  
 에 되지 않는다. 그리고 공간상에서 주소를 할당 할 때  
 겹쳐지는 부분이 너무 많으므로 주소할당이 되지 않는  
 공간이 많이 생기게 된다. 최대 라우터의 개수와 자식  
 노드의 개수가 가장 적은 2개로 봤을 때 이러한 결과가  
 나오지만 최대 라우터의 수와 최대 자식 노드의 개수를  
 증가하게 되면 더욱 적은 깊이가 생기게 되고 그만큼  
 코디네이터와 마지막 노드까지의 거리가 더욱 적어지게  
 되고, 할당되지 않는 주소도 더 많이 생기게 된다. 그림  
 3은 부모 노드가 가질 수 있는 자식 노드의 최대 개수  
 가 2이고, 최대 라우터의 개수가 2개라고 했을 때 최대  
 깊이가 7일 때 노드의 연결 상태를 나타낸다.

$$4 \times (1 + 2 + 3 + 4 + 5 + 6 + 7) = 112 \quad (4)$$

$$4 \times (1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 + 11 + 12 + 13 + 14 + 15) = 480 \quad (5)$$

그림 3에서 센서의 개수는 식 (4)에 의해서 112개가  
 되고, 여기에서 15층까지 센서를 확장하여 구성하였을  
 때 총 센서의 개수를 식 (5)에 의해 480개가 되어 공간  
 적으로 봤을 때 낭비되는 주소 공간이 너무 많은 것을

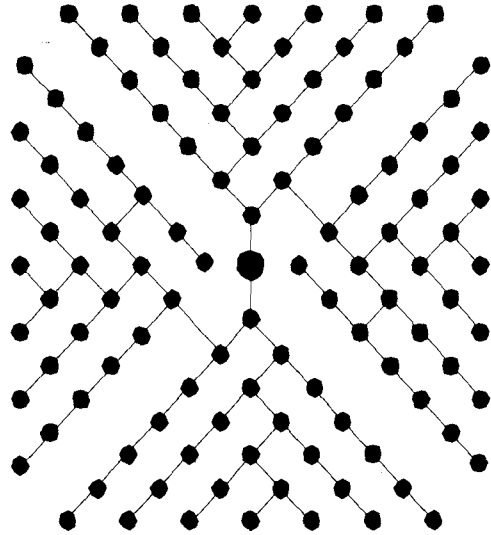


그림 3.  $C_m = R_m = 2$ 일 때 노드 연결 상태  
 Fig. 3. Node connection state when  $C_m = R_m = 2$ .

표 3.  $C_m = R_m = 2$ 일 때의  $Cskip(d)$   
 Table 3.  $Cskip(d)$  when  $C_m = R_m = 2$ .

d	$Cskip(d)$	d	$Cskip(d)$	d	$Cskip(d)$	d	$Cskip(d)$
0	32767	4	2047	8	127	12	7
1	16383	5	1023	9	63	13	3
2	8191	6	511	10	31	14	1
3	4095	7	255	11	15	15	0

알 수 있다. 표 3은 그림 3의 센서들의 주소 할당에 있  
 어서 깊이에 따른  $Cskip(d)$ 의 값을 나타낸 것이다.

### III. 제안하는 알고리즘

유선에서는 트리 형식으로 거리에 상관없이 선으로  
 연결을 하면 통신이 이루어진다. 하지만 Zigbee 통신은  
 무선 통신이기 때문에 공간에 대한 성분들을 고려해야  
 한다. 많은 센서 노드를 넓은 지역에 하나의 네트워크  
 로 구성을 하기 위해서는 다른 주소 할당 알고리즘이  
 필요하다. 그래서 본 논문에서 제안하는 알고리즘은  
 센서 네트워크를 구성하는 공간을 좌표 평면으로 개념  
 화 시킨다. 좌표의 중심점에는 코디네이터를 놓고 그것  
 을 중심으로 주소를 순차적으로 할당해 나간다. 주소를  
 표현할 때 8비트는 x축의 좌표로써 표현하고, 나머지 8  
 비트는 y축의 좌표로 표현한다. 이렇게 주소를 배치하  
 게 되면 (0, 0)에서부터 (255, 255)까지의 조합으로써  
 $2^{16}$ 인 65536개의 주소 값이 나오게 된다. 좌표 평면  
 에서 코디네이터를 중심점인 (0, 0)의 좌표 값을 가지도록  
 하기 위해서 먼저 좌표 평면의 중심점인 (128, 128)을  
 (0, 0)으로 만들기 위해 x와 y의 좌표 값에 각각 -128을

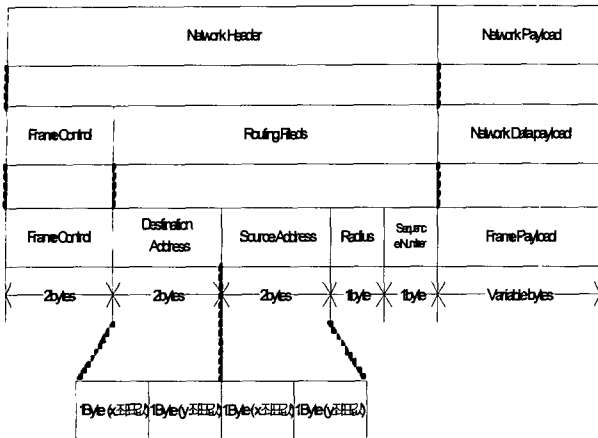


그림 4. 네트워크 계층의 프레임 포맷과 그에 따른 주소의 좌표 값 포맷

Fig. 4. Frame format of network layer and the coordinate value of address.

해준다. 그렇게 되면 코디네이터를 중심으로 사방으로 센서가 퍼지도록 할 수 있고, 그 값을 좌표 값으로 나타낼 수 있다.

그림 4에서 보듯이 목적지의 주소와 원 데이터 주소를 좌표로 나타내서 표현할 수 있다. 만약 원래 데이터 주소의 좌표 값이 (100, 40)이고, 목적지 주소의 좌표 값이 (20, 70)이면 4bytes의 주소 값의 비트 스트림이 11100111 10101000 10010100 11000110로 표현된다. 그림 5는 코디네이터를 중심으로 센서 노드들의 주소를 좌표평면으로 개념화 하여 나타낸 것을 보여준다. 일반적으로 16비트의 주소 값은 센서 노드의 ID 값을 보여준다. 하지만 이렇게 주소 값으로 표현을 함으로써 주소의 값을 가지고 코디네이터와 얼마나 떨어져 있는지를 알 수 있다. 그리고 트리 라우팅을 할 때도 좌표 값이 (0, 0)에 가까운 값으로 전달을 하면 되기 때문에 더욱 효율적이다. 기존의 네트워크 프레임 포맷에 포함되어 있는 16비트 주소 공간에 이런 의미를 부여함으로써 같은 오버 헤드를 가지고 효율적인 트리 라우팅이 가능하도록 한다.

또한 좌표 평면에 주소를 좌표 값으로 할당하고 각각의 센서와 센서 사이의 거리는 30m 정도로 놓으면 네트워크의 최대 깊이가 127이므로 코디네이터와 마지막 센서와의 거리는  $30 \times 127 = 3810m$ 가 된다. 그러므로 네 방향으로 3.8km 정도의 지역에 센서를 배치하여 각각의 센서 값을 읽어 들일 수 있다. 이와 같은 Zigbee 센서 네트워크는 계층적인 구조로 되어 있으므로 통신 경로를 설정해 주어야 한다. 그러기 위해서는 일정한 노드에 라우팅 기능과 센싱 기능을 동시에 하는 디바이

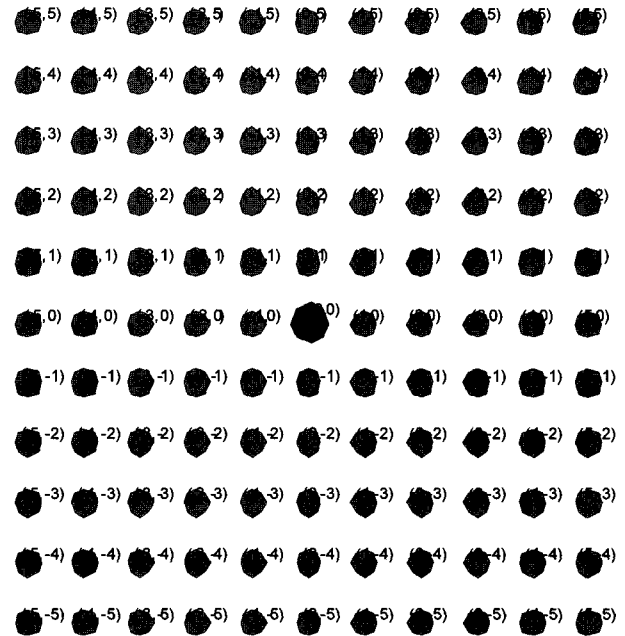


그림 5. 좌표 값으로 주소를 할당한 센서 배치

Fig. 5. Sensor arrangement according to the coordinate value.

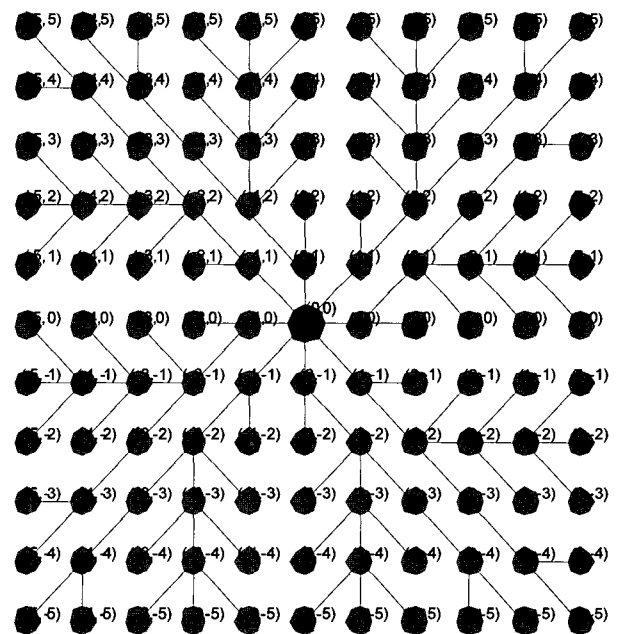


그림 6. 좌표 값에 따른 라우팅 경로

Fig. 6. Routing path according to the coordinate value.

스인 FFD(Full Function Device)를 두고 나머지 노드에는 센싱 기능만을 하는 RFD(Reduced Function Device)를 둔다. 그런데 어느 장소에 라우터 기능을 담당하는 디바이스를 놓을 것이냐에 대한 문제가 생긴다. 그래서 코디네이터에서부터 첫 번째 홉에 있는 디바이스는 모두 라우터 기능을 하는 FFD를 두고 그 8개의 디바이스가 각각 자식을 만들어서 뻗어나가면서 FFD

와 RFD를 형성해 나간다. 이렇게 고정된 네트워크에서는 관리자가 센서의 위치를 알고 있기 때문에 그 센서에 맞는 좌표 값을 할당할 수 있다. 여기에서 네트워크가 고정되어 있으므로 특정한 장소에 라우터 기능을 하는 디바이스를 두어서 라우팅이 이루어지도록 한다. 라우터의 특정한 장소는 평면적인 공간에 관리자가 임의로 정하여 설치한다. 이렇게 임의로 설치된 라우터에 의해 트리 라우팅이 이루어지고, 그것을 그림 6에서 도식화하였다.

위와 같은 방법으로 라우팅 경로를 설정하고 그 경로에 따라 라우터를 설치해 둔다. 이렇게 하여 좌표의 값과 센싱 된 값을 같이 전송하게 되면 라우터는 자신의 부모 노드의 라우터로 그 값을 전송하게 되고 그 값이 코디네이터에게로 전달되어 코디네이터는 이 값이 어디에서 전송이 되었는지 알고 그에 따른 처리를 하게 된다.

다음은 평면적인 공간에서 아닌 입체적인 공간에서 봤을 때 어떻게 주소 할당을 하는지에 대해 알아보겠다. 빌딩이나 아파트에 센서 네트워크를 구축한다고 했을 때 한 층의 개념은 평면이지만 층이 모이면 입체적인 공간이 되므로 평면적인 좌표 값 주소 할당방식을 적용하는 것은 적절하지 않다. 이러한 문제를 구분하기 위해 코디네이터의 주소 값을 (0, 0)으로 하고 코디네이터의 직계자식 노드는 x의 좌표 값은 0로 하고, y의 좌표 값을 1~255값을 주소 값으로 할당을 한다. 그리고 코디네이터의 첫 번째 라우터의 주소는 (1, 0)으로 하고 이것의 자식은 x의 좌표 값은 1이고, y의 좌표 값을 1~255값을 주소 값으로 할당을 하여 255개의 자식을 가지는 라우터가 될 수 있다. 이렇게 라우터의 주소를 (2, 0), (3, 0), (4, 0), (5, 0), ..., (255, 0)로 표현하고 각각의 자식 노드의 주소는 x의 좌표 값은 각각 라우터의 x의 좌표 값과 동일하고 y의 좌표 값이 바뀌면서 할당이 된다. 이렇게 주소할당을 하면 라우팅을 할 때 라우터가 어떤 목적지 주소를 받아서 이 주소가 자신의 자식 노드의 주소인지 아닌지를 빠르게 판단하여서 자신의 자식이면 바로 전달 해 주고, 아니면 상위 라우터에게 넘겨준다. 그림 7은 입체적인 공간에서 적용되는 좌표평면의 센서 노드와 주소의 좌표 값을 표현한 것이다.

이 방법에서는 코디네이터가 좌표 평면의 중심에 있을 필요가 없기 때문에 연산을 줄이기 위해 주소를 (0, 0)에서 (255, 255)값으로 할당을 한다. 예를 들어 빌딩에 네트워크를 구성한다고 하면 1층에 코디네이터를 설치

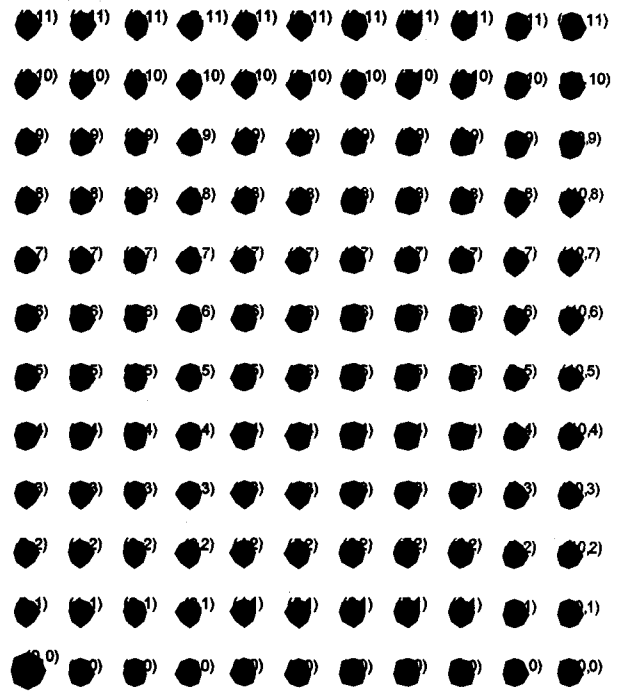


그림 7. 입체공간에 적용되는 좌표평면  
Fig. 7. The coordinate system applied at the solid geometry.

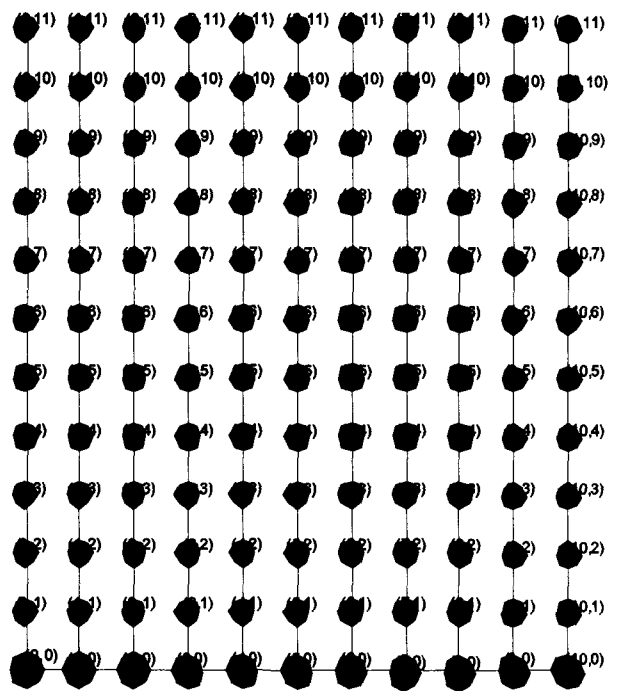


그림 8. 입체공간에서 연결된 노드  
Fig. 8. The connected nodes at the solid geometry.

하여 255개의 자식을 직접 관리하면서 다른 라우터와 센서 노드들의 통신을 도와주는 역할을 하고 2층에는 (1, 0)의 주소를 가지는 라우터를 설치하고 255개의 자식을 관리하도록 하고 3층에는 (2, 0)의 주소를 가지는

라우터를 설치하여 255개의 센서 노드들을 관리하도록 하고, 이와 같은 방법으로 4층, 5층, . . . , 255층까지 라우터를 설치하여 관리 할 수 있다. 그림 8은 11층까지의 라우터 연결 방식을 나타내 주고 있다.

여기에서 라우터 중에 (1, 0)이 관리하는 센서 노드들의 예를 들어보면, 제일 처음에 연결되는 노드의 주소의 좌표 값이 (1, 1)이고, 두 번째로 연결되는 센서 노드는 주소의 좌표 값이 (1, 2)이 되고, 세 번째는 (1, 3), 이렇게 센서 노드에게 주소의 좌표 값을 할당한다. 그리고 (1, 0), (2, 0), 등을 중심 라우터로 정의한다. 중심 라우터가 가질 수 있는 자식의 수가 255개가 되기 때문에 그 중에도 라우터가 연결될 수 있다. 그래서 (1,

0)의 중심 라우터와의 통신 거리가 너무 멀어서 다른 라우터가 필요하다면 (1, 0)의 관리 하에 라우터를 연결시킬 수 있다. 만약 (1, 0)의 중심 라우터 주소의 좌표 값이 (1, 4)인 라우터가 연결이 되면 이 라우터도 자식 노드들에게 주소 값을 할당하여 주어야 하기 때문에 주소 할당 방식을 알아야 한다. 여기에서 주소 할당 방식은(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), . . . , (1, 255) 이렇게 순차적으로 할당을 한다. 그렇기 때문에 지금 어디까지 주소가 할당이 되었는지 (1, 0)의 좌표 값을 가진 중심 라우터와 (1, 4)의 좌표 값을 가진 라우터가 지금 어느 좌표 값까지 주소로 할당이 되어 있는지 인지하고 있어야 한다. 이러한 방법에 의해 두 개의 라우터에 자식이 연결이 되면 그에 맞는 좌표 값을 할당 할 수 있기 때문에 공간적인 효율성을 높일 수 있다. 그리고 (1, 4)의 좌표 값을 가지는 라우터는 자식 노드에게 할당한 주소의 값을 저장하고 있어야 한다. 그렇지 않으면 라우팅을 할 때 어떤 목적지 주소를 받았는데 그 목적지 주소가 자식 노드의 주소인지 아닌지를 모르게

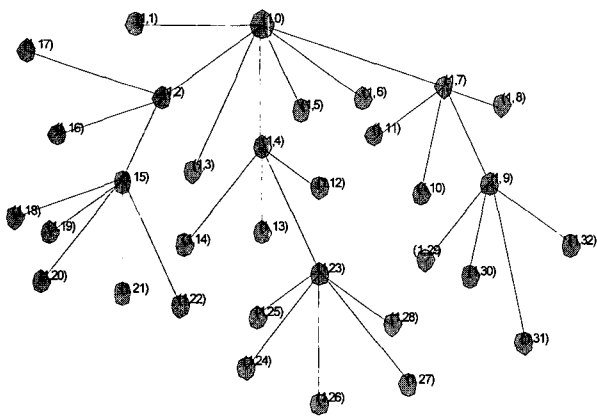


그림 9. 중심 라우터에 연결된 노드  
Fig. 9. The nodes connected at the root router.

된다. 하지만 그 자식 노드의 개수가 제일 크게 되는 경우가 이론상으로 254개이기 때문에 저장 공간에 대한 과부하는 걸리지 않는다. 그림 9는 하나의 예제로써 (1, 0)의 좌표 값을 갖는 중심 라우터에게 연결되는 라우터와 센서 노드들을 보여준다.

그림 9에서 중심 라우터에 32개의 센서 노드들이 연결되어 있고 그 중에 라우터로 연결된 노드의 주소의 좌표 값이 (1, 2), (1, 4), (1, 7), (1, 15), (1, 23), (1, 9) 이렇게 6개이다. (1, 2)의 좌표 값을 가진 라우터는 그 자식 노드인 (1, 17), (1, 16), (1, 15)과 (1, 15)의 자식 노드인 (1, 18), (1, 19), (1, 20), (1, 21), (1, 22)의 좌표 값을 저장하고 있어야 한다. 그리고 나머지 5개의 라우터도 자식 노드의 주소의 좌표 값을 모두 저장하고 있을 경우에 트리 라우팅을 적용할 수 있다.

#### IV. 성능 평가

본 논문에서 제안한 좌표 평면을 이용한 주소 할당 방식과 Cskip 알고리즘으로 주소를 할당한 방식을 비교하였다. 비교하는 파라미터는 평균 멀티 홉의 수와 평균 홉 수에 따른 전송 시간을 비교하였다. 라우팅을 하는데 있어서 몇 홉을 거쳐서 가느냐에 따라 라우팅 비용이 소요된다. 통신을 할 수 있는 한도 내에서 홉 수가 적을수록 효율적인 라우팅 기법이다. 두 가지의 알고리즘을 같은 조건에서 시뮬레이션을 하기 위해 노드의 개수에 따른 최대 홉 수가 같도록 설정을 하였다. 예를 들어 100개의 노드가 있다고 하면 그 각각의 노드들이 통신을 할 수 있는 후보들은 99개가 되고 그 각각의 홉

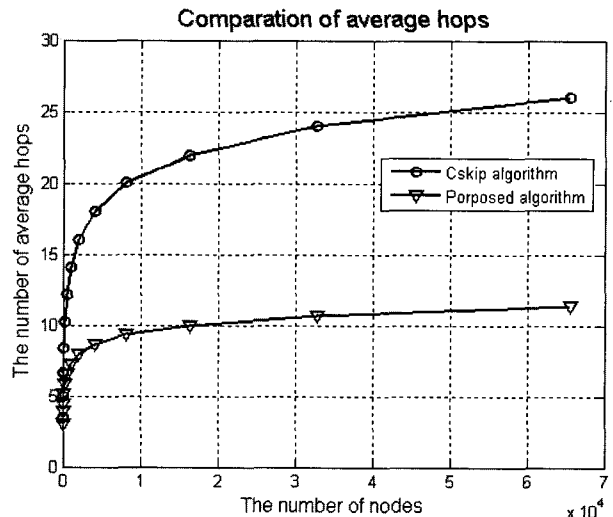


그림 10. 평균 홉 수 비교 결과  
Fig. 10. The comparison result of the average hops.

표 4. 평균 홉 수 비교표

Table 4. The comparison table of the average hops,

노드수	Cskip 알고리즘 평균	제안한 알고리즘 평균
	홉 수	홉 수
15	3.5	3.0
31	5.0	3.9
63	6.6	4.4
127	8.4	5.1
255	10.2	5.9
511	12.1	6.6
1023	14.1	7.3
2047	16.0	8.0
4095	18.0	8.6
8191	20.0	9.3
16383	22.0	10.0
32767	24.0	10.7
65535	26.0	11.3

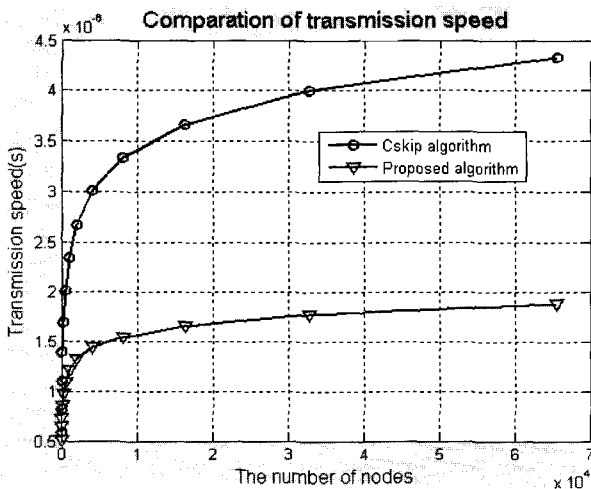


그림 11. 전송시간 비교 결과

Fig. 11. The comparison result of the transmission speed.

수를 다 더해서 100으로 나누어 주면 하나의 노드에 대한 평균 홉 수가 나온다. 나머지 99개 노드에 대해서도 같은 방식으로 하여 총 평균 홉 수를 구한다. 그림 10은 두 가지 알고리즘의 평균 홉 수의 비교 결과를 보여준다. 비교 결과에서 보듯이 본 논문에서 제안한 주소 할당 방식이 평균 홉의 수가 노드가 많아질수록 현저한 차이를 나타내는 것을 알 수 있다. 홉 수가 적은 것은

그만큼 라우팅 비용이 적게 드는 것을 뜻한다. 표 4는 노드의 개수에 따른 평균 홉 수의 차이를 보여주는 표이다.

전송 시간을 비교하기 위해 노드와 노드 사이의 거리가 50m라고 가정하였다. 노드와의 거리를 알고, 무선 통신에서 속도는 광속이므로 전송 시간을 알 수 있다. 그림 11은 평균 홉 수에 따른 전송 시간을 보여준다.

마지막으로 프로세싱 타임을 비교해 보면 Cskip 알고리즘으로 주소 할당을 하게 되면 라우터가 자식 노드에게서 받은 주소를 가지고 식 (3)에 의해서 2번의 16비트 비교 연산과 1번의 16비트 더하기 연산을 해야 한다. 그러나 본 논문에서 제안한 주소 할당 방식은 라우터가 자식 노드에게서 받은 목적지 주소의 x좌표 값만 비교하면 되기 때문에 8비트 비교 연산을 1번 수행하므로 프로세싱 타임이 6배 줄어든다. 그러므로 라우터가 더욱 효율적으로 라우팅을 할 수 있고 전송 시간도 줄일 수 있다.

### V. 결론 및 향후 연구 과제

본 논문에서는 Zigbee의 표준에서 정의하고 있는 계층 구조에서 노드들에게 16비트 주소를 할당하는데 있어서 공간을 좌표 평면으로 개념화 하여서 주소를 x와 y의 좌표 값으로 할당을 함으로써 주소 공간의 낭비를 줄이는 방법을 제안하였다. 이 알고리즘은 어느 특정 지역에 센서를 설치한다고 하면 그 공간에 효율적으로 센서를 배치하여 라우팅을 할 수 있는 알고리즘이다. 그리고 빌딩이나 아파트와 같은 입체적인 공간에 대해 좌표 평면을 적용하여 라우팅 하는 방법을 제안하였다. 그리고 시뮬레이션을 통해 이전의 주소할당 알고리즘과 비교하여 좌표 평면을 이용한 주소 할당 알고리즘이 더욱 효과적이라는 것을 증명하였다. 입체 공간에서 좌표 평면을 이용한 알고리즘을 적용 시킨다면 현재 Zigbee의 주소할당 알고리즘보다 더 큰 네트워크에 적용할 수 있다. 향후에 이 시스템을 실제 시스템에 적용하여 실제적인 결론을 도출할 것이다.

### 참고 문헌

- [1] G. Ding, Z. Sahinoglu, P. Orlik, J. Zhang, B. Bhargava, "Tree-Based Data Broadcast in IEEE 802. 15. 4 and Zigbee Networks", IEEE Trans. on mobile computing, Vol. 5, no. 11, pp. 1561-1574, November 2006.
- [2] Ran Peng, Sun Mao-heng, Zou You-min, "ZigBee Routing Selection Strategy Based on Data Services and Energy-balanced ZigBee Routing", Proceedings of the 2006 IEEE Asia-Pacific Conference on Services Computing, 2006.
- [3] Zigbee Alliance, "Zigbee specification", December 2006.



- [4] Ho-In Jeon, "Efficient Address Assignment for Mesh Nodes in Real-Time", 15-06-0437-01-0005- efficient- real- time- network- address- allocation- mechanisms- based- naa- concept- in- mesh- network, IEEE 802.15.5 Wireless PAN Mesh Network Task Group Face-to-Face meeting, November 14, 2006.
- [5] Zigbee Allinace, www.zigbee.org
- [6] 최동훈, 배성수, 최규태, "지그비 기술과 활용", 도서출판 세화

저 자 소 개



김 재 현(학생회원)  
 2007년 대구대학교 통신공학과 학사 졸업.  
 2007년 영남대학교 정보통신공학과 석사(과정).  
 <주관심분야 : 무선 센서 네트워크, Zigbee 센서 네트워크, 이동통신, 무선 통신>



강 원 석(정회원)  
 1998년 영남대학교 컴퓨터공학과 학사.  
 2000년 영남대학교 컴퓨터공학과 석사.  
 2000년~2004년 한국과학기술연구원 연구원.  
 2005년~현재 대구경북과학기술연구원 연구원.  
 <주관심분야 : 시뮬레이션/모델링, 분산처리, 임베디드소프트웨어>



허 수 정(학생회원)  
 2001년 대구대학교 통신공학 학사 졸업.  
 2007년 영남대학교 정보통신공학과 석사 졸업.  
 2007년~현재 영남대학교 정보통신공학과 박사 과정.  
 <주관심분야 : 이동통신, 무선자원관리, 무선센서 네트워크, 근거리 통신망>



이 동 하(정회원)  
 1985년 경북대학교 전자공학과 학사.  
 2001년 경북대학교 전자공학과 석사.  
 2005년 경북대학교 전자공학과 박사.  
 2006년 경북대학교 경영학과 T-MBA 수료  
 1987년~2005년 (주)LG전자 S/W연구실장  
 2004년~현재 영남대학교 겸임교수  
 2005년~현재 대구경북과학기술연구원 팀장.  
 <주관심분야 : 영상처리, DTV시스템, 임베디드 소프트웨어, 지능형로봇시스템>



박 용 완(정회원)  
 1982년 경북대학교 학사.  
 1984년 경북대학교 석사.  
 1989년 뉴욕주립대 석사.  
 1992년 뉴욕주립대 박사.  
 1992년~1993년 California Institute of Technology 전자공학 Research Fellow.

1994년~1996년 SK Telecom PCS 기술연구팀장  
 2001년 일본 NTT DoCoMo 연구소 초빙 교수  
 2003년~2004년 UC Irvine 방문교수  
 2004년~현재 지역혁신기술센터(RIC) 소장  
 2006년~현재 유비쿼터스 임베디드 센터 소장  
 1996년~현재 영남대학교 정보통신공학과 교수  
 <주관심분야: 이동통신, 무선멀티미디어 설계기술, 차세대 위치기반 기술>