

논문 2006-01-14

# 지그비 계층적 라우팅의 성능 향상 기법

( Improvement of Hierarchical Routing in ZigBee Networks )

김태홍\*, 김대영, 유성은, 성종우, 김영수

Taehong Kim, Daeyoung Kim, Seong-eun Yoo, Jongwoo Sung, Youngsoo Kim

**Abstract** : ZigBee is the emerging industrial standard for ad hoc networks based on IEEE 802.15.4. It is used for low data rate and low power wireless network applications. Expected applications of ZigBee include wireless sensor networks for remote monitoring, home control, and industrial automation. Since one of the most important design goals of ZigBee is to reduce the installation and running costs, the ZigBee stack is embedded in the cheap and small micro-controller unit. The hierarchical routing algorithm is used for ZigBee end devices which have very limited resources. Using the block addressing scheme, end devices can send data to the destination without a routing table. However, hierarchical routing has the problem that the packets follow the tree hierarchy to the destination even if the destination is located nearby. In this paper, we propose a scheme to improve the hierarchical routing algorithm in ZigBee networks by employing the neighbor table that is originally used together with the routing table. We suggest selecting the neighbor node that has the minimum remaining hops to the destination as the next hop node. Simulation result shows that the proposed scheme saves more than 30% of the hop counts compared with the original hierarchical routing.

**Keywords** : ZigBee, Hierarchical Routing, Tree Routing, Neighbor Table

## 1. 서론

지그비는 저속, 저가, 저전력의 홈네트워크 표준으로 2004년 12월 ZigBee Specification 1.0[1]이 발표된 이후, 국내외 우수 산업계를 중심으로 지그비 송수신 장치, 네트워크 스택, 지그비 응용 시스템 등 다양한 분야에서 상용화 노력이 활발히 진행되고 있다. 지그비는 장시간의 배터리 사용, 네트워크 자가 구성 및 자가 치유, 신뢰성 있는 통신, 네트워크 확장성, 무선 장치의 설치 및 운용 용이성 등의 특징을 가지고 있으며, 홈/빌딩 오토메이션, 가전, PC 주변장치, 의료 기기 등의 다양한 응용에 적합한 네트워크 기술이다.[2] 이에 따라 국내에서도 지그비를 활용하여 시스템을 구축하고 있으며, SK 텔레콤에서는 홈 네트워크 서

\* 교신저자(Corresponding Author)

논문접수 : 2006. 10. 16., 채택확정 : 2006. 11. 02.  
김태홍, 김대영, 유성은, 성종우, 김영수: 한국정보통신대학교 실시간 임베디드 시스템 연구실

※ 본 논문은 정보통신부와 한국정보사회진흥원의 지원 사업으로 수행된 연구 결과입니다.

비스, 삼성전자에서는 반도체 펌 생산관리 통합망에 지그비를 도입하였다.

지그비의 네트워크 계층에서는 네트워크의 구성 및 디스커버리 절차, 주소 할당 방식 및 이를 바탕으로 한 계층적 라우팅과 라우팅 테이블에 기반한 메쉬 라우팅 기법 등 네트워크 운용에 필요한 기능들을 정의하고 있으며, 특히, 계층적 라우팅은 메모리 자원이 매우 작은 단말장치(End device)에 적합한 라우팅 기법이다. 계층적 라우팅에서 각 노드는 라우팅 테이블을 사용하지 않고, 목적지 노드의 주소만을 이용하여 해당 노드의 자식 노드 또는 부모 노드를 중간 노드로 선택하여 데이터를 전달하게 된다. 그러나 부모 및 자식 링크 이외에 다른 이웃 노드와의 링크는 고려하지 않으므로, 목적지 노드가 소스 노드의 이웃 노드라 할지라도 트리 구조를 따라 데이터가 전달되므로 라우팅 비용에 비효율성의 문제가 나타나게 된다. 본 논문에서는 지그비 표준에서 정의하고 있는 이웃 노드 테이블을 계층적 라우팅에 활용함으로써, 계층적 라우팅의 비효율성 문제를 해결하고자 한다. 즉, 계층적 라우팅을 따라 목적지 노드로의 중간 노드를 선택하되, 목적지 노드까지의 라우팅 비

용을 줄일 수 있는 이웃 노드가 있을 경우에는 이웃 노드를 중간 노드로 선택함으로써 목적지 노드까지의 라우팅 비용을 절감하게 된다.

본 논문의 구성은 다음과 같다. II장에서는 지그비 및 IEEE 802.15.4, 계층적 라우팅 및 이웃 노드 테이블에 관하여 살펴보고, III장에서는 계층적 라우팅이 가지는 문제점을 파악한다. IV장에서는 계층적 라우팅의 성능을 향상시킬 수 있는 알고리즘을 제안하고, V장의 성능 평가와 함께 VI장에서 결론을 맺는다.

## II. ZigBee와 IEEE 802.15.4

지그비 얼라이언스는 저전력, 저비용의 무선 네트워크 기술의 구체적인 활용과 응용을 목적으로 모토로라, Freescale, 필립스, 삼성전자 등 14개의 프로모터 및 200여 개의 참여 회원사로 결성된 비영리 단체이다. 2004년 12월 무선을 이용한 차세대 홈 네트워크를 효율적으로 구축하고 적용 기간의 상호 운용성을 증진할 수 있는 산업 표준인 ZigBee Specification 1.0을 발표하였으며, 2006년 9월에는 멀티캐스트, 노드 이동성, 테이블 기반 라우팅의 성능 향상 등을 특징으로 한 새로운 버전의 지그비 표준이 발표되어 2007년 일반에 공개될 예정이다.[3]

### 1. IEEE 802.15.4

지그비는 저속, 저전력 무선인터페이스인 IEEE 802.15.4 표준을 물리 계층 및 데이터링크 계층으로 채택하였다. IEEE 802.15.4는 대체로 10~100미터의 무선 동작 영역과 낮은 전력 소모를 기반으로 동작하는, 간단한 저전력 기기들의 무선 통신을 지원하도록 물리계층 및 데이터링크 계층에 대하여 정의하는 저속 무선 개인 영역 네트워크(WPAN)에 대한 표준이다. 국제적으로 사용상 제약이 없는 ISM 2.4GHz 주파수 대역에서 16개, 900MHz 주파수 대역에서 10개, 868MHz 주파수 대역에서 1개 채널을 제공하며, 각 주파수 대역에 대해서 250Kbps, 40Kbps, 그리고 20Kbps의 데이터 전송 속도를 가진다. 하나의 네트워크는 64K개의 기기를 지원할 수 있으며, 각 기기는 64비트의 IEEE 표준 주소 또는 16비트의 지역 주소를 가질 수 있다. 이 외에도, 실시간 데이터 전송을 보장하기 위해 GTS(Guaranteed Time Slot)를 할당하며, CSMA/CA(Carrier Sense Multiple

Access with Collision Avoidance)의 매체 액세스와 에너지 수준의 감지(Energy Detection), 링크 품질 표시(Link Quality Indication)를 정의한다. 또한, 비컨 기반의 슈퍼 프레임 구조 방식을 사용하면, 전체 시간을 활성 영역(active region)과 비활성 영역(inactive region)으로 나눌 수 있고 비활성 영역에서 라디오를 끄으로써 전력 소모를 현저히 줄일 수 있다.[4]

## 2. ZigBee Network

지그비 네트워크 계층은 스타 및 트리 토폴로지뿐만 아니라 점대 점 방식의 메쉬 토폴로지를 지원한다. 또한, 네트워크를 스스로 형성할 수 있으며, 디바이스의 합류(Join) 및 이탈(Leave) 등 네트워크의 동적인 변화에 적응할 수 있는 자가치유의 특성을 갖는다. 네트워크의 주소 할당 방식은 Low cost & Low complexity를 지향하는 지그비 솔루션에 적합하도록 라우팅 테이블을 사용하지 않고 라우팅을 지원할 수 있는 Cskip 값 기반의 블록 어드레싱을 사용한다. 이렇게 할당된 주소는 계층적인 네트워크 토폴로지 정보를 담게 되고 라우팅 테이블을 지원할 수 없는 RFD(Reduced Function Device)들도 무선 환경에서 멀티 홉으로 데이터를 전송할 수 있게 된다. 동시에 통신 효율성을 지원하기 위해서 테이블 기반의 라우팅 방식인 On-demand 경로 탐색(Route Discovery) 방식을 지원하며, 여기에는 경로 탐색에 필요한 RREQ(Route Request) 및 RREP(Route Reply) 메시지의 타입 및 경로 탐색 알고리즘이 정의되어 있다.

### 2.1 계층적 라우팅 알고리즘

지그비의 5계층적 라우팅은 블록 어드레싱이라 불리는 주소 할당 방식을 기반으로 한다. 네트워크를 구성하는 각 노드들은 자신의 부모 노드로부터 16비트의 네트워크 주소를 할당받게 되고, 이 주소는 특정 네트워크 내에서 유일한 값이다. 또한, 할당받은 네트워크 주소 및 자신의 트리 레벨을 이용하여, 자신의 자식 노드에게 할당할 수 있는 주소 공간을 계산할 수 있다. 각 레벨의 노드들에게 주어진 주소 공간의 크기인 Cskip(d)는  $nwkmaxChildren(Cm)$ ,  $nwkMaxDepth(Lm)$ 과  $nwk-maxRouter(Rm)$ 을 이용하여 계산할 수 있으며, 그 공식은 (1)과 같다.

$$Cskip(d) = \frac{1 + Cm - Rm - Cm \cdot Rm^{Lm-d-1}}{1 - Rm} \quad (1)$$

네트워크에 합류한 노드들은 Cskip의 크기만큼의 자손 노드를 가질 수 있으며, 자식 노드의 디바이스 종류에 따라 그 할당방식이 다르다. 예를 들어, k번째에 합류하는 라우터 노드와 n번째에 합류하는 단말노드의 할당 방식은 아래와 같다.

$$A_k = A_{parent} + Cskip(d) \cdot (k-1) + 1 \quad (1 \leq k \leq Rm)$$

$$A_n = A_{parent} + Cskip(d) \cdot Rm + n \quad (1 \leq n \leq Cm - Rm) \quad (2)$$

이렇게 형성된 주소 기반의 계층 구조를 이용하면, 각 노드는 목적지 노드가 자신의 자손인지 아닌지 여부를 확인할 수 있으며, 예를 들어, 네트워크 주소 A를 가지는 레벨 d의 노드가 목적지 노드 D에 대해서 공식 (3)을 만족할 경우, 목적지 노드 D는 A의 자손 노드임을 알 수 있다.

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

지그비 계층적 라우팅에서는 위의 공식을 통해 목적지 노드가 자신의 자손 여부지만 판단하여, 맞을 경우 자식 노드에게 데이터를 보내고, 그렇지 않을 경우 부모 노드에게 전달한다.

## 2.2 이웃 노드 테이블

지그비 표준에 따르면 모든 노드들은 이웃 노드 테이블을 가지며, 이웃 노드 테이블에는 각 노드의 1홉 통신 거리 안에 있는 모든 노드들에 대한 정보가 저장된다. 또한, 사용자는 이웃 노드 테이블의 크기를 지정할 수 있으며, 이때에는 제한된 개수의 이웃 노드 정보만 저장될 수 있다. 이웃 노드 테이블에 저장되는 이웃 노드의 정보로는 PANid, 64비트 IEEE 주소, 16비트 네트워크 주소, 디바이스 종류, 그리고 해당 노드와의 논리적 관계가 있으며 선택적으로는 비컨 순서(Beacon Order), 트리 레벨, 합류 허가 여부(Permit joining) 등의 정보가 포함될 수 있다.

이웃 노드 테이블의 각 엔트리는 각 노드가 네트워크에 합류하기 전, 네트워크 디스커버리 절차를 거치면서 생성된다. 즉, 기존에 구성된 네트워크를 탐색하고자 하는 노드는 NLME-NETWORK-DISCOVERY.request 요청을 하며, 이에 대한

응답으로 수신한 비컨을 통해 기존에 이미 네트워크에 합류해 있는 노드들에 관한 정보를 파악하고 이 정보를 이웃 노드 테이블에 저장한다. 반대로, 이웃 노드가 네트워크를 이탈할 경우에는 이웃 노드 테이블의 해당 엔트리가 삭제되며, 이러한 사실은 NLME-LEAVE.indication 메시지 수신을 통해 알 수 있다. 또, 각 노드는 이웃 노드 엔트리에 해당하는 패킷을 수신할 때마다 관련 정보를 업데이트한다.

## III. 계층적 라우팅의 문제점

지그비의 계층적 라우팅은 다음 홉(Next Hop)을 선택할 때에 부모 및 자식의 링크만을 고려하게 된다. 즉, 목적지 노드가 자신에게 할당된 주소 공간에 포함될 경우, 자신의 자손 노드 중의 하나임으로 해당하는 자식 노드를 다음 홉의 노드로 선택하고, 그렇지 않을 경우에는 부모 노드를 다음 홉의 노드로 선택하여 데이터를 전달한다. 따라서 그림 1에서 보는 것과 같이 목적지 노드가 소스 노드의 1홉 통신 거리 안에 있다 하더라도 목적지 노드는 소스 노드와 부모-자식의 관계를 갖지 않으므로 트리의 계층구조를 따라 전달된다. 결국, 소스 노드가 보낸 패킷은 트리의 루트 노드인 코디네이터까지 거슬러 올라갔다가 목적지 노드로 전달된다. 그림 1의 예에서는 1홉에 가능한 패킷의 전달에 4홉의 라우팅 비용이 들게 된다. 그림 1의 예에서와 같이 목적지 노드가 소스 노드의 1홉 통신 거리 안에 있는 경우 외에도 많은 경우에서 계층적 라우팅은 부모-자식 간의 링크만을 고려하게 되는 근본적인 특성 때문에 라우팅 경로에 오버헤드가 생기게 된다.

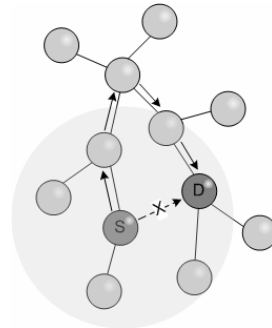


그림 1. 계층적 라우팅의 문제점

Fig. 1. The problem of hierarchical routing

### IV. 제안 알고리즘

본 장에서는 지그비의 계층적 라우팅의 비효율적인 라우팅 비용을 줄이고 성능을 향상시킬 수 있는 알고리즘을 제안한다. 즉, 지그비의 계층적 라우팅 방식을 기본적으로 활용함과 동시에 각 노드의 이웃 노드 테이블을 활용하여 목적지 노드까지의 라우팅 비용을 줄일 수 있는 이웃 노드가 있

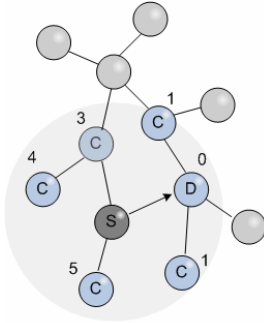


그림 2. 노드 D로의 중간 노드 후보 및 라우팅비용

Fig. 2. The candidates of next hop and remaining routing cost for node D

표 1. 주어진 목적지 주소에 대해 다음 홉의 노드를 선택하는 알고리즘

Table 1. Algorithm to choose next hop for the given destination address

<i>Find_NextHopAddr(dstAddr)</i>
Input: dstAddr
Output: NHDstAddr
begin
1. depth_dstAddr = Find_AddrRange(dstAddr, 0, 0)
2. Assign the next hop node of tree routing to NHDstAddr
3. Assign the remaining route cost when selecting NHDstAddr to minNHRouteCost
4. for each (neighbor $n_k$ in neighbor table)
5. $i = 0$
6. while ( $n_k$ is in AddrRange[i+1] && $i < \text{depth\_dstAddr}$ )
7. ++i
8. myRouteCost=(depth_dstAddr-i)+(depth( $n_k$ )-i)
9. if (minNHRouteCost > myRouteCost)
10. NHDstAddr = $n_k$
11. minNHRouteCost = myRouteCost
12. end if
13. end for each
end

을 경우에는 부모 또는 자식 노드 대신 이웃 노드를 다음 홉의 노드로 선택하는 것이다.

즉, 소스 노드 또는 중간 노드는 목적지 노드로 전송하기 위해 다음 홉의 노드를 선택해야하며, 그림2에서 보는 것처럼 여러 중간 노드의 후보들 중에서 하나의 노드를 선택하기 위해서는 일정한 기준이 필요하다. 제안하는 알고리즘에서는 각 후보 노드로부터 목적지 노드까지 계층적 라우팅 방식을 따른다고 가정하여, 각 후보 노드들에게 전달했을 경우 목적지 노드까지 남은 홉 수(Remaining Hops)를 계산한다. 그림 2에서 각 후보 노드들 위의 숫자는 각각에게 패킷을 전달했을 때 목적지 노드까지의 남은 홉 수를 나타내며, 목적지 노드로 직접 데이터를 전달했을 때에 라우팅 비용을 최소화한다는 사실을 알 수 있다.

표 1의 알고리즘은 주어진 목적지 노드에 대해, 목적지 노드까지의 라우팅 비용이 최소인 노드를 선택하여 다음 홉의 노드로 결정하는 알고리즘이다. 제안한 알고리즘에서는 계층적 라우팅 방식에 따라 다음 홉 노드를 결정하고, 그 때의 라우팅 비용을 최소 라우팅 비용으로 설정한다. 그 다음, 각 이웃 노드로부터 목적지 노드까지 계층적 라우팅을 따른다고 가정하여 라우팅 비용을 계산하고 최소 라우팅 비용과 비교한다. 만약, 기존의 최소 라우팅 비용보다 작을 경우에는 9-12라인과 같이, 다음 홉의 노드를 해당 이웃 노드로 변경하고, 최소 라우팅 비용도 갱신한다. 반대로, 계층적 라우팅을 따를 때보다 라우팅 비용이 적은 이웃 노드가 없을 경우에는 계층적 라우팅 방식에 따라 부모 또는 자식 노드에게 데이터를 보내게 된다.

이웃 노드로부터 목적지 노드까지의 남은 홉 수를 계산하기 위해서는, 이웃 노드의 주소와 목적지 노드의 주소를 공통으로 포함하는 서브 트리 중 루트의 레벨이 가장 높은 서브 트리를 찾는다. 왜냐하면, 이웃 노드로부터 계층적 라우팅을 통해 목적지 노드로 전달될 때에는 목적지 노드의 주소를 포함하는 노드까지 트리를 따라 거슬러 올라간 후에 목적지 노드로 전달되는 과정을 거치게 되기 때문이다. 즉, 이웃 노드와 목적지 노드 주소를 공통으로 포함하는 서브 트리의 루트가 이웃 노드가 전달한 패킷이 트리를 따라 올라가다가 다시 내려오는 전환점이 되며, 표 1의 8번째 라인에서처럼, 이 서브 트리의 루트 레벨을 기준으로 남은 홉 수를 계산할 수 있다.

표 1의 5-7번째 라인에서는 트리의 루트 즉,

레벨 0부터 시작하여, 트리 레벨을 1씩 증가시키면서 이웃 노드의 주소가 해당 트리 레벨에서 목적지 주소를 포함하는 노드의 주소 공간에 포함되지 않을 때까지 반복한다. 따라서 5-7라인을 수행한 후에는, 이웃 노드의 주소와 목적지의 주소가 공통으로 포함되는 서브 트리 중 가장 높은 레벨을 가진 서브 트리의 루트 레벨을 찾을 수 있게 되며, 이를 위해서는 목적지 주소를 포함하는 각 트리 레벨에서의 주소 공간 범위를 알아야 한다.

표 2. 목적지 주소가 포함되어 있는 각 트리 레벨에서의 주소 공간 범위 계산 알고리즘  
Table 2. Algorithm to find the address space in each level that contains the destination

---

```

Find_AddrRange(dstAddr, startAddr, curDepth)
Input: dstAddr, startAddr, curDepth
Output: depth_dstAddr, AddrRange[depth_dstAddr]
begin
1. if (dstAddr = startAddr)
2.   return curDepth
3. else
4.   for i = 1 to Rm
5.     if (dstAddr is in the address space of ith router)
6.       store address space of ith router to
         AddrRange[curDepth+1]
7.       Find_AddrRange(dstAddr, addr of ith router,
         curDepth+1)
8.     end if
9.   end for
10.  if (Cm-Rm > 0)
11.    if (dstAddr is the end device of startAddr)
12.      store dstAddr to AddrRange[curDepth+1]
13.      return curDepth+1
14.    end if
15.  end if
16. end if
end

```

---

표 2는 주어진 목적지 주소가 포함되어 있는 각 트리 레벨에서의 주소 공간 범위를 계산하는 알고리즘을 보여 준다. Find\_AddrRange() 함수의 인자인 dstAddr은 목적지 노드의 주소, startAddr은 각 레벨에서 주소 공간의 시작 주소, curDepth는 트리의 레벨을 의미한다. 이 함수는 startAddr과 curDepth를 각각 0으로 설정하여 호출되며, 트리의 루트에서부터 시작하여 목적지 노드를 포함하고 있는 라우터를 검색함으로써 레벨 1의 주소 공간을 찾는다. 다음 startAddr과 curDepth를 해당 서브트리가 가지는 주소 공간의 첫 번째 주소 및 1로 설정하여 Find\_AddrRange()를 호출한다. 이렇게 재귀적인 방식으로 목적지 노드가 포함되

어 있는 레벨까지 탐색해 내려가며, 각 트리 레벨에서의 주소 공간은 AddrRange[]에 저장되어, 처음 이 함수를 호출한 함수에서 사용된다.

본 장에서 제안된 알고리즘은 소스 노드 또는 중간 노드의 이웃 노드들 중 목적지 노드까지 가장 적은 라우팅 비용을 갖는 노드를 선택하도록 하는 것으로, 각 노드는 자신의 이웃 노드만을 고려하여 다음 홉의 노드를 선택하게 된다. 다음 홉 노드의 선택이 지역 최소화(Local Minimum) 방식을 따르게 되지만, 루프(Loop)와 같은 문제는 발생하지 않는다. 왜냐하면, 데이터 패킷이 전달될 때마다 목적지 노드까지의 남은 홉 수의 최대값 즉, 계층적 라우팅을 이용할 때의 라우팅 비용은 항상 감소하기 때문이다.

## V. 성능 평가

### 1. 시간복잡도 분석

정리 1. 제안한 알고리즘은 다항식 시간 (Polynomial Time) 내에 풀 수 있다.

증명 : 소스 노드 또는 중간 노드의 이웃 노드의 개수를  $n$ 이라고 가정한다. 이 때, nwkMaxDepth(Lm)은 상수이고, 표 1에서 각 이웃 노드에 대하여 레벨이 가장 높은 목적지 노드와의 공통 서브 트리를 검색하는 과정(라인 6)은 많아야 Lm번 호출된다. 즉, 이웃 노드  $n$ 개에 대하여 최대 Lm번씩 호출이 되므로, 제안하는 알고리즘의 시간 복잡도는  $O(Lm \cdot n)$ 이다. 따라서 다음 홉의 노드를 선택하는 문제는 다항식 시간 내에 풀 수 있다.

### 2. 라우팅 비용의 상한값

정리 2. 제안한 알고리즘에서 소스 노드로부터 목적지 노드까지의 라우팅 비용은 지그비의 계층적 라우팅의 라우팅 비용을 초과하지 않는다.

증명 : 모든 노드  $v$ 는 계층적 라우팅을 바탕으로 하되, 계층적 라우팅보다 적은 라우팅 비용을 갖는 이웃 노드를 다음 홉의 노드로 선택한다. 계층적 라우팅에서는 다음 홉의 노드를 결정할 때, 목적지 노드의 주소가 노드  $v$ 의 주소 공간 범위에 포함되어 있지 않을 경우 부모 노드  $p$ 를 다음 홉의 노드로 선택한다. 이 때, 부모 노드  $p$ 와 자식 노드  $c$ 에서의 남은 홉 수의 관계는 아래와 같다.

$$remaining\ hops(c) = remaining\ hops(p) + 2 \quad (4)$$

반대로, 목적지 노드의 주소가 노드  $v$ 의 주소 공간에 포함될 경우, 자식 노드  $c$ 를 다음 홉의 노

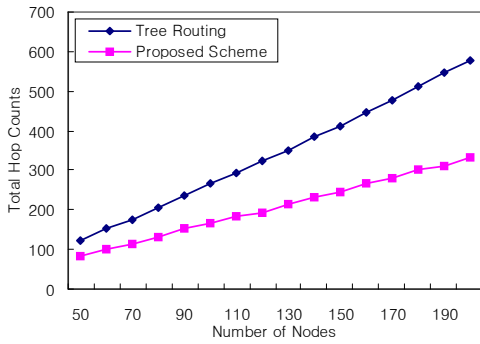


그림 3. 목적지 노드가 코디네이터이고 최대 이웃 노드 수가 5일 경우 모든 노드로부터 목적지까지의 평균 총 홉 수

Fig. 3. Average total hop counts of all nodes when the destination is coordinator and number of MaxNeighbor is 5

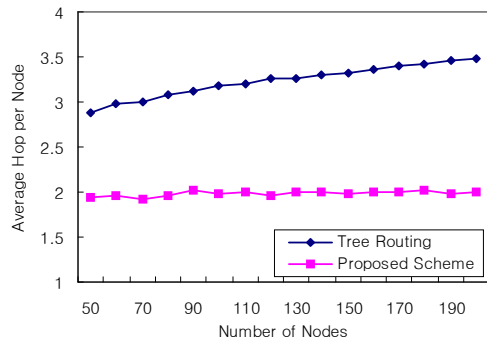


그림 4. 목적지 노드가 코디네이터이고 최대 이웃 노드 수가 5일 경우 목적지 도달까지의 노드 당 평균 홉 수

Fig. 4. Average hop counts per a node when the destination is coordinator and number of MaxNeighbor is 5

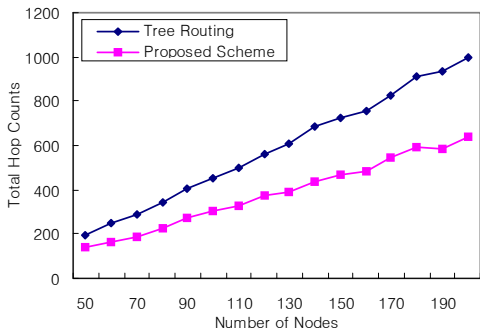


그림 5. 목적지 노드가 임의의 노드이고 최대 이웃 노드 수가 5일 경우 모든 노드로부터 목적지까지의 평균 총 홉 수

Fig. 5. Average total hop counts of all nodes when the destination is random and number of MaxNeighbor is 5

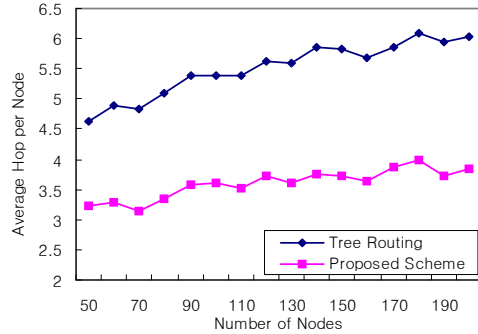


그림 6. 목적지 노드가 임의의 노드이고 최대 이웃 노드 수가 5일 경우 목적지 도달까지의 노드 당 평균 홉 수

Fig. 6. Average hop counts per a node when the destination is random and number of MaxNeighbor is 5

드로 선택하게 된다. 만약, 노드 v의 이웃 노드들 중 계층적 라우팅의 남은 홉 수보다 작은 라우팅 비용을 갖는 노드가 탐색되지 않을 경우, 계층적 라우팅을 따르게 된다. 그러므로 제안된 알고리즘은 최악의 경우 계층적 라우팅에 따라 데이터를 전달하게 되므로, 계층적 라우팅의 라우팅 비용을 초과할 수 없다.

### 3. 시뮬레이션 결과

시뮬레이션을 통한 성능 평가에서는 지그비의

계층적 라우팅과 제안 알고리즘의 라우팅 비용의 비교를 통해 성능 향상의 정도를 평가한다. 시뮬레이션에서 사용한 네트워크 크기는 100x100, 각 노드의 최대 통신거리는 20으로 동일하게 설정하였으며, 각 노드의 위치는 임의의 값으로 주어진다. 따라서 각 시뮬레이션에서의 네트워크 토폴로지는 항상 변하며, 노드의 위치에 따라 네트워크에 합류하지 못하는 노드가 생길 수 있다. 본 성능 평가에서는 주어진 네트워크의 총 노드 수 중 80% 이상의 노드가 네트워크를 구성할 경우에만 시뮬레이션 결과를 인정하였다. 또한, 지그비의 네트워

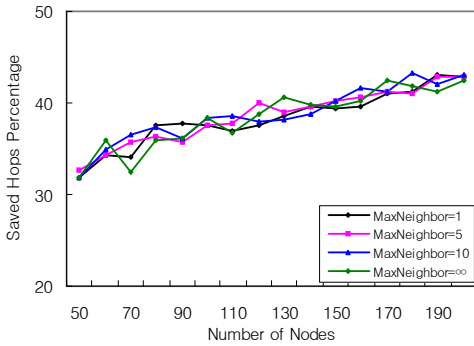


그림 7. 목적지 노드가 코디네이터일 경우 최대 이웃 노드의 개수에 따른 계층적 라우팅 대비 라우팅 비용 절감율

Fig. 7. The percentage of saved hops of proposed scheme according to number of MaxNeighbor (destination is coordinator)

크 구성 및 디스커버리 절차를 따르기 위하여  $C_m=4$ ,  $R_m=4$ ,  $L_m=5$ 로 설정하여, 각 노드가 가질 수 있는 자식 노드 수와 최대 트리 깊이를 제한하였다.

지그비 표준에서는 이웃 노드 테이블의 크기가 제한되어 있을 때, 이웃 노드 엔트리의 선택에 대하여 정의되어 있지 않다. 즉, 최대 이웃 노드 수 (MaxNeighbor)만큼 엔트리가 저장되어 있을 경우, 이웃 노드 엔트리를 업데이트하지 않는 방법, 링크가 가장 좋은 이웃 노드들로 계속해서 업데이트하는 방법 등 다양한 정책이 적용될 수 있다. 본 논문에서는 코디네이터와 나머지 노드들 간의 통신을 최적화하기 위하여 이웃 노드 테이블이 최대 이웃 노드 수만큼의 엔트리를 가지고 있다고 하더라도, 이웃 노드의 엔트리보다 낮은 트리 레벨을 갖는 이웃 노드가 있을 경우 이웃 노드 테이블을 업데이트하는 방법을 선택하였다. 또, 부모 및 자식 노드 이외에 순수 이웃 노드들만 이웃 노드 테이블에 저장하도록 하였다.

그림 3과 4는 최대 이웃 노드 수를 5, 네트워크에 합류한 모든 노드들의 목적지 노드를 코디네이터로 설정하여 모든 노드들의 라우팅 비용의 합을 계산함으로써, 지그비 계층적 라우팅과 제안한 방법의 라우팅 비용을 비교하였다. 이 때, 라우팅 비용은 동일한 네트워크 토폴로지에서 계산되어진다. 그림 3에서 보는 것처럼 네트워크의 총 노드 수가 증가하면서 총 라우팅 홉 수는 증가하는 현상을 확인할 수 있으며, 제안된 알고리즘에서의 라

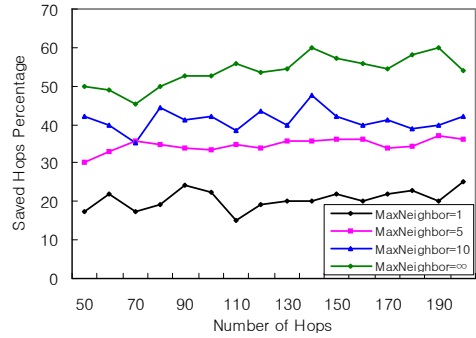


그림 8. 목적지 노드가 임의의 노드일 경우 최대 이웃 노드의 개수에 따른 계층적 라우팅 대비 라우팅 비용 절감율

Fig. 8. The percentage of saved hops of proposed scheme according to number of MaxNeighbor (destination is random)

우팅 비용이 계층적 라우팅에 비해 현저하게 낮은 사실을 발견할 수 있다. 또한, 그림 4에서는 네트워크의 총 노드 수에 따라 목적지 도달까지의 각 노드 당 평균 홉 수를 비교해 볼 수 있으며 계층적 라우팅에서의 각 노드 당 평균 홉 수가 3~4인 것에 비해, 제안한 알고리즘은 약 2홉만이 소모된다. 계층적 라우팅에서는 네트워크의 총 노드 수가 증가함에 따라 각 노드 당 평균 홉 수가 증가하는데 비해 제안한 방식에서는 일정한 경향이 나타나는 데, 이는 네트워크의 총 노드 수가 증가함에 따라 높은 트리 레벨을 가진 노드가 증가하는 반면 부모 노드보다 낮은 트리 레벨을 가진 이웃 노드 함께 증가하기 때문이다.

그림 5와 6에서는 최대 이웃 노드 수를 5, 목적지 노드를 임의의 노드로 설정하고 시뮬레이션을 50회 이상 반복하여 평균 라우팅 비용을 비교함으로써 성능을 평가해 보았다. 일반적으로 목적지 노드가 임의의 노드일 경우 패킷이 목적지까지 내려와야 되기 때문에, 모든 노드를 고려했을 경우 목적지가 코디네이터일 경우보다 라우팅 비용이 크다고 할 수 있다. 따라서 그림 5에서 보는 것처럼, 목적지 노드가 임의로 선택될 경우 계층적 라우팅에 소요되는 총 홉 수는 목적지 노드가 코디네이터인 경우보다 2배 가까이 많은 것을 확인할 수 있다. 또, 네트워크의 총 노드 수가 증가함에 따라, 각 노드 당 평균 홉 수가 4.5에서 6까지 증가하게 된다. 제안한 알고리즘에서 역시 목적지

노드가 코디네이터일 때보다 각 노드 당 평균 홉 수가 약 1에서 1.5홉 증가된다. 그러나 동일한 환경에서의 계층적 라우팅과 비교할 때에 목적지 노드가 코디네이터인 경우보다 노드 당 평균 홉 수의 차는 오히려 증가하는 것을 확인할 수 있다. 이는 제안한 알고리즘의 기본적인 목적과 부합하는 결과임을 확인할 수 있으며, 각 노드가 전송하는 패킷은 목적지 노드를 자식으로 하는 노드까지 트리를 따라 올라가는 것이 아니라, 목적지 노드까지의 라우팅 비용이 가장 작은 이웃 노드에게 패킷을 전달함으로써 라우팅 비용이 절감된 결과이다.

마지막으로 그림 7과 8에서는 각 노드가 가질 수 있는 최대 이웃 노드의 수를 조절하면서 계층적 알고리즘 대비 제안한 알고리즘의 성능 향상 정도를 측정하였다. 그림 7은 목적지 노드가 코디네이터일 때, 이웃 노드의 개수를 1, 5, 10, 무한대로 설정하여 계층적 알고리즘에 필요한 라우팅 비용 대비 절감된 라우팅 비용을 보여준다. 성능 평가에서는 각 경우에 대해 큰 차이가 나타나지 않는 것을 볼 수 있으며 이는 이웃 노드 테이블의 관리 정책이 낮은 레벨의 이웃 노드를 우선으로 저장하기 때문임을 알 수 있다. 즉, 코디네이터로 데이터를 전달하기 위해서는 이웃 노드들 중 트리 레벨이 가장 작은 노드를 다음 홉의 노드로 선택하게 되는데 최대 이웃 노드의 개수와 상관없이 트리 레벨이 가장 작은 이웃 노드는 같기 때문이다. 이 점을 분석해 보면 제안한 방식에서의 라우팅 비용은 지그비의 테이블 기반 라우팅의 성능과 유사하다는 점을 알 수 있다. 왜냐하면, 최단 거리를 통해서 코디네이터까지 데이터를 전달하기 위해서는 이웃 노드 내에 트리 레벨이 가장 낮은 노드가 중간 노드로 선택되어야 하기 때문이다.

그림 8에서는 목적지 노드가 임의의 노드이기 때문에 이웃 노드의 엔트리 수가 많을수록 계층적 라우팅을 이용할 때의 라우팅 비용보다 적은 라우팅 비용을 가지는 이웃 노드를 가질 가능성을 높아진다. 따라서 이웃 노드의 수가 증가함에 따라 절감된 총 홉 수의 비율은 약 20%에서 50%로 급격하게 증가되는 사실을 확인할 수 있다. 하지만 이웃 노드를 제한하지 않았을 때 절감된 총 홉 수의 비율이 약 50%인데 비해, 최대 이웃 노드의 수를 5~10정도만 설정하더라도 약 30~40%의 라우팅 비용을 줄일 수 있다는 점은 의미 있는 결과라고 할 수 있다.

## VI. 결 론

본 논문에서는 지그비 계층적 라우팅의 라우팅 비용의 비효율성 문제를 제기하고, 지그비 표준에서 정의된 이웃 노드 테이블을 이용하여 성능을 향상할 수 있는 방안을 제안하였다. 제안한 알고리즘은 이웃 노드 테이블의 각 엔트리에 대한 목적지 노드까지의 남은 홉 수를 검사함으로써 최소의 값을 가진 노드를 다음 홉의 노드로 선택하게 하였으며, 이는 최대 이웃 노드의 개수가 제한되어 있지 않을 경우에도 다항식 시간 안에 풀 수 있는 만큼 효율적인 알고리즘이다. 또한, 다음 홉의 선택이 지역 최소화 기법을 바탕으로 하고 있음에도 루프와 같은 문제가 발생하지 않는다.

시뮬레이션을 통한 성능 평가에서는 제안한 방식을 사용할 때에 계층적 라우팅에 필요한 라우팅 비용의 30% 이상을 절감할 수 있는 것을 확인하였으며, 특히, 목적지 노드가 코디네이터일 경우 이웃 노드 테이블을 활용하는 것만으로 지그비의 테이블 기반 라우팅의 성능과 유사한 효과를 낼 수 있다. 또한, 목적지 노드가 임의의 노드일 경우에는 각 노드가 얼마나 유용한 이웃 노드의 엔트리를 많이 가지고 있는지에 따라 라우팅 비용의 절감 정도가 다르게 나타났으며, 이는 네트워크의 응용에 따라 최대 이웃 노드의 수 및 이웃 노드 테이블의 관리 정책을 달리함으로써 성능을 최적화할 수 있다. 따라서 제안한 알고리즘의 지그비 적용은 적은 메모리를 사용하면서도 효율적인 네트워크 성능을 요구하는 많은 응용 분야에서 활용될 수 있을 것으로 기대된다.

## 참고문헌

- [1] ZigBee Alliance, "ZigBee Specification 1.0", Dec. 2004
- [2] J. Kim, H. Lee, D. Hwang, B. Kim, "Development Trend of Standards for Low-Rate, Low-Cost, and Low-Power Wireless PAN", Vol. 18, No. 2, pp.37, ETRI trends, 2003.
- [3] ZigBee Alliance, [www.zigbee.org](http://www.zigbee.org)
- [4] "Wireless Medium Access Control(MAC) and Physical Layer(PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)", IEEE Std 802.15.4-2003, IEEE Computer Society, 01 October 2003



- [5] Gang Ding et al, "Efficient and Reliable Broadcast in ZigBee Networks ", IEEE Transaction on Mobile Computing
- [6] D. Kim, Y. Doh, S. Yoo, K. Chang, W. Park, C. Seo, "Low Rate WPAN Technolgies and Standards ," KISS Information and Communications Journal, Dec. 2004.
- [7] C. Jung, D. Woo, Y. Kim, I. Chun, C. Lim, "Nano Esto: An IDE for USN Application Developers", Journal of IEMEK, Vol. 1, No. 1, June 2006.

### 저 자 소 개

#### 김 태 홍

2005년 아주대학교 컴퓨터공학과 학사. 현재, 한국정보통신대학교 컴퓨터공학 석사과정. 관심분야: 지그비, 센서네트워크 라우팅 프로토콜

Email: damiano@icu.ac.kr

#### 김 대 영

1990년 부산대학교 전산통계학과 학사. 1992년 부산대학교 전산통계학과 석사. 2001년 University of Florida 컴퓨터공학 박사. 현재, 한국정보통신대학교 조교수. 관심분야: 센서네트워크, 실시간 임베디드 시스템, 애드혹 네트워크

Email: kimd@icu.ac.kr

#### 유 성 은

2003년 한양대학교 전자전기공학부 학사. 2005년 한국정보통신대학교 컴퓨터공학 석사. 현재, 한국정보통신대학교 컴퓨터공학 박사과정. 관심분야: 센서네트워크, WPAN, 실시간통신

Email: seyoo@icu.ac.kr

#### 심 종 우

2002년 성균관대학교 정보통신공학부 학사. 2004년 한국정보통신대학교 컴퓨터공학 석사. 현재, 한국정보통신대학교 컴퓨터공학 박사과정. 관심분야: 홈 네트워크, 센서네트워크 미들웨어

Email: jwsung@icu.ac.kr

#### 김 영 수

1994년 공군사관학교 학사. 2001년 서강대학교 컴퓨터공학 석사. 현재, 한국정보통신대학교 컴퓨터공학 박사과정. 관심분야: 센서네트워크

Email: pineland@icu.ac.kr