

# 클러스터 라벨 기반의 지그비 메쉬 라우팅 프로토콜

정회원 이 광 국\*, 준회원 김 성 훈\*, 종신회원 박 흥 성\*\*°

## Cluster Label-based ZigBee Mesh Routing Protocol

Kwang-Koog Lee\* *Regular Member*, Seong-hoon Kim\* *Associate Member*,  
Hong-seong Park\*\*° *Lifelong Member*

### 요 약

기존의 지그비 표준 메쉬 라우팅 알고리즘은 노드 수의 증가로 인해 라우팅 오버헤드 및 중단간 지연 등의 네트워크 성능의 심각한 저하 현상에 따른 네트워크 범위성 및 효율성에 대한 문제점을 가지고 있다. 이에 따라 본 논문에서는 증가하는 노드 수에 따른 범위성 문제를 해결하기 위해 ZiCL(ZigBee Cluster Label)이라 불리는 클러스터 라벨 기반의 새로운 지그비 메쉬 라우팅 알고리즘을 제안하고자 한다. ZiCL은 지그비 네트워크를 한 개 이상의 논리적인 클러스터로 나누고 각각의 클러스터에게 유일한 라벨을 부여함으로써 클러스터 라벨 기반의 경로 탐색을 수행한다. 이를 통해 ZiCL은 소스 노드가 경로탐색을 통해 확립한 목적지 노드의 클러스터 라벨 정보를 같은 클러스터 멤버들과 함께 공유하게 함으로써 이후 발생할 수 있는 잠재적인 경로 탐색 횟수를 효과적으로 줄이는데 기여한다. 이러한 ZiCL의 성능을 분석하기 위해 NS-2 시뮬레이션을 수행하였고, 그 결과 라우팅 오버헤드, 패킷 전달율, 중단간 지연 및 메모리 사용률에 대한 네트워크 성능이 향상됨을 확인할 수 있었다.

**Key Words :** Wireless Mesh Networks, ZigBee, IEEE 802.15.4, ZigBee Cluster Label (ZiCL)

### ABSTRACT

To solve scalability problem in the ZigBee Network, this paper presents a new mesh routing protocol for ZigBee, called ZigBee Cluster Label (ZiCL). ZiCL divides the ZigBee network into one or more logical clusters and then assigns a unique Cluster Label to each cluster so that it discovers a route of a destination node based on Cluster Label. When a node collects new Cluster Label information of a destination node according to discovery based on Cluster Label, ZiCL encourages nodes with the same Cluster Label to share the information. Consequently, it contributes on reducing numerical potential route discoveries and improving network performances such as routing overhead, end-to-end delay, and packet delivery ratio. Simulation results using NS-2 show ZiCL performs well.

### I. 서 론

지그비<sup>[1]</sup>는 저전력, 저가격 및 사용의 용이성을 특징으로 갖는 무선 단거리 표준 기술의 하나로서 IEEE 802.15.4<sup>[2]</sup>를 기반으로 메쉬형(Mesh) 토폴로지를

를 지원한다. 지그비에서 이러한 메쉬 토폴로지를 지원하기 위한 메쉬 라우팅 알고리즘은 AODV(Ad hoc On-demand Distance Vector)<sup>[5]</sup>와 유사한 방식을 갖고 있으므로 멀티홉 환경에서 최적 경로를 통한 데이터의 전송을 가능하게 한다. 하지만 이러한 메쉬 라

※ 본 연구는 BK21 및 산업자원부의 지원에 의해 수행되었습니다.

\* 강원대학교 전자통신공학과 산업정보통신 연구실 ((kwangkooglee, shkim1980}@gmail.com)

\*\* 강원대학교 IT특성화학부 전기전자공학부 산업정보통신 연구실 (hspark@kangwon.ac.kr)(° : 교신저자)  
논문번호 : KICS2006-12-532, 접수일자 : 2006년 12월 14일, 최종논문접수일자 : 2007년 11월 5일

우팅 프로토콜은 일반적인 무선 이동 네트워크 (Mobile Ad Hoc Network, MANET)의 라우팅 프로토콜을 그대로 수용하고 있기 때문에, 저용량, 저전력 및 저성능을 특징으로 갖는 지그비 메쉬 네트워크에서 노드 수의 증가에 따라 네트워크의 성능 저하 측면에 있어 문제가 될 수 있다. K. Akkaya와 M. Younis는 이러한 문제에 대하여 무선 센서 네트워크 (Wireless Sensor Network)의 일반적인 특성을 나열 하면서 무선 이동 네트워크의 라우팅 프로토콜이 무선 센서 네트워크에 적합하지 않음을 언급하였다<sup>11</sup>. 또한, S. Lee와 E. M. Royer는 AODV 라우팅 프로토콜을 사용하여 노드 수에 따라 중단간 지연, 라우팅 오버헤드, 패킷 전달율 등의 네트워크 성능을 시뮬레이션 함으로써, 노드 수의 증가에 따른 심각한 네트워크 성능 저하현상에 대해 무선 라우팅 프로토콜에 대한 범위성(Scalability) 문제를 언급하였다<sup>12</sup>. 게다가, 저용량의 특징을 갖는 지그비 센서의 메모리를 고려한다면 AODV 기반의 라우팅 프로토콜은 연결되는 노드의 증가에 따라 라우팅 테이블이 커지기 때문에 잦은 테이블 교체 및 경로의 재탐색 등의 측면에서 불필요한 에너지 소비 및 네트워크의 성능 저하 등으로 인한 추가적인 문제가 발생할 수 있다. 결과적으로 지그비의 저용량 및 저전력 특징을 고려하면서 네트워크의 성능을 보장하기 위한 새로운 메쉬 라우팅 프로토콜의 개발이 필요하다.

따라서 본 논문에서는 지그비 네트워크에서 범위성 문제를 해결하기 위해 클러스터 라벨 기반의 지그비 메쉬 라우팅 프로토콜(ZigBee Cluster Label, ZiCL)을 제안한다. 본 논문은 기존의 연구<sup>8)</sup>되었던 지그비 메쉬 라우팅의 확장된 연구로써 게이트웨이 노드의 선정 방법과 2-홉 이웃 라우팅 기법 및 라벨 정보의 공유 방법 등을 추가로 소개한다. 또한 범위성에 관련하여 보다 다양한 시뮬레이션 시나리오를 포함한다. ZiCL은 지그비 네트워크를 한 개 이상의 논리적인 클러스터로 분할하고 각각의 클러스터에 대해 유일한 클러스터 라벨을 부여함으로써 클러스터 라벨 단위의 경로 탐색을 수행한다. 이러한 경로 탐색을 기반으로 ZiCL은 클러스터 내부의 임의의 노드가 경로 탐색을 통해 확립한 목적지 클러스터 라벨 정보에 대하여 같은 클러스터 멤버들과 공유하도록 함으로써, 다른 클러스터 멤버가 별도의 경로 탐색 없이도 동일한 목적지 클러스터 라벨을 갖는 목적지 노드에 대하여 데이터의 전송을 수행할 수 있게 한다. 이는 네트워크에서 노드의 개수가 증가함에 따라 잠재적인 경로 탐색의 횟수를 효과적으로 줄일 수 있기 때문에, 경로 확립에

따른 라우팅 오버헤드 및 패킷 전달율 등에서 표준 메쉬 라우팅 방식보다 향상된 네트워크 성능을 보여 준다. 또한 지그비 표준 메쉬 라우팅에 비해 메모리 사용률을 낮춤으로써 저용량 및 저전력 기반의 지그비 메쉬 네트워크에 적합함을 보여준다. NS-2<sup>10)</sup>를 사용한 시뮬레이션 결과는 ZiCL이 노드의 개수의 증가에 따라 기존의 방식에 비해 개선된 네트워크의 성능과 메모리 사용률을 보여준다.

본 논문의 구성은 다음과 같다. II장에서는 IEEE 802.15.4를 기반으로 하는 메쉬 라우팅 프로토콜들에 대하여 소개한다. III장에서는 제안한 ZiCL의 라우팅에 대하여 설명하고 IV장에서는 NS-2 시뮬레이션을 통한 ZiCL의 성능을 분석 및 평가한다. 마지막으로 V장에서 결론 및 차후과제를 끝으로 본 논문을 마무리한다.

## II. IEEE 802.15.4 기반의 메쉬 라우팅 프로토콜

### 2.1 지그비 네트워크

지그비 노드는 코디네이터, 라우터, 종단 디바이스의 세 가지 타입으로 분류된다. 코디네이터 및 라우터는 지그비 네트워크를 형성하거나 다중 홉 데이터 전송을 위한 라우터 기능을 갖고 네트워크 계층에서 설정되는 네트워크 환경 설정 파라미터를 기반으로 자식으로 참여하는 노드들에 대하여 주소를 할당하거나 최적 경로를 설정하기 위한 경로 탐색 과정에 참여하는 노드를 말한다. 이와 달리 종단 디바이스는 코디네이터 및 라우터의 기능을 최소화한 노드로써 주소를 할당하는 능력이 없을 뿐만 아니라 경로 탐색 과정에 참여하지 못하고 데이터의 다중 홉 전송을 지원하지 못하기 때문에 라우팅 과정에서 중간노드의 역할을 수행하지 못한다.

지그비 네트워크는 계층적으로 형성되는 16비트의 네트워크 주소를 통해 깊이에 따른 트리 구조를 갖는다. 깊이 레벨  $d$ 에서 네트워크 주소는  $C_{skip}(d) = (1 + C_m - R_m - C_m \times R_m^{L_m - d - 1}) / (1 - R_m)$  공식에 의해 구분된다. 여기서  $C_m$ 은 최대 자식의 노드 수,  $R_m$ 은 최대 라우터 개수,  $L_m$ 은 최대 자식의 깊이 값을 의미한다. 위 공식을 통해서 참여하는 라우터들은 일정 주소 공간을 갖는 주소 블록을 할당 받는다. 주소 블록을 할당받은 라우터는 블록 상에서 최우선한 주소 값을 자신의 네트워크 주소로 설정하고 동일한 방식으로 참여하는 자식들에게 블록을 할당한다. 반면에, 코디네이터 및 라우터 이하 종단 디바이스로 참

여하는 지식 노드들은  $A_n = A_{parent} + Cskip(d) \times R_m + n$  공식에 의해 한 개의 네트워크 주소를 할당 받는다. 여기서 n은 주소를 할당하는 부모 입장에서 n번째로 참여하는 중단 디바이스를 의미한다.

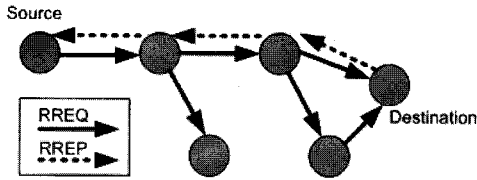


그림 1. 경로 탐색 과정

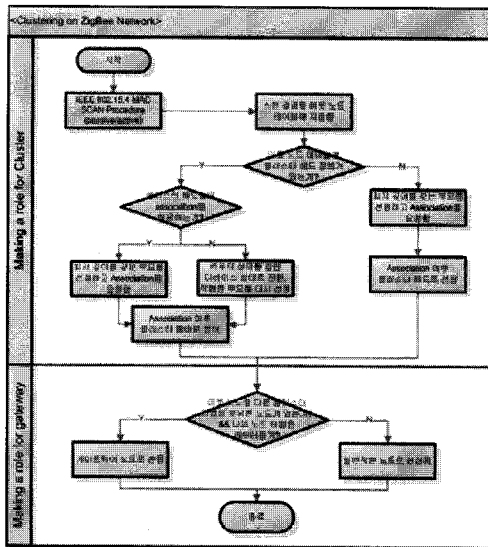


그림 2. 클러스터의 형성 및 게이트웨이 역할 선정

### 2.2 지그비 AODV 라우팅 기법

지그비는 멀티홉 환경의 메쉬 토폴로지를 지원하기 위해 MANET의 AODV<sup>[5]</sup>와 유사한 라우팅 방식을 사용한다. AODV는 그림 1과 같이 소스 노드와 목적지 노드 간 최적의 경로를 형성하기 위해 소스 노드로부터 RREQ(Route Request) 메시지를 네트워크상에 브로드캐스트 한다. 해당 메시지는 목적지 노드가 받을 때까지 중간 노드들을 통해서 계속해서 브로드캐스트 된다. 만일 목적지 노드가 RREQ 메시지를 수신하면 역경로를 통해 RREP(Route Reply) 메시지를 유니캐스트 함으로써 경로를 형성하게 된다.

앞서 언급된 바와 같이 지그비 AODV는 노드의 확장에 따라 네트워크 성능이 저하되거나 충분한 메모리를 요구하기 때문에 저전력 및 저용량 특징을 갖는 지그비에서 문제가 된다.

## III. ZiCL 라우팅 프로토콜

### 3.1 클러스터 형성 알고리즘

ZiCL은 그림 3에서 보는 바와 같이 지그비 네트

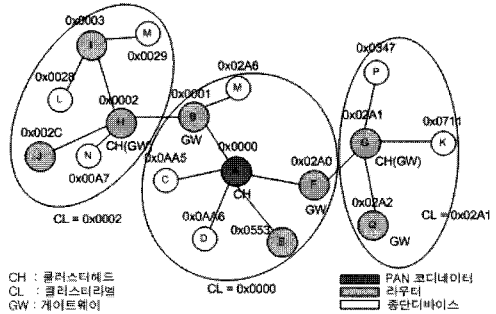


그림 3. ZiCL의 클러스터링

워크를 하나 이상의 중첩되지 않은 논리적인 클러스터로 나눈다. 각 클러스터는 하나의 클러스터 헤드와 다수의 클러스터 멤버를 가질 수 있으며 클러스터 내에서 게이트웨이 역할을 수행하는 노드들을 통해 연결될 수 있다. 또한 클러스터간의 식별을 위해 각 클러스터는 클러스터 라벨(Cluster Label)을 소유한다. 여기서 클러스터 라벨 정보는 클러스터 헤드로 선정된 노드의 16비트 네트워크 주소를 그대로 사용함으로써 주소의 유일성을 보장한다.

ZiCL은 클러스터 형성에 따른 오버헤드를 최소화하기 위해 지그비의 네트워크 참여 특성을 이용하여 클러스터를 형성하는 절차를 갖는다. 일반적으로 지그비 노드는 IEEE 802.15.4 기반의 MAC 계층으로부터 passive/active scan 과정을 거친 이후 적합한 부모를 선정하여 association을 수행해야 네트워크에 참여할 수 있다. 따라서 지그비 네트워크에 참여된 라우터 및 코디네이터들은 그림 2의 흐름도의 과정을 통해 클러스터 헤드와 클러스터 멤버로 나누어진다.

흐름도에서 보는 바와 같이 네트워크에 참여한 노드는 IEEE 802.15.4 MAC에서 active 및 passive scan 과정을 수행한다. 네트워크 계층에서는 MAC의 scan 과정이후 생성된 리스트를 통해 이웃 노드 테이블을 생성한다. 여기서 생성된 이웃 노드 테이블 내에 클러스터 헤드의 정보가 있다면 참여 노드는 해당 클러스터 헤드를 자신의 부모 노드로 선정하고 association 과정을 통해 클러스터 멤버로 참여한다. 만일 선정된 부모가 참여하는 노드들에 대해

association을 허가하지 않는 경우 자신을 종단 디바이스의 상태로 전환한 후 적합한 부모를 선정하여 클러스터 멤버로써 네트워크에 참여한다. 한편 참여 노드가 이웃 노드 테이블 내에 어떠한 클러스터 헤드 정보도 포함되어 있지 않는 경우 자신을 클러스터 헤드로 선정한다. 마지막으로 종단 디바이스들은 association 과정 이후 클러스터 멤버로써 네트워크에 참여하며 항상 부모 노드와 동일한 클러스터 라벨을 갖는다.

클러스터 헤드 및 클러스터 멤버의 역할이 구분되면 노드들은 게이트웨이 역할의 유무를 결정한다. 클러스터에 포함된 노드들은 자신의 클러스터 라벨 정보를 알기 때문에 생성된 이웃 테이블 내에 다른 클러스터 라벨 정보가 있는 경우 자신이 게이트웨이 노드인지 아닌지를 알 수 있다.

결국 클러스터 및 게이트웨이 선정을 통해 지그비 네트워크는 그림 3과 같은 구조를 형성한다. 각 클러스터 내부는 지그비의 계층적인 구조를 바탕으로 형성되며 클러스터 헤드로 선출되는 노드들은 2홉의 간격을 유지하며 게이트웨이를 통해 연결된다. 클러스터 내에서 부모가 다른 종단 디바이스 간의 연결을 고려하면 클러스터 내에서 최대 4홉을 유지되며 라우터 간에는 2홉을 유지한다.

### 3.2 목적지 노드의 클러스터 라벨 계산 알고리즘

일반적인 클러스터 기반의 라우팅<sup>[7]</sup>은 임의의 소스 노드가 목적지 노드에게 데이터를 전송하기 위해 목적지 노드가 속한 클러스터의 주소를 알아야 한다. 따라서 대부분의 라우팅은 목적지 노드의 클러스터 주소를 미리 알고 있다고 가정을 하거나 목적지 노드의 클러스터 주소를 알아내기 위한 경로 탐색과정을 수반한다. 하지만 ZiCL은 목적지 노드의 16비트 주소를 통해 목적지 노드의 클러스터 라벨을 계산할 수 있다. 클러스터 라벨을 계산하려면 지그비의 주소할당체계와 네트워크 설정을 위한 파라미터들을 사용해야 한다. 이러한 파라미터는  $C_m$  (자식의 최대 개수),  $R_m$  (라우터의 최대 개수),  $L_m$  (깊이의 최대 개수)으로써 지그비 표준<sup>[4]</sup>에서 정의하고 있다. 클러스터 형성 규칙에 의하면 코디네이터와 2홉 떨어진 위치의 라우터들이 클러스터헤드로 선출되기 때문에 코디네이터가 할당하는 주소를 시작으로 주소의 할당값 깊이에 따라 추적함으로써 목적지 노드의 깊이, 타입 및 부모의 주소에 대하여 계산할 수 있다.

### 3.3 ZiCL의 라우팅

ZiCL의 라우팅은 2홉(hop) 이웃 라우팅기법과 클러스터 내 라우팅과 클러스터 간 라우팅 방식으로 나누어진다. 이 두 라우팅 방식의 세부 동작은 그림 4에 표시되어 있다.

2홉 이웃 라우팅 기법은 클러스터의 구조와 관계

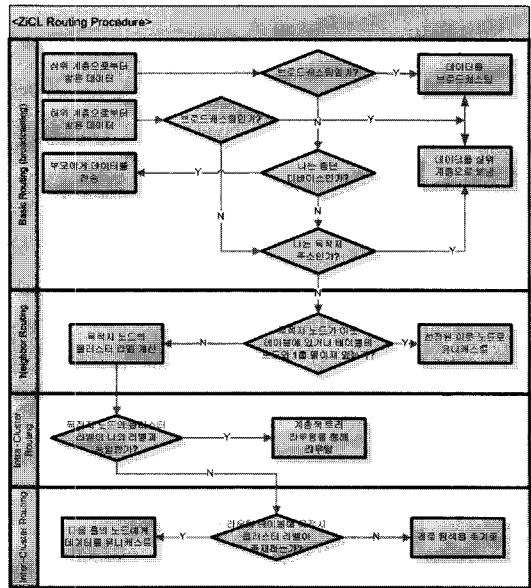


그림 4. ZiCL의 라우팅 절차

없이 이웃 노드 테이블의 정보를 바탕으로 1홉 또는 2홉 바깥에 있는 노드들에게 바로 데이터를 포워딩할 수 있는 방법을 말한다. 만일 목적지 노드가 이웃 노드 테이블에 속하거나 해당 테이블 내에 속하는 노드의 부모 노드 혹은 자식 노드인 경우 라우팅 테이블 없이 그대로 데이터를 포워딩 할 수 있게 된다. 하지만 2홉 바깥에 위치한 목적지 노드가 근원지 노드의 이웃 노드 테이블에 저장된 노드들과 계층성이 없으면 아래의 클러스터 내 라우팅 혹은 클러스터 간 라우팅을 과정을 수행해야 한다.

클러스터 내 라우팅(Intra-cluster routing)의 경우 같은 클러스터 라벨을 소유한 멤버들 간의 데이터 전송을 위한 라우팅을 의미한다. ZiCL은 목적지 클러스터 라벨의 계산에 의해 목적지 노드가 같은 클러스터 라벨에 속하고 있음을 확인할 수 있다. 따라서 ZiCL은 같은 클러스터 라벨을 갖는 목적지에 대하여 지그비의 계층형 구조를 이용한 트리 라우팅을 통해 별도의 경로탐색 없이 데이터를 즉시 포

워딩한다.

클러스터 간 라우팅(Inter-cluster routing)은 서로 다른 클러스터 라벨을 소유한 노드 간의 라우팅을 의미한다. 목적지 클러스터 라벨 계산 수행을 통해 목적지 노드가 다른 클러스터 라벨에 속하는 경우 소스 노드는 라우팅 테이블을 검색하여 라우팅을 수행한다. 만일 라우팅 테이블 내에 목적지 노드의 클러스터 라벨 정보가 없는 경우 경로탐색 과정을 통해 목적지 노드와 경로를 확립하여 데이터를 전송한다.

### 3.4 ZiCL의 경로 탐색

ZiCL의 경로 확립을 요약하면 다음과 같다<sup>[8]</sup>. 은 지그비 표준과 유사하게 RREQ(Route Request) 메시지를 네트워크상에 브로드캐스팅 하는 것으로부터 시작된다. 이때 해당 RREQ를 받은 노드들은 새로운 클러스터 라벨에 대한 라우팅 엔트리를 생성하고 목적지 노드의 네트워크 주소 대신 소스 노드로부터 계산된 목적지 클러스터 라벨 정보를 저장한다. 이와 동시에 노드는 목적지 노드의 주소에 대하여 지그비 표준과 동일한 경로 탐색 테이블 엔트리를 생성한다. 최종적으로 해당 RREQ 메시지를 수신한 목적지 노드는 기존의 라우팅 방식과 동일한 RREP(Route Reply) 메시지를 RREQ 메시지에 의해 형성된 경로 탐색 테이블의 역경로를 통해서 유니캐스트 한다. 이때 RREP 메시지를 수신한 중간 노드들은 RREQ에 의해 생성된 라우팅 테이블을 활성화시키고 Next Hop Address 필드에 RREQ를 보낸 노드의 주소를 저장한다.

### 3.5 클러스터 라벨 정보의 공유

ZiCL은 클러스터 라벨 정보를 공유함으로써 노드들의 잠재적인 경로 탐색을 줄일 수 있다. 클러스터 라벨 정보의 공유는 네트워크 참여를 통한 방법과 경로탐색을 통한 방법으로 나누어지며 두 방법 모두 1홉 브로드캐스팅 방법을 사용한다. 여기서 1홉 브로드캐스팅이란 브로드캐스트 패킷을 1홉의 노드들에게만 제한하는 브로드캐스트 방식을 의미한다.

우선 네트워크 참여를 통한 클러스터 라벨 공유 방법에서 네트워크에 새롭게 참여한 노드는 클러스터 멤버가 된 이후 RASK(Route Ask) 메시지를 1홉 브로드캐스트한다. 또한 클러스터 헤드가 게이트웨이 역할을 동시에 수행할 경우 혹은 클러스터 멤버가 게이트웨이로 선정된 경우 역시 RASK 메시지를 1홉 브로드캐스트하고 해당 메시지 내에 이웃한

클러스터 라벨의 정보를 포함시킨다. 이후 RASK 메시지를 수신받은 1홉 이내의 노드들 중 같은 라벨을 소유한 클러스터 헤드는 RASK 메시지로부터 새로운 라벨 정보를 갱신하고 RUPT(Route Update) 메시지를 1홉 브로드캐스트한다. RUPT 메시지를 수신한 같은 라벨을 갖는 클러스터 멤버들은 자신의 라우팅 테이블을 갱신한다. 반면에 RASK 메시지를 수신한 노드들이 같은 클러스터 멤버가 아닌 경우

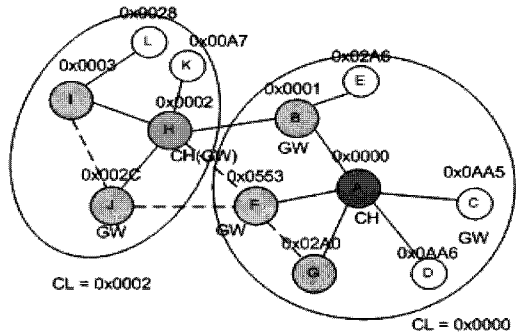


그림 5. 네트워크 참여를 통한 라벨정보 공유

새로운 라벨 정보를 RNOT(Route Notify) 메시지를 통해 클러스터 헤드에게 통보한다.

예를들면 그림 5에서 보이는 네트워크에서 노드 H가 추가되는 경우 노드 H는 association 과정 이후 이웃 노드 테이블에 저장하는 노드 B와 F를 갖는다. 노드 H는 현재 자신의 이웃에 클러스터 헤드가 없기 때문에 두 노드 중 적합한 부모 노드를 선정하고 자신이 클러스터 헤드(라벨: 0x0002)로 선정된다. 이후 노드 H는 이웃 노드 테이블을 통해 자신이 게이트웨이 역할을 수행해야 하므로 RASK 메시지를 1홉 브로드캐스트한다. 브로드캐스트 메시지가 노드 B와 F에 전달됨으로써 두 노드들은 자신들이 게이트웨이 노드로 선정되었음을 알게된다. 따라서 이웃 노드 테이블에 노드 H의 정보를 추가하고 RNOT를 통해 새로운 0x0002 라벨 정보를 클러스터 헤드 A에게 전달한다. 이후 해당 메시지를 수신한 노드 A는 RUPT를 통해 0x0002로 가기위한 게이트웨이가 B와 F가 존재함을 클러스터 내부의 멤버들에게 알린다. 결국 노드 G의 경우 노드 A로부터 전송된 RUPT를 수신받고 0x0002 라벨의 정보를 라우팅 테이블에 유지한다.

경로탐색을 통한 방법은 임의의 한 근원지 노드가 경로탐색을 통해 확립한 목적지 노드의 클러스터

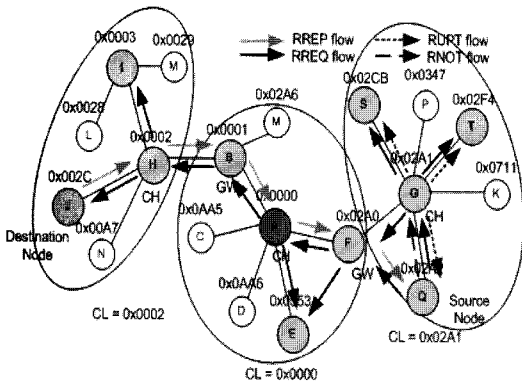


그림 6. 경로탐색을 통한 라벨정보 공유

터 라벨 정보에 대해 클러스터 멤버들과 공유하는 방법을 말한다. 이 과정은 근원지 노드가 RREP를 수신한 이후 RNOT 메시지를 클러스터헤드에게 통보하면서 이루어진다. 이 메시지에는 새로운 클러스터 라벨 정보가 들어간다. 만일 최종적으로 RREP를 수신한 노드가 클러스터헤드 자신이면 이 메시지는 전송되지 않는다. 이후 해당 RNOT를 수신한 클러스터헤드는 동일한 클러스터 라벨을 갖는 노드들과 해당 클러스터 라벨 정보를 공유하기 위해 RUPT 메시지를 클러스터 내에 1홉 브로드캐스트한다. 이를 수신하는 클러스터 멤버들은 각자 해당되는 원격 클러스터 라벨에 대한 라우팅 엔트리를 생성함으로써 해당 클러스터 라벨을 소유한 임의의 노드에게 경로 탐색 없이 데이터를 전송할 수 있게 된다.

예를들면 그림 6과 같이 노드 Q가 노드 J에게 데이터를 보내려 하는 경우, 노드 Q는 우선 노드 J의 클러스터 라벨을 계산한다. 계산된 클러스터 라벨 정보가 라우팅 테이블에 없으면 노드 Q는 노드 J를 찾기 위한 RREQ 메시지를 네트워크에 브로드캐스트 한다. 이때 중간 노드들은 해당 메시지를 받을 때 자신의 라우팅 테이블 내에 클러스터 라벨 0x0002를 저장한다. 이후 노드 J는 해당 RREQ 메시지를 수신하면 RREQ를 통해 생성된 역경로를 통해 RREP를 유니캐스트 한다. H를 지나는 RREP 메시지는 클러스터 라벨 0x0000에 속한 노드 B, A, F에게 수신되고 이들은 RREQ 수신시 생성된 0x0002의 라우팅 엔트리를 활성화한다. 이후 노드 F는 RREQ를 통해 최적의 역경로가 노드 Q를 가리키므로 클러스터헤드 G를 거치지 않고 노드 Q에 바로 RREP를 전송한다. 결국 최종적으로 RREP를 수신한 노드 Q 또한 목적지 클러스터 라벨에 대한 라우팅 엔

트리를 활성화하고 노드 J와 경로를 확립함으로써 데이터를 보내기 시작한다. 추가적으로 노드 J는 클러스터 멤버들과 해당 클러스터 라벨 정보를 공유하기 위하여 클러스터헤드 G에게 RNOT 메시지를 전송한다. 이후 RNOT를 수신한 클러스터헤드 G는 RUPT 메시지를 라벨 브로드캐스트를 하여 라우팅 기능을 갖는 클러스터 멤버들과 새로운 클러스터 라벨 0x0002를 공유하도록 한다. 따라서 노드 S와 T 역시 클러스터 라벨 0x0002를 소유한 임의의 노드에 대하여 경로 탐색 없이도 데이터를 전송할 수 있게 된다.

#### IV. ZiCL의 성능분석

본 논문에서는 제안된 ZiCL의 성능 분석을 위해 NS-2<sup>[10]</sup> 시뮬레이터를 사용하였다. 시뮬레이션 환경은 트래픽으로써 CBR(constant bit rate)를 사용하였고, 150초 동안 전체 노드의 개수를 40, 60, 80, 100, 120, 140의 순서로 차례대로 증가시키면서 각 시나리오에 대하여 시뮬레이션을 수행하였다. 패킷 사이즈는 80 bytes로 평균 패킷 전달율은 0.70 packets/sec인 지그비 네트워크 환경을 가정하였고, 지그비 네트워크는 코디네이터를 중심으로 노드의 개수에 따라 최대 반경 150m 이내로 조성하였다. 데이터 트래픽 소스의 개수는 각 시나리오마다 20, 40, 60로 설정하였다. 그리고 각 시나리오의 전체 노드의 개수에 대하여 라우터와 중단 디바이스의 비율은 0.25로 일정하게 유지하였다. 예를 들어, 시나리오의 전체 노드의 개수가 40개인 경우 코디네이터를 포함한 라우터는 10개, 중단 디바이스의 경우는 30개가 된다. 시뮬레이션에서 모든 노드들은 움직임이 없는 것으로 가정하였고 제안된 지그비 ZiCL과 지그비 AODV의 성능 비교를 위해 메모리 사용량, 중단 지연, 라우팅 오버헤드 및 패킷 전달율 등을 측정하였다.

##### 4.1 라우팅 오버헤드율

라우팅 오버헤드율은 시뮬레이션 시간동안 데이터 패킷과 라우팅 패킷과의 비율을 나타낸 값으로 값이 클수록 네트워크 내에 오버헤드가 커짐을 나타낸다. 그림 7에서 볼 수 있듯이 지그비 AODV와 ZiCL 모두 동일하게 노드의 증가에 따라 오버헤드율이 증가함을 알 수 있다. 데이터 트래픽 소스가 20개인 경우에는 정해진 시뮬레이션 시간동안

라우팅 패킷의 비율이 데이터 비율보다 높아지기 때문에 트래픽 소스의 개수가 감소할수록 오히려 오버헤드가 높은 것으로 측정되었다. 관찰된 결과를 통해 알 수 있듯이 ZiCL은 지그비 AODV에 비해 오버헤드가 적고 특히 노드 수가 140개인 경우 각 소스 트래픽에 대해 두 배 가량의 차이가 있음을 알 수 있다. 이는 ZiCL의 특성상 한 번의 경로 탐색으로 여러 노드들이 경로를 획득하고 같은 클러스터 내부 4홉 이내의 범위에서는 경로 탐색이 없는 계층적인 트리 라우팅을 수행하기 때문이다.

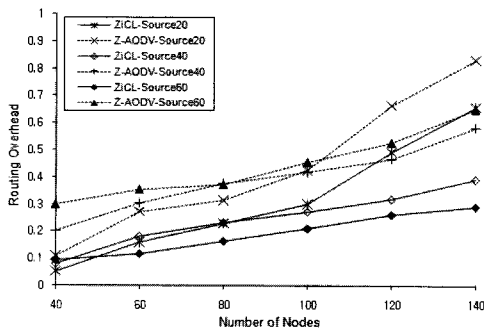


그림 7. 라우팅 오버헤드

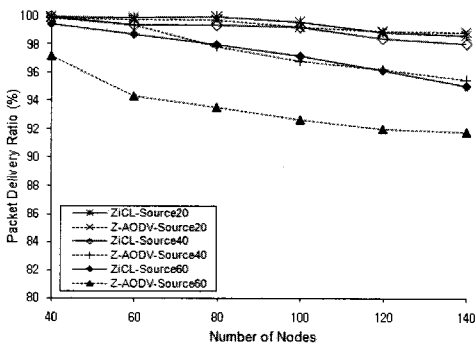


그림 8. 패킷 전달율

#### 4.2 패킷 전달율

패킷 전달율은 소스 노드에서 발생한 데이터 패킷의 수와 최종 목적지에 도달한 패킷 수의 비율을 의미한다. 그림 8에서 보는 바와 같이 전체적으로 노드의 개수가 증가함에 따라 각 트래픽 소스 별로 패킷 전달율이 낮아지는 것을 알 수 있다. 이는 노드의 수가 증가함에 따라 지그비 네트워크 내에 트래픽이 증가하면서 경로 탐색 등으로 인한 패킷의 충돌 확률이 높아지기 때문이다. 특히 트래픽 소스

가 가장 많은 60개의 경우 데이터 흐름이 많기 때문에 충돌 등의 영향이 다른 트래픽 소스의 개수에 비해 현저히 감소하는 것을 볼 수 있다. 전체적으로 ZiCL은 각 트래픽 소스의 개수에 따라 경로 탐색으로 인한 영향이 적기 때문에 지그비 AODV에 비해 더 나은 패킷 전달율을 보인다.

#### 4.3 중단간 지연

중단간 지연은 소스 노드에서 데이터 패킷이 발생된 시점부터 다중 홉을 거쳐 목적지 노드에 데이터 패킷이 도착한 시점까지의 평균 지연 시간이다. 그림 9에서 보는 바와 같이 중단간 지연은 노드의 개수가 증가함에 따라 전체적으로 증가되는 양상을 보인다. 이는 노드의 개수가 커짐에 따라 거치는 홉의 개수가 증가하기 때문이다. 노드의 개수가 증가함에도 불구하고 트래픽 소스 40개인 경우가 20개인 경우보다 더 짧은 중단간 지연을 보이는 것은 네트워크 내에 배치된 토폴로지에서 차이를 보이기 때문이다. 트래픽 소스 별로 보면 ZiCL의 경우 지그비 AODV에 비해 큰 차이를 보이지 않으나 보다 나은 성능을 보인다. 이는 ZiCL이 클러스터 구조를 사용함에도 불구하고 지그비 AODV와 동일한 최적

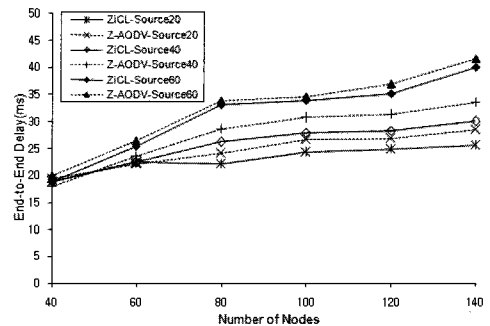


그림 9. 중단간 지연

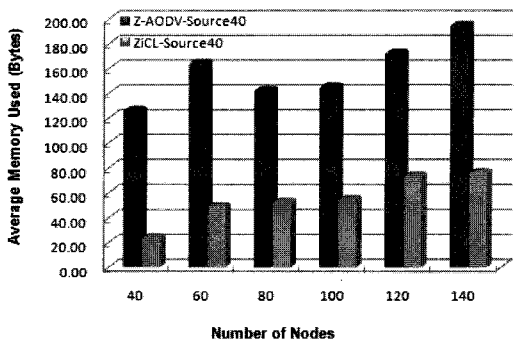


그림 10. 메모리 사용률

경로 기반의 라우팅을 수행하면서 상대적으로 줄어드는 경로 탐색 횟수에서 경로 확립에 따른 패킷의 큐잉시간이 차이가 있기 때문이다.

#### 4.4 메모리 사용률

본 논문에서는 노드 수의 증가에 따른 메모리 사용률을 알아보기 위하여 트래픽 소스가 40개인 경우에 대하여 각 시나리오에서 평균 라우팅 테이블 메모리 사용량을 측정하였다. ZiCL의 노드들은 한 개의 라우팅 엔트리에 대하여 목적지 노드의 클러스터 라벨(2바이트), 다음 홉 주소(2바이트), 상태값(3비트)을 설정하므로 이를 테이블 엔트리 수와 곱하여 각 시나리오에 존재하는 라우터의 개수만큼 나눈 평균값을 측정하였다. 지그비 AODV의 경우 표준에 따라 최종목적지 주소(2바이트), 다음 홉 주소(2바이트), 상태값(3비트)의 동일한 크기의 라우터 테이블을 갖는다. 시뮬레이션 결과 지그비 AODV는 그림 10과 같이 메모리 사용률은 ZiCL에 비해 3배 이상의 메모리를 더 요구하는 것을 볼 수 있다. 반면의 ZiCL의 경우 트래픽에 관여하지 않는 노드들이 라우팅 엔트리를 생성함에도 불구하고 클러스터 라벨만을 라우팅 테이블의 데이터로 사용하므로 메모리 사용률에 있어 지그비 AODV보다 적은 메모리를 요구한다.

### V. 결론

지그비 표준에서 정한 메쉬 라우팅은 AODV 기반의 라우팅 프로토콜을 지그비의 계층적 특성에 맞추어 오버헤드를 줄인 형태로써 지그비의 저전력 환경에 사용될 수 있도록 정의되어 있다. 하지만 지그비 표준 메쉬 라우팅은 시뮬레이션 결과를 통해 알 수 있듯이 네트워크의 사이즈가 커지고 노드의 개수가 증가할수록 노드간의 연결 링크가 증가함에 따라 패킷 손실률이 심하고 라우팅 오버헤드가 커지는 단점을 갖고 있었다. 이것은 경로 확립을 위한 라우팅 패킷들의 간섭 현상이 다수의 연결 링크에서 발생하는 데이터 전송에 영향을 미치면서 라우팅 오버헤드 및 종단간 지연과 같은 네트워크 성능을 저하시키기 때문이다. 따라서 본 연구에서는 기존의 메쉬 라우팅의 범용성(Scalability) 문제를 해결하기 위한 ZiCL을 제안하였다. ZiCL은 지그비 네트워크를 한 개 이상의 클러스터로 나누고 클러스터의 유일한 식별자인 클러스터 라벨을 라우팅 시 이용함으로써 다수의 연

결 링크가 형성되어도 클러스터 라벨을 통해 불필요한 경로 탐색을 줄이기 때문에 표준 메쉬 라우팅에 비해 오버헤드가 적은 향상된 성능을 보여주었다. 또한 ZiCL은 라우팅 엔트리 내에 노드의 네트워크 주소가 아닌 클러스터 라벨 단위로 저장함으로써 목적지 노드의 개수만큼 라우팅 엔트리를 설정해야 하는 지그비 AODV에 비해 보다 적은 메모리를 요구하였다.

### 참고 문헌

- [1] K. Akkaya and M. Younis, "A Survey on Routing Protocols for Wireless Sensor Networks", *In International Journal on Elsevier Ad Hoc Network*, Vol. 3/3 pp. 325-349, 2005.
- [2] S. Lee, E. M. Royer, and C. E. Perkins, "Scalability study of the ad hoc on-demand distance vector routing Protocol", *In International Journal on Network Management (IJNM)*, 2003
- [3] P. Kinney, "ZigBee Technology: Wireless Control that Simply Works", Communication Design Conference, October, 2003
- [4] ZigBee Alliance, "ZigBee Specification: ZigBee Document 053474r06 Version 1.0", 14 Dec. 2004.
- [5] C. E. Perkins, E. M. Royer and S. R. Das, "Ad hoc On-demand Distance Vector (AODV) Routing", Internet-Draft, IETF, March, 2002,
- [6] I. Chakeres, E. M. Royer, and C. E. Perkins "Dynamic MANET On-demand Routing Protocol", IETF Internet Draft, October 2005.
- [7] M. Jiang, J. Li, and Y. C. Tay, "Cluster Based Routing Protocol (CBRP)", Internet-Draft, IETF, Jul. 1999, Work in progress.
- [8] K. K. Lee, S. H. Kim, Y. S. Choi, and H. S. Park, "A Mesh Routing Protocol using Cluster Label in the ZigBee Network" - IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS06), Canada, Oct. 2006
- [9] IEEE Computer Society, "Part 15.4: Wireless Medium Access Control and Physical Layer Specifications for Low-Rate Wireless Personal Area Networks(LR-WPANs)", 1 Oct. 2003.
- [10] The Network Simulator - ns-2, <http://www.isi.edu/nsnam/ns/>



이 광 국 (Kwang-Koog Lee)

정회원



2006년 2월 강원대학교 컴퓨터  
공학과 졸업  
2006년 3월~현재 강원대학교 전  
자통신공학과 석사과정  
<관심분야> 무선통신, 유비쿼터  
스 컴퓨팅, 분산처리

박 흥 성 (Hong-seong Park)

종신회원



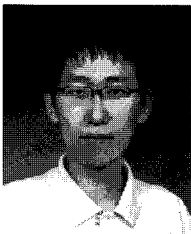
1983년 2월 서울대학교 제어계측  
공학과 졸업  
1986년 2월 서울대학교 제어계측  
전공 석사  
1992년 2월 서울대학교 제어계측  
전공 박사  
1992년~현재 강원대학교 전기전

자정보통신공 학부 교수

<관심분야> 실시간 통신, 무선데이터통신, 미들웨어

김 성 훈 (Seong-hoon Kim)

준회원



2006년 2월 강원대학교 전기전자  
정보통신공학부 졸업  
2006년 3월~현재 강원대학교 전  
자통신공학 전공 석사과정  
<관심분야> 무선 통신, 미들웨어