

# 무선 센서 네트워크를 위한 퍼지 라우팅 프로토콜

이 병 권\* · 전 중 남\*\*

## 요 약

무선 센서 네트워크에서 네트워크 수명을 오래 유지하기 위하여 라우팅 경로를 네트워크 전체에 분산시키는 방법이 연구되고 있다. 본 논문은 퍼지 제어 규칙을 기반으로 라우팅 경로를 선택하는 퍼지 라우팅 프로토콜을 제안한다. 여러 가지 라우팅 경로에 대하여 에너지, 거리, 홉 수를 입력으로 하여 퍼지 추론 엔진으로 적합도를 산출하고, 적합도가 높은 한 개의 경로를 최종 라우팅 경로로 선정한다. 한 번 데이터 패킷을 전달한 경로에 속한 노드들의 에너지를 감소시킴으로써 다음에 경로를 선정할 때 동일한 경로를 선택하지 않도록 설계하였으며, 결과적으로 라우팅 경로가 분산되어 센서 네트워크의 유지시간을 길게 만드는 효과가 있다.

TinyOS 시뮬레이터에서 실험한 결과, 제안하는 라우팅 프로토콜이 기존의 방법보다 에너지 효율 면에서 우수함을 입증하였고, 또한 센서 네트워크의 유지시간이 길어짐을 확인하였다.

키워드 : 라우팅 프로토콜, 퍼지 제어, 퍼지 라우팅, 무선 센서 네트워크

## A Fuzzy Routing Protocol for Wireless Sensor Network

ByongKwon Lee<sup>\*</sup> · Joongnam Jeon<sup>\*\*</sup>

### ABSTRACT

Distributing the routing path over the entire network is an important factor to maintain the lifetime of wireless sensor network as long as possible. This paper proposes a fuzzy routing protocol that decides a routing path based on the fuzzy control rules. The fuzzy controller receives the energy values, distances, and hop counts of possible route paths as input, and the inference engine produces the contribution factors for each of route paths. The route path with the largest contribution factor is elected as the final routing path. The nodes contained in the routing path reduce their energy after transmitting a data packet so as to prevent the same route path from being selected repeatedly. It makes the network traffic spreaded over the network resulting longer network lifetime.

The computer simulations on TinyOS have shown that the fuzzy routing protocol is more energy efficient and has longer network lifetime compared to the existing routing protocols.

Key Words : Routing Protocol, Fuzzy Control, Fuzzy Routing, Wireless Sensor Network

### 1. 서 론

무선 센서 네트워크(WSN, Wireless Sensor Network) 설계 이슈 중 하나는 제한된 자원과 열악한 환경에서 '얼마나 오래 동안 네트워크 유지시간(life time)을 길게 유지할 수 있느냐'이다[1]. 센서 네트워크 유지시간에 중요한 요인 중 하나가 라우팅 프로토콜이다.

많은 무선 센서 네트워크의 라우팅 프로토콜이 개발되고 제안되고 있다. 기존에 발표된 DSR(Dynamic Source Routing)[4], GBR(Gradient Based Routing)[6], 그리고 AODV(Ad Hoc on Demand Distance Vector Routing)[7] 등

의 센서 네트워크 라우팅 프로토콜들은 서로 경로 선택 방법은 서로 다르지만, 한 번 네트워크 상태가 결정된 이후에는 동일한 조건으로 라우팅 경로를 결정한다. 따라서, 센서 네트워크의 환경 변화에 적응하기 어려운 단점이 있다. 그리고 센서 노드로 구성된 네트워크는 일부 노드의 에너지가 고갈되면 전체 네트워크 구성이 끊겨 패킷 송·수신이 불가능할 가능성이 높다.

따라서, 데이터 전송 후 네트워크 환경의 변화를 반영하여 다른 조건에서 라우팅 경로를 결정함으로써 센서 노드의 에너지를 균등하게 사용하도록 라우팅 경로를 분산하는 라우팅 프로토콜이 필요하다[2].

본 논문에서는 이러한 문제점을 해결하기 위하여 퍼지 제어 이론[8]을 센서 네트워크 라우팅에 결합한 퍼지 라우팅 기법을 제안한다. 퍼지 라우팅 프로토콜은 데이터 패킷 전달이 가능한 모든 경로에 대하여 노드의 잔여 에너지, 거리,

\* 이 논문은 2006학년도 충북대학교 학술연구지원사업의 연구비지원에 의하여 연구되었음.

† 정 회 원 : 휴먼앤컴퓨터정보통신(주) 선임연구원

\*\* 종 신 회 원 : 충북대학교 전기전자컴퓨터공학부 교수(교신기자)

논문접수: 2007년 5월 29일, 심사완료: 2007년 11월 20일

홉 수를 퍼지 입력 변수를 조건부 규칙과 퍼지 연산하고, 연산 결과를 결론부 퍼지 집합과 합성 연산하고, 비퍼지화 과정을 거쳐 적합도를 산출된다. 이 과정 중 많은 연산이 필요한 비퍼지화 과정을 간소화 하는 방법도 제안한다. 최종적으로 만들어진 적합도는 데이터 패킷을 전달할 최종 경로를 선택하는 척도로 사용된다. 또한, 한 번 선택된 경로에 대하여 잔여 에너지를 감소시킴으로써, 다음 경로 선정 시 동일한 경로가 선택될 가능성이 줄어들어 경로가 분산되는 효과를 기대할 수 있다.

논문의 구성으로 2장에서는 관련 연구로 기존의 무선 센서 라우팅 프로토콜들을 조사 분석하고, 3장에서는 퍼지 라우팅 프로토콜을 제안하고, 4장에서는 제안된 프로토콜과 기존의 라우팅 프로토콜과의 성능을 비교 분석한 결과를 제시한다. 마지막 5장에서 결론을 맺는다.

**2. 무선 센서 네트워크 라우팅**

무선 센서 네트워크 라우팅은 전파 방식에 따라 노드의 위치 기반의 'Flat Networks Routing', 가상 계층을 형성하는 'Hierarchical Networks Routing', 그리고 노드가 자신의 위치를 알고 있을 경우에 위치 정보를 라우팅에 활용하는 'Location Based Routing'으로 구분할 수 있다[3]. 본 연구에서 제안하고자 하는 퍼지 라우팅 프로토콜은 'Flat Networks Routing' 프로토콜에 해당하며 여기에서는 이 방식의 무선 센서 네트워크 라우팅 프로토콜들의 특성을 분석하고 비교하여 제시한다.

Flooding[4] 프로토콜은 모든 노드들이 자신이 수신한 패킷들 중에서 목적지가 자신이 아닌 패킷들을 다시 방송(broadcast)하여 인접 노드로 패킷을 전파하는 방법이다. 대부분의 라우팅 프로토콜에서 비콘 신호를 모든 센서 노드들로 전달하고자 할 때 사용한다.

DSR[5] 프로토콜은 패킷 안에 경유한 노드의 ID를 포함하여 비콘 패킷을 전달하고 경로의 역순으로 데이터 패킷을 전달하는 방법이다. 이 방법은 패킷 경로에 포함되는 노드의 수에 비례하여 패킷의 크기가 증가한다는 단점을 가지고 있어 노드 수가 많은 네트워크의 경우 불리하다.

GBR[6] 프로토콜은 비콘 패킷이 노드를 경유할 때마다 싱크에서 목적지 노드까지의 홉 수를 저장하여 데이터 전달

〈표 2.1〉 라우팅 프로토콜

라우팅 프로토콜	경로선택요소	데이터 수집	테이블 작성
Flooding	모든 경로	분산 경로	X
DSR	패킷정보	고정 경로	X
GBR	홉	고정 경로	X
AODV	랜덤	분산 경로	O

을 위한 라우팅 경로를 선정 시 홉 수가 낮은 경로를 선택하면 싱크로 되돌아온다는 개념을 사용한 방법이다. 하지만 데이터 패킷은 홉 수가 낮은 단일 경로를 통하여 전달되기 때문에 센서 노드 전체 중 일부 노드에서 국부적인 에너지 손실이 발생할 수 있다.

AODV[7] 프로토콜은 방향성을 갖는 벡터 형태의 라우팅 테이블을 작성하면서 동작한다. 비콘 패킷이 전달될 때 패킷을 수신한 노드의 ID를 저장하고, 데이터 패킷은 이 인접 노드를 거쳐 싱크로 전달된다. 인접 노드가 한 개 이상일 때 처리하는 방법으로 대상 노드들 중 하나를 선정하는 방법과 인접 노드들로 모두 전송하는 방법이 있을 수 있다.

〈표 2.1〉은 위에서 서술한 네 가지의 무선 센서 라우팅 프로토콜들의 특징을 비교한 것이다.

**3. 퍼지 라우팅**

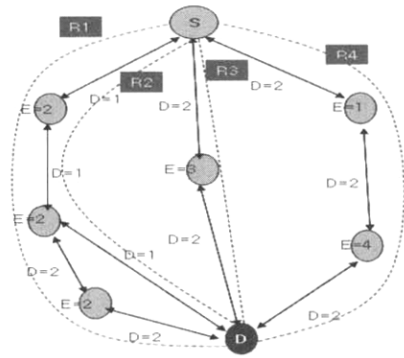
퍼지 라우팅은 퍼지 제어 이론을 이용하여 적합도가 가장 높은 갖는 경로를 찾는 방법이다. 이 절에서는 퍼지 개념을 사용한 효율적 라우팅 기법에 대하여 기술하고, 퍼지 추론에 필요한 퍼지화기, 퍼지 규칙베이스 생성과 퍼지 추론 엔진의 설계 과정을 설명한다. 또한 퍼지 연산 시 소요되는 시간 줄이는 기법에 대하여 제시하고, 마지막으로 퍼지 라우팅 프로토콜에 대하여 기술한다.

**3.1 효율적 라우팅**

무선 센서 네트워크의 라우팅 경로를 결정하는 요소는 잔여 에너지의 양, 노드 간의 거리, 홉 수, 전파의 세기, 노드의 위치 등 여러 가지 요소가 있다. 본 논문에서는 라우팅

〈표 3.1〉 라우팅 경로 조건

수집된 값				퍼지화			우선순위
경로	에너지	거리	홉수	에너지	거리	홉수	
R1	6	6	4	많다	많다	보통	3
R2	4	3	3	보통	조금 적다	조금 적다	1
R3	3	4	2	조금 적다	보통	적다	2
R4	5	6	3	조금 많다	많다	조금 적다	3



(그림 3.1) 라우팅 경로

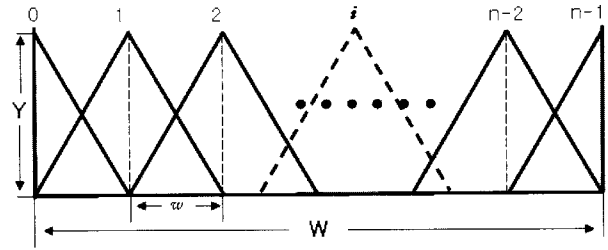
경로를 결정하는 조건으로 에너지(E), 거리(D), 홉 수(H)의 세 가지 요소들을 고려한다.

(그림 3.1)에서 S가 싱크 노드이고 D가 목적지 노드일 때, 에너지, 거리, 홉 수에 대하여 만들어질 수 있는 경로는 R1(E=6, D=6, H=4), R2(E=4, D=3, H=3), R3(E=3, D=2, H=2), R4(E=5, D=6, H=3)의 네 가지이다. 최대 에너지만을 고려하는 경우 선택된 라우팅 경로는 에너지의 합이 큰 R1이다. S와 D 간의 최소 거리로 라우팅 경로를 선택하면 거리가 가장 짧은 R2가 선정된다. 그리고, 최소 홉 수를 기준으로 선택된 라우팅 경로는 R3이다. 이와 같이 라우팅 경로 선택 조건에 따라 선정된 경로가 다르므로, 경로 선택 조건들을 서로 보완할 수 있는 의미적인 해석이 필요하다.

<표 3.1>은 (그림 3.1)의 라우팅 경로들에 대하여 라우팅 조건들을 의미적으로 해석하여 우선순위를 작성한 것이다. 작성 기준은 에너지, 거리, 홉 수의 합에 대하여 작성하였고, 퍼지화에 대한 언어집합의 기준으로 “1: 매우적다, 2: 적다, 3:조금적다, 4: 보통, 5: 조금 많다, 6: 많다, 7:매우 많다”로 표현하였다. 조건으로 에너지는 크면 좋고, 거리와 홉 수는 낮을수록 좋다. R1과 R4의 경우 우선순위는 3이다. 이 경우 에너지 면에서 R4보다 R1이 1만큼 우수하지만 홉 수면에서 R4가 1만큼 우수해 동등한 우선순위를 갖는다. R2와 R3을 비교할 때 에너지, 거리 면에서 R2가 R3보다 우수하다. 최종적으로 에너지, 거리, 홉 수를 고려해 볼 때 R2의 경로가 우수하다고 판단된다. 본 논문에서 제안하는 퍼지 라우팅 프로토콜은 의미 해석이 가능한 퍼지 연산에 의하여 라우팅 경로를 찾는다.

### 3.2 퍼지화기

퍼지화기는 수집된 에너지, 거리, 홉 수에 대하여 퍼지 입력에 대하여 퍼지 입력으로 변환하는 역할을 한다. 라우팅 과정에서 수집된 에너지, 거리 홉 수는 신뢰성이 있는 값이라고 가정하여, 본 연구에서는 단일 입력 함수를 사용하였다.



(그림 3.2) 삼각함수 퍼지 집합

### 3.3 퍼지 집합 표현

이 절에서는 조건부와 결론부의 퍼지 집합에 대한 표현 방법을 설명한다. 일반적으로 조건부와 결론부 퍼지 집합을 (그림 3.2)와 같은 삼각함수로 표현한다. 퍼지 집합의 수를  $n$ , 퍼지 소속 함수 제어 값의 범위를  $Y$ , 입력 값의 폭을  $W$ , 입력 또는 출력 변수를  $x$  ( $0 \leq x < W$ ), 소속 함수  $f(x)$  ( $0 \leq f(x) < Y$ )로 정의한다. (그림 3.2)의  $w$ 는  $(n-1)$  개의 구간 중 하나의 폭을 의미하며 (식 3.1)과 같이 표현될 수 있다.

$$w = \frac{W}{n-1} \quad (0 \leq i < n-1) \quad (\text{식 3.1})$$

(그림 3.2)에서  $i(0 \leq i < n-1)$ 번째 퍼지 집합에 대한 소속 함수  $f_i(x)$ 는 양의 기울기를 갖는 일차 방정식과 음의 기울기를 갖는 일차 방정식의 조합으로 표현되고 (식 3.2)와 같다.

$$f_i(x) = \begin{cases} \frac{Y}{w}x - (i-1)Y & \text{if } x = ((i-1)w \leq x < iw), \\ -\frac{Y}{w}x + (i+1)Y & \text{elseif } x = (iw \leq x < (i+1)w), \\ 0 & \text{otherwise.} \end{cases} \quad (\text{식 3.2})$$

### 3.4 퍼지 규칙베이스 생성

퍼지 규칙베이스는 입력 변수들에 대한 조건부 퍼지 집합을 결론부 퍼지 집합으로 연결하는 규칙들의 집합이다. 에너지에 대한 규칙수를  $R_E$ , 거리에 대한 규칙수를  $R_D$ , 홉 수에 대한 규칙 수를  $R_H$ , 그리고 결론부 규칙수를  $R_O$ 로 가정한다. 퍼지화기를 통하여 입력되는 값 중 에너지의 값  $e$ 에 대한 입력 범위를  $W_e$ , 거리  $d$ 와 홉 수  $h$ 에 대한 입력 범위를 각각  $W_d, W_h$ 로 가정한다. 이때 입력된 값  $e, d, h$ 가 속하는 규칙 번호는 (식 3.3)과 같다. 여기서  $[x]$ 는  $x$ 를 넘지 않는 정수이고,  $i = (0 \leq i < R_E), j = (0 \leq j < R_D), k = (0 \leq k < R_H)$ 이다.

$$\begin{aligned}
 i &= \left\lfloor \frac{(R_E - 1)e}{W_e} \right\rfloor, \\
 j &= \left\lfloor \frac{(R_D - 1)d}{W_d} \right\rfloor, \\
 k &= \left\lfloor \frac{(R_H - 1)h}{W_h} \right\rfloor.
 \end{aligned}
 \tag{식 3.3}$$

본 논문에서는 이 세 가지의 입력 변수를 3차원 공간으로 해석하여 원점에서 해당 규칙 번호  $(i, j, k)$ 까지의 거리를 결론부 규칙 수로 나누어 결론부 규칙 번호를 결정하였다. 이와 같이 구한 결론부 규칙번호  $l$ 은 (식 3.4)와 같다.

$$l = \left\lfloor \frac{\sqrt{i^2 + j^2 + k^2}}{\sqrt{R_E^2 + R_D^2 + R_H^2}} \cdot R_O \right\rfloor, \text{ 여기서 } 0 \leq l < R_O, \tag{식 3.4}$$

예를 들어,  $R_E = R_D = R_H = 4$ ,  $W_e = W_d = W_h = 0 \sim 60$ ,  $R_O = 11$ ,  $e = 10$ ,  $d = 40$ ,  $h = 24$ 일 때  $i, j, k$ 는 각  $[0.5] = 0, [1.2] = 1, [2.0] = 2$ 이고,  $l = [4.73] = 4$ 이다. 그러므로 결론부 규칙 4번째와 연결된다.

이 과정은 실수 연산을 포함하고 있어, 센서 노드에서 수행하는 경우에 부담이 될 수 있다. 이 문제를 해결하는 방법은 다음과 같이 두 가지 방법이 있을 수 있다. 첫째, 매핑 테이블은 초기화 과정에서 한 번만 수행하므로, 매핑 테이블을 미리 구하여 고정 배열(static array)로 구현하는 방법이다. 두 번째는 3차원 공간 거리  $\sqrt{i^2 + j^2 + k^2}$  대신에 직선 사각 거리  $(i + j + k)$ 를 사용하여 계산을 간소화 하는 것이다.

### 3.5 퍼지 추론 및 제어

퍼지 추론은 퍼지화기를 통하여 입력된 에너지, 거리, 흡수에 대하여 퍼지 연산을 수행하여 적합도를 산출하고, 합성 연산을 통하여 한 개의 퍼지 제어 값을 생성한다. 라우팅을

위하여 퍼지 제어 이론을 적용하는 경우에 여러 라우팅 경로에 대한 적합도를 산출한 후 값이 가장 큰 적합도를 갖는 경로를 선택하는 것이 목적이기 때문에, 비퍼지화 작업을 할 필요가 없다. 3.5절에서는 일반적인 퍼지 제어 이론에 따라 한 개의 퍼지 제어 값을 생성하여 라우팅 경로를 결정하는 과정에 대하여 설명하고, 3.6절에서 간소화된 퍼지 라우팅 알고리즘을 제시한다.

(그림 3.3)은 입력된 퍼지 변수  $e, d, h$ 에 대하여 조건부 퍼지 집합과의 MAX 연산 알고리즘이다. 라인 (2)에서 에너지는 흡수, 거리와는 역관계를 가지고 있어 입력 변수의 폭인 IWIDITH에서 에너지를 역으로 취해서 연산하였다. 라우팅 경로 선택시 에너지가 높고, 거리와 흡수가 낮으면 경로 선택에 있어 우선순위가 높고, 반대로 에너지가 낮고, 거리와 흡수가 크면 우선순위가 낮다. 라인 (6)의 함수 fuzzyCalc는 입력된  $e, d, h$ 에 대하여 퍼지 집합과 Max 연산을 수행하여 세 가지 변수에 대한 적합도를 FE, FD, FH에 저장한다. 라인 (8)~(15)은 (식 3.2)의 퍼지 함수 값을 계산하는 부분이다. 라인(9)의 "region"은 퍼지 삼각 함수를 연산하기 위한 부분으로 입력 퍼지 변수  $x$ 에 대하여  $i$  번째 삼각 함수의 양쪽 두 개의 기울기 중에 어느 변에 해당하는지 판단하는 구분으로 -1인 경우 기울기가 양수인 일차방정식에 해당되고 0인 경우는 기울기가 음수인 일차방정식에 해당함을 나타낸다. 이와 같은 방법으로 입력된 모든 퍼지 제어 변수( $e, d, h$ )에 대하여 적합도를 계산한다.

(그림 3.4)는 MAX 값 저장 버퍼인 FE, FD, FH에 대하여 MIN 값을 선정하고 비퍼지화를 수행하는 과정이다. 라인(7)에서 함수 MIN을 이용하여 최소값을 구한다. 라인(10)에서 integrate 함수를 이용하여 min 값에 해당하는 적합도를 산출한다. 합성 연산을 통하여 만들어진 FZ는 비퍼지화 과정에서 무게 중심을 구하는데 이용된다. 최종적으로 라인 (19)에서 무게 중심인 제어 점 G를 찾게 된다.

위 과정은 한 개의 경로에 대한 퍼지 제어 값을 산출하는

```

void fuzzy_edh(int e, int d, int h) {
    fuzzyCalc(IWIDITH - e, FE);
    fuzzyCalc(d, FD);
    fuzzyCalc(h, FH);
}
void fuzzyCalc(int x, int *f) {
    inti, region
    for (i=0; i<NO_IRULES; i++) {
        region = x/IW - i;
        switch (region) {
            case -1:f[i] = IHEIGHT*x/IW - (i-1)*IHEIGHT; break;
            case 0:f[i] = -IHEIGHT*x/IW + (i+1)*IHEIGHT; break;
            default: f[i] = 0; break;
        }
    }
}
    
```

(그림 3.3) 퍼지 추론 MAX 연산

```

void fuzzy_min() { (1)
    intWX=0, W=0, G=0; (2)
    for (x=0; x<OWIDTH; x++) FZ[x] = 0; (3)
    for (x=0; x<NO_IRULES; x++) (4)
        for (y=0; y<NO_IRULES; y++) (5)
            for (z=0; z<NO_IRULES; z++) { (6)
                min=MIN(FE[x], FD[y], FH[z]); (7)
                if (min > 0) { (8)
                    c = IRULE_TABLE[x][y][z]; (9)
                    integrate(c, min); (10)
                } (11)
            } (12)
    for (x=0; x< OWIDTH; x++) { (13)
        if (FZ[x] != 0) { (14)
            WX += FZ[x] * x; (15)
            W += FZ[x]; (16)
        } (17)
    } (18)
    G = WX / W; (19)
} (20)

```

(그림 3.4) 퍼지추론 MIN 연산

```

MAX = 0; (1)
for (x=0; x<NO_IRULES; x++) (2)
    for (y=0; y<NO_IRULES; y++) (3)
        for (z=0; z<NO_IRULES; z++) { (4)
            min=MIN(FE[x], FD[y], FH[z]); (5)
            if (min > 0) { (6)
                c = IRULE_TABLE[x][y][z]; (7)
                if (min > MAX) { (8)
                    MAX = min; (9)
                    Rank = c; (10)
                } (11)
            } (12)
        } (13)
}

```

(그림 3.5) 퍼지 추론 랭크 방법

알고리즘이다. 퍼지 라우팅 알고리즘은 각 경로에 대하여 제어 값  $G$ 를 구하고,  $G$  값이 가장 큰 경로를 최종 라우팅 경로로 선택한다.

### 3.6 퍼지 엔진 간소화

센서 네트워크의 라우팅 연산은 특정 장비나 기기에 대하여 제어 값을 구하는 문제가 아니기 때문에 비퍼지화 과정이 필요하지 않다. 단지 만들어진 라우팅 정보로 경로의 우선순위를 구하면 충분하다. 또한 무선 센서 네트워크의 경우 제한된 자원을 가지고 있어 연산 속도가 느린 소형의 센서 노드에서 복잡한 퍼지 연산을 그대로 사용하면 연산에 따른 부하가 더 많이 발생할 수 있다.

본 연구에서는 두 단계로 간소화된 퍼지 라우팅 알고리즘

을 제안한다. 첫 번째 단계는 비퍼지화 과정을 수행하지 않고 랭크(rank)를 구하여 직접 경로를 결정하는 과정이고, 두 번째 단계는 이 과정에 포함되어 있는 다중 루프 연산을 제거하는 방법이다.

원래는 (그림 3.4)와 같이 임의의 노드에서 싱크 노드로 도달할 수 있는 각 경로에 대하여 적합도에 대하여 제어점에 해당하는 무게 중심을 구하고, 그 중에서 가장 큰 값을 갖는 경로를 선택해야 한다. 즉, 무게 중심이 최종 구하려는 해가 아니기 때문에 최대 적합도를 갖는 출력 규칙 번호를 구하는 것으로 간소화할 수 있다. 따라서, (그림 3.5)과 같이 라인 (5)에서 적합도를 구하고, 라인 (10)에서 최대 적합도에 해당하는 출력 규칙 번호를 입력 조합에 대한 랭크로 간주한다. 최종적으로 랭크 값을 비교하여 가장 랭크가 큰

```

X = Y = Z = 0; (1)
for (x=1; x<NO_IRULES; x++) { (2)
    if (FE[X] < FE[x]) X = x; (3)
    if (FD[Y] < FD[x]) Y = x; (4)
    if (FH[Z] < FH[x]) Z = x; (5)
} (6)
Rank = IRULE_TABLE[X][Y][Z]; (7)
    
```

(그림 3.6) 단일 루프 연산 알고리즘

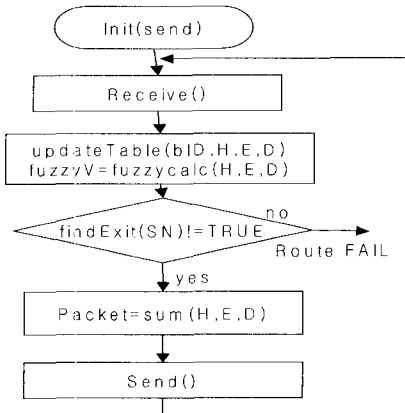
경로를 선택한다. 이와 같은 과정을 거쳐 비퍼지화 과정을 제거할 수 있다.

두 번째 단계는 (그림 3.5)의 3중 루프를 제거하는 과정이다. (그림 3.5)의 3중 루프는 에너지, 홉 수, 거리 공간에서 각 축에 대하여 최대값에 해당하는 좌표에 대한 출력 규칙을 구하는 과정을 포함하고 있다. 이 과정은 각 축에 대하여 최대값을 구한 후, 각 축에서의 최대값에 대하여 출력 규칙을 구하는 것과 동일하다. 따라서, (그림 3.5)은 (그림 3.6)와 같이 한 개의 루프로 대체할 수 있으며, 계산량 면에서는 복잡도  $O[n^3]$ 이 복잡도  $O[n]$ 으로 감소한다. 여기서  $n$ 은 입력 규칙 수이다.

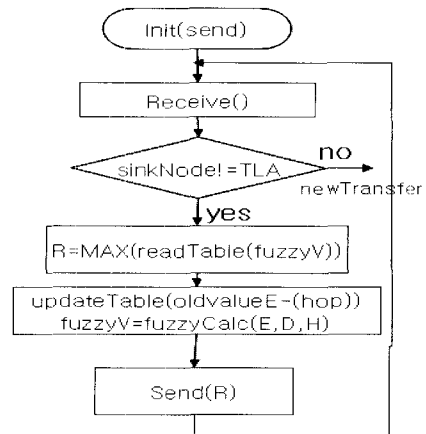
3.7 퍼지 라우팅 프로토콜

제안하는 퍼지 라우팅 프로토콜은 비콘을 방송하여 경로를 설정하는 전방향 전달 과정과 목적지로부터 데이터를 전송하는 데이터 수집 과정으로 구성된다.

전방향 전달 과정의 흐름도를 (그림 3.7)에 제시하였다. 전방향 전달 과정에서 싱크를 포함한 노드들은 자신의 위치 좌표, ID, 목적지 정보를 포함하는 비콘 패킷을 flooding 방식으로 센서 네트워크에 전파한다. 비콘 패킷을 수신한 노드는 인접 노드를 거쳐 싱크에 도달하는 경로에 정보인 인접 노드의 ID와 에너지, 거리, 홉 수의 합들을 저장하고(updateTable()), 그 경로에 대하여 (그림 3.6)로 구한 랭크 값을 계산하여 저장한다(fuzzyV = fuzzyCalc()). 목적지가 자기 자신이 아닌 경우, 각 노드는 자신의 정보를 추가한 패킷을 만들어 인접 노드로



(그림 3.7) 전방향 전달 과정



(그림 3.8) 데이터 수집 과정

전달한다. 이와 같은 과정을 통하여 각 센서 노드는 자신으로부터 싱크까지 도달할 수 있는 경로들에 대한 인접 노드의 ID와 에너지, 거리, 홉 수의 합들, 그리고 그 경로에 대한 랭크 값을 라우팅 정보로 저장한다.

목적지에서 원하는 데이터를 수집해 싱크 노드로 전달하는 과정은 (그림 3.8)의 흐름도를 따른다. 데이터 패킷을 수신한 노드는 싱크가 아니면 퍼지 랭크 값을 근거로 라우팅을 수행한다. 먼저, 가능한 경로들에 대하여 최대 랭크 값을 갖는 경로 R을 선정한다( $R=Max(readTable())$ ). 그리고, 선정된 경로 R에 대하여 경로 R을 거쳐 싱크 노드까지 도달하는 홉 수에 비례하여 경로 R에 대한 에너지를 감소시킨다(updateTable()). 마지막으로 다음 데이터 패킷의 라우팅을 위하여 선정된 경로에 대한 랭크 값을 다시 계산한( $fuzzyV=fuzzyCalc()$ ) 후 데이터를 경로 R을 통하여 전송한다(Send()).

한 번의 데이터 패킷의 전달이 끝나고 다음 데이터 패킷을 전송할 때 새로 계산된 랭크 값을 참고하므로, 동일한 경로가 다시 선정될 가능성이 줄어든다. 따라서, 데이터 패킷을 전달할 때에는 분산 경로가 만들어진다.

4. 시뮬레이션 및 성능 분석

시뮬레이션 및 성능 분석은 센서 네트워크 운영체제인 TinyOS[10] 기반으로 동작하는 nesC 언어로 퍼지 라우팅

<표 4.1> 실험 조건

항 목	조 건
토폴로지	그리드(grid), 랜덤(random)
라우팅 결정 요소	에너지(E), 거리(D), 홉 수(H)
퍼지 집합의 수	조건부 10개, 결론부 10개
노드 간의 전송 범위	전파 감도 100%
실험방법	각 노드의 초기 에너지 양 100 전송시에너지 2감소, 수신시에너지 1감소
데이터 수집 방법	싱크노드를 고정시키고 모든 노드가 싱크로 데이터 전달
비교 대상 프로토콜	DSR, GBR, AODV

<표 4.2> 조건부 퍼지 집합 결정표

퍼지 집합 수	조건부		결론부	
	평균	표준편차	평균	표준편차
2	70.6	43.6	74.0	54.5
4	68.8	42.1	74.2	55.4
6	67.4	44.8	74.1	49.0
8	69.1	43.1	74.1	50.4
10	69.1	40.1	70.0	47.7
12	70.4	43.7	70.6	48.5
14	70.4	44.5	69.3	47.9

<표 4.3> 프로토콜 성능평가 (n: 이웃 노드의 수 m: 퍼지 집합 수)

프로토콜 항목	DSR	GBR	AODV	Fuzzy라우팅
라우팅 정보	• 경유한 노드의 ID	• 이웃한 노드의 ID • 홉 수	• 이웃한 노드 ID	• 이웃한 노드의 ID • 에너지의 합 • 홉 수의 합 • 거리의 합 • 적합도
사용 메모리	라우팅정보 × 경로 수	라우팅정보 × 이웃노드의 수	라우팅정보 × 이웃노드의 수	라우팅정보 × 이웃노드의 수
복잡도	O(1)	O(n)	O(1)	O(n)+O(m)

프로토콜을 구현하고 TOSSIM 시뮬레이터와 TinyViz를 이용하여 실험하였다. 성능비교 척도로 라우팅 프로토콜들의 연산 복잡도, 센서 네트워크에 배포된 노드의 에너지 사용에 대한 에너지 히스토그램, 센서 노드의 소비 에너지 평균 및 표준 편차를 사용하였다.

실험 조건은 <표 4.1>과 같이 센서 네트워크의 토폴로지를 센서 노드를 그리드와 랜덤으로 배치하여 실험하였다. 퍼지 라우팅의 경우 라우팅 정보는 최대 인접 노드들의 수 만큼 필요하므로, 그리드 배치인 경우에는 4개, 실험 결과 랜덤 배치인 경우 인접 노드의 수는 2~7개임을 확인하고 10개의 원소를 라우팅 테이블에 할당하였다.

데이터 패킷은 TinyOS가 제공하는 TOS\_Msg 구조를 그

대로 사용하였고, 한 개의 패킷의 크기는 패킷 헤더 7바이트와 데이터 영역 29바이트를 합친 36바이트이다. 데이터 영역에는 자신의 주소, 센싱 시간, 센싱 값이 들어간다. 한 개의 데이터 패킷을 송수신하는 데 필요한 에너지는 전송시 2, 수신시 1로 가정하였다.

제안된 프로토콜을 실험하기 전에 라우팅을 위한 조건부와 결론부 퍼지 집합 수를 결정해야 한다. <표 4.2>는 여러 가지 퍼지 집합 수에 대하여 퍼지 라우팅 프로토콜 동작 시 노드가 소비한 에너지의 평균과 표준편차를 나타낸 것이다. 퍼지 집합 수를 변경하면서 실험한 결과 소비한 에너지의 평균이 낮고 평균에 대한 표준편차가 적은 조건부 퍼지 집합의 수는 10개의 퍼지 조건부 집합을 갖는 경우로 나타났다. 또한 조건부 규칙수를 10개로 고정하고 결론부 퍼지 집합 수를 변경하면

〈표 4.4〉 노드 소비 에너지의 평균 및 표준 편차

토폴로지 (노드 수)	항 목	Fuzzy Routing	AODV	DSR	GBR
Grid(49)	평 균	86.0	89.0	86.6	88.0
	표준편차	21.9	22.3	21.4	22.1
Random(49)	평 균	84.2	93.6	86.7	85.1
	표준편차	23.3	17.6	22.5	22.5
Grid(100)	평 균	59.3	61.1	59.8	62.5
	표준편차	25.4	25.5	27.8	26.1
Random(100)	평 균	56.3	71.7	59.3	63.8
	표준편차	26.3	23.2	25.7	29.4

서 실험한 결과는 10개의 결론부 퍼지 집합을 갖는 경우였다. 따라서 이 값으로 퍼지 라우팅 프로토콜을 구현하여 다른 라우팅 프로토콜들과 성능을 비교하였다.

퍼지 라우팅 프로토콜과 비교 대상 프로토콜들의 연산 복잡도 및 소요되는 메모리량은 <표 4.3>과 같다. DSR는 경유한 노드의 ID 정보만을 포함하고 있어 메모리 사용은 경로의 수만큼 사용한다. 하지만 연산의 복잡도는 경로가 패킷 안에 미리 정해져 있으므로 O(1)이다. GBR는 이웃한 노드의 ID와 홉 수의 정보를 가지고 홉 수가 낮은 한 개의 경로를 선정하여 동작하므로 연산의 복잡도는 O(n)이다. AODV 이웃한 노드의 ID 정보로 테이블을 생성하여 동작하므로 연산의 복잡도는 O(1)이다. 퍼지 라우팅 기법은 적합도가 가장 좋은 것을 선정하기 때문에 O(n)이고 에너지 변화에 따른 적합도 계산 시 'Rank' 값을 찾는 동작을 수행하므로 O(m)이 추가된다.

4.1 에너지 히스토그램

에너지 히스토그램을 작성하기 위해 일부 센서 노드의 에너지가 고갈되어 더 이상 패킷이 전송되지 않을 때까지 계속 패킷을 전송하여 소모된 에너지에 대한 노드 수를 에너지 히스토그램으로 작성하여 (그림 4.1)에 제시하였다. AODV(b)의 경우 에너지를 많이 사용한 노드들이 많게 나

타나 가장 좋지 않은 결과를 보였다. 퍼지 라우팅(a)의 경우 주로 사용된 에너지의 범위가 10~40% 범위에 존재하여 다른 라우팅 프로토콜(c)(d)과 비교하여 에너지를 적게 사용한 노드들이 많음을 알 수 있다.

4.2 평균 및 표준 편차

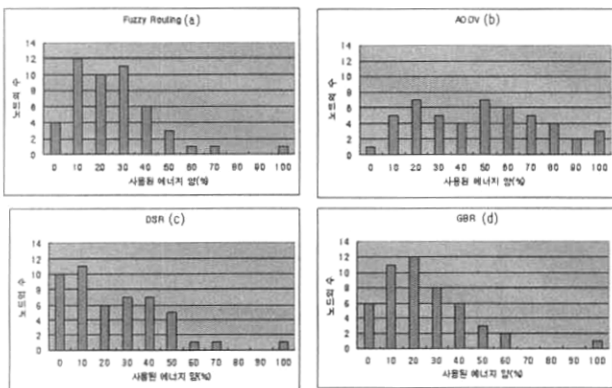
에너지 히스토그램을 통하여 퍼지 라우팅 기법이 전체 노드가 에너지를 다른 프로토콜보다 균등하게 사용함을 확인할 수 있다. 이러한 결과를 뒷받침할 수 있는 척도가 소비 에너지에 대한 평균 및 표준편차이다. <표 4.4>의 실험 결과에 의하면, 퍼지 라우팅 기법의 소비 에너지에 대한 평균과 표준편차 값이 다른 프로토콜들에 비하여 적은 값을 보여, 라우팅 경로가 분산되는 것을 확인할 수 있다.

4.3 네트워크 유지시간

센서 네트워크 유지시간을 측정하기 위해 100개의 노드를 랜덤하게 배포하고 네트워크가 구성되고 끊어진 시간의 변화에 따른 에너지의 평균 및 표준 편차를 측정하여 (그림 4.3)와 같은 결과를 얻었다.

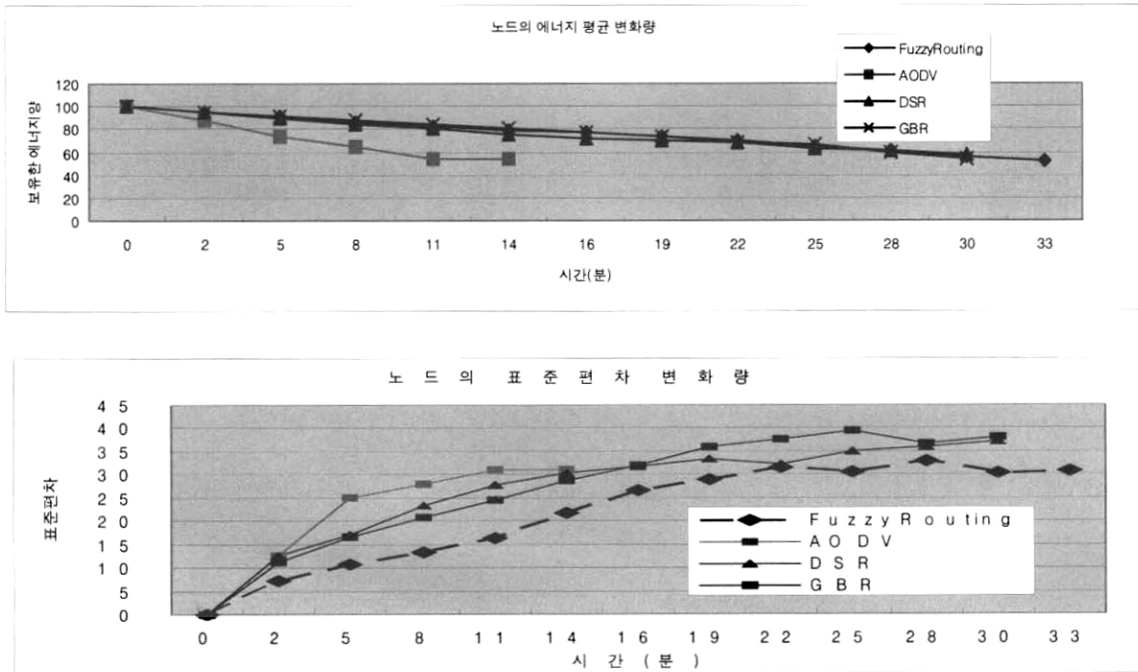
(그림 4.2)은 노드의 최초 에너지 보유량을 100으로 설정하고 에너지 평균 변화량을 시간 단위로 구분하여 작성된 결과로 제안된 퍼지 라우팅 프로토콜과 기존의 AODV, DSR, GBR 프로토콜과 비교한 것이다. 제안된 퍼지 라우팅 프로토콜과 DSR, GBR 프로토콜은 에너지 평균 변화량이 비슷하게 감소하지만 네트워크 유지시간은 제안된 프로토콜이 더 오래 지속됨을 확인할 수 있다. AODV 프로토콜의 경우 네트워크 유지시간이 14분 정도로 짧았고, 나머지 DSR과 GBR의 경우 30분 정도에서 네트워크가 끊겼다. 하지만 제안된 프로토콜의 경우 33분으로 2분 이상이 지속됨을 확인하였다.

결론적으로, 퍼지 연산을 통한 분산 라우팅을 하는 퍼지 라우팅 프로토콜이 네트워크 유지시간이 더 오래 지속됨을 확인할 수 있었다.



(그림 4.1) 에너지 히스토그램





(그림 4.2) 에너지 평균 및 표준편차

### 5. 결론

무선 센서 네트워크에서 라우팅 방법의 선택은 네트워크의 유지시간을 얼마나 오래 지속할지를 결정하는 중요한 요인이 될 수 있다. 기존의 라우팅 프로토콜은 특정한 한 개의 요소를 고려하여 패킷을 송·수신한다. 이로써, 반복된 경로를 사용함으로써 발생하는 국부적인 노드의 에너지 소비는 네트워크 전체 유지시간을 감소시키는 원인이 된다.

본 연구에서는 경로 선택시 고려될 수 있는 여러 요소 중 에너지, 거리, 홉 수를 선정하고 퍼지 연산으로 에너지를 균등하게 사용하면서 네트워크 유지시간을 오래 지속할 수 있는 라우팅 프로토콜을 제안하였다. 사람의 인지와 비슷하게 판단할 수 있는 퍼지 제어 기술을 접목하여 경로 판단력을 높였다. 근본적으로 복잡한 퍼지 제어 이론을 라우팅 기법에 적용하기 위하여 퍼지 제어 값을 산출하지 않고 입력 변수들에 대한 랭크 값만으로 경로를 결정할 수 있게 만들었으며, 이 과정을 단일 루프로 구현하는 기법을 제안함으로써 자원의 제한이 있는 센서 모드에서도 적용할 수 있게 만들었다. 그리고, 데이터 패킷 전송 후 사용한 경로의 환경을 재계산함으로써 분산 라우팅이 가능해졌고, 그 결과 네트워크 유지시간이 길어짐을 확인하였다.

현재는 시뮬레이션에 의하여 퍼지 라우팅 기법의 성능을 평가하였으나, 실제 모드에 적용하여 성능을 평가할 필요가 있으며, 또한 기존에 사용한 척도 이외에 전파의 세기도 고려해 볼 필요가 있다. 그리고, 조건부에 대한 결론부 규칙 매핑 과정에 근사치 방법을 적용하여 연산 복잡도를 더 줄일 수 있는 방법이 연구되어야 할 것이다.

### 참고 문헌

- [1] C.S.Raghabendra, Krishna M. Sisvailngam and Taieb Znati, *Wireless Sensor Networks*, KAP, 2004.
- [2] 이상학, 정태충, *유비쿼터스 컴퓨팅의 핵심 무선 센서 네트워크 기술*, 경희대학교, 2005.
- [3] I.F.Akyildiz, W.Su, Y.Sankarasubramaniam and E.Cayirci, *A Survey on Sensor Networks*, Proc. IEEE, pp.102-114, August 2002.
- [4] S. Hedetniemi and A. Liestman, "A survey of gossiping and broadcasting in communication networks", *IEEE Networks*, Vol. 18, No. 4, pp.319-349, 1988.
- [5] D. B. Johnson and D. A. Maltz. "Dynamic Source Routing in Ad-hocWireless Networks," In T. Imielinksi and H. Korth, *Mobile Computing*, pp.153 - 181. Kluwer Academic Publishers, 1996.
- [6] J.Gao, *Energy efficient routing for wireless sensor networks*. Ph.D. Dissertation, UCLA 2000.
- [7] C. E. Perkins, E. M. Royer, and S. R. Das, "Ad Hoc On Demand Distance Vector (AODV) Routing," *IETF Internet Draft*, draftietf-manet-aodv-08.txt, March 2001.
- [8] L.-X. Wang, *A Course in Fuzzy System and Control*, Eng-lewood Cliffs, NJ: Prentice-Hall, 1997.
- [9] 이광형, 오길록 공저, *퍼지이론 및 응용*, 1995, 홍릉과학출판사
- [10] TinyOS Home Page, <http://webs.cs.berkeley.edu/tos/index.html>, April 2004.



### 이 병 권

e-mail : sonic747@hanmail.net  
1995년 대덕대학 전자과(전문학사)  
1999년 한밭대학교 전자계산학과(학사)  
2002년 한남대학교 컴퓨터공학과(석사)  
2007년 충북대학교 전자계산학과(박사)  
2007년~현재 휴먼엔퓨터정보통신(주)  
선임연구원

관심분야: 임베디드시스템, 무선센서네트워크, 퍼지이론,  
컴퓨터구조



### 전 중 남

e-mail : joongnam@cbu.ac.kr  
1981년 연세대학교 전자공학과(학사)  
1985년 연세대학교 전자공학과(석사)  
1990년 연세대학교 전자공학과(박사)  
1990년~현재 충북대학교 전기전자컴퓨터  
공학부 교수

관심분야: 임베디드시스템, 컴퓨터구조, 병렬처리,  
무선센서네트워크