韓國컴퓨터情報學會 論文誌
第14卷 第4號, 2009. 4.

2009-14-4-3-1

무선 센서 네트워크에서 보이드 문제 해결을 위한 위치 기반 데이터 전송 기법

김 석 규 *

A Geographic Routing based Data Delivery Scheme for Void Problem Handling in Wireless Sensor Networks

Seog Gyu Kim *

요 약

무선 센서 네트워크에서 위치 기반 그리디 포워딩 기법(Geographic greedy forwarding) 은 한 홉 내의 주위 노드들의 위치 정보만 가지고 데이터 전송 경로를 설정하기 때문에 자신보다 목적지에 더 가까운 노드가 존재하지 않는 경우 데이터를 전달할 수 없는 경우가 발생한다. 본 논문에서는 이러한 보이드 문제가 발생한 환경에서 데이터를 효과적으로 전송하기 위해 향상된 위치 기반 라우팅 알고리즘(CGR: Cost based Geographic Routing) 을 제안한다. 제안하는 CGR 알고리즘은 그림자 경계 영역을 설정하고 설정된 영역에 존재하는 노드들끼리 비용 함수 연산 알고리즘을 통하여 효과적인 그리디 포워딩이 수행되도록 한다. 시뮬레이션 결과, 제안한 알고리즘이 기존의 알고리즘보다 평균 전체 라우팅 경로 길이가 줄어들 뿐만 아니라, 데이터 전달 성공 비율이 높고 에너지 절감 효과가 뛰어남을 확인 하였다.

Abstract

In wireless sensor networks(WSNs), geographic greedy forwarding fails to move a packet further towards its destination if the sender does not have any closer node to the destination in its one hop transmission region. In this paper, we propose a enhanced geographic routing, called CGR(Cost based Geographic Routing) for efficient data delivery against void problem environment. CGR first establishes *Shadow Bound Region* and then accomplishes *Renewing Cost Function Algorithm* for effective greedy forwarding data delivery. Our simulation results show significant improvements compared with existing schemes in terms of routing path length, success delivery ratio and energy efficiency.

▸ Keyword : 무선 센서 네트워크(Wireless Sensor Network), 그리디 포워딩(Greedy Forwarding), 위치 기반 라우팅(Geographic Routing), 보이드 문제(Void Problem)

• 제1저자 : 김석규
• 투고일 : 2009. 03. 30, 심사일 : 2009. 04. 09, 게재확정일 : 2009. 04. 20.
* 안동대학교 전자정보산업학부 조교수

Ⅰ. 서 론

무선 센서 네트워크(Wireless Sensor Network; WSN)는 일반적으로 제한된 전원과 리소스를 갖는 많은 센서 노드들과 이러한 센서 노드들로부터 정보를 수집하고 처리하여 백본망 혹은 다른 네트워크에 전달하는 싱크 노드들로 구성된다. 이러한 제한된 배터리 전원 사용과 센서 노드들의 교체 불가능 때문에, 무선 센서 네트워크에서는 에너지 효율적인 데이터 전송 방식과 네트워크의 전체 수명을 오랫동안 유지하는 것이 최우선적인 설계 주목적이다. 이러한 특징과 더불어, 최근에는 LBS(Location Based Services)와 같은 위치 기반 응용 서비스 및 VANETs(Vehicular Ad Hoc Networks)과 같은 응용 기술에 대한 관심 증가로 인하여 위치 기반 라우팅(Geographic Routing) 기술이 활발히 연구 개발되고 있다. 위치 기반 라우팅 기술은 일반적인 망 토폴로지 정보를 이용하는 대신에 기본적으로 노드들의 상호 위치 정보와 목적지 위치 정보를 이용하여 데이터를 효율적으로 전송하는 방법이다. 위치 기반 라우팅 기법은 빈번한 네트워크 토폴로지 변화에도 쉽게 적응할 수 있으며, 또한 일반적인 토폴로지 기반의 라우팅 기법에 비해 라우팅 오버헤드가 적어 보다 에너지 효율적인 기법이라 할 수 있다 [1][2].

한편, 이러한 위치 기반 라우팅은 일반적으로 위치 정보 교환을 위해 상호 위치를 결정하는 기법과 이러한 위치 정보를 이용하여 목적지로 데이터를 전달하기 위해 다음 전달 노드를 결정하는 그리디 포워딩(Greedy Forwarding) 기법으로 구성된다 [1]. 그리디 포워딩이란, 전송 노드가 목적지 노드로 데이터를 전송하고자 할 때, 이웃 노드들 중에서 전송 노드의 통신 반경 내에 목적지까지 거리가 가장 가까운 이웃 노드를 다음 홉(next hop)으로 선택하여 데이터를 전송하는 방식이다. 〈그림 1〉에서 보여지듯이, 소스 노드로부터 목적지 노드까지 데이터를 전송 하고자 할 때, 통신 반경 내에서 목적지 노드까지의 거리가 가장 가까운 노드를 다음 전달 노드(next hop)로 선택하여 최종 목적지까지 데이터를 전송하는 방식이다. 그러나 무선 센서 네트워크에서는 에너지가 완전히 소모되어 동작하지 않은 노드들이나, 고장 난 노드들, 이동성이 있는 노드들 또는 통신 거리 내에 데이터를 전달 할 수 있는 노드들 자체가 없는 경우 등이 발생하여 데이터를 목적지까지 전송할 수 없는 영역, 즉 보이드 (Void, 혹은 보이드 영역(Void area))가 존재할 수 있다 [2]. 〈그림 2〉에서 보여지듯이 노드 n1의 경우, 자신의 통신 반경 내에 목적지 노드까지의 거리가 자기 자신과 목적지 노드까지의 거리보다 짧은 노드가 존재하지 않아 그리디 포워딩에 의한 데이터 전송이 실패하게 된다. 하지만, n2와 n3의 경우에 비록 목적지까지의 거리가 n1의 경우보다 길기는 하지만, n2와 n3 주위에 목적지까지의 데이터 전달을 가능케하는 이웃 노드들이 존재하다. 따라서 n2 혹은 n3로 데이터를 전달하게 되면 결과적으로 최종 목적지 노드까지 데이터 전송이 성공적으로 이루어지게 된다. 이처럼 보이드가 발생한 상황에서 목적지로의 데이터 전달 장애를 초래하므로 이러한 문제를 해결하는 것이 위치 기반 그리디 포워딩 방식의 라우팅에서 매우 중요하다 [1][2]. 본 논문에서는 이러한 보이드 문제(void problem)를 해결하기 위하여 비용 함수 연산을 이용한 위치 기반 라우팅 알고리즘(CGR; Cost based Geographic Routing)을 제안한다. 제안된 CGR은 보이드 문제를 해결하는데 있어서 전체 라우팅 경로 길이를 최소화하여 결과적으로 에너지 소모를 줄이면서도 데이터 전달의 성공률을 높이는 방식이다.

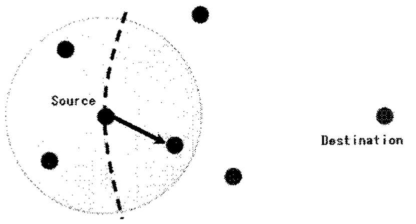

그림 1. 그리디 포워딩
Fig 1. Greedy forwarding

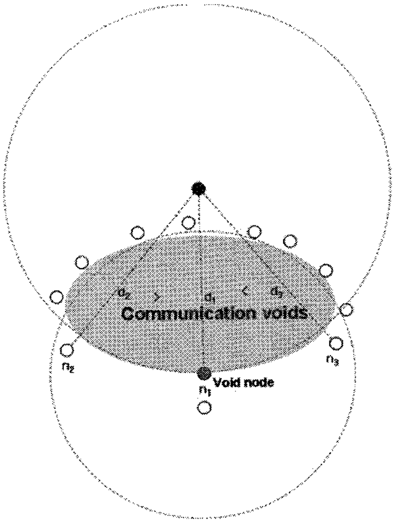

그림 2. 위치 기반 그리디 포워딩에서의 보이드 문제
Fig 2. Void problem in geographic greedy forwarding

본 논문의 구성은 다음과 같다. 2장에서는 기존에 연구되었던 위치 기반 그리디 포워딩 기법들에 대해 간단히 설명하고 3장에서는 CGR 알고리즘에 대해서 자세히 소개한다. 4장에서는 제안한 CGR 알고리즘의 성능을 시뮬레이션을 통하여 분석 및 평가하고, 마지막으로 5장에서 결론을 맺는다.

II. 관련연구

위치 기반의 라우팅 방식은 모두 다음과 같은 가정을 가지고 있다.

- 노드는 자신의 위치 정보를 알고 있다.
- 노드는 목적지 노드의 위치 정보를 알고 있다.
- 노드는 통신 거리를 계산하여 파악가능하며 양방향 링크를 사용한다.

위치 기반의 라우팅 방식의 장점은 오직 자신과 자신의 이웃 노드에 대한 위치 정보만을 바탕으로 데이터 전송을 하기 때문에 네트워크 전체에 대한 토폴로지 관리를 할 필요가 없다는 점이다. 하지만, 앞서 언급했듯이 보이드 문제가 발생할 수 있으므로 이것을 해결하기 위해 GPSR [3], GOAFR+ [4], GOAFR++ [5], BOUNDHOLE [6], PAGER [7], YAGR [8], Node-MAP [9] 같은 알고리즘들이 연구되어 제안되었다.

GPSR [3], GOAFR+ [4], GOAFR++ [5] 에서 제안하는 방식은 그래프 기반의 라우팅 알고리즘으로써 기본적으로 전체 네트워크를 평면 그래프로 구성하여 데이터를 전송하는 방식이다 (Planarization technique). 이 알고리즘들은 통신 거리 내에 목적지까지의 거리가 더 작은 이웃 노드가 없어서 데이터를 전달하지 못하게 되면, 목적지 방향을 기준으로 오른손 법칙을 써서 목적지에 가까운 면으로 데이터를 전달하도록 경로를 설정한다. 오른손 법칙(right hand's rule)을 사용하면 노드와 노드 사이에 수많은 링크가 형성되게 되는데, 두 노드 사이를 연결한 링크와 또 다른 노드 사이의 링크가 서로 교차되지 않도록 해야 한다. 그러나 이렇게 각 노드가 교차되지 않도록 하기 위해서는 각 노드에서 계산량이 증가되며 각각의 면(face)를 따라 우회하는 과정에서 최단 경로(shortest path)와 비교하여 패킷의 전달 횟수가 많게 되어, 전달 과정 중에 패킷을 잃어버릴 확률을 높이고, 각 노드에서의 에너지 소비가 증가된다.

BOUNDHOLE [6]은 교차되는 링크를 제거하지 않고도 보이드가 모여 이루어진 홀(hole) 주위를 선회하는 방법을 제안하였다. 그러나 이 방법 역시 홀 주위를 오른손 법칙을 사용하여 탐색하게 되며, 이 홀의 모양을 기억하고 있도록 하기 때문에, 홀이 커질수록 각 노드에서의 에너지 소비가 증가하며, 교차되는 링크를 제거하지 않아도 홀 주변을 성공적으로 탐색이 되지만, 각 노드에서 오버헤드는 증가하게 된다.

PAGER [7]과 YAGR [8]의 기법들은 그래프 기반의 알고리즘이 아닌 보이드 문제가 발생할 가능성을 줄이기 위해서 보이드 노드들에 관한 정보를 업데이트 하고 또한 주위 노드들에게 알려주는 방식이다. 이러한 정보를 기반으로 하여 주위 노드들은 보이드 문제 발생을 피하면서 성공적인 그리디 포워딩 데이터 전송을 수행하게 된다. 하지만, 이러한 방식은 심각한 라우팅 오버헤드가 존재하며 상당한 라우팅 수렴 시간(Converging time)이 발생하게 된다. Node-MAP [9]은 이웃 노드들의 위치 정보들에 기반을 둔 전체 네트워크 맵을 구성하여 효율적인 그리디 포워딩 방식을 수행하는 알고리즘이다. 구성된 전체 네트워크 맵으로 인하여 그리디 포워딩 방식이 효과적으로 수행이 되지만, 전체 네트워크 맵을 구성하는데 있어서 마찬가지로, 심각한 라우팅 오버헤드가 존재하며 상당한 라우팅 수렴 시간(Converging time)이 발생하게 된다.

III. CGR 알고리즘

본 논문에서 제안하는 CGR(Cost based Geographic Routing) 알고리즘은 보이드 문제를 해결하기 위해 보이드 노드를 포함하는 그레이 노드들과 그 경계 노드들로 구성된 그림자 경계 영역(Shadow Bound Region)을 설정하고 설정된 영역에 존재하는 노드들끼리 비용 함수 연산 알고리즘을 통하여 서로의 라우팅 비용을 갱신하여 보이드 영역으로의 데이터 전송을 회피하여 최종 목적지로의 데이터 전달을 위한 그리디 포워딩이 효과적으로 수행할 수 있게 한다. 우선 3.1절에서 몇가지 가정 및 네트워크 모델을 정의하고, 3.2절에서 전체적인 CGR 알고리즘 동작에 대해서 설명한다. 마지막으로 3.3절과 3.4절에서 CGR 알고리즘의 구체적인 동작과 구성 단계에 대해서 기술한다.

3.1 네트워크 모델

무선 센서 네트워크는 다수의 센서 노드들과 하나의 싱크 노드로 구성되어 있다. 센서 노드들은 센싱한 데이터를 멀티홉으로 싱크 노드에게 전달한다. 센서 노드들은 2차원 공간에 랜덤하게 배치 되어 있으며, 모든 노드들은 자신의 위치와 싱크 노드의 위치를 알고 있다. 센서 노드들은 unit disk graph로 표현될 수 있는데, 이것은 모든 노드들의 통신거리

(transmission range)가 r로 동일함을 의미한다. 한편, 무선 센서 네트워크 모델을 G(V, E, S), V는 전체 노드들의 집합(V = S ∪ N, N는 노드들의 집합, S는 싱크 노드의 집합), E는 노드들(싱크 노드 포함) 사이의 양방향 통신 링크를 의미하는 그래프로 모델링할 수 있다. 노드는 자신의 이웃 노드와 서로 통신이 가능하며 자신의 위치 정보와 목적지까지의 거리 정보를 이웃 노드에게 주기적으로 전송한다.

3.2 알고리즘 개요

구체적인 CGR 알고리즘을 설명하기 전에, 본 논문에서 사용되는 용어들에 대해서 간략하게 언급한다.

- Def. 1 - 거리 d(x) : 2차원 공간의 위치를 기준으로 노드 x와 싱크 노드 사이의 유클리드 거리(Euclidean distance)를 의미한다.
- Def. 2 - 이웃 노드 N(x) : 노드 x의 통신 반경 내에 존재하는 이웃 노드들의 집합을 의미한다.
- Def. 3 - 보이드 노드(Void node) : 오목 노드(concave node)로써 그리디 포워딩 방식으로 더 이상 패킷을 목적지로 전달할 수 없는 노드를 의미한다.
- Def. 4 - 그레이 노드(Gray node) : 이미 보이드노드가 된 노드를 제외한 이웃 노드들 중에서 더 이상 싱크 노드까지의 거리가 자기 자신보다 작은 노드가 없는 경우에 해당하는 노드로 정의한다. 주목할 점은 이 정의 4에 의해서 보이드 노드 또한 그레이 노드 집합에 포함된다는 것을 의미하는 것이다.
- Def. 5 - 경계 노드(Border node) : 그레이 노드들과 일반 노드 사이의 경계에 존재하는 노드로 정의한다. 다시 말하면, 경계 노드는 자신을 제외한 원 홉 반경 내에서 그레이 노드이외에 반드시 목적지로의 거리가 짧은 노드가 존재한다.
- Def. 6 - 그림자 경계 영역(Shadow Bound Region) : 그레이 노드들과 경계 노드들로 구성된 영역을 의미한다. 이 그림자 경계 영역 안에 존재하는 노드들끼리 비용 함수를 갱신하는 알고리즘을 수행하여 효과적인 그리디 포워딩이 가능케 한다.

기본적으로 CGR 알고리즘은 노드 x의 패킷을 그리디 포워딩 기법을 통해 데이터를 전송하는데 보이드 문제가 발생했을 경우 우회경로를 통해서 효과적으로 패킷을 목적지(싱크) 노드에게 전달하는 방식이다. 앞서 설명했듯이, 그리디 포워딩 방식에서는 노드 x의 통신 반경 내에 존재하는 이웃 노드들 중 목적지까지 거리가 보다 더 가까운 노드가 존재하지 않은 경우 패킷 전송을 실패하게 되는데 이러한 그레이 노드들과 경계 노드들로 구성된 그림자 경계 영역(Shadow Bound Region)을 사전에 설정하고 이 영역에 속해 있는 노드들끼리 위치 정보를 활용하는 비용 함수를 갱신하여 그레이 노드들에게 상대적으로 높은 위치 정보의 비용함수를 설정하여 주위 노드들이 그레이 노드들로의 데이터 전송을 회피하고 우회하여 전체적인 네트워크에서 거리에 따른 위치기반 그리디 포워딩이 원활하게 이루어지도록 함으로써 보이드 문제를 해결한다.

그레이 노드들로 인하여 그리디 포워딩 방식에 의한 데이터 전달이 실패하면, 그리디 포워딩이 아닌 다른 방식을 통해 우회하게 되는데 일반적으로 오른손 법칙(right hand's rule)에 의해서 우회 경로를 찾은 후에 목적지로의 데이터 전달을 재개한다. 이러한 방식은 전체 라우팅 경로 길이가 증가되고 우회 경로를 찾는 방식의 비효율성으로 인한 오버헤드가 증가하여 전체적으로 에너지 비효율성과 패킷 손실의 위험성이 커지게 된다. 제안하는 CGR 알고리즘 방식에 의해서, 그레이 노드들로 패킷을 전달할 가능성이 높은 경계 노드들로 구성된 그림자 경계 영역을 설정하면 영역 밖의 노드는 그 영역 안으로 패킷을 전송할 가능성을 없애고 비용 함수 연산에 의한 영역 안에서의 효과적인 그리디 포워딩 방식을 가능하게 하여 결과적으로 소스 노드의 데이터를 효율적으로 안전하게 목적지 노드까지 전달할 수 있다. 이러한 그림자 경계 영역 설정 및 비용 함수 연산 수행은 네트워크 생성 초기 혹은 중간에 노드들의 주기적인 위치 정보 메시지 교환으로 영역을 구성하는 노드들의 위치 변화 혹은 홀 발생이 감지되었을 때 수행된다. 이러한 알고리즘을 수행하기 위해서 CGR에서는 제어 패킷을 교환하게 되는데 제어 패킷에는 {상태정보, 위치, Cost(x)} 정보를 포함하게 된다. 이때 상태정보는 노드 x의 그레이 노드(Gray) 또는 경계 노드(Border)임을 나타내는 요소이며, 위치는 노드 x의 위치 정보를 나타내며, Cost(x)는 비용 함수 연산 알고리즘에서 사용하는 위치 정보를 기반으로 하는 비용 함수를 나타낸다. 구체적인 CGR 알고리즘의 그림자 경계 영역 설정과 비용 함수 갱신 설정 방법에 대해서 다음 절에서 자세히 설명한다.

3.3 그림자 경계 영역 설정

네트워크 설정 초기 그리디 포워딩 방식으로 패킷을 전송하기 이전에 보이드 노드가 발생하면, 이러한 보이드 노드와 주위의 그레이 노드들을 전체 네트워크 토폴로지 상에서 따로 분리하여 관리하는 과정이 필요하다. CGR에서는 일단, 어떤 노드 x가 자신이 그레이 노드(보이드 노드 포함)임을 인식하

면 그림자 경계 영역 설정(Establishing shadow bound region phase) 알고리즘을 수행하게 된다. 이러한 그림자 경계 영역을 설정하여 그리디 포워딩을 실패 없이 수행하여 이에 따른 에너지 효율성을 높이면서 데이터를 성공적으로 목적지에 전달한다. 자신이 그레이 노드임을 인식한 노드는 자신의 상태를 그레이 노드임을 의미하는 'Gray'를 명시한 후 자신의 위치 정보와 거리 정보를 포함하는 제어 패킷을 이웃 노드들에게 전송한다. 이 제어 패킷을 수신한 노드는 전송 반경 내에 있는 이웃 노드들 중 그레이 노드들을 제외하고 그리디 포워딩을 통해 패킷을 전송할 수 있는 지를 확인한다. 즉, 그레이 노드들을 제외한 나머지 이웃 노드들 중 자신 보다 목적지로의 거리가 가깝거나 같은 노드가 있는지를 확인한다. 만약 존재하지 않으면, 이전과 마찬가지로, 자신이 그레이 노드임을 인식한 노드는 자신의 상태를 그레이 노드임을 의미하는 'Gray'를 명시한 후 자신의 위치 정보와 거리 정보를 포함하는 제어 패킷을 이웃 노드들에게 전송한다. 만약 존재하면, 이것은 그리디 포워딩을 수행할 수 있는 다른 노드가 존재한다는 것을 의미하므로, 자신의 상태를 그림자 영역 경계임을 의미하는 'Border'를 명시한 후 자신의 위치 정보와 거리 정보를 포함하는 제어 패킷을 이웃 노드들에게 전송한다. 그리고 최종적으로 그림자 경계 영역 설정 단계를 끝마친다. 그림자 경계 영역 설정에 대한 알고리즘동작이 〈그림 3〉에서처럼 요약될 수 있다. 〈그림 4〉에서 보여지듯이, 보이드 노드로부터 그림자 경계 영역 설정 알고리즘을 통해 주위 노드들의 상태를 업데이트 하여 경계 노드들까지 갱신 한 후, 이들 노드들로 이루어진 그림자 경계 영역이 최종적으로 설정된다.

Algorithm 1. Establishing shadow bound region phase

```
status(x) = Gray;
if receive control packet from a neighbor y then
  if d(x) ≤ d(y)
     status(x) = Gray;
  else
     status(x) = Border;
  end if
end if
send out a control packet containing status and location
information to all one hop neighbors
```

그림 3. 그림자 경계 영역 설정
Fig 3. Establishing shadow bound region phase

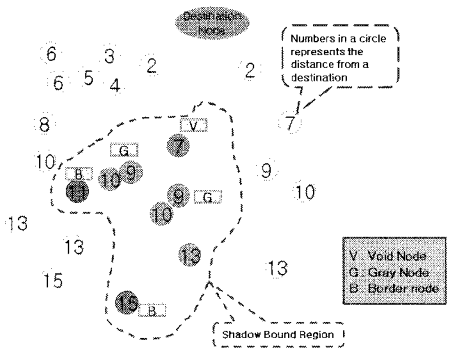

그림 4. 그림자 경계 영역 설정
Fig 4. Establishing Shadow Bound Region

3.4 비용 함수 갱신 설정

그림자 경계 영역 설정이 끝난 후, 경계 영역 밖의 노드들이 경계 영역 안의 그레이 노드들에게 데이터를 전달을 방지하고, 또한 경계 영역 안의 노드들이 보이드 노드를 회피해서 우회 경로를 찾아 데이터를 전달 할 수 있도록 하는 알고리즘이 추가적으로 수행 되어야 한다. CGR에서는 이러한 것이 수행 될 수 있도록 위치 정보 기반의 비용 함수를 도입하여 이러한 비용 함수를 라우팅 경로에 사용하여 보이드 문제를 해결하면서 그리디 포워딩이 원활하게 수행되도록 한다. 앞선 단계인 그림자 경계 영역 설정이 끝난 후, 비용 함수 갱신 설정 단계는 다음의 순서대로 동작한다. 우선, 첫째 노드 x는 자기 자신이 경계 노드임을 확인하자마자, 자신의 거리값(d(x): 자신과 목적지 노드와의 유클리드 거리)을 비용함수(cost(x))로 설정한다. 그리고 자신의 위치 정보와 상태(경계 노드), 비용 함수 값을 포함하는 제어 패킷을 주위 이웃 노드들에게 전송한다. 이 제어 패킷을 받은 노드는 자신이 그레이 노드 상태인지를 확인 한 후, 자신의 비용 함수 값과 수신한 제어 패킷안의 비용 함수 값을 비교하여 자신의 값이 작거나 같은 경우 비용함수를 다음과 같이 설정한다.

$$cost(x) = \max(cost(x), cost(y)) + |d(x) - d(y)| \qquad (1)$$

위 비용 함수의 의미는 그림자 경계 상태에 있는 모든 노드들의 거리 값을 역으로 증가 시키는 기능을 하게 된다. 이렇게 경계 영역 안의 노드들의 거리 값을 증가시킴으로써 경계 밖의 노드들은 경계 영역 안의 노드들의 거리 값이 상대적으로 크기 때문에 목적지로의 거리가 멀다는 의미를 갖게 되어 데이터를 전달하지 않고 다른 노드들에게 데이터를 전달

하게 된다. 또한 위의 비용 함수 연산으로 인하여 그림자 경계 안의 노드들 사이에도 보이드 노드에 가까울수록 비용함수가 증가 되어 보이드 노드로부터 데이터 전송이 그림자 경계 영역 밖으로 탈출하여 결과적으로 그리디 포워딩이 원활하게 수행될 수 있도록 한다. 일단 비용 함수를 수식 (1)과 같이 설정한 후, 다시 자신의 상태(그레이 노드), 위치 정보, 그리고 갱신된 비용 함수를 포함하는 제어 패킷을 이웃 노드들에게 전송한다. 만약 위의 자신의 비용 함수 값과 수신한 제어 패킷안의 비용 함수 값을 비교하여 자신의 값이 큰 경우, 수신한 제어 패킷을 버린다. 위 비용 함수 갱신 설정은 경계 노드로부터 그림자 경계 영역 안의 모든 노드들로의 비용 함수 갱신이 완료된 후 끝나게 된다. 결과적으로 이렇게 비용 함수 갱신이 완료 되면, 그레이 영역의 모든 노드들이 위치 정보에 기반 한 비용 함수 값이 높아 우회 경로를 통해 그리디 포워딩이 가능할 뿐만 아니라 경계 밖의 노드들에게는 사전에 경계 영역 안으로 패킷을 전송하지 않고 다른 우회 경로의 노드들에게 전송하여 최종적으로 목적지에게 데이터를 전달 할 수 있게 된다. 여기서 사용하는 비용 함수는 기본적으로 그리디 포워딩에서 사용하는 거리 값에 기반한 비용 함수이며, 그림자 경계 영역 설정 된 후, 그레이 노드들과 경계 노드들 사이에서만 사용된다. 비용 함수 갱신 설정에 대한 알고리즘 동작이 <그림 5>에서처럼 요약될 수 있다. <그림 6>은 <그림 4>의 예에서 비용 함수 갱신 알고리즘을 통해 그림자 경계 영역 안의 노드들끼리 경계 노드로부터 위치 정보 값이 비용 함수를 통해 갱신되어 최종적으로 위치 정보가 수정 업데이트 되어 그리디 포워딩에 의한 성공적인 데이터 전송을 가능케 만든다.

```
Algorithm 2. Renewing cost function phase
1: status(x) = Border;
2:     cost(x) = d(x);
3:     send out a control packet containing status, location
       information and cost function to all one hop neighbors
4: status(x) = Gray;
5:     cost(x) = d(x);
6: if receive control packet from a neighbor y then
7:    if cost(x) ≤ cost(y)
8:       cost(x) = max(cost(x), cost(y)) + |d(x) - d(y)|
9:       send out a control packet containing status, location
         information and cost function to all one hop neighbors
10:   else
11:      drop the packet
12:   end if
13: end if
```

그림 5. 비용 함수 갱신 설정
Fig 5. Renewing cost function phase

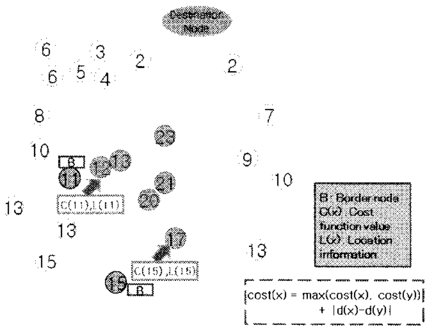

그림 6. 비용 함수 갱신
Fig 6. Renewing Cost Function

IV. 성능 평가

제안하는 CGR 알고리즘과 그리디 포워딩의 대표 알고리즘인 GPSR [3]의 성능 비교분석을 위하여 우리는 MATLAB [10]을 사용하여 실험하였다. 시뮬레이션은 다양한 토폴로지 크기의 변화(100x100, 110x110, 120x120, 130x130, 140x140 m*m)와 노드 밀집도의 변화(density; 하나의 홉 변경 내에 존재하는 노드 수; 5~20)에 따른 반복적인 실험을 하였다. 또한 모든 노드의 전송 반경은 5m로 동일하며, 노드들과 보이드들은 랜덤하게 네트워크에 위치되어 있다. <표 1>은 본 논문에서 사용된 기본적인 시뮬레이션 파라미터 값들을 나타낸 것이다.

표 1. 시뮬레이션 파라미터
Table 1. Simulation parameters

파라미터	값
프로토콜	GPSR, CGR
토폴로지 크기	100x100, 110x110, 120x120, 130x130, 140x140 (m*m)
통신 전송 반경	5 (m)
송신 에너지 소모량	14.9 (mW)
수신 에너지 소모량	12.5 (mW)
밀집도 (Density; 하나의 홉 반경 내에 존재하는 노드 수)	5 ~ 20 (nodes/π*5m*5m)

4.1 패킷 전달 비율

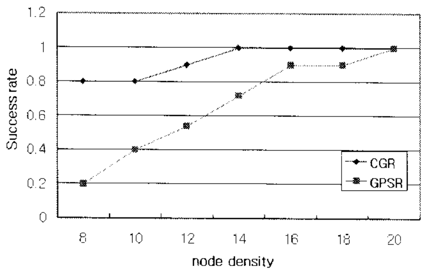

그림 7. 패킷 전달 성공률
Fig 7. Packet delivery ratio

<그림 7>은 노드의 밀집도 변화에 따른 소스 노드로부터 목적지 노드(싱크노드)로까지의 패킷 전달 성공률 결과를 나타낸 것이다. 일반적으로 노드 밀집도가 상당히 높은 경우 보이드 문제가 발생 했을 경우일지라도 주위의 대체 전송 노드를 발견할 가능성이 높으며 이에 따른 데이터 지연 그리고 전송 실패율이 적게 된다. 따라서 GPSR이나 CGR 알고리즘의 성능이 거의 비슷하게 됨을 알수 있다. 반대로 노드 밀집도가 낮은 경우 보이드 문제가 발생 했을 경우 주위의 대체 전송 노드를 발견할 가능성이 상대적으로 적으며 이에 따른 데이터 지연 그리고 전송 실패율이 높게 된다. 이것은 GPSR의 경우 그래프 이론에서의 평면화(planarization) 실패에 따른 결과라고 볼 수 있다. 따라서 본 실험 결과에서 보이듯이 노드 밀집도 변화에 상관없이 전체적으로 데이터 전달 성공률이 제안하는 CGR이 GPSR 보다 높음을 확인 할 수 있다.

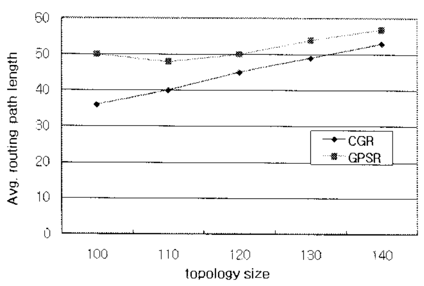

그림 8. 토폴로지 크기에 따른 평균 라우팅 경로 길이
Fig 8. Average routing path length with topology size

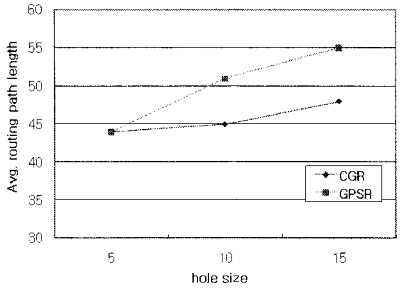

그림 9. 홀 크기에 따른 평균 라우팅 경로 길이
Fig 9. Average routing path length with hole size

4.2 라우팅 경로 길이

<그림 8>, <그림 9>은 각각 토폴로지 크기 변화(00x100, 110x110, 120x120, 130x130, 140x140 m*m)와 보이드 노드로 인해 생긴 홀 크기 변화에 따른 평균 라우팅 경로 길이에 대산 실험 결과이다. 평균 라우팅 경로 길이란, 소스 노드로부터 목적지 노드까지의 패킷 전송시 전체 라우팅 경로 길이 즉, 홉 수에 대한 평균 값을 의미한다. 일반적으로 네트워크 토폴로지 크기가 클수록 라우팅 경로 길이가 증가 되는데, 그림 7에서 보여주듯이 CGR의 그레이 영역 경계 설정과 비용 함수 갱신 설정 알고리즘에 의해서 전체적으로 GPSR보다 라우팅 경로 길이가 적음을 알 수 있다. 또한 그림 8에서 보여주듯이 홀의 크기가 클수록, CGR 알고리즘은 보이드 문제가 발생한 환경에서도 전체적인 그리디 포워딩이 원활하게 해 주기 때문에 패킷 전송시 전체 라우팅 경로 길이가 GPSR 알고리즘보다 짧음을 보여주게 된다.

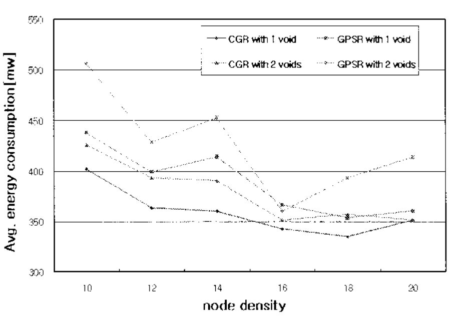

그림 10. 평균 에너지 소모량
Fig 10 Average energy consumption

결과적으로, 이러한 전체 평균 라우팅 경로 길이가 짧다는 것은 그만큼 데이터 전달시 지연도 줄게 되고 이에 따른 신뢰

성 및 에너지 효율성이 높다는 것을 의미한다. 따라서 CGR 방식이 GPSR보다 뛰어남을 확인할 수 있다.

4.3 에너지 소모

<그림 10>는 노드 밀집도 변화에 따른 평균 에너지 소모량에 대한 결과를 나타낸다. 이 실험에서는 전체 네트워크 토폴로지 크기는 100x100 (m*m)로 고정한 상태에서 노드 밀집도를 변화시켜 실험하였다. 일반적으로 보이드 문제가 발생한 상황에서 데이터 전달시 노드 밀집도가 적은 환경에서 보다 더 에너지 소모량이 높다. 이것은 노드 밀집도가 높을수록 그만큼 우회 경로를 위한 대체 노드를 보다 신속하게 찾을 가능성이 높기 때문이다. CGR 알고리즘은 GPSR에 비해서 그레이 영역 경계 설정과 비용 함수 갱신 설정 단계에서 보다 제어 패킷을 주고 받게 되므로 이에 따른 오버헤드의 에너지 소모가 발생한다. 하지만, GPSR의 경우 보이드 문제를 해결하는데 보다 더 에너지 소모가 많아 결과적으로 제안하는 CGR 방식이 GPSR보다 에너지 효율성이 뛰다는 것을 본 실험에서 확인 할 수 있다. 그림 9에서 보여지듯이 보이드 수가 많을수록 에너지 소모량이 증가되며, 전체적으로 CGR이 GPSR보다 평균 에너지 소모량이 적음을 확인 할 수 있다.

V. 결 론

일반적으로 위치 기반 그리디 포워딩(Geographic greedy forwarding) 방식은 한 홉내의 노드들의 위치정보만을 가지고 데이터를 전송하기 때문에 에너지 효율성이 적은 오버헤드가 필요한 무선 센서 네트워크에 적합하다. 그러나 보이드 문제가 발생하는 경우 그리디 포워딩이 실패하는 단점이 존재한다.

본 논문에서는 이러한 보이드 문제를 해결하기 위해 CGR(Cost based Geographical Routing) 알고리즘을 제안하여 보이드 문제가 발생한 상황에서 그리디 포워딩 방식이 원활하게 수행될 수 있도록 한다. CGR 알고리즘은 패킷을 목적지로 전송하기 이전에 보이드 노드와 그레이 노드들로 구성된 경계 영역을 설정하여 이 영역안에 존재하는 노드들의 위치 정보를 비용 함수를 이용하여 갱신함으로써 전체적으로 그리디 포워딩 방식이 문제없이 동작할 수 있도록 한다. 그림자 경계 영역 밖의 노드들에게는 그림자 경계 안의 노드들에게 데이터를 전달하지 못하도록 비용 값을 조정하고 또한, 경계 안의 노드들에게는 우회 경로를 찾아 보이드 노드를 회피하여 그리디 포워딩이 가능하도록 비용 값을 갱신하게 된다. 실험을 통하여 CGR의 경우 GPSR 방식 보다 노드 밀집도가 낮은 경우에도 데이터 전달 성공률이 높음을 확인 하였으며, 데이터 전달시 전체적인 평균 라우팅 경로 길이가 상대적으로 짧음을 확인 하였다. 결과적으로 이러한 라우팅 경로 길이가 짧음은 전체적인 라우팅 오버헤드를 줄어들게 되고 이에 따른 에너지 소비 및 전송 지연 시간이 감소된다는 것을 확인할 수 있다. 따라서 CGR 방식이 이러한 보이드 문제를 해결하는 데 뛰어난 성능을 가지고 있음을 알 수 있다.

참고문헌

[1] M. Mauve, J. Widmer, and H. Hartenstein, "A Survey on Position-based Routing in Mobile Ad Hoc Networks," IEEE Network Magazine, Vol. 15, No. 6, pp. 30-39, Nov. 2001.

[2] N. Ahmed, S. S. Kanhere, and S. Jha, "The Holes Problem in Wireless Sensor Networks: A Survey," ACM Mobile Computing and Communications Review, Vol. 9, No. 2, pp. 4-18, Apr. 2005.

[3] B. Karp and H. T. Kung, "Greedy Perimeter Stateless Routing for Wireless Networks," Proc. ACM MobiCom, Boston, USA, pp. 243-254, July 2000.

[4] F. Kuhn, R. Wattenhofer, and Y. Zhang, "Geometric Ad-hoc Routing: of Theory and Practice," Proc. ACM Symposium on Principles of Distributed Computing, Boston, USA, pp. 63-72, 2003.

[5] H. Frey and I. Stojmenovic, "On Delivery Guarantees of Face and Combined Greedy Face Routing in Ad Hoc and Sensor Networks," Proc. ACM/IEEE Mobicom, pp. 390-401, Sept. 2006.

[6] Q. Fang, J. Gao, and L. J. Guibas, "Locating and Bypassing Routing Holes in Sensor Networks," Proc. IEEE INFOCOM, Hong Kong, pp. 2458-2468, Mar. 2004.

[7] L. Zou, M. Lu, and Z. Xiong, "A Distributed Algorithm for the Dean End Problem of Location Based Routing in Sensor Networks," IEEE Transactions on Vehicle Technology, Vol. 54, No. 4, pp. 1509-1522, July 2005.

[8] J. Na, D. Soroker, and C. K. Kim, "Greedy Geographic Routing Using Dynamic Potential

Field for Wireless Ad Hoc and Sensor Networks," IEEE Communication Letter, Vol. 11, No. 3, pp. 243-245, Mar. 2007.
[9] 장우석, "Ad Hoc 네트워크에서 이웃노드 정보를 이용한 전체 노드 맵 구현," 한국 컴퓨터정보논문지, 제11권 제6호, 221-226쪽, 2006년 12월.
[10] MATLAB, *http://www.mathworks.com.*

저 자 소 개

김 석 규
1990년: 연세대학교 전자공학과 학사
1992년: 연세대학교 전자공학과 석사
1997년: 연세대학교 전자공학과 공학박사
1997년-2004년: SK텔레콤 선임연구원
2006년-현재: 안동대학교 전자정보산업학부 정보통신공학전공 조교수
〈관심분야〉 센서네트워크, 모바일통신