

# 센서네트워크에서의 프록시 트라젝토리 기반 데이터 저장 기법

임 화 정<sup>†</sup> · 이 헌 길<sup>††</sup>

## 요 약

센서네트워크에서 데이터를 효율적으로 전송하는 것은 매우 중요한 문제이다. 수집된 데이터를 외부의 저장소나 시스템에 저장하기보다는 센서네트워크 내에 있는 노드들에 직접 저장하는 데이터 중심 저장소 기법은 높은 접근성을 제공할 수 있다. 센서네트워크에서 사용되는 데이터 저장소 기법을 사용하면 데이터를 외부 저장소로 전송하기 위한 지연시간과 에너지 낭비없이 실시간으로 수집된 데이터에 접근할 수 있는 것이다. 그러나 데이터를 저장하는 일부 노드에 질의가 집중될 경우 질의에 대한 응답시간이 길어질 수 있으며, 부하집중으로 노드의 에너지가 빠르게 고갈되어 네트워크 수명이 단축될 수 있다. 본 논문에서는 질의를 처리하는 저장소 노드에 발생하는 부하를 주위의 여러 노드들로 분산시켜 질의에 대한 응답지연시간을 줄이고, 사용자의 이동성을 지원하는 프록시개념의 프록시 트라젝토리를 형성하는 프록시 트라젝토리 저장기법(APT)기법을 제안한다. 시뮬레이션을 통하여 제안기법의 노드별 질의 응답시간의 단축과 응답 홉 수의 감소를 검증하였다.

키워드 : 센서네트워크, 프록시, 트라젝토리

## An Proxy Trajectory Based Storage in Sensor Networks

Hwa-Jung Lim<sup>†</sup> · Heon-Guil Lee<sup>††</sup>

## ABSTRACT

Efficient data dissemination is one of the important subjects for sensor networks. High accessibility of the sensed data can be kept by deploying the data centric storage approach in which data is stored over the nodes in the sensor network itself rather than external storages or systems. The advantage of this approach is its direct accessibility in a real-time without the severe burden on delay and power dissipation on the data path to the external storages or systems. However, if the queries from many users are concentrated to the few nodes with data, then the response time could be increased and it could lead to the reduction of network life time by rapid energy dissipation caused by concentrated network load. In this paper, we propose a adaptive data centric storage scheme based on proxy trajectory (APT) mechanism. We highlight the data centric storage mechanism by taking account of supporting large number of users, and make it feasible to provide high-performance accessibility when a non-uniform traffic pattern is offered. Storing data around the localized users by considering spatial data-access locality, the proxy trajectory of APT provides fast response for the users. The trajectory, furthermore, may help the mobile users to roams freely within the area they dwell.

Keywords : Sensor Network, Proxy, Trajectory

## 1. 서 론

유비쿼터스 환경에서 실시간으로 정보를 얻기 위해서는 외부저장소에 있는 데이터에 접근하는 것이 아니라 사용자가 센서네트워크에 직접 접속하여 데이터를 얻어오는 방식을 취하면 실시간성을 보장할 수 있다. 실시간성을 보장하기 위해서는 센서노드들이 수집한 데이터에 효율적으로 접근

근하기 위한 메커니즘이 필요하다[1,2].

네트워크상 저장된 데이터 접근방안으로 분산 해쉬 테이블(Distributed Hash Table, DHT)과 같은 네트워크상의 노드에 데이터를 저장하고, 저장된 데이터를 얻기 위해 노드에 접근하는 기법이 연구되어 왔다[3,4].

DHT를 센서네트워크에 적용하면 사용자가 외부 저장소를 거치지 않아도 센서노드에 저장된 데이터에 쉽게 접근할 수 있지만 센서네트워크에서 노드의 ID를 가지고 Hash Table을 구성하면 해당 노드까지 데이터를 전송하기 위한 오버헤드가 커질 수 있다. 이러한 문제점을 해결하기 위한 방안으로 노드의 ID가 아니라 노드가 위치하고 있는 지리적

\* 2006년도 강원대학교 학술연구조성비로 연구하였음.  
† 준 회 원 : 강원대학교 컴퓨터정보통신공학과 박사수료  
†† 정 회 원 : 강원대학교 컴퓨터학부 교수(교신저자)  
논문접수 : 2008년 5월 20일  
수정일 : 1차 2008년 6월 27일, 2차 2008년 7월 4일  
심사완료 : 2008년 7월 7일

좌표(위치)정보와 연관된 Hash Table을 구성하는 방법이 있다[5].

센서노드들이 센싱한 데이터에 Hash를 적용하여 저장할 위치정보를 찾아 전달하면 해당 위치에 있는 노드가 데이터를 관리한다. 센서네트워크 사용자는 데이터의 Hash값으로 데이터가 저장된 위치로 질의를 전송하여 원하는 데이터를 구할 수 있다. 데이터 접근용이성을 제공할 수 있는 방법이긴 하지만, 센서네트워크 특성상 이를 구현하는데 다음과 같은 문제가 있다. 첫째, 노드가 불규칙하게 설치되는 센서네트워크의 경우가 특정 좌표주변에 매우 많은 노드가 밀집되거나, 존재하지 않을 가능성이 있다. 둘째, 매우 제한된 자원을 가진 센서 노드들은 하나의 노드에 너무 많은 부하가 집중되면 에너지 자원이 빨리 고갈되어 네트워크 수명이 단축된다.

본 논문에서는 문제해결을 위해 기존의 분산 해쉬 테이블 기반의 데이터 저장소 기법에 트라젝토리 기반 전달(Trajectory Based Forwarding, TBF)기법을 도입하는 데이터의 인기도와 질의 특성에 따라 트라젝토리를 설정하여 사용자의 응답 시간 단축 및 이동성을 지원할 수 있는 프록시 트라젝토리 저장기법(Adaptive Proxy Trajectory based Storage, APT)을 제안한다.

TBF기법은 수학적 함수를 이용하여 메시지가 전달될 방향과 경로를 결정하는 방법으로 여러 가지 모양의 경로를 설정할 수 있다[7,8,9]. APT에서는 DHT를 통해 구해진 좌표 즉, 타깃 포인트(Target Point)를 둘러싸는 트라젝토리(Trajectory)를 형성하여 노드들이 트라젝토리에 속하는 노드들에 데이터를 저장하도록 한다.

APT기법에서는 DHT에 비해 트라젝토리에 속하는 여러 노드들이 데이터를 저장하고 질의에 응답하기 때문에 부하가 분산되는 효과를 가진다. 또한 데이터에 대한 질의빈도가 높아지는 경우 트라젝토리가 형성되는 거리(H)를 조절하여 트라젝토리를 넓혀 부하가 분산되도록 한다. 만약 질의를 전송하는 사용자들이 특정영역에 밀집해 있는 경우 또는 사용자가 이동하는 경우에 응답지연시간을 줄이기 위하여 해당 영역 주변에 새로운 트라젝토리를 형성하여 프록시 서버처럼 동작하도록 한다.

본 논문은 1장의 서론에 이어, 2장에서 관련 연구의 문제점 및 해결 방안에 대해 분석한다. 3장에서는 제안 기법에 대해 설명하고 4장에서 시뮬레이션으로 구현하여 성능평가를 하고, 5장에서 결론 맺는다.

## 2. 관련연구

데이터를 송·수신하는 노드가 중심이 되는 일반 네트워크와는 달리 센서네트워크에서는 노드들이 수집한 데이터가 통신의 중심이 된다. 센서네트워크 연구가 시작된 초기부터 Direct Diffusion과 같은 데이터 중심의 라우팅 기법들이 연구되어져 왔다[1].

Direct Diffusion의 문제점은 첫째, 동일한 데이터에 대하

여 중복적인 질의나 응답이 여러 곳에서 발생하는 경우 많은 오버헤드를 유발할 수 있다. 둘째, 수집된 데이터들이 모두 베이스스테이션으로 전송되어 외부저장소에 저장되는 기법에서는 불필요한 데이터 전송으로 인한 에너지낭비를 초래할 수 있다.

따라서 광범위한 지역에 대한 다양한 정보를 사용자에게 효율적으로 제공하기 위한 방안으로 센서노드들을 데이터 저장소로 이용하는 기법들이 연구되고 있다[3,5].

데이터 중심 저장소 기법의 대표적인 예로 GHT기법이 있다[5]. GHT는 DHT기법을 센서네트워크에 적용한 것으로, 위치기반 라우팅 프로토콜인 GPSR프로토콜을 기반으로 동작한다[6].

GPSR은 조밀하게 분포되어있는 센서네트워크 환경에서 좌표를 찾을 때, 해쉬함수를 통해 생성한다. 따라서 실제 생성되는 좌표는 실제 노드의 배치와 상관없이 생성되므로 해당 좌표에 노드가 없는 경우가 발생한다. 이러한 문제의 해결책으로 GHT는 먼저 좌표와 인접한 노드를 홈 노드(Home Node)로 설정하고 데이터를 저장하도록 한다. 또한 홈 노드에 저장 공간이 부족할 경우를 대비하여 계층적으로 데이터를 저장한다.

GHT의 문제점은 첫째, 노드가 밀집해 있는 경우 다양한 경로를 통해 전달되는 질의가 홈 노드(Home Node)에 전달되지 못할 가능성이 있다. 둘째, 계층적으로 데이터를 저장할 때, 저장 공간에 대한 부하분산만 고려하고 네트워크상의 부하는 고려하지 않았기 때문에 홈 노드에 질의응답으로 인한 부하가 집중될 수 있다는 문제점을 지닌다. 따라서 질의 빈도가 높은 데이터를 저장하고 있는 노드들이 다른 노드들에 비해 부하량이 증가할 수 있으며, 이로 인하여 에너지 소모량도 증가하게 됨으로써 해당 노드들의 가용시간이 줄어들게 된다.

## 3. APT기법

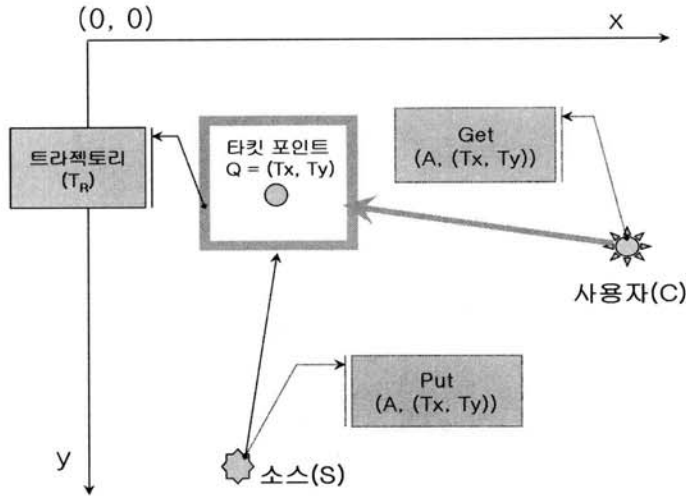
### 3.1 APT기법의 기본 개념

본 논문에서 제안하는 기법 APT(Adaptive Proxy Trajectory based Storage)의 기본 개념은 다음과 같다.

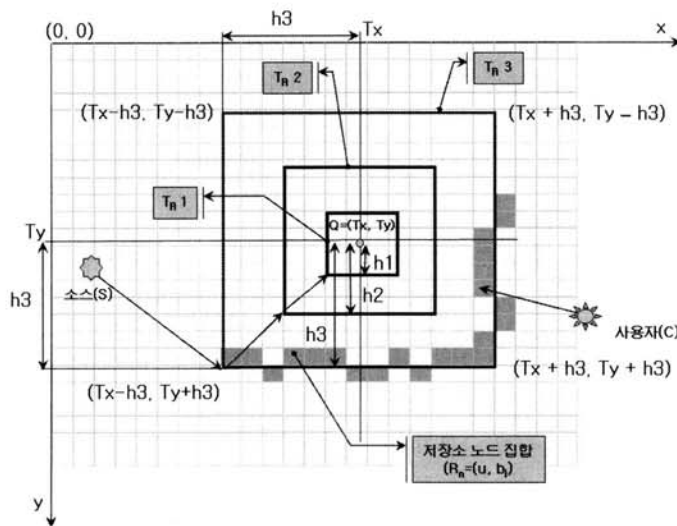
앞서 살펴본 바와 같이 GHT는 데이터 전송과 질의 전송으로 인한 오버헤드가 적지만 한 노드에 부하가 집중될 수 있다는 단점이 있다. 이런 문제를 해결하기 위하여 제안하는 기법 APT에서는 그림1과 같이 GHT를 통해 생성된 좌표인 타깃 포인트(Target Point, Q)주변에 트라젝토리(Trajectory)를 형성하여 응답율을 높이고 부하를 분산시킨다.

(그림 1)에서 데이터를 수집하는 소스노드(s)는 데이터 A를 해쉬하여 Q의 좌표값(Tx, Ty)을 얻어 해당 위치로 데이터를 Put()을 한다. 해당 데이터를 얻고자 하는 사용자도 A를 해쉬하여 좌표값(Tx, Ty)를 얻어 해당 위치로 Get()을 한다. 제안기법에서는 Q를 둘러싸는 트라젝토리(Trajectory, T<sub>R</sub>) 형성한다.

T<sub>R</sub>는 데이터 조회율과 같은 질의빈도 수로 산출된 질의



(그림 1) APT 동작 방식



(그림 2) APT에서 계층적 TR구성

빈도(popular,  $\rho$ )에 계산된 거리(Distance, H)만큼 떨어진 위치에 형성된다. 형성된  $T_R$ 에는 Q의 데이터를 저장할 수 있는 저장소 노드 집합( $R_n$ )이 있다.

저장소노드 집합( $R_n$ )은 백본노드(backbone, b)와  $T_R$ 경계선(boundary, B)사이 에 있는 노드(u)로 구성된다. 따라서 사용자는 Q까지 접근하지 않고  $T_R$ 내에 있는 저장소 노드로부터 데이터를 가져갈 수 있다.

### 3.2 계층적 트라젝토리

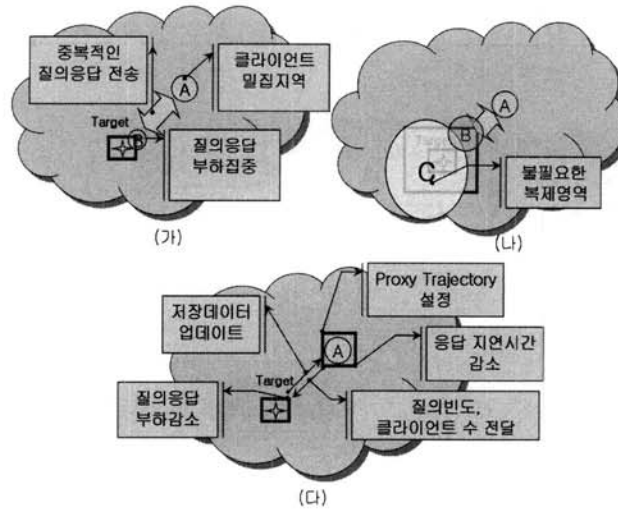
데이터를 수집하는 S는  $T_R$ 에 속하는 노드들이 취합한 데이터의 질의빈도( $\rho$ )정보를 바탕으로 H를 결정한다. 데이터의 질의빈도( $\rho$ )가 일정수준 이상 증가하면 S는 H를 증가시켜서 새로운  $T_R$ 를 형성한다. 반대로 데이터의 질의빈도( $\rho$ )가 일정수준 이하로 감소하면 S는 범위를 감소시킨다.

(그림 2)에서 S는 해쉬함수를 이용하여 데이터를 저장할 위치인 Q를 구한다. 초기의 거리 H는 기본거리  $h_1$ 으로 설

정한다. S는 Q의 좌표값( $T_x, T_y$ )와 기본거리  $h_1$ 을 Q를 향하여 전송한다. 노드들은 Q와  $h_1$ 을 가지고  $T_R$ 을 계산하여  $T_{R1}$ 을 형성한다. 일정주기 동안 취합된 데이터의 질의빈도( $\rho$ )가 일정수준 이상이면 S는  $h_1$ 을  $h_2$ 로 증가시킨다.  $h_2$ 에 의한 새로운  $T_{R2}$ 를 형성한다. 이런 방식으로  $T_{R2}$ 에서 일정 주기 동안 취합된 질의빈도( $\rho$ )가 일정 수준 이상이면 거리는  $h_3$ 로 증가하여  $T_{R3}$ 을 형성한다. 반대로 질의빈도( $\rho$ )가 일정 수준 이하로 감소하는 경우에는 거리는 다시  $h_2$ 로 감소하여  $T_{R2}$ 로 돌아간다. 데이터의 질의빈도( $\rho$ )에 따라 거리 H가 변하고 거리에 따라  $T_R$ 는 동적으로 증가하거나 감소하여 노드에 발생하는 부하를 분산시킨다.

### 3.3 프록시 트라젝토리

APT에서 프록시 트라젝토리(Proxy Trajectory,  $PT_R$ )도 일반 네트워크상의 프록시와 비슷하게 동작한다[10]. 사용자들 주변에 노드들에 자주 접근하는 데이터를 저장하여  $T_R$



(그림 3) PTR의 사용 예시

의 부하를 줄이고 사용자들의 응답지연 시간을 줄여준다.

(그림 3)의 (가)에서 네트워크상의 A지역에 사용자들이 밀집해 있는 경우  $T_R$ 의 B지역에 있는 노드들에 질의가 중복적으로 전송되어 부하 집중이 발생한다.

앞서 설명한 계층적 트라젝토리 기법을 사용하면 데이터에 대한 전체적인 인기도가 높기 때문에 소스노드는  $T_R$ 의 범위를 확장한다. 하지만 그림의 (나)에서처럼 사용자들의 질의와 이에 대한 응답은 B지역에 집중되며  $T_R$ 에서 B지역을 제외한 나머지 C영역의 노드들에는 부하가 거의 발생하지 않는다. 따라서 C지역의  $T_R$ 는 불필요하게 형성된 것이며 에너지의 낭비를 초래한다.

사용자가 특정지역에 밀집해 있는 경우 발생할 수 있는 문제점은 동일한 데이터에 대한 질의와 응답이 중복적으로 전송된다는 것이다. 이러한 중복 전송은  $T_R$ 에 속한 노드들 뿐만 아니라 이를 중계해주는 노드들의 에너지 낭비로 이어진다. 이러한 문제를 해결하기 위해서 사용자가 밀집해 있는 지역에 데이터를 저장해놓는다면 질의응답의 중복적 전송을 줄일 수 있을 뿐만 아니라 질의에 대한 응답시간도 줄일 수 있다.

그림 (다)에서처럼 사용자 밀집지역인 (A)지역을 둘러싸는  $T_R$ 를 형성하여 사용자들의 질의를 처리하도록 하면 중복적인 질의응답 횟수를 줄이면서 질의에 응답하는 시간 또한 줄어드는 효과를 볼 수 있다. 본 논문에서는 이러한  $T_R$ 를 프록시 트라젝토리(PTR)라 부른다.

### 3.3.1 특징 및 동작방식

프록시 트라젝토리의 장점은  $T_R$ 가 불필요하게 확장하는 것을 방지하고 중복적인 질의응답 메시지가 전송되는 것을 방지하며 사용자에 빠른 응답시간을 제공한다는 것이다. PTR는 프록시 서버개념을 센서네트워크에 적용하여 사용자들에게 빠른 서비스를 제공하면서 네트워크에 부하를 분산시킨다. 사용자 주변에  $T_R$ 를 생성하는 PTR는 사용자들 시스템의 중심으로 간주하는 개념이다.

TPS나 GHT와 같은 기존의 기법들은 사용자의 위치와는 상관없이 데이터를 어떻게 효율적으로 저장하고 전달할 것인가에만 중점을 두고 있다. 따라서 사용자들이 네트워크상의 특정지역에 밀집해 있더라도 데이터 저장 시 이에 대해 전혀 고려하지 않고 저장한다.

데이터가 저장되는 위치가 우연히 사용자들이 밀집해 있는 지역과 가깝다면 사용자들에 질의에 대한 응답시간은 짧을 것이다. 하지만 데이터가 사용자 밀집지역과 동떨어진 위치에 저장된다면 많은 수의 사용자들이 긴 응답시간을 가질 것이다.

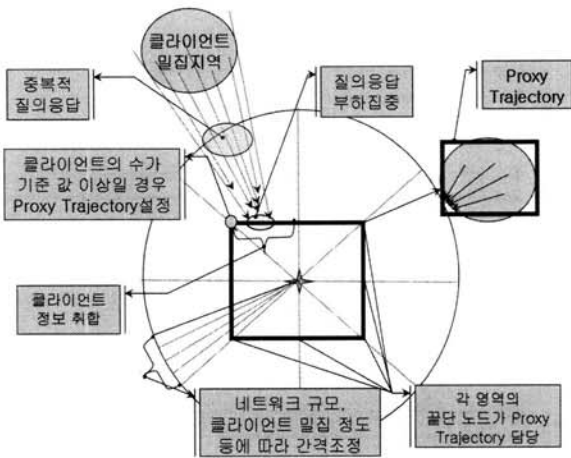
인터넷에서 인접 지역에 있는 사용자들은 비슷한 데이터를 원하기 때문에 프록시 서버를 사용하면 사용자들이 빠른 응답을 받을 수 있다[11]. 센서네트워크에서도 인접 지역에 있는 사용자들은 동일한 데이터를 원할 가능성이 높기 때문에 사용자들이 많이 원하는 데이터는 사용자들 주변에 저장하는 것이 사용자들에게 빠른 응답시간을 제공할 수 있을 것이다.

PTR는 데이터를 효율적으로 저장할 것인가에 중점을 두는 것이 아니라 사용자들에게 빠르고 효과적으로 전달할 것인가에 중점을 두는 개념이다.

PTR는 사용자의 지역별 밀도를 기반으로 형성된다. 특정 지역에 사용자들이 밀집되어 있는 경우 해당 지역을 둘러싸는 PTR를 형성한다.

지역별 사용자의 밀도는  $T_R$ 에 속하는 노드들의 정보를 취합하면 구할 수 있다. 사용자들은 저장 위치를 향해 질의를 전송하기 때문에 저장 위치를 중심으로 같은 방향에 있는 노드들이 전송하는 질의는  $T_R$ 의 같은 구간으로 전달된다. 따라서  $T_R$ 상의 특정 구간에 많은 질의가 전송된다면 해당 방향에 많은 사용자들이 밀집해 있다는 것을 확인할 수 있다.

특정구간에 질의가 집중되는 경우 해당 구간으로 질의를 전송하는 노드들의 좌표값 정보를 이용하여 사용자들이 밀집해 있는 위치를 구할 수 있다.  $T_R$ 는 위치기반 라우팅을 기반으로 하기 때문에 질의를 전송하는 노드들에 좌표값은 질의 메시지에서 구할 수 있다.



(그림 4) PTR 동작방식

APT에서는  $T_R$ 를 일정한 거리의 구간으로 나누어 구간별 사용자의 밀도를 구하여  $PT_R$ 를 형성한다. (그림 4)에서 보듯이 저장 위치를 중심으로 네트워크를 방사형으로 나누어  $T_R$ 의 구간별로 네트워크의 해당 부분을 담당하도록 한다. 네트워크의 크기와 사용자의 밀집도 등에 따라 네트워크를 나누는 간격을 조절하여  $PT_R$ 를 적응적으로 형성할 수 있도록 한다. 구간별 사용자의 밀도는 해당 구간의 끝단 노드가 취합하도록 한다. 구간의 사용자 밀집도가 기준치 이상일 경우 끝단 노드는 사용자들이 밀집해있는 중심과 범위를 계산하고 해당 위치를 중심으로  $PT_R$ 를 형성한다.

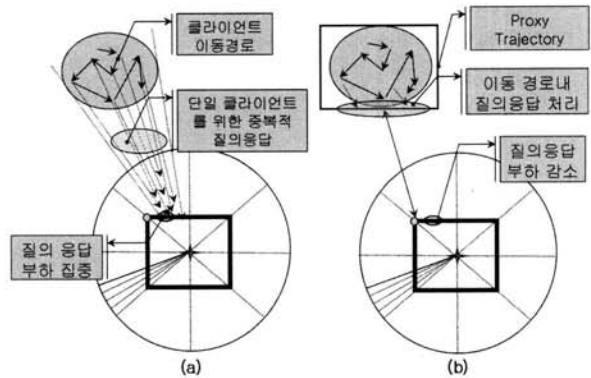
### 3.3.2 이동 사용자 지원

사용자가 특정 지역에서 이리저리 이동하면서 질의를 전송하는 경우  $PT_R$ 를 이용하여 사용자에게 데이터를 빠르게 전달할 수 있다.

사용자가 이동하면서 반복적으로 질의를 전송하는 것은 여러 사용자들이 고정 위치에서 질의를 중복적으로 전송하는 것과 비슷하게 생각할 수 있다. 사용자의 이동지역을 둘러싸는  $PT_R$ 를 형성하면 사용자는 저장 위치까지 반복적으로 질의를 전송하지 않더라도 빠르게 데이터를 전달받을 수 있다.

(그림 5)는  $PT_R$ 를 이용하여 이동하는 사용자를 지원하는 것을 보여준다. 사용자가 이동하면 메시지에 삽입되는 위치 정보가 달라지기 때문에  $T_R$ 에 속하는 노드들은 질의별로 독립적인 하나의 사용자로 처리하게 된다. 따라서 사용자가 특정지역에서 이동하면서 질의를 반복적으로 전송하면  $T_R$ 상의 특정 구역에 질의가 집중된다. 구간에 속하는 노드들은 각 질의를 서로 다른 사용자로 취급하여 많은 사용자가 질의를 전송하는 것으로 간주하며 사용자의 밀도를 높게 평가한다. 따라서 사용자의 이동지역을 중심으로  $PT_R$ 가 자연스럽게 형성된다.

$PT_R$ 를 이용하면 사용자가 저장 위치로부터 멀리 떨어져 있거나 자주 움직이는 경우 효과적일 수 있다. 사용자가 저장 위치로부터 멀리 떨어져 있는 경우 사용자가 전송한 질



(그림 5) Proxy Trajectory를 이용한 사용자의 이동성 보장

의에 대한 응답이 사용자까지 전송되는데 긴 시간이 걸릴 수 있는데  $PT_R$ 를 이용하면 사용자 주위에서 응답이 전달되기 때문에 응답지연시간을 크게 줄일 수 있다. 사용자가 자주 움직이는 경우에는 질의와 응답이 전파되는 동안에 사용자가 이동하면 응답이 사용자로 전달되지 못할 수도 있다.  $PT_R$ 를 이용하면 질의와 응답이 전달되는 시간이 짧기 때문에 데이터가 사용자로 전달될 가능성이 높아진다.

## 4. 성능평가

### 4.1 계층적 트라젝토리

#### 4.1.1 실험환경

본 논문에서 제안된 APT기법을 NS-2를 이용하여 구현하였다. 802.11 MAC과 GPSR 라우팅 프로토콜을 사용하였으며, NS-2에 구현된 에너지 모델을 사용하여 에너지 효율성을 비교하였다. 또한 어플리케이션으로 Ping 프로그램을 사용하여 Ping에 대한 응답메시지를 보내는 것을 데이터 전송하는 것으로 가정하여 실험하였다.

센서노드가 패킷을 송·수신할 때 사용하는 에너지는 각각 0.5W, 0.2W로 설정하고, 노드의 통신 거리는 30m 정도로 설정하였다. 실험은 소규모 네트워크와 대규모 네트워크 두 가지 환경에서 실시하였다. 소규모 네트워크 실험환경은 600m × 600m 크기의 영역에 20m 간격으로 900개의 노드들을 그리드 형태로 배치하였으며, 대규모 네트워크 실험환경은 2000m × 2000m 크기의 영역에 20m간격으로 10000개의 노드들을 그리드로 배치하여 광범위한 영역에 설치되는 밀도가 높은 네트워크를 모델링 하고자 하였다.

GHT를 비교대상으로 설정하였으며 데이터 저장소의 주위에만 데이터를 저장하고 응답하도록 하였다. 데이터 저장 기법의 성능을 측정하기 위하여 0번 노드가 데이터를 센싱하여 전달하는 역할을 담당하도록 하였으며 저장소는 네트워크의 중심점으로 설정하였다. 그리드 형태의 네트워크에서 태두리에 위치하는 노드들이 질의를 전송하도록 하였다. 600m × 600m크기의 네트워크에서는 총 116개의 노드들이 질의를 전송하며 2000m × 2000m크기의 네트워크에서는 총 396개의 노드들이 질의를 전송한다. 질의빈도는 시간에 따

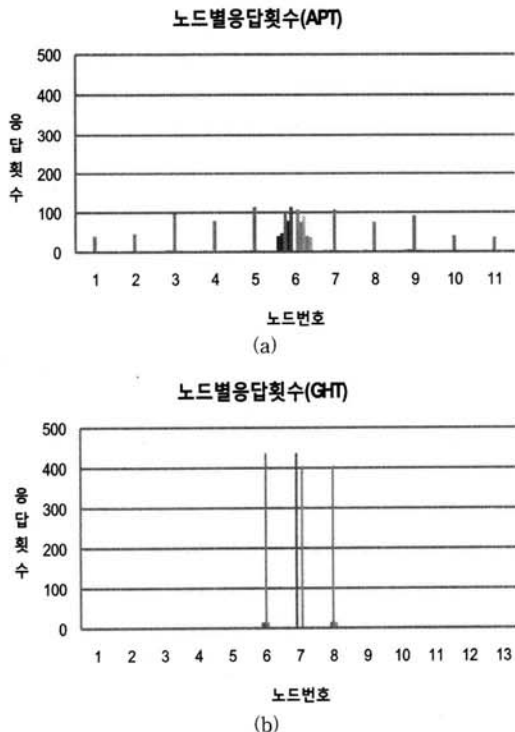
라 최소 0.5초 간격에서 최대 4초 간격 사이에서 질의 빈도를 다르게 하여 성능을 측정하였다.

4.1.2 노드별 응답 횟수

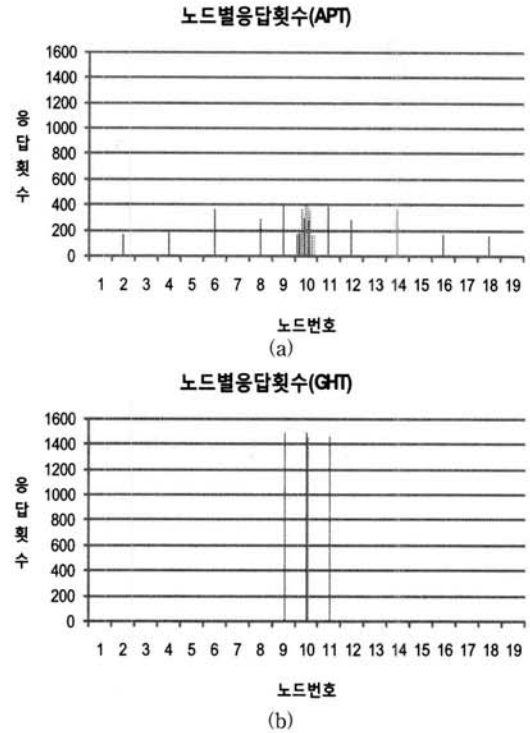
각 실험환경에서 응답횟수 0인 노드들을 제외한 노드들 대상으로 응답횟수를 나타낸 실험결과이다.

(그림 6)는 노드별로 질의를 수신하여 응답을 처리한 횟수를 보여준다. 총 600개 노드 중에서 GHT(b)의 경우 응답 노드 수 8개에 비해 제한한 APT(a)에서는 응답 노드 수 120개로 약 15배정도 응답 노드 수가 많은 것으로 나타났다. 이는 앞에서 살펴보았듯이 GHT(b)의 경우 질의빈도에 상관없이 저장소 범위가 고정적이기 때문에 질의에 대한 응답이 매우 적은 수의 노드들로 집중되는데 비해 APT(a)에서는 데이터 저장소의 범위가 동적으로 변하기 때문에 GHT (b)에 비해 많은 노드들이 응답에 참여한 것을 알 수 있다. APT에서 같은 범위에 속하는 노드들의 응답횟수가 다른 것은 노드들의 배치가 그리드 형태이기 때문에 GPSR에서 라우팅시 일부의 노드로 집중되는 현상이 발생하였기 때문이다. 노드들이 랜덤하게 분포하는 경우에는 같은 범위에 속하는 노드들은 비슷한 응답횟수를 가질 것이다.

(그림 7)은 대규모 네트워크에서 노드별로 질의를 수신하여 응답을 처리한 횟수로, 응답 횟수가 0인 노드를 제외한 노드들의 응답횟수를 그래프로 나타내었다. (그림 7)의 GHT(a)에서 질의에 대한 응답을 한 노드들은 8개로 (그림 6)과 같지만, ALT(c)에서 응답에 참여한 노드들의 수는 168개로 GHT에 비해 약 21배의 노드들이 응답에 참여한 것을 알 수 있다.



(그림 6) 노드별 응답횟수



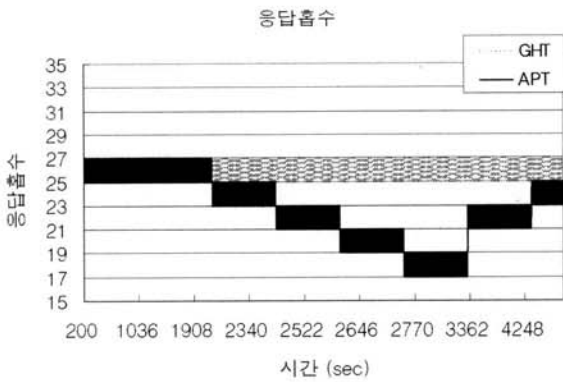
(그림 7) 대규모 네트워크 노드별 응답횟수

따라서 네트워크의 크기가 커질수록 제한한 방법이 특정 노드에 집중되는 부하를 분산시켜 네트워크의 부하분산과 응답횟수가 집중되는 노드의 에너지 소모를 줄일수 있음을 알 수 있다. 단, 위 실험에서 ALT에서 같은 범위에 속하는 노드들의 응답횟수가 다른 것은 노드들의 배치가 그리드 형태이기 때문에 GPSR에서 라우팅시 일부의 노드로 집중되는 현상이 발생하였기 때문이다. 노드들이 랜덤하게 분포하는 경우에는 같은 범위에 속하는 노드들은 비슷한 응답횟수를 가질 것이다.

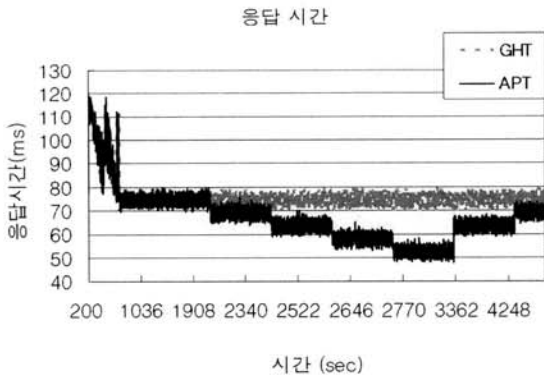
4.1.3 응답 홉 수 및 응답 시간

사용자에게 얼마나 좋은 성능을 제공하느냐를 평가하기 위해서 질의를 전송하고 응답을 받는데 걸리는 응답 시간을 측정하였다. 질의에 대한 응답시간은 질의가 데이터 저장소까지 전송되는 시간과 저장소 노드에서 질의를 처리하는 시간 그리고 응답이 질의노드까지 도달하는 시간으로 구성된다. 데이터 저장소 노드에서 질의를 처리하는 시간은 어플리케이션에 의존적일 수 있기 때문에 이는 배제하고, 실험에서는 질의를 받으면 바로 응답메시지를 전송하도록 하였다. 질의 빈도에 따라 응답시간이 변하는 원인을 확인하기 위하여 질의와 응답 메시지가 전송되면서 거치는 노드들 수를 홉 수로 측정하였다. 질의 빈도에 따라 질의와 응답 메시지가 거치는 홉 수를 보면 APT기법이 정상적으로 동작하는지 확인할 수 있다.

(그림 8)은 질의 메시지가 저장소 노드까지 가서 응답메시지가 질의노드로 돌아오는데 거치는 홉 수를 나타낸다. 그래프는 질의별 응답시간을 측정한 것이기 때문에 질의가



(그림 8) 응답 흡수



(그림 9) 노드의 응답 시간

적은 부분보다 질의가 많은 부분에 대한 데이터가 많아 시간 간격이 균등하지 않다. (그림 7)에서 2100초경에 질의빈도가 증가함에 따라 데이터 저장소의 범위가 40m에서 80m로 증가하는 것을 확인하였다. 그 결과 질의노드와 데이터 저장소간의 거리가 줄어들기 때문에 응답 흡수가 감소하는 것을 알 수 있다.

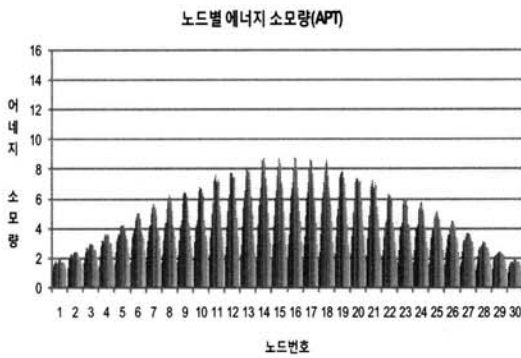
2100초부터 2800초까지 질의빈도가 급격히 증가하고 데이터 저장소 범위도 증가하기 때문에 응답 흡수도 이에 따라 감소하며 2800초 이후로는 다시 증가하는 것으로 보아 데이터 저장소에 속하는 노드들이 정상적으로 질의를 처리하는 것으로 생각할 수 있다.

아래 (그림 9)는 질의에 대한 응답시간을 보여준다. GHT의 경우 데이터 저장소의 범위가 고정적이기 때문에 질의에 대한 응답이 도달하는데 걸리는 시간이 일정한 것을 알 수 있다. 하지만 APT의 경우 질의빈도의 변화에 따라 데이터 저장소의 범위도 변하기 때문에 응답시간도 변하는 것을 확인할 수 있다.

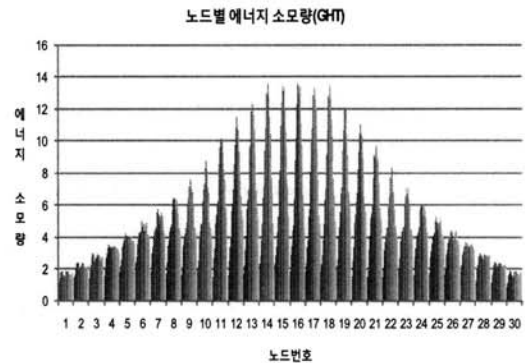
#### 4.1.4 네트워크의 노드별 에너지 소모량

(그림 10)에서 (a)와 (b)는 900개의 노드들의 에너지 소모량을 비교한 것이고, (c), (d)는 10000개의 노드들의 에너지 소모량을 제안 방식 APT와 비교 방식 GHT를 비교한 것이다.

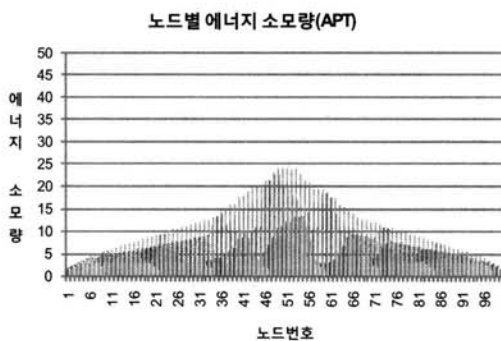
GHT(c, d)의 경우를 보면 에너지 소모가 네트워크의 중심부분에 있는 노드들에 집중되는 것을 볼 수 있다. 이는 질의처리



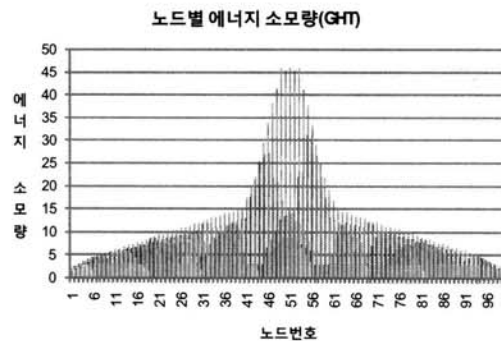
(a)



(b)



(c)



(d)

(그림 10) 노드별 에너지 소모량

가 모두 네트워크 중심에서 이루어지기 때문이다. 반면 APT (a, b)의 경우 여러 레벨에 걸쳐 노드들이 질의를 처리하기 때문에 에너지 소모량이 많이 완만해진 것을 볼 수 있다. 노드들의 수가 많으면 많을수록 GHT에서 에너지소모가 중앙에 집중된다는 것을 확연하게 보여준다. ALT와 GHT의 최대 에너지 소모량이 약 두 배 이상 차이 나는 것을 볼 수 있다.

4.2 프록시 트라젝토리(Proxy Trajectory, PTR)

4.2.1 실험 환경

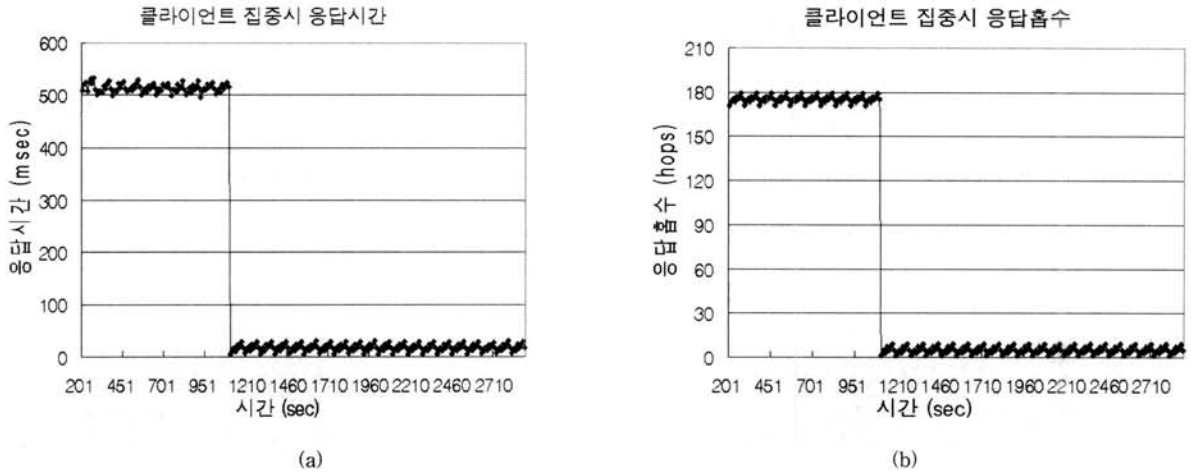
PT<sub>R</sub>의 성능은 (그림 11)과같이 사용자가 특정 지역에 밀집해 있는 경우(A)와 (그림 12)와 같이 좁은 지역에서 이동하는 경우(B)로 나눠 PT<sub>R</sub>설정 시 응답시간과 응답 홉 수의 변화를 측정하였다.

A의 경우 5000m × 5000m크기의 네트워크에 2500개의 노드를 배치하였으며 B의 경우에는 3000m × 3000m 크기의 네트워크에 900개의 노드를 배치하였다. 두 경우 모두 노드를 100m간격으로 배치하였으며 노드의 라디오 범위도 100m로 설정하여 노드 당 4개의 이웃노드를 갖도록 하였다. A

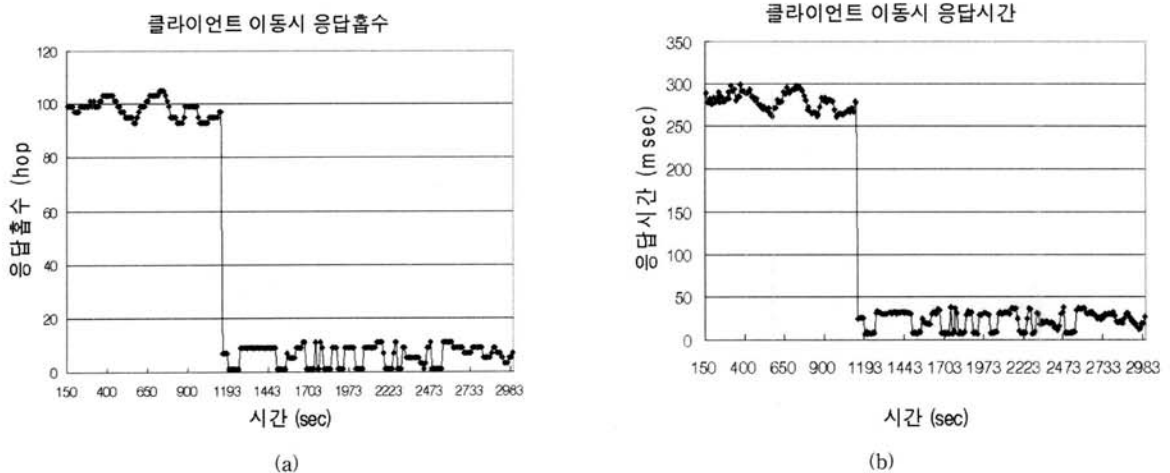
실험에서는 데이터 저장위치를 (200,200)로 설정하였으며 (4500, 4500)에서부터 (4900, 4900)사이 에 있는 노드들을 사용자 로 설정하고 일정한 간격으로 질의를 전송하도록 하였다. B 실험에서도 데이터 저장위치는 (200, 200)으로 설정하였으며 한 노드가 (4500, 4500)에서부터 (4900, 4900)사이에서 Random Way Point 이동모델에 따라 이동하도록 하였다. 노드의 속도는 1~15m/s로 설정하였으며 정지시간은 50초로 하였다.

4.2.2 사용자 집중 시(A)

(그림 11)의 실험 결과 노드들이 가로 세로로 50개씩 그리드 형태로 배치되어 있는 환경에서 양끝 모서리에서 대각선으로 패킷이 전송되기 위해서는 약90~100개 정도의 노드들을 거쳐야 한다. 사용자가 한쪽 모서리에서 대각선 방향에 있는 저장위치로 질의를 전송하여 이에 대한 응답이 사용자까지 도달하는 데에는 약180~200개 정도의 노드를 거쳐야 하므로 긴 응답시간을 필요로 한다. 하지만 사용자 주위에 PT<sub>R</sub>를 설정하면 사용자와 T<sub>R</sub>사이 에 1~5개 정도의 노드만 거치면 되므로 질의 메시지와 이에 대한 응답 메시지



(그림 11) 사용자 집중시 응답시간과 응답홉수 변화



(그림 12) 사용자 이동시 응답시간과 응답 홉 수 변화



가 돌아오는데 거치는 노드도 10개 이내로 줄어든다. 거치는 노드수의 감소는 응답시간의 단축으로 볼 수 있다.

#### 4.2.3 사용자의 이동시(B)

(그림 12)는 사용자가 일정지역에서 랜덤하게 움직이는 경우  $PT_R$ 가 형성되어 응답시간과 응답 홉 수가 줄어드는 것을 보여준다. 사용자가 이동하면 좌표값이 바뀌기 때문에  $T_R$ 에서 다수의 사용자로부터 질의가 전송되는 것으로 간주되어 이동 사용자 주위로  $PT_R$ 를 형성한다. 이는 사용자들이 밀집해 있는 영역에  $PT_R$ 가 형성되는 경우와 비슷하다.

그래프에서 응답시간과 응답 홉 수가 일정치 않은 것은 사용자가 이동하여  $T_R$ 까지의 거리가 바뀌기 때문이다. 이 실험을 통하여  $PT_R$ 가 이동 사용자에게 빠른 응답시간을 제공함을 알 수 있다.

### 5. 결 론

본 논문에서는 광범위한 영역에 배치되어 있는 센서네트워크에서 노드에 발생하는 부하를 분산시키고 사용자에게 빠른 응답시간을 제공하기 위한 APT기법을 제안하였다. APT기법은  $T_R$ 를 기반으로 데이터 저장소를 형성하도록 하여 다수의 노드에서 데이터를 저장하고 질의에 응답함으로써 부하를 분산시킨다.

실험결과와 같이 센서네트워크에서 대표적인 DCS기법인 GHT와의 성능비교를 통하여 APT기법이 질의 빈도에 따라 데이터 저장소의 범위를 조절을 통하여 응답시간을 줄이고 노드들에 발생하는 부하와 에너지소모량도 분산시킬 수 있음을 보였다. 또한 사용자가 특정영역에 밀집해 있거나 일정한 영역에서 이동하는 경우  $PT_R$ 를 통하여 매우 짧은 응답시간을 제공함을 보였다.

### 참 고 문 헌

[1] H. Henn, S. Hepper, K. Rindtorff, T. Schack and J. Burkhardt (Editor), "Pervasive Computing: Technology and Architecture of Mobile Internet Applications," Addison-Wesley Professional; 1st edition Jan., 15, 2002.

[2] D. Niculescu, "Communication Paradigms for Sensor Networks," Communications Magazine, IEEE, Vol.43, pp.116-122, Mar., 2005.

[3] K. Seada, A. Helmy, "Rendezvous Regions: A Scalable Architecture for Service Location and Data-Centric Storage in Large-Scale Wireless Networks," In Proceedings of WMAN 2004.

[4] J. B. Tchakarov and N. H. Vaidya, "Efficient Content Location in Wireless Ad Hoc Networks," In Proceedings of IEEE International Conference on Mobile Data Management (MDM'04), pp.74, 2004.

[5] S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan and S. Shenker, "GHT: A Geographic Hash-table for Data centric Storage In Sensor nets," In Proceedings of the First ACM International Workshop on Wireless Sensor Networks and Applications (WSNA), pp.78-87, Sept., 2002.

[6] Brad Karp and H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," In Proceedings of International Conference on Mobile Computing and Networking (MOBICOM'00), 2000.

[7] D. Niculescu and B. Nath, "Trajectory based forwarding and its applications," in Proceedings of the MobiCom'03, ACM/IEEE, San Diego (2003), pp.260-272. 2003.

[8] A. Capone, L. Pizziniaco and I. Filippini, A. Capone, L. Pizziniaco and I. Filippini, "A SiFT: an efficient method for trajectory based forwarding," In Proceedings of wireless Communication Systems, 2005, 2nd International Symposium on, pp.135-139, Sept., 2005.

[9] Yeonghwan Tscha, Mehmet Ufuk C and ag'layan, "Query slipping prevention for trajectory-based publishing and subscribing in wireless sensor networks," pp.1979-1991, Computer Communications, 29, 2006.

[10] W. Zhang, G. Cao and T. La Porta, "Dynamic Proxy Tree-Based Data Dissemination Schemes for Wireless Sensor Networks," ACM/Springer Wireless Networks Journal, 13:5, Oct., 2007.

[11] Y.Y. Lim, M. Messina, F. Kargl, L. Ganguli, M. Fischer and T. Tsang, "SNMP-Proxy For Wireless Sensor Network," In Proceedings of ITNG 2008, 7-9, Apr., 2008.



#### 임 화 정

e-mail : hylim@kangwon.ac.kr

1999년 2월 상지대학교 행정학과(학사)

2003년 8월 상지대학교 컴퓨터정보공학과  
(공학석사)

2007년 8월 강원대학교 컴퓨터정보통신공학과  
(박사수료)

관심 분야: 센서네트워크, 유비쿼터스, 시스템 및 보안



## 이 헌 길

e-mail : hglee@kangwon.ac.kr

1983년 2월 서울대학교 컴퓨터공학  
(공학학사)

1985년 2월 한국과학기술원(KAIST)  
전산학과(공학석사)

1993년 2월 한국과학기술원(KAIST)  
전산학과(공학박사)

1989년~1993년 삼성전자 선임연구원

1993년~1995년 정보통신부 전산관리소 전산사무관

1995년~현 재 강원대학교 컴퓨터학부 교수

관심분야: 무선 센서 네트워크, 보안 및 분산처리