

# 센서네트워크를 위한 적응적 지역 트라젝토리 기반의 데이터 저장소 기법

임 화 정<sup>†</sup> · 이 좌 형<sup>††</sup> · 양 동 일<sup>†††</sup> · 차 영 환<sup>††††</sup> · 이 현 길<sup>†††††</sup>

## 요 약

센서 네트워크에서 데이터 저장소 기법은 해쉬 테이블을 기반으로 네트워크상의 센서노드들을 저장소로 간주한다. 센서노드들은 해쉬 테이블에서 추출된 노드에 데이터를 저장하며 질의를 전송하는 사용자들도 해쉬 테이블을 이용하여 원하는 데이터가 저장된 센서노드에 접근한다. 이러한 데이터 저장소 기법의 문제점들 중 하나는 데이터를 저장하는 노드에 질의가 집중될 경우 질의에 대한 응답시간이 길어질 수 있으며 부하집중으로 에너지가 빠르게 고갈되어 네트워크 수명이 단축될 수 있다는 것이다. 본 논문에서는 적응적 지역 트라젝토리 기반의 데이터 저장소 (Data Centric Storage based on Adaptive Local Trajectory, ALT)기법을 제안한다. ALT기법에서는 데이터를 저장할 노드를 중심으로 트라젝토리를 형성하며 질의 빈도의 변화에 따라 적응적으로 트라젝토리 범위를 변화시킨다. 실험을 통하여 제안하는 ALT기법이 기존의 데이터 저장소 기법에 비해 질의 빈도에 따라 빠른 응답시간을 제공함을 보이며 질의처리를 위한 부하가 여러 노드로 분산됨을 보인다. 또한 이를 통하여 노드의 에너지 소모도 분산됨으로써 네트워크 수명을 연장할 수 있음을 설명한다.

**키워드 :** 데이터 저장소, 부하분산, 트라젝토리, 적응적, 질의 빈도

## A Data Centric Storage based on Adaptive Local Trajectory for Sensor Networks

Hwa-Jung Lim<sup>†</sup> · Joa-Hyoung Lee<sup>††</sup> · Dong-il Yang<sup>†††</sup> · Yeong-Hwan Tscha<sup>††††</sup> · Heon-Guil Lee<sup>†††††</sup>

## ABSTRACT

Sensor nodes are used as a storage space in the data centric storage method for sensor networks. Sensor nodes save the data to the node which is computed by hash table and users also access to the node to get the data by using hash table. One of the problems which the data centric storage method has is that queries from many users who are interested in the popular data could be concentrated to one node. In this case, responses for queries could be delayed and the energy of heavy loaded node could be dissipated fast. This would lead to reduction of network life time. In this paper, ALT, Data Centric Storage based on Adaptive Local Trajectory, is proposed as scalable data centric storage method for sensor network. ALT constructs trajectory around the storage node. The scope of trajectory is increased or decreased based on the query frequency. ALT distributes the query processing loads to several nodes so that delay of response is reduced and energy dissipation is also distributed.

**Key Words :** Data Storage, Load Distribution, Trajectory, Adaptive, Query Frequency

### 1. 서 론

최근 MEMS(Micro Electro Mechanical Systems)와 마이크로프로세서 그리고 무선 통신 기술의 발전으로 센서노드들을 이용하여 넓은 지역에 걸쳐 정확한 정보를 얻고자하는 센서네트워크에 대한 연구가 활발히 진행되고 있다. 이러한 센서네트워크는 새롭게 부각되고 있는 유비쿼터스 환경을

구성하기 위한 필수적인 기반구조가 되고 있다[1, 2]. 주변 환경과 유기적으로 연동되는 유비쿼터스 환경에서는 주변으로부터 정보를 얻는 과정이 실시간으로 이루어져야 한다. 따라서 외부저장소에 있는 데이터에 접근하는 것이 아니라 사용자가 센서네트워크에 직접 접속하여 데이터를 얻어오는 방식을 취하면 실시간성을 보장할 수 있다. 이를 위해서는 사용자가 센서노드들이 수집한 데이터에 효율적으로 접근하기 위한 메커니즘이 필요하다[2, 3].

네트워크상에 저장된 데이터에 용이하게 접근하기 위한 방안의 하나로 분산 해쉬 테이블(Distributed Hash Table, DHT) 개념을 기반으로 네트워크상의 노드에 데이터를 저장하고 접근하는 기법이 연구되어 왔다[4, 5]. DHT를 센서네트워크에 적용하면 사용자가 외부 저장소를 거치지 않아도

\* 2006년도 강원대학교 학술연구조성비로 연구하였음.  
<sup>†</sup> 준 회원 : 강원대학교 컴퓨터정보통신공학과 박사수료  
<sup>††</sup> 준 회원 : 강원대학교 컴퓨터정보통신공학과 박사과정  
<sup>†††</sup> 정 회원 : 한림성심대학 인터넷비즈니스과 겸임교수  
<sup>††††</sup> 정 회원 : 상지대학교 컴퓨터정보공학부 교수  
<sup>†††††</sup> 정 회원 : 강원대학교 컴퓨터학부 교수(교신저자)  
 논문접수 : 2007년 9월 21일, 심사완료 : 2008년 1월 7일

센서노드에 저장된 데이터에 쉽게 접근할 수 있다. 하지만 센서네트워크에서 노드의 ID를 가지고 해쉬 테이블(Hash Table)을 구성하면 해당 노드까지 데이터를 전송하기 위한 오버헤드가 커질 수 있다.

이러한 문제점을 해결하기 위한 방안으로 노드의 ID가 아니라 노드가 위치하고 있는 지리적 좌표(위치)정보를 활용하여 해쉬 테이블을 구성하는 방법이 있다[6]. 센서노드들은 센싱한 데이터에 해쉬를 적용하여 저장할 위치정보를 찾아 전달하면 해당 위치에 있는 노드가 데이터를 관리한다. 따라서 센서네트워크에서 원하는 데이터가 있는 사용자는 해쉬를 통하여 데이터가 저장된 위치로 질의를 전송하여 데이터를 구할 수 있다.

이는 데이터에 대한 접근용이성을 제공할 수 있지만 센서네트워크 특성상 이를 구현하는데 몇 가지 어려움이 따른다. 첫째, 센서네트워크는 불규칙하게 설치되는 경우가 많기 때문에 특정 좌표주변에 매우 많은 노드가 존재하거나 노드가 존재하지 않을 가능성이 있다. 둘째, 센서네트워크를 구성하는 센서노드들이 매우 제한적인 자원을 가지기 때문에 하나의 노드에 너무 많은 부하가 집중되면 안 된다는 것이다. 즉, 인기 있는 데이터를 저장하고 있는 노드에는 질의에 대한 응답으로 인하여 많은 부하가 발생할 가능성이 높으며 이는 노드의 에너지자원을 빨리 고갈 시키는 결과를 초래할 수 있다.

본 논문에서는 이러한 문제점들을 해결하고자 분산 해쉬 테이블기반 데이터 저장소 기법에 트라젝토리 기반 전달 (Trajectory Based Forwarding, TBF) 기법을 도입하는 적응적 지역 트라젝토리 기반 데이터 저장 기법(Data Centric Storage based on Adaptive Local Trajectory, ALT)을 제안한다[7, 8, 9, 10].

TBF기법은 수학적 함수를 이용하여 메시지가 전달될 방향과 경로를 결정하는 방법으로 여러 가지 모양의 경로를 설정할 수 있다[9]. ALT에서는 DHT를 통해 구해진 좌표 즉, 타겟 포인트(Target Point)를 둘러싸는 트라젝토리(Trajectory)를 형성하여 노드들이 트라젝토리에 속하는 노드들에 중복하여 데이터를 저장하도록 한다. 따라서 단 하나의 노드에 데이터를 저장하는 DHT에 비해 ALT기법에서는 트라젝토리에 속하는 여러 노드들이 데이터를 저장하고 질의에 응답하기 때문에 부하가 분산되는 효과를 가진다. 또한 데이터에 대한 질의빈도가 높아지는 경우 트라젝토리가 형성되는 거리를 조절하여 트라젝토리를 넓혀 부하가 분산되도록 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 결론을 맺는다.

## 2. 관련 연구 비교 분석

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 연구되어 왔다[1, 11].

Direct Diffusion 동작 방식은 관심 있는 데이터에 대한 질의가 전체 네트워크로 플러딩되고, 해당 데이터가 여러 경로를 통해 베이스스테이션으로 전송되면 그 중 제일 전송율이 좋은 경로로 데이터를 받는다[11]. 이러한 기법의 문제점은 첫째, 동일한 데이터에 대하여 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. 이러한 기법에서 센서노드들이 수집한 데이터는 외부저장소가 아닌 센서노드들에 저장되고 사용자들은 데이터를 얻기 위해 센서노드로 데이터를 요청한다[6, 12].

데이터 중심 저장소 기법의 대표적인 예로 GHT(Geographic Hash-table for DCS)기법이 있다[6]. GHT는 DHT기법을 센서네트워크에 적용한 것으로, 위치기반 라우팅 프로토콜인 GPSR 프로토콜을 기반으로 동작한다[4, 5, 6, 13]. GHT는 해쉬 함수를 통해 생성된 위치에 GPSR 라우팅 프로토콜로 좌표 상에서 제일 근접한 노드에 데이터를 저장한다. GPSR은 조밀하게 분포되어 있는 센서네트워크 환경에서 좌표를 찾을 때, 해쉬 함수를 통해 좌표를 생성한다[13]. 따라서 실제 생성되는 좌표는 실제 노드의 배치와 상관없이 생성되므로 해당 좌표에 노드가 없는 경우가 발생할 수 있다. 이러한 문제의 해결책으로 GHT는 먼저 좌표와 인접한 노드를 홈노드(Home Node)로 설정하고 데이터를 저장하도록 한다. 또한 홈노드에 저장 공간이 부족할 경우를 대비하여 계층적으로 데이터를 저장한다.

그러나 GHT에서 제시한 해결책은 첫째, 노드가 밀집해 있는 경우 다양한 경로를 통해 전달되는 질의가 홈노드(Home Node)에 전달되지 못할 가능성이 있다. 둘째, 계층적으로 데이터를 저장할 때, 저장 공간에 대한 부하분산만 고려하고 네트워크상의 부하는 고려하지 않았기 때문에 홈노드에 질의응답으로 인한 부하가 집중될 수 있다는 문제점을 지닌다.

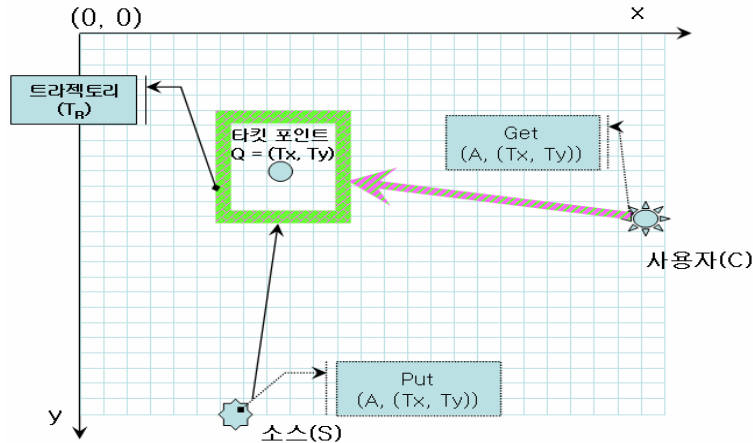
이러한 이유로 질의 빈도가 높은 데이터를 저장하고 있는 노드들이 다른 노드들에 비해 부하량이 증가할 수 있으며, 이로 인하여 에너지 소모량도 증가하게 됨으로써 해당 노드들의 가용시간이 줄어들게 된다.

## 3. ALT기법

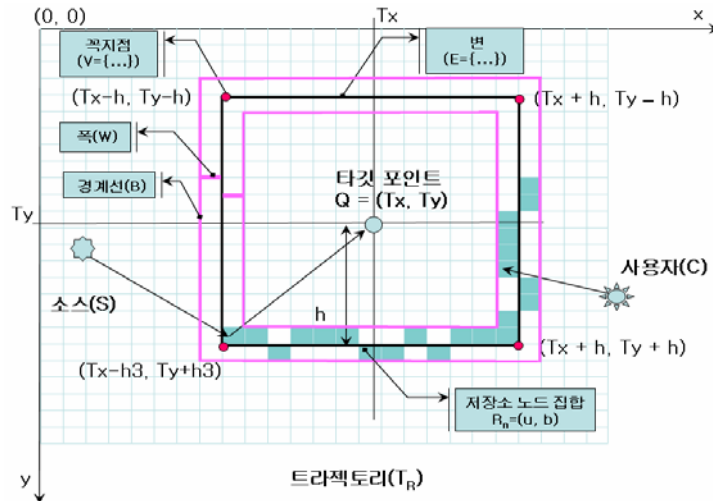
### 3.1 ALT기법의 기본 개념

본 논문에서는 GHT와 TBF 기법의 장점들을 취합하여 새로운 ALT 기법을 제시한다[6, 7, 8, 9, 10].

앞서 살펴본 바와 같이 GHT는 데이터 전송과 질의 전송으로 인한 오버헤드가 적지만, 한 노드에 부하가 집중될 수



(그림 1) ALT 동작방식



(그림 2) ALT에서의 트라젝토리( $T_R$ ) 형성과정

있다는 단점이 있다. 이런 문제를 해결하기 위하여 본 논문에서 제안하는 ALT 기법에서는 GHT를 통해 생성된 좌표인 타겟 포인트(Target Point)주변에 트라젝토리(Trajectory)를 형성하여 응답 율을 높이고 부하를 분산시킨다.

(그림 1)에서 데이터를 수집하는 소스노드(s)는 데이터 A를 해쉬하여 타겟 포인트(Target Point, Q) 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. 해당 위치와 제일 근접한 노드에 데이터를 저장하는 GHT기법과 달리 제안된 ALT 기법은 Q를 둘러싸는 트라젝토리(Trajectory,  $T_R$ ) 저장 공간을 형성한다.

트라젝토리( $T_R$ )는 해쉬함수로 구한 Q의 방향으로 질의빈도나 데이터 조회율을 나타내는 질의빈도(popular,  $\rho$ )에 따라 구해지는 거리(Distance, H)만큼 떨어진 위치에 형성된다. 형성된  $T_R$ 에는 Q의 데이터를 저장할 수 있는 저장소 노드 집합( $R_n$ )이 있다.

저장소 노드 집합( $R_n$ )은 백본노드(b)와  $T_R$  경계선(boundary,

B)사이에 있는 노드 (u)로 구성된다. 따라서 사용자는 Q까지 접근하지 않고  $T_R$ 내에 있는 저장소 노드로부터 데이터를 가져갈 수 있다.

### 3.2 트라젝토리(Trajectory)

ALT는 Q를 둘러싸는 트라젝토리(Trajectory,  $T_R$ ) 저장 공간을 형성한다. 트라젝토리( $T_R$ )를 형성하기 위해서는 타겟 포인트(Q)와 Q로부터 꼭지점 집합(V)까지의 거리(h)정보만 있으면 된다. 이 때, 노드들은 자신의 위치를 알고 있다고 가정하기 때문에 노드가 트라젝토리의 범위( $T_R$ )에 속하는지 확인할 수 있다.

트라젝토리( $T_R$ )는 해쉬 함수로 구한 Q의 방향으로 질의 빈도나 데이터 조회율을 나타내는 질의빈도(popular,  $\rho$ )에 따라 구해지는 거리(Distance, H)만큼 떨어진 위치에 형성된다. 트라젝토리( $T_R$ )에 대한 정의는 다음과 같다.

**정의 1.**  $T_R$ 의 기준 변E와 E에 속하는 백본노드(backbone) b는 다음과 같이 정의된다.

$$Q = (Tx, Ty)$$

$$V = \{(Tx-h, Ty-h), (Tx+h, Ty-h), (Tx-h, Ty+h), (Tx+h, Ty+h)\}$$

먼저, 꼭지점 집합(V)을 구성하고 있는 네 좌표 간에는 다음과 같은 식이 성립한다.

$$V = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)\}$$

각 좌표를  $\begin{cases} x_1 = Tx-h, y_1 = Ty-h \\ x_2 = Tx+h, y_2 = Ty-h \\ x_3 = Tx-h, y_3 = Ty+h \\ x_4 = Tx+h, y_4 = Ty+h \end{cases}$  치환하면  $\begin{cases} x_1 \equiv x_3, y_1 \equiv y_3 \\ x_2 \equiv x_4, y_2 \equiv y_4 \end{cases}$  이 성

립한다.

(그림 1)과 같이 네 꼭지점(Vertex, V)이 트라젝토리( $T_R$ )의 기준이 된다. V의 원소를 꼭지점으로 하는 사각형의 변(Edge) E라고 한다.

따라서 V를 연결한 변(E)의 범위는  $E = \begin{cases} ① y = y_1 \text{ and } x_1 < x \leq x_2 \\ ② x = x_1 \text{ and } y_1 < y \leq y_3 \\ ③ y = y_3 \text{ and } x_1 < x \leq x_2 \\ ④ x = x_2 \text{ and } y_1 < y \leq y_3 \end{cases}$

이다.

고로 E에 존재하는 노드를 백본노드(backbone, b)라고 한다.

**정의 2.** 트라젝토리의 전체 폭( $T_R$  Width)은 아래와 같이 정의된다.

$W = r \times t$  ( $0 \leq t \leq 1$ ) W: 트라젝토리의 폭 r: 노드의 전송범위, t: 트라젝토리( $T_R$ )의 폭(W) 조절인수

$$T_R \text{ Width} = 2 \times W$$

트라젝토리( $T_R$ )는 일정한 폭(width, W)을 유지한다. 폭(W)은 노드의 전송범위(r)와 트라젝토리( $T_R$ )의 폭(W) 조절인수(t)에 의해 구해진다. (그림 1)의 E의 양쪽 폭(W)만큼의 영역으로 형성된다.

따라서 트라젝토리( $T_R$ )의 경계선(boundary, B)은 E로부터 W만큼 떨어진 곳이다.

$$B = E \pm W$$

(그림 1)에서 E와 W사이에 있는 노드들을 u라고 한다.

**정의 3.** E와 W사이에 있는  $T_R$ 의 노드 u는 다음과 같이 정의할 수 있다.

$$u = \{D \leq W\}$$

노드로부터 E까지의 최단거리(Distance, D)가 폭(W)보다 작거나 같으면 노드 u에 속한다.

E까지의 거리 D는 각 E의 위치로부터 아래와 같이 구할 수 있다.

$$E = \begin{cases} ① y = y_1 \text{ and } x_1 < x \leq x_2 \\ ② x = x_1 \text{ and } y_1 < y \leq y_3 \\ ③ y = y_3 \text{ and } x_1 < x \leq x_2 \\ ④ x = x_2 \text{ and } y_1 < y \leq y_3 \end{cases} \quad D = \begin{cases} ① \text{의 경우 } D = |y_1 - y| \\ ② \text{의 경우 } D = |x_1 - x| \\ ③ \text{의 경우 } D = |y_3 - y| \\ ④ \text{의 경우 } D = |x_2 - x| \end{cases}$$

E에 있는 백본노드 b와 u의 집합을 저장소 노드집합(Repository,  $R_n$ )라고 한다.

**정의 4.** 저장소 노드의 집합(Repository,  $R_n$ )은 아래와 같이 정의된다.

$$R_n = (b \vee u) \quad \begin{matrix} b: \text{백본노드 집합,} \\ u: E \text{와 } W \text{사이에 있는 } T_R \text{의 노드} \end{matrix}$$

$$R_n = \{n | n \in A \wedge n = (b \vee u)\} \quad A: \text{전체노드 집합, } n: \text{저장소 노드}$$

### 3.3 계층적 부하분산

#### 3.3.1 데이터에 대한 질의빈도 수집

센서네트워크에서 수집될 수 있는 데이터의 종류는 매우 다양하며 데이터의 종류와 특성에 따라 데이터에 대한 질의빈도와 질의를 전송하는 사용자 수도 달라진다.

특정한 데이터를 원하는 사용자 수가 많을 경우 해당 데이터를 저장하고 있는 노드에 많은 질의가 전송되고 이를 처리하기 위한 부하가 해당 노드에 집중될 수 있다. 하나의 노드에 부하가 집중될 경우 해당 노드의 에너지 자원이 빠르게 소모될 수 있고, 심지어 노드가 동작을 멈추는 상황까지 초래할 수 있다. 이러한 문제를 해결하기 위해서는 데이터에 대한 질의빈도와 사용자 수와 같은 정보들을 이용하여 노드에 발생하는 부하를 분산시킬 필요가 있다.

ALT 기법에서는 두 단계에 걸쳐서 노드에 발생하는 부하를 분산한다. 첫째, 타깃 포인트(Q)를 중심으로 트라젝토리( $T_R$ ) 저장 공간을 형성하여 노드 간에 부하가 분산되도록 한다. 둘째, 데이터에 대한 조회율 즉, 질의빈도( $\rho$ )에 따라 거리(H)를 조절하여 트라젝토리( $T_R$ )를 조절하여 부하를 분산시킨다. 저장 공간인 트라젝토리( $T_R$ )의 크기에 따라  $T_R$ 에 속하는 저장소 노드의 수가 달라지기 때문에 질의빈도( $\rho$ )에 따라 범위를 조절하면 부하가 발생하는 노드의 수를 조절할 수 있다.

먼저 특정 데이터에 대한 조회수( $P_i$ )가 높아지면 이를 토대로 전체 데이터 조회수( $P_t$ )를 구할 수 있다.

**정의 5.** 노드의 전체 데이터 조회수  $P_t$ 는, 특정 데이터에 대한 전체 조회수의 합계로 나타낸다.

$$P_t = \sum_{i=1}^m P_i \quad P_i: \text{전체 데이터 조회수, } P_i: i\text{번째까지의 데이터 조회수}$$

전체 데이터 조회수  $P_t$ 를 이용하여 아래와 같은 질의빈도( $\rho$ )를 추출할 수 있다.

**정의 6.** 질의빈도( $\rho$ )는 다음과 같이 정의할 수 있다.

$$\rho = \frac{P_t}{P_m} \quad (0 \leq \rho \leq 1) \quad P_m: \text{최대(max) 데이터 조회수, } \rho: \text{질의빈도}$$

질의빈도  $\rho$ 는 전체 데이터 조회수  $P_t$ 를 서비스 가능한 최대값 즉, 최대 데이터 조회수인  $P_m$ 으로 나눈 값이 된다. 이

때,  $\rho$ 의 범위는  $(0 \leq \rho \leq 1)$ 이다.

따라서 질의빈도  $\rho$ 에 따라서 트라젝토리( $T_R$ )가 생성되는 거리( $H$ )를 아래 정의와 같이 조절 할 수가 있다.

**정의 7.** 거리  $H$ 는 다음과 같이 정의한다.

$$H = h1 + i_{max} \times \rho \quad (0 \leq \rho \leq 1) \quad h1: \text{기본 거리}, i_{max}: \text{최대팽창 길이 수}$$

따라서 트라젝토리( $T_R$ )가 생성되는 거리( $H$ )에 의해  $T_R$ 의 크기가 변하게 된다.

**정의 8.** 거리  $H$ 가 주어졌을 때,  $T_R$ 의 면적크기( $A_{Tr}$ )는 다음과 같이 정의한다.

$$A_{Tr} = A_{out} - A_{in} \quad A_{out}: \text{바깥 면적}, A_{in}: \text{안쪽 면적}$$

트라젝토리( $T_R$ )의 면적은  $T_R$ 이 생성되는 거리  $H$ 에서 생성되는 바깥쪽 폭( $W$ )까지의 면적에서 안쪽 폭( $W$ )까지의 면적을 뺀  $8WH$ 가 된다.

거리  $H$ 가 두 배 증가한  $2H$  일 경우, 트라젝토리( $T_R$ )의 면적은  $16WH$ 로 약 2배 증가한다. 즉, 트라젝토리의 면적은 거리  $H$ 에 비례하여 증가한다.

따라서 폭  $W$ 가 고정일 경우, 거리  $H$ 에 비례하여 면적이 커짐을 알 수 가 있다.

$$\begin{aligned} A_{Tr} &= (H+2W)^2 - (H-2W)^2 \\ &= H^2 + 4WH + 4W^2 - H^2 + 4WH - 4W^2 \\ &= 8WH \end{aligned}$$

즉, 면적당 노드의 밀도가 균일하다고 가정한다면,  $T_R$ 의 생성범위가 커지면 노드 수도 증가하기 때문에 노드 당 발생하는 부하 량을 균등하게 조절할 수 있다.

ALT 기법에서는 특정 데이터에 대한 조회수( $P_i$ )가 증가

하면 부하 량에 따라  $T_R$ 를 조절하기 위해서 해당 데이터에 대한 질의빈도( $\rho$ )를 산출한다. 데이터에 대한 질의빈도( $\rho$ )는 질의를 전송하는 사용자의 수와 특정 데이터에 대한 데이터 조회수( $P_i$ ), 그리고 전체 데이터 조회수( $P_t$ )로 구한다.

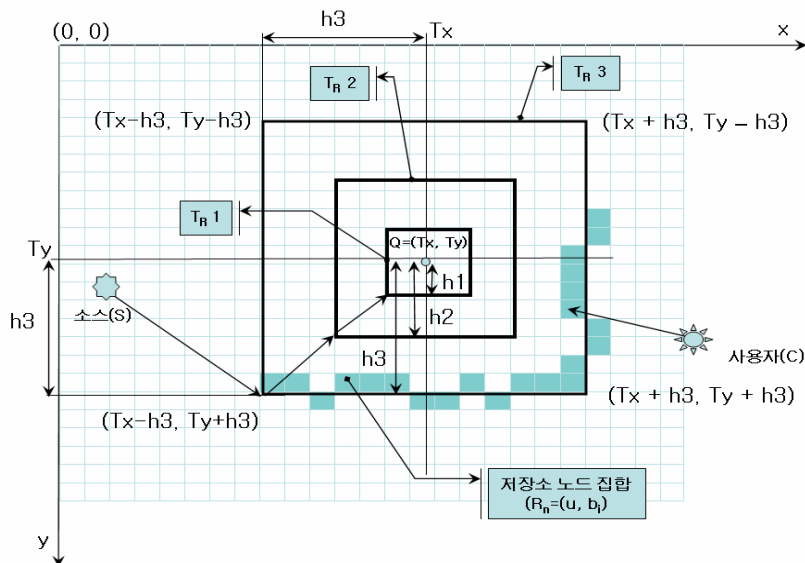
질의빈도( $\rho$ )에 대한 정보를 소스노드(S)가 취합하여  $T_R$ 를 설정한다. 소스노드(S)가 데이터를 타깃 포인트(Q)로 전송하면 Q 주위의 노드들은  $T_R$ 를 형성한다.  $T_R$ 에 속하는 노드들은 데이터를 저장하고 있다가 데이터에 대한 질의가 있으면 응답을 하면서 사용자 수와 질의 빈도 수 ( $P_i$ )에 대한 정보를 저장한다. 소스(S)에서 업데이트 정보를  $T_R$ 로 전송하면  $T_R$ 에 속하는  $R_n$ 의 저장소 노드들은 데이터를 업데이트하면서 데이터에 대한 질의빈도( $\rho$ )를 같이 전달한다.

ALT에서 데이터와 질의빈도( $\rho$ )의 흐름은 트라젝토리에 속하는 노드들이 개별적으로 질의빈도( $\rho$ )에 대한 정보를 소스노드로 전송하는 것이 아니라 여러 노드에 정보를 취합해서 전송하기 때문에 많은 오버헤드가 발생하지는 않는다. 질의빈도( $\rho$ )에 대한 정보는 사용자 수의 합, 질의 빈도의 평균 또는 분산 등으로 표현할 수 있으며 노드들이 이웃노드로 데이터를 전송하면서 질의빈도( $\rho$ )에 대한 정보도 같이 전송하여 구할 수 있다.

### 3.3.2 가변적 트라젝토리

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

(그림 3)에서 S는 해쉬함수를 이용하여 데이터를 저장할 위치인 Q를 구한다. 초기의 거리  $H$ 는 기본 거리  $h1$ 으로 설정한다. S는 Q의 좌표값 (Tx, Ty) 과 기본 거리  $h1$ 을 Q를 향하여 전송한다. 노드들은 Q와  $h1$ 을 가지고  $T_R$ 을 계산하



(그림 3) ALT에서 가변적  $T_R$  구성

여  $T_R$  1을 형성한다. 일정주기 동안 취합된 데이터의 질의 빈도( $\rho$ )가 일정수준 이상이면  $S$ 는 기본 거리를  $h_2$ 로 증가시킨다.  $h_2$ 에 의한 새로운  $T_R$  2를 형성한다. 이런 방식으로  $T_R$  2에서 일정주기 동안 취합된 질의빈도( $\rho$ )가 일정 수준이상이면 거리는  $h_3$ 로 증가하여  $T_R$  3을 형성한다. 반대로 질의빈도( $\rho$ )가 일정 수준 이하로 감소하는 경우에는 거리는 다시  $h_2$ 로 감소하여  $T_R$  2로 돌아간다. 데이터의 질의빈도( $\rho$ )에 따라 거리  $H$ 가 변하고 거리에 따라  $T_R$ 는 동적으로 증가하거나 감소하여 노드에 발생하는 부하를 분산시킨다.

### 4. 성능평가

#### 4.1 실험환경

본 논문에서 제안된 ALT기법을 네트워크 시뮬레이터 NS-2를 이용하여 구현하였다. 에너지 효율성을 비교하기 위해서 NS-2에 구현된 에너지 모델을 사용하였다. 에너지 모델에서 센서 노드가 패킷을 전송할 때 사용하는 에너지는 0.5W, 수신할 때 사용되는 에너지는 0.2W로 설정하였다. MAC계층으로는 802.11을 사용하였다. 노드의 통신 거리는 30m정도로 설정하였다. 노드간 라우팅 프로토콜로는 GPRS를 사용하였으며 어플리케이션으로는 Ping 프로그램을 사용하였다. 질의 노드에서 데이터 저장소로 Ping 메시지를 전송하면 이에 대한 응답메시지를 보내는 것으로 데이터를 전송하는 것으로 가정하였다. GHT를 비교대상으로 설정하였으며 데이터 저장소의 주위에만 데이터를 저장하고 응답하도록 하였다.

네트워크의 크기를 소규모 네트워크(a)와 대규모 네트워크(b)로 나누어서 실험을 진행하였다. (a)는 600m×600m 크기의 영역에 20m 간격으로 900개의 노드들을 그리드 형태로 배치하여 좁은 영역에 설치되는 네트워크를 모델링 하고자 하였다. (b)는 2000m X 2000m 크기의 영역에 20m간격으로 10000개의 노드들을 그리드로 배치하여 광범위한 영역에 설치되는 밀도가 높은 네트워크를 모델링 하고자 하였다. 노드의 통신 거리가 30m이고 노드 간 간격이 20m 이기 때문에 노드별로 8개의 이웃노드를 가지게 된다.

데이터 저장기법의 성능을 측정하기 위하여 0번 노드가 데이터를 센싱하여 전달하는 역할을 담당하도록 하였으며 저장소는 네트워크의 중심점으로 설정하였다. 그리드 형태의 네트워크에서 테두리에 위치하는 노드들이 질의를 전송하도록 하였다. 600m X 600m크기의 네트워크에서는 총 116개의 노드들이 질의를 전송하며 2000m X 2000m크기의 네트워크에서는 총 396개의 노드들이 질의를 전송한다. 질의 빈도는 시간에 따라 최소 0.5초 간격에서 최대 4초 간격사이에서 질의 빈도를 다르게 하여 성능을 측정하였다.

성능평가의 지표로는 사용자에게 얼마나 좋은 성능을 제공하느냐의 측면과 이 성능을 제공하는 시스템의 부하 량을 측정하였으며 최종적으로 네트워크의 에너지 소모량을 비교하였다.

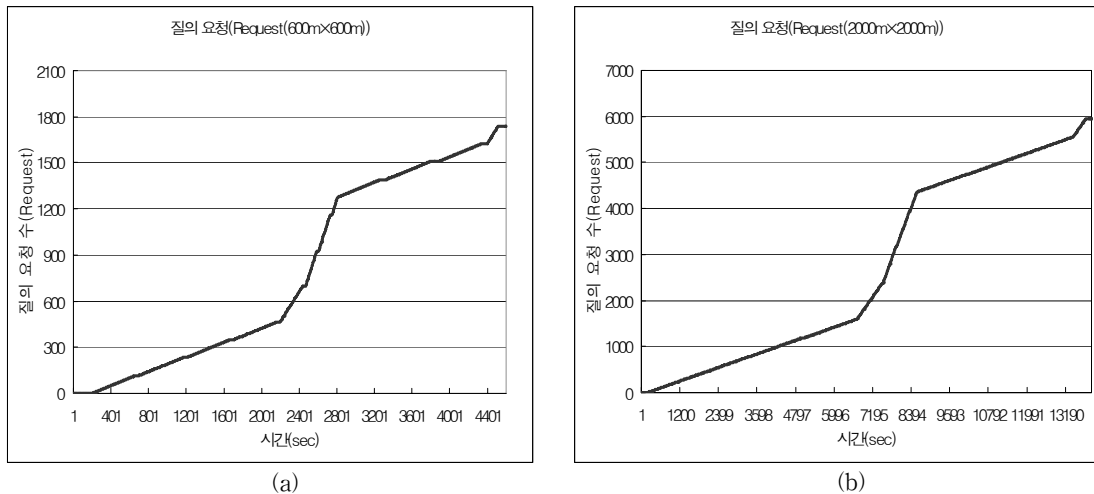
#### 4.2 질의빈도

시스템에 발생하는 부하 량은 노드별로 질의에 대해 응답하는 횟수와 데이터 저장소의 범위를 측정하였다. GHT의 경우에는 질의 빈도에 상관없이 하나의 노드에서 모든 질의를 처리하여 부하가 집중된다. 질의가 하나의 노드로 집중되면 질의를 처리하는데 걸리는 시간이 늘어나며 에너지소모량도 증가하는 단점을 가진다. ALT에서는 질의 빈도에 따라 저장소의 범위를 변화시키기 때문에 질의처리 부하가 여러 노드로 분산되는 장점을 가진다. 질의빈도에 따라 데이터 저장소가 어떻게 변하고 이에 따라 노드별로 질의를 처리하는 횟수에 변화를 비교하면 ALT의 성능을 확인할 수 있다.

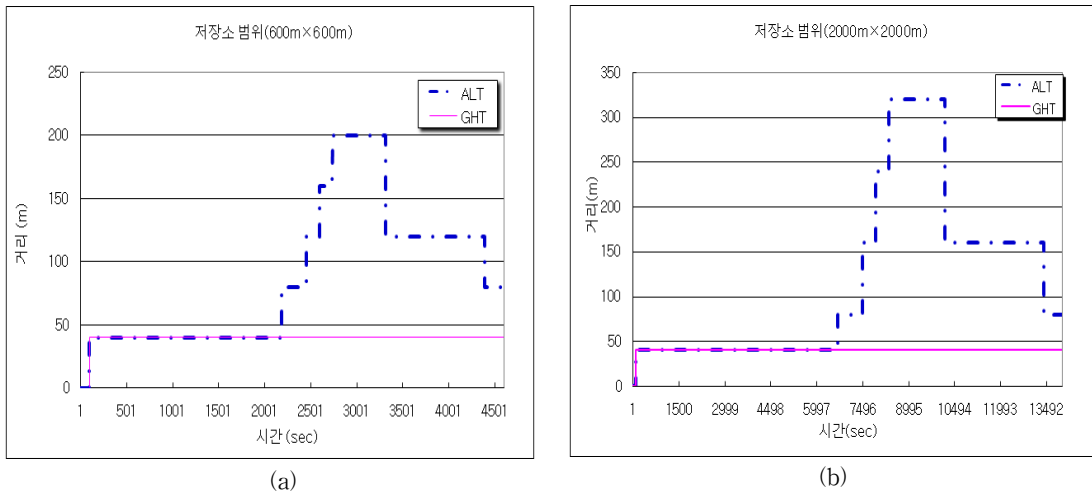
(그림 4)는 질의노드들에서 발생하는 질의빈도를 보여준다. (a)는 소규모 네트워크에서의 질의 빈도이며 (b)는 대규모 네트워크에서의 질의빈도이다. 실험에서 질의 빈도는 일정하게 증가하다가 한 시점에 급격히 증가하도록 하여 질의 빈도가 증가할 경우 ALT에서 데이터 저장소의 범위가 증가하는지 확인할 수 있도록 하였다. 급격한 증가 후에는 질의빈도를 줄여서 데이터 저장소의 범위가 감소하는 것을 확인하였다.

#### 4.3 데이터저장소의 범위와 응답횟수

(그림 5)는 질의빈도에 따라 저장소의 범위가 어떻게 변



(그림 4) 질의 빈도



(그림 5) 저장소 범위

하는지를 보여준다. GHT의 경우 질의빈도에 상관없이 저장소의 범위가 고정적이지만 ALT의 경우 질의빈도가 증가하면 이에 따라 저장소범위도 증가하는 것을 확인할 수 있다. GHT는 위치기반 라우팅인 GPSR을 기반으로 DHT로 계산된 위치에 가장 근접한 노드를 홈 노드(Home Node)를 설정하고 홈 노드에 데이터를 저장한다. 홈 노드에 장애가 발생할 경우에 대비하여 홈 노드 주변에도 데이터를 저장한다. 이는 ALT에서 트랙을 설정하는 것과 비슷하지만 DCS에서는 일정한 거리에 있는 노드가 아니라 홈 노드 주변의 임의의 노드들에 데이터를 저장한다.

실험에서는 노드 간 간격을 20m로 설정하였기 때문에 홈 노드를 중심으로 8개의 노드들이 데이터를 저장하며 범위는 40m이 된다. GHT는 홈 노드에 장애가 발생하여 새로운 홈 노드가 설정될 때까지 고정적인 범위를 갖는다. 이에 비해 ALT에서는 질의빈도가 커지면 데이터저장소의 범위를 증가시켜 많은 노드에서 질의를 처리하도록 한다. 앞의 질의횟수를 나타내는 그림을 보면 (a)의 경우 2100초정도에서 질의 빈도 증가율이 커지는 것을 볼 수 있다. 즉, 해당 시점에 저장소의 범위가 40m에서 80m로 증가하는 것을 확인할 수 있다. 이는 (a)에서 질의빈도에 따라 저장소의 범위를 40m씩 증가시키도록 설정하였기 때문에 질의빈도가 급격히 증가함에 따라 저장소의 범위도 빠르게 증가하는 것을 볼 수 있다. 질의빈도가 급격히 증가하다가 2800초 정도에서는 다시 초기와 비슷한 수준으로 감소하기 때문에 데이터 저장소의 범위도 감소하는 것을 확인할 수 있다. (b)에서는 6700초 정도에서 질의빈도가 급격히 증가하다가 4300초 정도에 감소하며 ALT도 이에 따라 데이터 저장소의 범위를 증가시키다 감소시키는 것을 볼 수 있다. (b)에서는 질의노드수가 많고 전체 노드수도 많기 때문에 (a)에 비해 데이터 저장소의 범위를 더 크게 늘리도록 설정하였다. 이에 따라 초기에는 40m에서 80m로 40m이 증가하지만 이후에는 80m씩 증가하는 것을 볼 수 있다.

(그림 6)은 소규모 네트워크에서 노드별로 질의를 수신하

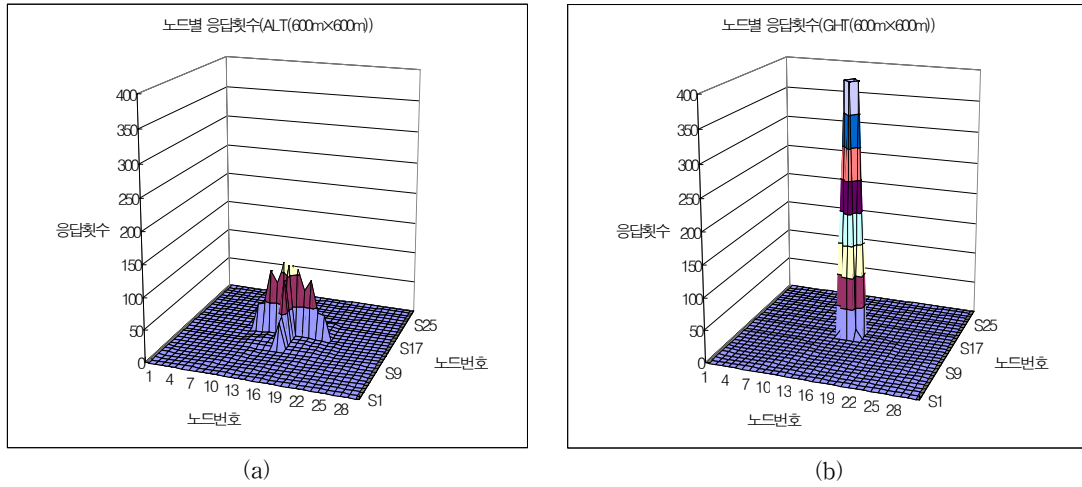
여 응답을 처리한 횟수를 보여준다. 앞에서 살펴보았듯이 GHT(b)의 경우 질의빈도에 상관없이 저장소 범위가 고정적이기 때문에 질의에 대한 응답이 매우 적은 수의 노드들로 집중되는 것을 확인할 수 있다. 이에 비해 ALT(a)에서는 데이터 저장소의 범위가 동적으로 변하기 때문에 GHT에 비해 많은 노드들이 응답에 참여한 것을 알 수 있다.

(그림 7)은 대규모 네트워크에서 노드별로 질의를 수신하여 응답을 처리한 횟수로, GHT(d)질의에 대한 응답이 매우 적은 수의 노드들로 집중되는 것에 비해, ALT(c)에서는 GHT에 비해 상대적으로 많은 노드들이 응답에 참여한 것을 알 수 있다.

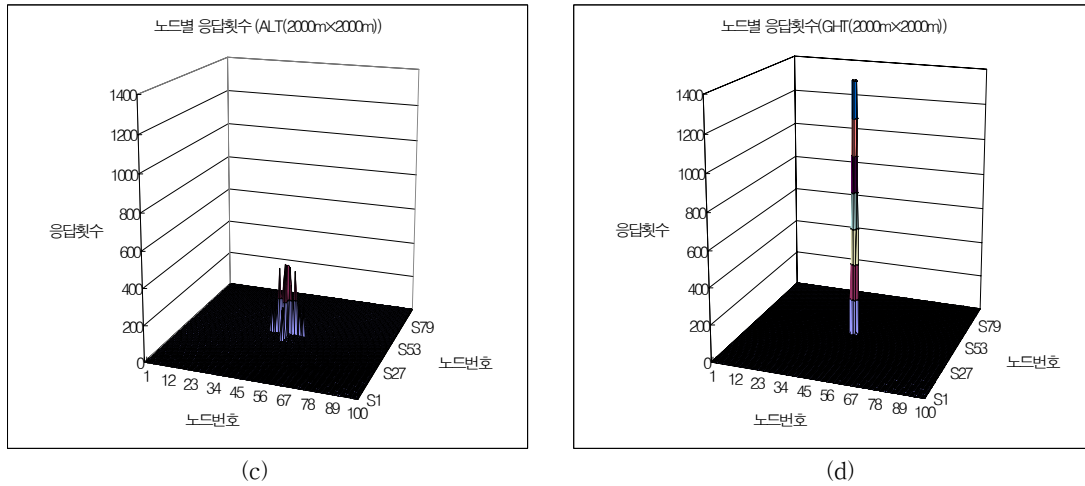
ALT에서 같은 범위에 속하는 노드들의 응답횟수가 다른 것은 노드들의 배치가 그리드 형태이기 때문에 GPSR에서 라우팅시 일부의 노드로 집중되는 현상이 발생하였기 때문이다. 노드들이 랜덤하게 분포하는 경우에는 같은 범위에 속하는 노드들은 비슷한 응답횟수를 가질 것이다.

#### 4.4 응답시간과 홉 수

사용자에게 얼마나 좋은 성능을 제공하느냐를 평가하기 위해서 질의를 전송하고 응답을 받는데 걸리는 시간과 거치는 홉 수를 측정하였다. 아무리 성능이 좋은 시스템이라도 사용자에게 결과가 전달되는 시간이 오래 걸린다면 좋은 시스템이라 하기 어렵다. 사용자의 데이터에 대한 관심도에 따라 질의 빈도수가 달라진다고 볼 수 있으며 따라서 질의 빈도수가 높아질수록 보다 빠른 응답시간을 제공할 필요가 있다. 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 노드들에 수를 홉



(그림 6) 소규모 네트워크에서의 응답 횟수



(그림 7) 대규모 네트워크에서의 응답 횟수

수로 측정하였다. 질의 빈도에 따라 질의와 응답 메시지가 거치는 홉 수를 보면 ALT기법이 정상적으로 동작하는지 확인할 수 있다.

(그림 8)은 질의 메시지가 저장소 노드까지 가서 응답 메시지가 질의노드로 돌아오는데 거치는 홉 수를 나타낸다. 그래프는 질의별 응답시간을 측정한 것이기 때문에 질의가 적은 부분보다 질의가 많은 부분에 대한 데이터가 많아 시간 간격이 균등하지 않다. (a)에서는 2100초경에 질의빈도가 증가하며 이에 따라 데이터 저장소의 범위가 40m에서 80m로 증가하는 것을 앞서 확인하였다. 그 결과 질의노드와 데이터 저장소간의 거리가 줄어들기 때문에 응답 홉 수가 감소하는 것을 위의 그림을 통해 알 수 있다. 2100초부터 2800초까지 질의빈도가 급격히 증가하고 데이터 저장소 범위도 증가하기 때문에 응답 홉 수도 이에 따라 감소하며 2800초 이후로는 다시 증가하는 것으로 보아 데이터 저장소에 속하는 노드들이 정상적으로 질의를 처리하는 것으로 생각할 수 있다. (b)에서는 데이터 범위가 80m씩 증가하기 때문에 평

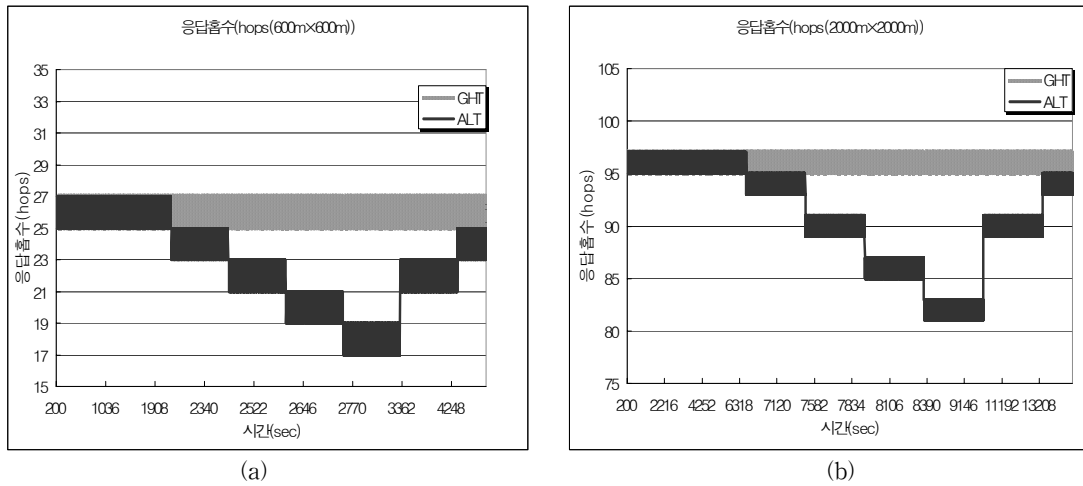
균적으로 2홉 정도씩 증가하는 것을 볼 수 있다.

위의 (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 ALT의 경우 질의빈도의 변화에 따라 데이터 저장소의 범위도 변하기 때문에 응답시간도 변하는 것을 확인할 수 있다.

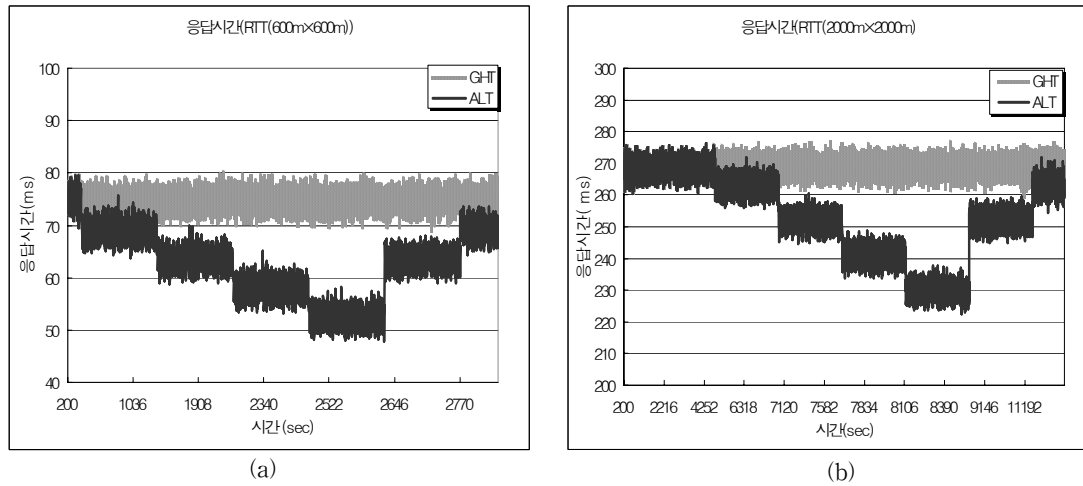
#### 4.5 에너지 소모량

센서네트워크의 노드들은 매우 제한적인 에너지자원을 가지고 있기 때문에 에너지소모량을 최소화 시키는 많은 노력들이 있어왔다. 네트워크 전체에 발생하는 부하량을 줄이는 것도 중요하지만 부하가 일부의 노드로 집중되는 것도 방지할 필요가 있다. 데이터 저장소 기법에서 메시지가 일부의 노드들로 집중될 경우 해당 노드들의 에너지가 빠르게 고갈되어 노드가 동작을 멈추는 상황이 발생할 수 있다. 네트워크의 일부 지역에 있는 노드들이 동작을 멈출 경우 해당 지





(그림 8) 응답 홑 수



(그림 9) 응답시간

역을 센싱하지 못하게 된다. GHT기법에서는 질의가 하나의 노드로 집중되지만 ALT에서는 질의빈도에 따라 여러 노드로 부하가 분산되기 때문에 에너지 소모도 분산시키게 된다.

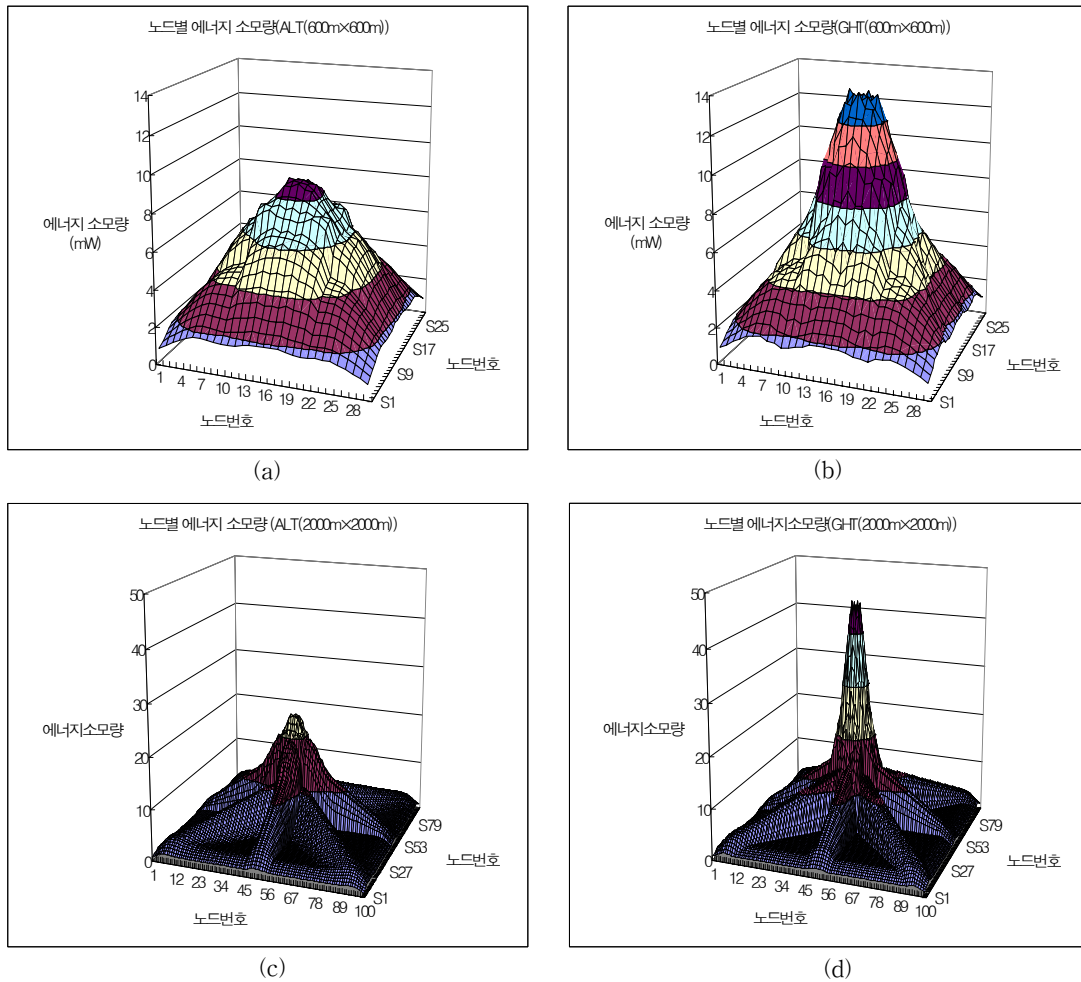
(그림 10)에서 (a, b)의 경우를 보면 GHT에서는 에너지 소모가 네트워크의 중심부분에 있는 노드들에 집중되는 것을 볼 수 있다. 이는 질의처리가 모두 네트워크 중심에서 이루어지기 때문이다. ALT에서는 여러 레벨에 걸쳐 노드들이 질의를 처리하기 때문에 에너지 소모량이 많이 완만해진 것을 볼 수 있다. (c, d)의 경우 GHT에서 에너지소모가 중앙에 집중된다는 것을 확연하게 보여준다. ALT와 GHT의 최대 에너지 소모량이 두 배 이상 차이 나는 것을 볼 수 있다.

### 5. 결 론

주변 환경에서 데이터를 수집하는 센서네트워크에는 많은 데이터들이 있으며 많은 어플리케이션에서 데이터를 많은 사용자에게 빠르게 전달할 필요가 있다. 기존의 센서네트워

크 방식들은 센서노드들이 센싱한 데이터를 베이스스테이션을 전송한 후 외부의 네트워크로 전달하는 과정을 거쳤다. 필드에서 즉각적인 데이터를 원하는 어플리케이션에는 이러한 방식은 많은 지연시간을 가지기 때문에 실시간적인 데이터를 얻기가 어렵다. 이런 문제점을 해결하기 위한 방안으로 센서네트워크상의 센서노드들을 데이터 저장소로 사용하는 기법들이 최근 제시되고 있다. 센서노드에 데이터를 저장하면 사용자가 원하는 데이터를 원하는 시점에 빠르게 제공할 수 있다는 장점을 가진다. 하지만 센서노드들은 매우 제한적인 자원을 가지고 있기 때문에 부하가 일부의 노드로 집중될 경우 에너지가 고갈되어 네트워크 수명이 빠르게 단축되는 악영향을 가져올 수 있다.

본 논문에서는 데이터 저장소 기법에 트라젝토리 기법을 접목하여 하나의 노드가 아니라 여러 노드에서 질의를 처리할 수 있도록 하고 이 트라젝토리를 상황에 적응적으로 변화시키는 ALT기법을 제안하였다. ALT에서는 DHT를 기반으로 데이터를 저장할 위치를 계산하여 해당 위치를 중심으로 트라젝토리를 형성한다. 질의를 하는 사용자들은 저장위



(그림 10) 에너지 소모량

치로 질의를 전송하며 트라젝토리에 속하는 노드들 중 질의를 수신하는 노드가 질의에 응답한다. 사용자의 질의는 시간과 상황에 따라 바뀔 수 있기 때문에 트라젝토리를 고정적으로 설정하는 것보다는 질의빈도에 따라 적응적으로 적용하는 것이 효율적이다. ALT에서는 트라젝토리에 속하는 노드들이 질의정보를 수집하여 이를 바탕으로 질의 빈도에 따라 트라젝토리의 범위를 조절하도록 한다. 질의 빈도가 증가할 경우 트라젝토리의 범위를 넓혀 부하가 여러 노드로 분산되도록 하며 질의에 대한 응답이 질의노드에 도달하는 응답시간도 줄어들도록 한다. 부하를 분산시킴으로써 에너지소모도 분산되며 이로써 센서네트워크의 수명을 연장할 수 있다. 센서네트워크에서 대표적인 DCS기법인 GHT와의 성능비교를 통하여 ALT시스템이 질의 빈도에 따라 데이터 저장소의 범위를 조절을 통하여 응답시간을 줄이고 노드들에 발생하는 부하와 에너지소모량도 분산시킬 수 있음을 보였다.

본 논문에서 어느 정도의 데이터 중복성을 이용하여 사용자에게 빠른 응답을 줄 수 있도록 데이터 저장소 중심의 ALT기법을 제안하였다. 향후 연구에서는 제안 방식을 이용하여 데이터량 또는 노드 수 등에 따라 데이터의 중복성을 최소화하는 방안에 대해 연구할 예정이다.

### 참고 문헌

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "A Survey on Sensor Networks," IEEE Communications Magazine, pp.102-114, Aug. 2002.
- [2] H. Henn, S. Hepper, K. Rindtorff, T. Schack, J. Burkhardt, "Pervasive Computing: Technology and Architecture of Mobile Internet Applications," Addison-Wesley Professional; 1st edition Jan. 15, 2002.
- [3] D. Niculescu, "Communication Paradigms for Sensor Networks," IEEE Communications Magazine, Vol.43, pp.116-122, Mar. 2005.
- [4] 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.
- [5] J. B. Tchakarov, 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.
- [6] 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.

- [7] M. D. V. Machado, O. Goussevskaia, R. A. F. Mini, C. G. Rezende, A. A. F. Loureiro, G. R. Mateus, and J. M. S. Nogueira, “Data Dissemination in Autonomic Wireless Sensor Networks,” IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, Vol.23, No.12, DEC. 2005.
- [8] A. Capone, L. Pizziniaco and I. Filippini, “A SiFT: an efficient method for trajectory based forwarding,” In Proceedings of wireless Communication Systems, 2nd International Symposium on, pp. 135-139, Sept. 2005.
- [9] M. Yuksel, R. Pradhan and S. Kalyanaraman, “An Implementation Framework for Trajectory-Based Forwarding in Ad-Hoc Networks,” In Proceedings of Vol.7 pp.4062-4066, Jun. 2004.
- [10] Y. Tscha , M. U. C, ag˘layan, “Query slipping prevention for trajectory-based publishing and subscribing in wireless sensor networks,” pp.1979-1991, JOURNAL OF Computer Communications 29 . 2006.
- [11] K. Akkaya, M. Younis, “A survey on routing protocols for wireless sensor networks,” JOURNAL OF Ad Hoc Networks, pp. 325-349, 2005.
- [12] X. Liu, Q. Huang and Y. Zhang, “Combs, Needles, Haystacks: Balancing Push and Pull for Discovery in LargeScale Sensor Networks,” In Proceedings of SenSys '04. ACM Press, New York, NY, pp.122-133, 2004.
- [13] B. Karp, H. T. Kung, “GPSR: Greedy Perimeter Stateless Routing for Wireless Networks,” In Proceedings of ACM International Conference on Mobile Computing and Networking (MOBICOM'00),2000.



**임 화 정**

e-mail : hjlim@kangwon.ac.kr  
 1999년 2월 상지대학교 행정학과(학사)  
 2003년 8월 상지대학교 컴퓨터정보공학과  
 (공학석사)  
 2007년 8월 강원대학교 컴퓨터정보통신공학과  
 (박사수료)

관심분야: 센서네트워크, 유비쿼터스, 시스템 및 보안



**이 작 형**

E-mail : jinnie4u@kangwon.ac.kr  
 2003년 2월 강원대학교 정보통신공학과  
 (공학사)  
 2005년 2월 강원대학교 컴퓨터정보통신공학과  
 (공학석사)  
 현 재 강원대학교 컴퓨터정보통신공학과  
 (박사과정)

관심분야: 멀티미디어 시스템, 센서 네트워크



**양 동 일**

e-mail : saneya@kangwon.ac.kr  
 2002년 2월 삼척대학교 컴퓨터공학과  
 (공학사)  
 2005년 2월 강원대학교 컴퓨터공학과  
 (이학석사)  
 2007년 8월 강원대학교 컴퓨터공학과  
 (이학박사)

2004년~2007년 강원대학교, 한림성심대학, 동원대학 시간강사  
 2004년~현재 안산공과대학 시간강사  
 2007년~현재 한림성심대학 인터넷비즈니스과 겸임교수  
 관심분야: 온톨로지, 유비쿼터스, 지식 표현



**차 영 환**

e-mail : yhtscha@sangji.ac.kr  
 1983년 2월 인하대학교 전자계산학과  
 (학사)  
 1985년 2월 한국과학기술원(KAIST)  
 전산학과(석사)  
 1993년 2월 인하대학교 대학원 전자계산학과  
 (박사)

1986년 3월~1987년 2월 미국 NIST(NBS) 객원과학자  
 1993년 3월~1994년 2월 서울대학교 컴퓨터신기술공동연구소  
 특별연구원  
 2004년 3월~2005년 2월 터키 Bogazici 대학교 NETLAB  
 객원교수  
 2003년 3월~2003년 12월 한국전자통신연구원 축약연구원  
 1994년 3월~2008년 1월 현재 상지대학교 컴퓨터정보공학과  
 교수

관심분야: 네트워크 구조, 통신 프로토콜, 위치정보 보호 라우팅



## 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~현재 강원대학교 컴퓨터학부 교수

관심분야: 무선 센서 네트워크, 보안 및 분산처리