

---

# GDCS : 센서네트워크를 위한 에너지 효율적인 그리드 기반 데이터 중심 저장 시스템

GDCS : Energy Efficient Grid based Data Centric Storage for Sensor Networks

---

신재룡\*, 유재수\*\*, 송석일\*\*\*

광주보건대학 의약정보관리과\*, 충북대학교 전기전자정보공학부\*\*, 충주대학교 컴퓨터공학과\*\*\*

Jae-Ryong Shin(sjr@mail.ghc.ac.kr)\*, Jae-Soo Yoo(yjs@cbnu.ac.kr)\*\*,  
Seok-Il Song(sisong@cjnu.ac.kr)\*\*\*

---

## 요약

이 논문에서는 동적으로 작업부하의 변화에 대응하는 새로운 데이터 중심 스토리지를 제안한다. 제안하는 데이터 중심 스토리지는 가상의 다중 레벨 그리드를 기반으로 구성되며 동적으로 핫스팟 영역의 부하를 분산할 수 있다. 또한, 제안하는 데이터 중심 스토리지는 기존의 GPSR (Greedy Perimeter Stateless Routing)과 같은 라우팅 기법을 적용하여 수집한 데이터를 특정 센서 노드에 저장할 수 있다. 시뮬레이션을 수행하여 기존에 제안된 데이터 중심 스토리지에 비해 데이터를 저장하는데 소요되는 에너지를 크게 줄일 수 있음을 보인다. 그리고, 제안하는 방법을 실제 센서 네트워크용 운영체제를 기반으로 구현하고 시뮬레이션 도구를 이용해서 실행한 후 제안하는 방법이 부하를 효과적으로 분산할 수 있음을 보인다.

■ 중심어 : | 데이터 중심 저장구조 | 센서 네트워크 | 색인 구조 |

## Abstract

In this paper, new data centric storage that is dynamically adapted to the change of work load is proposed. The proposed data centric storage distributes the load of hot spot area by using multilevel grid technique. Also, the proposed method is able to use existing routing protocol such as GPSR (Greedy Perimeter Stateless Routing) with small changes. Through simulation the proposed method enhances the lifetime of sensor networks over one of the state-of-the-art data centric storages. We implement the proposed method based on a operating system for sensor networks, and evaluate the performance through running based on a simulation tool.

■ keyword : | Data Centric Storage | Sensor Network | Index Structure |

---

## I. 서론

무선 센서 네트워크는 실세계에 대한 연속적이고 질

시간적인 모니터링과 상호작용을 가능하게 한다. 이런 점 때문에 무선 센서 네트워크의 응용범위는 객체 추적, 환경 모니터링, 물류관리, 교통 모니터링 등 매우 넓

---

\* 본 논문은 2005년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원(KRF-2005-202-D00395) 과 과학기술부 한국과학재단의 지원(No.R01-2006-000-10809-0)을 받아 수행된 연구입니다.

접수번호 : #090112-006

접수일자 : 2009년 01월 12일

심사완료일 : 2009년 01월 20일

교신저자 : 송석일, e-mail : sisong@cjnu.ac.kr

으며 이에 대한 관심이 매우 높다. 센서 네트워크를 구성하는 센서 노드는 응용에 따라서 지속적으로 물리적 데이터(위치, CO<sub>2</sub>, 온도, 습도 등)를 수집한다. 수집된 데이터는 응용에 따라 센서 네트워크에 저장되거나 특정 서버로 전송되어 저장 및 질의 처리 또는 분석에 사용된다.

센서 네트워크에서 센서 노드가 수집한 데이터를 처리 방법을 설계할 때는 센서 노드의 자원제약을 고려하여야 한다. 일반적으로 무선 센서 노드란 배터리에 의해 전력공급을 받으며 통신기능이 있는 상대적으로 낮은 성능을 갖는 컴퓨터이다. 센서 노드는 전력을 비롯한, 계산 능력, 저장공간 등의 자원이 제한적이다. 특히, 센서 네트워크가 설치되는 장소가 쉽게 접근할 수 없는 장소인 경우에는 소모된 전력을 재충전하거나 배터리를 교체하는 것이 매우 어렵거나 불가능하다.

센서 노드가 수집한 데이터를 처리하는 방법은 크게 두 가지로 나누어 볼 수 있다. 첫 번째는 센서 노드가 수집한 데이터를 바로 서버로 전송하여 서버가 데이터 저장 및 질의처리를 수행하는 것이다. 이 방법에서는 센서 노드가 수집한 데이터를 모두 서버로 전송해야 하므로 많은 송신비용을 필요로 한다. 센서 노드에서 전력소모가 가장 많은 부분이 바로 데이터 송신부분이므로 송신되는 데이터의 크기를 줄이기 위해 압축이나, 제거를 이용하는 방법들이 제안되었다. 두 번째는, 센서 노드가 데이터를 서버로 전송하지 않고 센서 네트워크 내부에 저장하여 질의를 처리하는 방법이다. 센서 네트워크는 각 노드가 수집한 데이터를 서버로 전송하기 위한 전력소모를 피할 수 있다. 두 번째 방법의 가장 대표적인 것이 데이터 중심 저장 구조(Data Centric Storage, 이하 DCS) 이다.

DCS는 센서노드가 수집한 데이터를 데이터의 값에 따라서 센서 네트워크의 특정 노드에 저장한다. 센서 노드가 수집한 데이터를 저장하는 노드를 소유노드라 할 때, 소유노드는 데이터의 값을 기반으로 하는 해시 함수에 의해서 결정된다. 소유노드가 결정된 데이터는 데이터를 수집한 노드로부터 소유노드까지 GPSR (Greedy Perimeter Stateless Routing)[1]과 같은 라우팅 프로토콜에 의해서 전송된다. 이렇게 하면 같은 값

을 갖는 데이터는 모두 같은 소유노드에 저장된다.

최근 제안된 DCS 들은 핫스팟(hot-spot) 문제를 해결하는데 초점을 두고 있다. 핫스팟 문제는 각 센서 노드가 수집한 데이터들 중 많은 수가 적은 수의 센서 노드로 사상(mapping)되어 저장되는 경우에 발생한다. 핫스팟문제가 발생한 지역의 소유노드들은 저장공간 제약 때문에 전송되어 데이터들 중 일부 데이터는 저장하지 못할 수도 있다. 또한, 핫스팟 영역으로 전달되는 질의 처리가 핫스팟 지역의 센서 노드와 그 주변의 센서노드에서 발생하는 충돌로 인해 지연될 수 있다. 핫스팟 지역의 센서 노드들은 질의와 저장에 대한 부하가 높아지면서 전력소모가 급격히 증가하고 이로 인해 더 이상 동작할 수 없는 센서 노드들이 증가하게 되어 센서 네트워크 전체가 마비되는 문제가 초래될 수도 있다.

DIM (Distributed Index for Multidimensional Data) [2]는 잘 알려진 DCS 중 하나이다. DIM은 K-D 트리를 기반으로 해서 여러 속성을 갖는 데이터에 대한 범 위질의를 효과적으로 처리하기 위해 제안되었다. 하지만 DIM은 핫스팟 문제에 대한 해결책을 제시하지 않고 있다. KDDCS (K-D tree based Data-Centric Storage) [3]는 DIM에서 발생했던 핫스팟문제를 해결하기 위한 방안을 제시하고 있다. KDDCS 는 DIM과 마찬가지로 K-D 트리를 기반으로 하고 있으며 핫스팟이 발행하면 부하의 균등을 위해서 K-D 트리를 재분배할 수 있도록 한다. KDDCS 의 재분배 방법은 센서노드에 저장된 데이터를 이웃하는 노드로 이동시켜야 한다. 또한, KDDCS 는 K-D 트리를 이용하여 라우팅을 수행하므로 GPSR과 비해 라우팅에 소요되는 메시지의 수가 더 많다.

DIM 과 KDDCS 외에도 다양한 DCS 들이 제안되었다. [4]에서는 ZP (Zone Partitioning)과 ZPR (Zone Partial Replication)을 제안하였다. ZP 는 핫스팟에 대한 저장의무를 보다 많은 센서 노드에게 나누어 주는 방법이며 ZPR 은 핫스팟을 이웃하는 노드에 복제하는 방법이다. DBAS (Dynamic BALanced data-centric Storage) [5]는 센서 네트워크를 위한 서버(base station) 과 센서 노드들이 공동작업을 통해서 부하를 균등하게 하는 방법을 제안하고 있다.

이 논문에서는 그리드 기반의 데이터 중심 저장시스템인 GDCS (Grid based Data Centric Storage) 를 제안한다. 제안하는 GDCS는 DIM과 KDDCS와 마찬가지로 다차원의 속성을 갖는 센싱 데이터에 대한 범위질의 처리를 효과적으로 처리 할 수 있다. 하지만, GDCS 는 K-D트리를 기반으로 하지 않고 가상의 그리드를 기반으로 데이터를 저장하고 질의를 처리한다. 또한, GDCS 는 DIM 이나 KDDCS처럼 K-D트리를 기반으로 라우팅을 하지 않고 GPSR을 일부 수정하여 적용한다. 마지막으로, GDCS는 다중 레벨 그리드를 응용한 효과적인 부하 균등 기법을 가진다.

이 논문의 구성은 다음과 같다. 2장에서는 이 논문에서 주로 참고하고 있는 DIM과 KDDCS는 물론 기존에 제안된 DCS에 대해서 보다 자세하게 살펴본다. 3장에서는 제안하는 GDCS의 구조 및 연산방법에 대해서 설명하고 4장에서는 시뮬레이션을 통해서 GDCS의 우수성을 입증한다. 마지막으로 5장에서 결론을 맺는다.

## II. 관련연구

이 장에서는 기존에 제안된 DCS들에 대해서 보다 자세하게 설명한다. 먼저 GHT (Geographic Hash Table) [6]는 센서노드가 수집한 데이터의 값을 기초로 지리적 위치를 생성하여, 그 위치에 가장 가까운 센서에 데이터를 저장하는 방식이다. 이 방법은 정합 질의(exact match query)나 접두사 질의(prefix query)는 효율적으로 수행할 수 있으나, 범위 질의(range query)를 처리하는데 있어서는 매우 비효율적이다. GHT는 유사한 값을 지리적으로 멀리 떨어진 위치에 저장할 수도 있기 때문에, 영역 질의를 처리하기 위해서는 네트워크 전역에 흩어진 여러 센서들에게 부분 질의를 전송해야 하기 때문이다.

DIM (Distributed Index for Multi-dimensional data) [2]은 센서 노드들이 온도, 습도, 조도 등 다차원의 속성을 갖는 데이터를 저장하고 질의를 처리하는 것을 효과적으로 지원하기 위한 색인구조이다. DIM은 [그림 1]처럼 2차원 평면상에 배치되는 센서의 위치에 따라서

센서 네트워크 영역을 1차원과 2차원으로 번갈아 가면서 균등하게 분할하여 하나의 영역에 하나의 센서만 남도록 분할한다. 센서 노드가 수집한 데이터는 GPSR 방식에 따라서 소유노드로 전송된다. 이 방식은 데이터가 많이 발생했을 때 외부 저장 공간에 가까운 센서들에게 부하가 집중되거나, 질의를 수행할 때 필요치 않은 센서가 질의 처리에 관여되는 것을 방지할 수 있다.

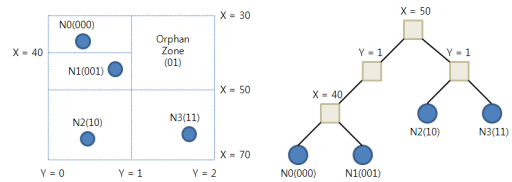


그림 1. DIM 의 구조

하지만 DIM은 범위 질의를 효율적으로 처리하기 위해 유사한 값을 가진 데이터가 동일한 혹은 지리적으로 인접한 센서에 저장되도록 한다. 이로 인해 특정 영역의 센서 노드들에 많은 양의 데이터가 집중이 되는 핫스팟문제가 발생할 수 있으며 이에 따라 데이터를 저장하지 못하게 되거나 특정 노드의 전력 소모가 급격히 증가하는 문제가 발생하게 된다. 또한, [그림 1]에서 01b 에 해당하는 영역처럼 센서 노드가 존재하지 않는 영역(Orphan Zone)이 발생한다.

KDDCS (K-D tree based Data-Centric Storage)[3]는 DIM 의 핫스팟 문제와 센서 노드가 존재하지 않는 영역(Orphan Zone) 문제를 해결하기 위해 제안되었다. KDDCS 는 DIM과 같이 K-D 트리를 기반으로 하고 있지만, [그림 2]에서처럼 영역을 분할할 때 단순하게 2등분하지 않고 센서 노드가 존재하지 않는 영역이 발생하지 않도록 임의의 지점에서 분할을 한다. 이렇게 하면 모든 영역에 센서 노드가 포함되도록 할 수 있지만, 라우팅시 DIM과는 다르게 K-D 트리의 하위노드에 대한 분할정보를 가지고 있는 상위 노드들을 방문해야 다음 노드로 이동할 수 있다. 즉, GPSR을 그대로 사용할 수 없으며 GPSR에 비해 보다 많은 메시지 전송을 필요로 하게 된다.

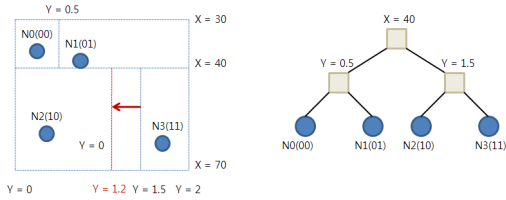


그림 2. KDDCS의 구조

KDDCS 는 분할정보를 확인한 후 다음 노드로 이동하므로 특정 센서 노드가 핫스팟으로 판명이 되면 분할 위치를 조정하여 센서 노드에 대한 부하를 이웃하는 노드로 분산할 수 있다. [그림 2]에서 N2가 핫스팟 센서 노드라면 분할위치를 Y=1.5 에서 Y=1.2 로 조정하여 Y 값이 1.2~1.5 에 해당하는 데이터를 N3에 저장하도록 하면 자연스럽게 부하가 분산된다. 하지만, 분할 위치를 조정한 후에 질의를 정상적으로 처리하기 위해서는 N2 에 저장되어 있던 Y 값이 1.2~1.5 에 해당하는 데이터를 N3로 이동시켜야 한다. 이로 인해 부하 분산을 하기 위한 비용이 다소 높은 편이다.

### III. GDCS(Grid based Data Centric Storage)

이 장에서는 논문에서 제안하는 GDCS 에 대해서 설명한다. 먼저, 센서 네트워크를 설치한 후 초기의 GDCS 를 구성하는 방법과 이후에 데이터를 삽입하는 방법, 핫스팟이 발생할 때 다중 레벨 그리드를 이용해서 해결하는 방법을 설명한다.

#### 1. 초기 GDCS 구성

이 논문에서는 제안하는 GDCS 는 방법은 기본적으로 가상 그리드를 기반으로 한다. 센서 네트워크가 설치된 영역을  $2^b$  개의 사각형 셀들로 분할한다.  $b$  는 비트의 개수로 임의의 값이다. 분할 후에는 각 셀에 길이가  $b$ 인 유일한 비트열을 할당한다. 셀에 할당된 비트열은 정수로 변환되어 각 셀에 대한 주소로 사용된다. [그림 3]에서 원안에 있는 센서 노드들이 포함된 셀에 할당된 비트열은 1111b 이며 이것은 이 셀에 포함된 센서 노드들의 주소가 된다.

각 셀에는 센서 노드가 하나도 포함이 안 되거나 하나 이상이 포함될 수 있다. 하나 이상의 노드가 포함된 셀에서는 셀의 헤더를 선출하고 셀 헤더가 셀 내의 센서 노드에 데이터를 저장하는 것을 관리하게 한다. 센서 네트워크를 설치한 후에 센서 네트워크의 영역 정보와 비트의 개수  $b$ , 센싱하는 데이터의 각 차원에 대한 속성의 도메인을 포함한 그리드 정보를 센서 네트워크의 모든 센서 노드에 브로드캐스트 한다. 그리고 센서 네트워크의 모든 센서 노드들은 자신의 위치와 이웃하는 노드들의 위치를 알고 있다고 가정한다. 각 센서 노드는 자신의 위치와 그리드 정보를 이용하여 자신의 셀 ID (CID)를 계산한다.

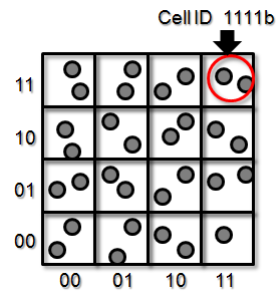


그림 3. 가상 그리드 및 셀 ID

#### 2. 데이터 삽입

센서 노드가 데이터를 수집하면 수집한 데이터의 값을 그리드 정보를 이용해서 CID로 사상한다. 데이터를 전송할 셀의 CID가 결정이 되면 GDCS는 GPSR을 이용해서 CID 셀의 중심을 향해 데이터를 전송한다. 전송하는 패킷은 (target\_CID, data, target\_XY) 로 구성되며 target\_CID 는 data 의 값에 의해 사상된 셀의 CID 이며, target\_XY는 CID 셀의 중심 좌표이다.

이미 설명한 것처럼 각 센서 노드는 기본적으로 GPSR 기법을 기반으로 소유노드가 있는 셀의 중심을 향해 패킷을 전송한다. 하지만, 셀의 중심에 항상 센서 노드가 위치하지 않고, 한 셀에는 여러 센서 노드들이 포함될 수 있으므로 GPSR을 그대로 사용하면 패킷을 효과적으로 전송할 수 없다. GDCS에서는 이를 보완하기 위해 GPSR을 일부 수정한다. GPSR 의 중개 노드들

은 다음 전송할 노드를 선택할 때 패킷의 target\_CID가 자신의 CID와 같은지를 비교한다. 만일 같으면 더 이상 다음 노드로 전송하지 않고, 자신이 소유노드가 되어서 데이터를 저장한다. 이렇게 하면 셀에 포함된 여러 노드에 데이터가 분산되는 효과를 얻을 수 있을 뿐만 아니라, CID의 중심에 센서 노드가 위치하지 않을 때도 가장 근접한 센서 노드를 찾기 위해서 불필요한 메시지를 교환하지 않아도 된다.

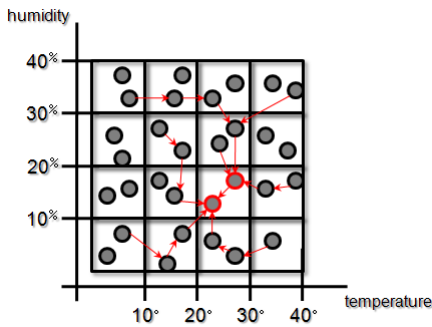


그림 4. GDCS에 데이터 삽입

[그림 4]의 예를 보면, 온도가 20°C ~ 30°C이면서 습도가 20% ~ 30% 인 데이터를 수집한 센서 노드들은 사상을 통해 target\_CID를 1001b로 결정한다. 수집된 데이터는 (1001b, data, target\_XY) 패킷으로 만들어져서 GPSR에 의해 1001b의 중심으로 라우팅 된다. 그림에서 보는바와 같이 1001b의 중심에는 센서 노드가 존재하지 않는다. 하지만, 각 센서 노드들은 자신의 CID와 패킷의 1001b를 비교하므로 1001b 셀에 포함된 센서 노드들 중 가장 먼저 패킷을 수신하는 노드가 데이터를 저장하게 된다. 그림에서는 7개의 패킷이 1001b 셀에 포함된 센서 노드 두 개에 골고루 나뉘어 저장된다.

이 방법을 이용하면 간단하게 수집한 데이터를 해당 센서노드에 저장하고 질의를 처리할 수 있다. 질의처리역시 질의의 술어를 사상해서 CID 들로 변환한다. 사상된 CID들에 해당하는 셀의 중심으로 질의를 전송한다. 만일 어떤 셀에 센서노드가 2개 이상이라면 질의는 셀 헤더로 전송되어 질의가 처리된다.

### 3. 다중 레벨 그리드 기반의 GDCS

제안하는 방법은 핫스팟 문제를 해결하지 못한다. 많은 수의 데이터가 특정 영역의 셀로 사상되면 해당 셀의 센서 노드들은 과부하가 걸리게 되어 핫스팟 문제가 발생한다. GDCS에서는 핫스팟 문제를 해결하기 위해서 다중 레벨의 그리드 기법을 이용한다. 즉, 센서 네트워크가 설치된 영역에 단일 레벨의 그리드가 아니라 여러 레벨의 그리드를 계층적으로 설치한다. [그림 5]의 예를 보면 두 레벨의 그리드가 생성되어 있다. 레벨 2는 b를 4로 한 그리드를 레벨 2라 하고, b가 2인 그리드를 레벨 1로 하여 계층적으로 구성되어 있다. 레벨 1에서 전체 영역을 4개의 하위 영역(셀)으로 나누고 각 영역(셀)을 다시 4개의 하위 영역(셀)으로 나눈다. 그리고 각 레벨에서 셀의 CID는 해당 셀이 속한 영역의 하위 레벨의 CID들을 연결하여 만들어 낸다. [그림 5]에서 빗금이 쳐진 셀의 CID는 레벨 1의 CID 011b와 레벨 2의 10b를 연결하여 0110b가 된다.

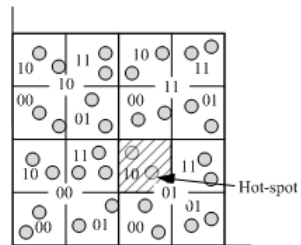


그림 5. 다중 레벨 그리드를 이용한 GDCS

GDCS 초기 구성 시에는 가장 높은 레벨의 그리드를 기반으로 구성한다. 하지만, 특정 셀이 핫스팟으로 판명되면 그 셀과 이웃하는 셀들을 한 레벨 낮은 그리드의 셀로 병합한다. [그림 5]에서 레벨 2의 0110b 셀(빗금친 셀)이 핫스팟 영역이라고 하자. GDCS는 핫스팟이 발생한 셀에 포함된 센서 노드들의 부하를 분산하기 위해서 이웃하는 셀들 (0100b, 0101b, 0100b)과 핫스팟 셀을 병합하여 [그림 6]과 같이 레벨 1의 셀 01b를 생성한다.

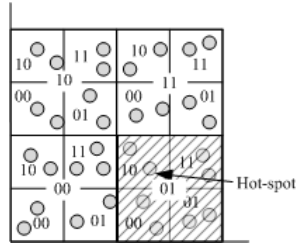


그림 6. 셀의 병합

10b 셀에 포함된 센서 노드들은 0110b 셀의 센서 노드들에 저장되어야 할 데이터들에 대한 저장 책임을 분담한다. 동시에 10b 셀에 포함되는 센서 노드들의 CID는 10b로 할당된다. 이후에는 각 셀 노드는 레벨 1 CID와 레벨 2 CID를 동시에 갖게 된다. 병합 후에 GDCS에 데이터를 삽입하기 위해 패킷을 전송할 때는 패킷의 CID와 각 센서 노드들의 각 레벨별 CID를 비교한다. 가장 높은 레벨의 CID부터 비교해 가면서 일치하는 것을 찾는다. 일치하는 CID를 갖는 센서 노드가 있으면 데이터를 저장한다.

패킷의 CID보다 비교하려는 CID의 레벨이 높을 때는 패킷의 CID를 비교 대상 CID의 레벨로 변환한 후에 비교한다. 예를 들면, 센서 노드의 레벨 1의 CID가 01b이고 레벨 2의 CID가 0110b라고 하자. 전송되는 패킷의 CID가 0100b 이라면 레벨 2의 CID 0110b 과 먼저 비교를 하고 일치하지 않으므로 레벨 1의 CID 01b로 변환한 후에 비교를 수행한다.

병합을 한 셀에는 하나 이상의 센서 노드들이 포함되므로 셀 헤더를 선출하게 된다. 셀 헤더는 해당 셀에 포함되는 센서 노드들에 대한 정보를 가지고 있으며, 질의를 처리할 때 질의를 셀 내의 모든 센서 노드들에 전송하여 질의 결과를 수집한 후 질의를 내린 센서 노드로 전송하는 역할을 수행한다. 병합된 셀에 대해서는 질의를 처리할 때 보다 많은 메시지 전송을 필요로 하게 된다. 하지만, 병합된 셀 내에는 유사한 데이터들이 저장되므로 범위질의를 처리하는데 소요되는 비용은 크지 않을 것으로 예상할 수 있다.

#### IV. 성능평가

이 논문에서는 GDCS의 성능을 기존에 제안된 KDDCS와 비교하기 위해서 시뮬레이션을 수행하였다. 시뮬레이션 파라미터는 [표 1]과 같다. 노드의 개수는 400개로 하였으며 센서 네트워크가 설치되는 전체 영역은 가로 세로 각각 200m로 가정하였다. 각 센서 노드의 초기 에너지 레벨은 1000으로 하였고, 메시지를 전송하는데 0.1이 소모되고 메시지를 수신하는 데는 0.05가 소모된다고 가정하였다. 마지막으로 각 센서 노드의 라디오 범위는 15m로 가정하였다.

표 1. 시뮬레이션 파라미터

노드 개수	400
센서 네트워크 설치 영역	200 x 200 m <sup>2</sup>
노드의 초기 에너지	1000 units
메시지 송신에 필요한 에너지 소모	0.1 units
메시지 수신에 필요한 에너지 소모	0.05 units
저장 용량	100 units
라디오 범위	15m

모든 노드는 정규분포의 데이터를 100개씩 수집하여 GDCS와 KDDCS 방식으로 저장한다. 한 노드가 수집하는 전체 데이터의 80%는 20%의 범위에 분포한다. 시뮬레이션에서는 GDCS와 KDDCS에서 각 센서 노드의 평균 에너지 레벨을 측정하였다. 또한, 두 방법에 대해서 핫스팟에 해당하는 센서 노드들의 평균 에너지 레벨을 측정하여 비교하였다.

[표 2]는 시뮬레이션을 수행한 후 각 노드들의 평균 에너지 레벨과 핫스팟 영역의 센서 노드들의 에너지 레벨을 비교한 것이다. 표에서 볼 수 있는 것처럼, 각 센서 노드의 평균 에너지 레벨은 KDDCS가 제안하는 GDCS에 비해서 약 73% 정도 더 소모된 것을 알 수 있다. 특히, 핫스팟 영역에서는 KDDCS가 GDCS에 비해서 약 97% 더 소모된 것을 볼 수 있다.

GDCS보다 KDDCS가 더 많은 에너지를 소모하는 원인은 다음과 같이 분석해 볼 수 있다. KDDCS는 다음 노드로 패킷을 전송하기 위해서 영역 분할정보를 저장하고 있는 센서 노드를 먼저 방문한 후 다음 노드로

전송을 해야 하므로 GDCS에 비해 더 많은 메시지를 전송한다. 또한, KDDCS는 부하를 분산하기 위해서 센서 노드간 데이터의 이동을 필요로 하기 때문에 상대적으로 많은 에너지를 소모하게 된다.

표 2. KDDCS와 GDCS의 에너지 레벨 비교

에너지 레벨	KDDCS	GDCS
전체 노드 평균	752.38 units	857.69 units
핫스팟 영역 노드 평균	331.13 units	611.01 units

또한, GDCS를 센서 노드용 운영체제인 TinyOS[7]를 기반으로 구현한 후 TOSSIM[8]에서 수행하여 부하 분산 효과를 살펴보았다. 부하 분산 효과를 살펴보기 위해 핫스팟영역의 센서 노드가 수신하는 메시지의 양이 다중 레벨의 그리드를 이용할 때 얼마나 줄어들었는지를 측정하였다. TOSSIM 상에서 실행시킬 때의 파라미터는 [표 3]과 같다. 전체 노드의 개수는 256개로 하였으며 센서 네트워크가 설치된 영역은 가로 세로 각각 100m 로 하였다. 라디오 범위는 10m 로 하고 그리드를 만들기 위한 비트의 개수는 8로 하였다.

표 3. TOSSIM 파라미터

노드 개수	256
센서 네트워크 설치 영역	100 x 100 m <sup>2</sup>
라디오 범위	10 m
b (비트의 개수)	8

구현한 GDCS를 TOSSIM을 기반으로 수행한 후에 핫스팟 영역의 센서 노드들이 수신하는 메시지가 전체 메시지에 비해 얼마나 되는지 측정하였다. 첫 번째 실행에서는 다중 레벨 그리드를 적용하지 않았고, 두 번째 실행에서는 레벨 2의 다중 레벨 그리드를 이용하였다. [표 4]에서 알 수 있는 것처럼, 하나의 레벨로 그리드를 구성했을 때는 핫스팟 영역의 센서 노드들이 수신하는 메시지의 비율은 전체를 기준으로 40% 였다. 하지만, 두 개의 레벨에 걸쳐 그리드를 구성했을 때는 동일한 핫스팟 영역의 센서 노드가 수신하는 메시지의 비율이 전체기준의 21%로 반 정도 줄어드는 것을 볼 수

있다. 이 실험을 통해서 다중 레벨 그리드를 적용한 GDCS가 핫스팟에 대한 부하를 효과적으로 분산함을 알 수 있다.

표 4. 핫스팟 영역의 메시지 수신율

	GDCS (Level 1)	GDCS (Level 2)
메시지 비율	40%	21%

## V. 결론

이 논문에서는 다중 레벨 그리드 기반의 데이터 중심 저장시스템인 GDCS를 제안하였다. 제안한 GDCS는 핫스팟이 발생하는 영역에 대해서 레벨을 변경시키면서 동적으로 부하를 분산할 수 있으며 추가적인 데이터의 이동이 불필요하다. 또한, 기존 제안된 KDDCS와는 다르게 부하 분산이 가능하면서도 기존의 GPSR을 이용해서 데이터 전송이 가능하다. 이 논문에서는 시뮬레이션을 통해 제안하는 GDCS가 기존의 KDDCS에 비해 에너지 소모가 핫스팟에서는 97%까지 줄일 수 있음을 보였다. 마지막으로, GDCS를 실제 센서 네트워크 운영체제인 TinyOS를 기반으로 구현하고 TOSSIM상에서 실행하여 다중 레벨 그리드 방식이 핫스팟 영역의 부하를 적절히 분산할 수 있음을 보였다.

## 참고 문헌

- [1] B. Karp and H. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," Proc. of International Conference on Mobile Computing and Networking, pp.243-254, 2000.
- [2] X. Li, Y. J. Kim, R. Govindan, and W. Hon, "Multi-dimensional Range Queries in Sensor Networks," Proc. of SenSys, pp.63-75, 2003.
- [3] M. Aly, K. Pruhs, and P. K. Chrysanthis, "KDDCS: a Load-balanced in-network Data-centric Storage Scheme for Sensor

Networks," Proc. of CIKM, pp.317-326, 2006.

[4] M. Aly, N. Morsillo, P. Chrysanthis, and K. Pruhs, "Zone Sharing: a Hot-spots Decomposition Scheme for Data Centric Storage in Sensor Networks," Proc. of International Workshop on Data Management for Sensor Networks, pp.21-26, 2005.

[5] Y. Lai, H. Chen, and Y. Wang, "Dynamic Balanced Storage in Wireless Sensor Networks," Proc. of International Workshop on Data Management for Sensor Networks, pp.7-12, 2007.

[6] P. Levis, *TinyOS Programming*, csl.stanford.edu/~pal/, 2006.

[7] P. Levis and N. Lee, *TOSSIM: A Simulator for TinyOS Networks*, www.cs.berkeley.edu/~pal, 2003.

- 1995년 3월 ~ 1996년 8월 : 목포대학교 전산통계학과(전임강사)
- 1996년 8월 ~ 현재 : 충북대학교 전기전자컴퓨터공학부 및 컴퓨터정보통신연구소 교수

<관심분야> : 데이터베이스시스템, 정보검색, 센서네트워크 및 RFID, 멀티미디어데이터베이스, 분산객체컴퓨팅

저 자 소 개

신 재 룡(Jae-Ryong Shin)

정회원



- 1996년 2월 : 충북대학교 정보통신공학과(공학사)
  - 1998년 2월 : 충북대학교 정보통신공학과(공학석사)
  - 2001년 2월 : 충북대학교 정보통신공학과(공학박사)
  - 2003년 3월 ~ 현재 : 광주보건대학 의약정보관리과 교수
- <관심분야> : 실시간데이터베이스, 내용기반검색

유 재 수(Jae-Soo Yoo)

중신회원



- 1989년 2월 : 전북대학교컴퓨터공학과(공학사)
- 1991년 2월 : 한국과학기술원 전산학과(공학석사)
- 1995년 2월 : 한국과학기술원 전산학과(공학박사)

송 석 일(Seok-II Song)

정회원



- 1998년 2월 : 충북대학교 정보통신공학과(공학사)
  - 2000년 2월 : 충북대학교 정보통신공학과(공학석사)
  - 2003년 2월 : 충북대학교 정보통신공학과(공학박사)
  - 2003년 8월 ~ 현재 : 충주대학교 컴퓨터공학전공 교수
- <관심분야> : 데이터베이스 시스템, 센서 네트워크, 이동객체데이터베이스