

# 센서 네트워크에서 저장 공간의 활용성과 에너지 효율성을 위한 시간 매개변수 기반의 데이터 중심 저장 기법

(A Time-Parameterized Data-Centric Storage Method for Storage Utilization and Energy Efficiency in Sensor Networks)

박 용 훈 <sup>†</sup>      윤 종 현 <sup>†</sup>      서 동 민 <sup>\*\*</sup>      김    준 <sup>\*\*\*</sup>      유 재 수 <sup>\*\*\*\*</sup>  
 (Yong Hun Park)    (Jong Hyun Yoon)    (Dong Min Seo)    (June Kim)    (Jae Soo Yoo)

**요 약** 최근 무선 센서 네트워크 환경에서 감지되는 데이터를 네트워크상에 효과적으로 저장하고 처리하기 위한 다양한 기법들이 제안되었다. 데이터 중심 저장(Data-Centric Storage : DCS) 기법은 분할된 데이터 영역을 센서들에게 할당하고, 수집된 데이터를 해당 데이터 영역을 담당하는 센서에 저장하는 기법이다. 기존 연구된 DCS 기반 기법들은 데이터 전송을 위한 통신비용의 감소와 정합 질의(Exact Query) 및 영역 질의(Range Query)의 효과적인 처리를 위해 제안되었다. 하지만 방대한 데이터가 발생하는 데이터 영역을 담당하는 센서의 저장 부하 문제는 고려되지 않았다. 최근 이와 같은 저장 부하 문제를 해결하기 위해 K-D 트리를 기반으로 동적 분할 데이터 영역을 사용하는 KDDCS가 제안되었다. 하지만 KDDCS를 포함한 모든 기존 DCS 기반 기법들은 인기 있는(많은 질의 요청을 받는) 데이터 영역에 대한 질의 부하가 고려되지 않았기 때문에, 결과적으로 전체 센서 네트워크의 수명이 단축된다. 그래서 본 논문에서는 저장 부하뿐만 아니라 질의 부하도 효과적으로 해결할 수 있는 새로운 DCS 기반 기법인 TPDCS(Time parameterized DCS)를 제안한다. 제안하는 기법은 데이터 차원뿐만 아니라 시간 차원을 함께 고려하여 센서에 할당되는 데이터 영역을 분할한다. 제안하는 방법은 데이터 저장과 질의 처리 분산을 통해 전체 센서 네트워크의 수명을 크게 연장시킨다. 또한 본 논문에서는 대표적인 기존 DCS기반 기법들과 다양한 성능 비교 평가를 통해 제안하는 기법의 우수성을 보인다.

**키워드** : 센서 네트워크, 데이터 중심 저장, 시간 매개변수, 에너지 효율

**Abstract** In wireless sensor networks, various schemes have been proposed to store and process sensed data efficiently. A Data-Centric Storage(DCS) scheme assigns distributed data regions to sensors and stores sensed data to the sensor which is responsible for the data region overlapping the data. The DCS schemes have been proposed to reduce the communication cost for transmitting data and process exact queries and range queries efficiently. Recently, KDDCS that readjusts the distributed data regions dynamically to sensors based on K-D tree was proposed to overcome the storage hot-spots. However, the existing DCS schemes including KDDCS suffer from Query Hot-Spots that are formed if the query regions are not uniformly distributed. As a result, it causes reducing the life time of the sensor network. In this paper, we propose a new DCS scheme, called TPDCS

· 본 연구는 교육과학기술부와 한국산업기술재단의 지역혁신인력양성사업 그리고 2008년 교육과학기술(지역거점연구단육성사업/충북BIT연구중심대학육성사업 단)으로부터 지원받아 수행된 연구임

논문접수 : 2008년 8월 18일  
 심사완료 : 2008년 12월 16일

<sup>†</sup> 학생회원 : 충북대학교 정보통신공학과  
 yhpark@netdb.chungbuk.ac.kr  
 blulette@netdb.chungbuk.ac.kr

<sup>\*\*</sup> 학생회원 : KAIST 전산학과 박사후연구원  
 dmseo@netdb.chungbuk.ac.kr

<sup>\*\*\*</sup> 종신회원 : 충북대학교 정보통신공학과  
 jkim@testmidas.com

<sup>\*\*\*\*</sup> 종신회원 : 충북대학교 정보통신공학과 교수  
 yjs@chungbuk.ac.kr  
 (Corresponding author임)

Copyright©2009 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 데이터베이스 제36권 제2호(2009.4)

(Time-Parameterized DCS), that avoids the problems of storage hot-spots and query hot-spots. To decentralize the skewed data and queries, the data regions are assigned by a time dimension as well as data dimensions in our proposed scheme. Therefore, TPDCS extends the life time of sensor networks. It is shown through various experiments that our scheme outperform the existing schemes.

**Key words** : Sensor Network, DCS(Data-Centric Storage), Time Parameter, Energy Efficiency

## 1. 서론

저전력 무선 통신의 발전과 다기능, 저가의 스마트 센서는 원격에서 상태정보를 감지할 수 있는 센서 네트워크(sensor network)의 실현이 가능하도록 하였다. 센서 네트워크는 광범위하게 설치되어 있는 유무선 네트워크 인프라에 상황인지를 위한 다양한 센싱 디바이스를 통합하여 감지된 환경 데이터를 응용서비스 서버와 연동하는 기술이다. 이 기술은 군사, 지능형 물류관리, 모바일 헬스케어, 실시간 보안 분야 등 다양한 분야로 활용 범위를 넓혀 가고 있다[1-4].

센서 네트워크는 기존 무선 네트워크와는 달리 설계 및 개발 시 고려해야 할 고유의 특성을 가지고 있다. 첫째, 센서 네트워크는 계산 능력, 저장 공간, 전력 등의 자원이 매우 제한적이다. 특히 고갈된 전력을 재충전하거나 배터리를 교환하는 것이 거의 불가능하기 때문에, 센서의 활동 중 가장 많은 전력을 필요로 하는 데이터의 무선 전송을 줄이는 것이 매우 중요하다. 둘째, 각각의 센서는 수동적으로 배치되거나 설정될 수 없으며 무작위로 배포된다. 그래서 각 센서는 전체 네트워크 상황을 알 수 없기 때문에 지역적인 정보만으로 네트워크를 구성해야 한다. 셋째, 한 번 배포된 센서는 재사용되지 않는다. 따라서 센서 네트워크의 동작 시간을 최대한 연장할 수 있는 데이터 처리 방법이 필요하다[5]. 센서 네트워크의 에너지 효율성은 가장 에너지 소모가 많은 몇몇 센서의 동작 시간에 의해 결정되기 때문에, 네트워크 내의 모든 센서가 균일하게 전력을 소모하도록 하는 것이 에너지 효율성을 향상시키는 방법이 될 수 있다.

센서 네트워크는 응용에 따라 사용되는 환경이 전혀 다르기 때문에 환경에 맞는 데이터 저장 방식을 적절하게 선택해야 한다. 센서 네트워크의 저장 기법은 크게 세 가지로 분류될 수 있다. 첫째, 외부 저장(external storage) 기법은 센서에서 새로운 데이터를 감지하거나 주기적으로 생성한 데이터를 지정된 외부 장치로 보내는 방식이다. 이 방식은 데이터가 센서에 저장되지 않기 때문에 센서 자체의 저장 공간은 필요하지 않다. 그리고 질의는 외부 저장 공간에서 직접 이루어지기 때문에 네트워크 비용이 소모되지 않는다. 하지만 외부 저장 공간과 연결된 센서 네트워크 토폴로지에서 상위 센서 노드일수록 많은 데이터를 전송해야하는 문제와 많은 데이터

가 동시에 발생하여 중앙 시스템으로 집중되기 때문에 데이터 전송 주기가 짧을 경우 네트워크 병목현상이 발생하는 문제를 가진다. 둘째, 직접 저장(local storage) 기법은 데이터를 감지한 센서가 직접 데이터를 저장하는 방식이다. 이 방식은 데이터 저장을 위한 전송 비용이 존재하지 않기 때문에 많은 데이터가 발생하는 환경에 적합하다. 하지만 질의를 처리하기 위해서는 네트워크상의 모든 센서에 질의를 보내야 하기 때문에 질의 처리 비용이 네트워크의 크기에 비례하여 증가하고, 질의 처리 시간이 매우 큰 문제를 가진다. 셋째, 데이터 중심 저장(DCS: data-centric storage) 기법은 직접 저장 방식과 마찬가지로 각 센서가 데이터 저장 공간을 갖는다. 이 방식에서 데이터 저장 위치는 해당 데이터를 담당하는 센서에 저장된다. DCS는 데이터를 저장하기 위해 여러 센서 노드를 거쳐 전송하기 때문에, 직접 저장 방식에 비해 데이터 저장 비용이 추가로 소요된다. 하지만 질의를 처리할 때는 네트워크 전역에 질의를 보내지 않고, 특정 센서에게만 전송하기 때문에 효율적이다[6].

최근 센서 네트워크의 에너지 효율을 향상시키기 위해 감지된 데이터를 센서 네트워크상에서 효과적으로 저장하고 처리하기 위한 DCS 기반의 다양한 기법들이 제안되었다. 대표적인 기법들로 DIM[7]과 KDDCS[8]가 있다. DIM(Distributed Index for Multi-dimensional data)은 K-D 트리를 기반으로 센서와 데이터를 균등 분할하는 방식으로 색인한다. DIM에서 각 센서는 K-D 트리의 단말 노드에 해당하는 다차원 데이터 영역을 맡는다. 그리고 DIM에서 사용되는 색인은 데이터의 지역성(locality)을 보존하는 위치 기반의 해싱 기법을 사용하기 때문에 적은 수의 라우팅으로 영역 질의를 처리할 수 있다. 하지만 특정 데이터 영역에 대한 데이터가 집중적으로 발생되는 저장소 핫스팟(storage hot-spot) 문제를 고려하지 않았기 때문에, 특정 센서에 대한 저장 공간 부족 문제 및 이로 인한 빠른 에너지 소모 문제를 가진다. KDDCS(K-D tree based Data-Centric Storage)는 DIM을 개선하기 위해, 저장소 핫스팟이 발생하는 센서의 데이터 영역을 동적으로 조정할 수 있는 K-D 트리 기반의 색인 기법을 제안하였다. 하지만 KDDCS를 포함한 모든 기존 DCS 기반의 기법들은 인기있는 데이터 영역에 대한 질의 핫스팟(query hot-spot) 문제를 고려하지 않았기 때문에, 결과적으로 전체 센서 네트워크의 에

너지 효율이 크게 감소하는 문제를 가진다.

따라서 본 논문에서는 저장소 핫스팟과 질의 핫스팟을 효과적으로 해결할 수 있는 TPDCS(Time Parameterized DCS) 기법을 제안한다. 제안하는 TPDCS는 질의 핫스팟을 고려한 최초의 DCS 기반의 기법으로, 분할된 데이터 영역만을 센서에게 할당한 기존 DCS 기반의 기법들과는 다르게 시간 영역과 데이터 영역을 함께 고려해 분할된 시간-데이터 영역을 센서에게 할당하는 기법이다. 즉, 감지된 데이터뿐만 아니라 데이터가 감지된 시간을 함께 고려해 저장 위치를 결정한다. 제안하는 기법은 시간에 따라 센서가 담당하는 데이터 영역이 변경되므로, 어떤 상황에서도 균등한 데이터 저장 공간의 사용 및 질의 분산이 가능해 센서 네트워크의 에너지 효율성이 증가된다. 또한 데이터의 지역성을 보장하기 위해 Z-Ordering[9] 기법을 통해 센서의 데이터 영역을 교차함으로써, 영역 질의 처리 비용을 크게 감소시킨다.

본 논문의 구성은 다음과 같다. 2장에서는 기존에 제안된 DCS 기반의 기법들에 대해 서술한다. 3장에서는 제안하는 TPDCS 기법에 대해 기술한다. 4장에서는 다양한 실험을 통해 기존에 제안된 기법들과의 성능 평가를 수행하고 마지막 5장에서는 결론 및 향후 연구 방향에 대해 기술한다.

## 2. 관련 연구

최근 DCS에 대한 연구가 많이 진행되고 있다. GHT[10]는 DCS를 제안한 첫 번째 기법으로서 해시 함수를 이용하여 발생한 데이터를 저장할 센서의 좌표를 결정하고 그 좌표와 가장 근접한 센서에 그 데이터를 저장하는 방식이다. GHT는 데이터를 균등하게 분포 시키지 못할 뿐더러 데이터의 지역성을 보장하지 못하기 때문에 영역 질의 처리에는 적합하지 않다. DIM[7]은 데이터의 지역성을 보장하여 효율적인 영역 질의 처리를 위해 제안되었다. 이후 DIM의 저장소 핫스팟을 해결하기 위해 센서가 담당하는 데이터 영역의 범위를 동적으로 조절하는 KDDCS[8]가 제안되었다. 그리고 DBS[11]는 기지국을 중심으로 센서들의 부하를 수집하여 영역을 재할당 하는 기법을 제안하였지만 질의 처리와 영역 변경에 기지국이 항상 중심되어야 하는 제약이 있다. 또한 질의 핫스팟 문제를 해결하기 위해서 인기 있는 데이터를 복제하여 동일한 질의를 여러 센서가 분산 처리 하는 방식인 ZPR[12]이 제안되었다. 하지만 ZPR은 저장소 핫스팟을 고려하지 않고 오직 질의 핫스팟만을 고려한다. 이 외에 DCS 환경에서 데이터 전송을 효과적으로 수행하여 전송 부하를 줄이는 기법들도 제안되었다[13-17]. 본 장에서는 DCS의 가장 대표적인 기법으로서 성능 평가의 대상이 되는 DIM과 KDDCS에 대해서 설명한다.

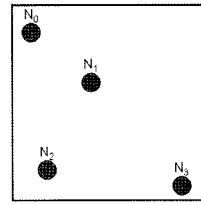
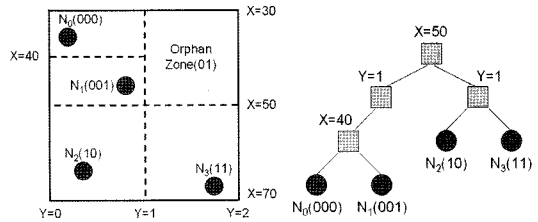


그림 1 초기 센서 노드들의 물리적 위치



(a) DIM 노드 주소 (b) DIM K-D 트리

그림 2 DIM K-D 트리

DIM은 분포 종속 분할 전략[18]을 사용하는 K-D 트리를 기반으로 차원을 번갈아 가면서 균등 분할을 수행하여 각 센서의 데이터 영역을 결정한다. 센서 네트워크 환경에 분포된 각 센서에게 저장할 데이터 영역을 할당한다. 그리고 이와 같은 영역을 할당받은 각 센서를 식별하기 위해 영역 비트 코드(Zone Bit Code)를 각 센서에 부여한다. 그림 1은 센서 네트워크 환경에 분포된 센서들을 보여준다. 각 센서에 부여된 영역 코드 비트들에 대한 K-D 트리를 모든 센서에 유지함으로써 자신이 감지한 데이터들을 해당 데이터 영역을 담당하는 센서로 전달할 수 있다. 그림 2(a)와 그림 2(b)는 DIM을 통해 그림 1의 각 센서에게 저장할 데이터 영역을 분할하고 영역 비트 코드를 할당한 것과 비트 코드들에 대한 K-D 트리를 나타낸다. 그리고 한 센서가 데이터를 수집하면 그림 2(b)의 DIM K-D 트리를 통해 수집된 데이터를 저장하는 센서 노드를 찾고 GPSR[19] 기법을 통해 데이터를 전달한다.

DIM은 데이터 수가 아닌 공간 중심으로 영역을 분할하기 때문에 분할된 영역에 센서가 할당되지 않는 고아 영역(orphan zone)이 발생한다. 고아 영역은 해당 영역에 대해 발생하는 데이터들을 인접 센서들로 전달해야 하는 비용과 질의 처리 시 인접 센서들을 접근해 질의를 처리해야 하는 비용이 요구된다. 또한 DIM은 센서들이 센서 네트워크 환경에서 불균등하게 분포될 경우, 넓은 데이터 영역이 할당된 센서에서 저장소 핫스팟이 발생한다. 저장소 핫스팟이 발생한 센서에서는 저장소의 제한으로 인한 데이터 손실뿐만 아니라 많은 데이터 수집 비용과 질의 처리 비용이 요구되기 때문에 전체 센서네트워크의 수명을 단축시키는 요인이 된다.

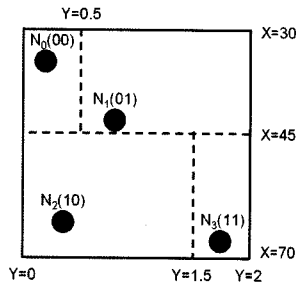


그림 3 KDDCS K-D 트리

KDDCS는 DIM의 고아 영역과 저장소 핫스팟 문제를 해결하기 위해 제안된 기법이다. 그림 3은 KDDCS를 통해 그림 1의 각 센서에게 저장할 데이터 영역을 분할하고 영역 비트 코드를 할당한 것을 보여준다. KDDCS의 데이터 중속 분할 전략[18]을 통한 영역 분할은 각 분할 영역에 한 개의 센서를 필히 배치하기 때문에 DIM의 고아 영역 문제가 발생하지 않고 센서에 저장된 데이터 수가 같도록 분할 축의 위치를 결정하기 때문에 저장소 핫스팟 문제가 초기 분할 시 고려된다. 하지만, 초기 영역 분할 후 특정 센서에 저장되는 데이터의 수가 많아지면 여전히 저장소 핫스팟 문제가 발생하기 때문에 KDDCS는 센서에 저장된 데이터 수가 같아지도록 분할 축이 동적으로 조정되는 균형 K-D 트리를 제안한다. 예를 들어, 초기 분할 후  $N_3$ 에 저장된 데이터 수가 많아져  $N_2$ 와  $N_3$ 에 저장된 데이터 비율이 임계값을 넘으면  $N_2$ 와  $N_3$ 에 저장된 데이터 수가 같아지도록 분할 축이 조정되고, 조정된  $N_3$ 의 데이터 영역에 포함되지 않는  $N_3$ 의 일부 데이터들이  $N_2$ 로 전달된다.

KDDCS는 DIM의 고아 영역 문제와 저장소 핫스팟 문제를 개선하여 전체 센서 네트워크의 수명을 연장시켰다. 하지만, 균형 K-D 트리를 유지하기 위해 주기적으로 센서에 저장된 데이터 비율이 임계값을 넘는지 검사해야하는 비용과 분할 축 조정에 따른 관련 센서들의

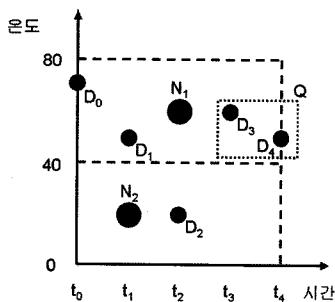
데이터 전송 비용이 요구된다. 또한, 분할 축 변경은 관련 센서들의 균형 K-D 트리에 대한 정보 갱신과 센서에 저장된 정보의 이동을 야기한다.

### 3. 제안하는 TPDCS(Time Parameterized Data-Centric Storage) 기법

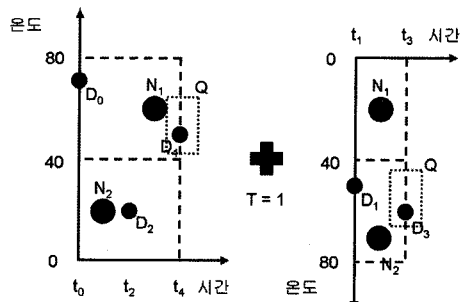
#### 3.1 제안하는 기법의 특성

DCS 기반의 DIM과 KDDCS는 센서 네트워크의 에너지 효율을 향상시키기 위해 감지된 데이터를 센서 네트워크상에서 효과적으로 저장하고 처리하기 위해 최근에 제안된 기법들이다. 하지만 DCS 기반의 기존 기법들은 데이터 영역만을 고려한 데이터 분할 정책을 사용하기 때문에, 인기있는 데이터 영역에 대한 질의 핫스팟이 발생해 에너지 효율이 크게 향상되지 못한다. 또한, 제한적인 저장 공간을 가지는 센서의 특성을 완벽하게 고려하지 않았기 때문에 저장소 핫스팟을 피하려는 많은 비용이 요구된다. 그래서 TPDCS는 시간 영역과 데이터 영역을 함께 고려해 분할된 시간-데이터 영역을 센서에게 할당하는 기법을 제안한다. 즉, TPDCS는 감지된 데이터뿐만 아니라 데이터가 감지된 시간을 함께 고려해 저장 위치를 결정한다.

그림 4는 DIM, KDDCS 그리고 TPDCS에서의 센서들이 감지한 데이터를 저장하는 방법을 보여준다. 그림 4에서  $N_i$ 는 센서 노드,  $D_i$ 는 시간  $t_i$ 에 감지된 데이터를 나타낸다. 그림 4(a)는 DIM과 KDDCS에서  $N_1$ 과  $N_2$ 가 각각 (0~40]과 (40~80]의 데이터 영역에 해당하는 데이터를 저장하는 것을 보여준다. 만약 (0~40]과 (40~80]의 데이터 영역에 해당하는 데이터가 2:8의 비율을 가지고 감지되면,  $N_1$ 은  $N_2$ 에 대해 저장소 핫스팟 문제를 갖게 된다. 이 문제는 DIM에서 저장소 핫스팟이 발생한 센서의 데이터 손실을 야기한다. 또한 KDDCS에서는 균형 K-D 트리의 분할 축 조정이 발생하기 때문에 추가적인 데이터 전송 비용이 발생한다. 그래서 TPDCS는 이



(a) DIM과 KDDCS의 데이터 영역



(b) TPDCS의 데이터 영역

그림 4 TPDCS 기본원리

문제를 해결하기 위해, 임의의 시간 주기로 센서들이 담당하는 데이터 영역을 재할당하는 기법을 제안한다. 그림 4(b)는 TPDCS에서 시간 주기 1을 기준으로 데이터 영역을 재할당하는 기법의 기본원리를 보여준다. TPDCS에서  $N_1$ 과  $N_2$ 는  $t_1$ ,  $t_3$  그리고  $t_5$ 에 각각 (0~40]과 (40~80]의 데이터 영역에 해당하는 데이터를 저장하고  $t_2$ 와  $t_4$ 에 각각 (40~80]과 (0~40]의 데이터 영역에 해당하는 데이터를 저장한다. 즉, 임의의 시간을 주기로 특정 센서의 저장소 핫스팟 문제를 다른 센서들에게 분담하여 감지되는 모든 데이터를 전체 센서 네트워크에 균등 분포시킨다. 또한, 제안하는 방법은 영역 질의  $Q$  처리 시 동일한 데이터 영역에 대한 질의도 질의 시간 영역에 따라 다른 센서에서 수행이 되기 때문에 질의 핫스팟에 대한 센서의 비 균등한 에너지 소모를 줄일 수 있다.

**3.2 상대 위치 기반의 데이터 영역 할당 알고리즘**

DIM과 KDDCS에서 초기 각 센서들의 데이터 영역 할당은 센서들의 공간상 절대적 위치와 데이터 영역이 밀접한 관계를 갖고 할당되기 때문에, 센서 분포 상태에 따라 데이터 영역을 할당하고 조정하는데 많은 한계가 있다. 따라서 TPDCS는 센서들의 분포 형태에 독립적으로 센서들에게 균등한 데이터 영역을 할당하기 위해, 센서들의 공간상 상대적 위치와 데이터 영역의 관계를 고려하여 각 센서들의 데이터 영역을 할당하는 기법을 제안한다. TPDCS에서 초기 센서들의 데이터 영역 할당은 아래와 같은 과정을 통해 수행된다.

**(과정 1) 센서들의 공간상 절대 위치를 기반으로 공간 영역 분할.** 초기 모든 센서의 위치 정보를 수집한 기지국은 KDDCS와 마찬가지로, 균형 K-D 트리를 통해 각 센서가 한 개의 분할된 공간에 위치하도록 영역을 분할한다. 즉, 분할된 영역의 개수  $RN$ 과 센서의 개수  $SN$ 은 같다. 그림 5(a)는 균형 K-D 트리 정책에 의해 분할된 공간 영역을 보여준다. 그리고 이때 만들어진 K-D 트리를  $ST$ 라 하자. 균형 K-D 트리의 특성 때문에  $ST$ 의 단말 노드 간 최대 트리 레벨(level) 차이는 1

이다. 그래서  $SN$ 이 주어졌을 때,  $ST$ 의 최대 트리 높이  $ST.Hmax$ 는  $\lceil \log_2 SN \rceil$  이고, 최저 트리 높이  $ST.Hmin$ 은  $\lceil \log_2 SN \rceil - 1$ 이다. 예를 들어,  $SN$ 이 15일 때  $ST.Hmax$ 는 4이고,  $ST.Hmin$ 는 3이다.

**(과정 2) 분할된 공간 영역을 기반으로 데이터 영역 분할.** 과정 1에서 공간 영역을 분할하면서 만들어진 K-D 트리와 동일한 구조로 데이터 영역을 분할한다. 하지만 공간 영역 분할과 다르게 센서의 공간상 절대 위치를 고려하여 분할하지 않고, 영역을 균등 분할하는 형태로 분할을 수행한다. 그림 5(b)는 그림 5(a)의 분할된 공간 영역을 기반으로 분할된 데이터 영역을 보여준다. 이때 만들어진 K-D 트리를  $DT$ 라 하자. 영역을 균등 분할하기 때문에, 트리의 레벨에 따라 데이터 영역의 크기가 다르고 같은 레벨의 데이터 영역의 크기는 모두 같다. 그래서 모든 센서는 레벨이  $ST.Hmax$  또는  $ST.Hmin$ 에 대한 데이터 영역의 크기를 가진다. 또한, 레벨이  $ST.Hmax$ 인 데이터 영역 크기는 레벨이  $ST.Hmin$ 인 데이터 영역의 크기에 비해 최대 두 배 크다.

**(과정 3) 각 센서에 분할된 데이터 영역 매핑.** 과정 1과 과정 2를 통해  $ST$ 와  $DT$ 가 생성되며, 이 두 K-D 트리는 분할 축의 위치는 틀리지만 분할 구조는 동일하다.  $ST$ 에 위치하는 각 센서는  $DT$ 상에서 동일한 위치를 가지는 데이터 영역을 할당 받는다. 제안하는 데이터 영역 매핑 방법은 고아 영역이 존재하지 않으면서, 각 센서들이 가지는 데이터 영역의 크기를 최대한 균등하게 한다. 그림 5는 TPDCS에서 초기 각 센서들에 대한 데이터 영역 매핑 예제를 보여준다. 그림 5(a)의 각 영역에 위치하는 센서는 동일한 번호를 가진 그림 5(b)의 데이터 영역을 할당 받는다. 본 논문에서 초기 센서  $N_i$ 에 매핑된 데이터 영역 번호를  $DR(N_i)$ 로 표기한다.

**3.3 데이터 영역 정보 배포**

데이터 영역 분할과 할당 작업은 기지국에서 수행한다. 그러기 위해서 기지국은 센서 네트워크의 각 센서들로부터 위치 좌표표를 전송 받는다. 데이터 영역 분할 연

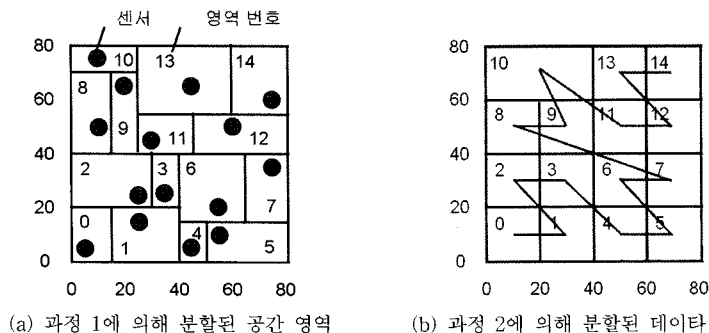


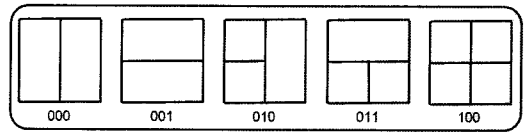
그림 5 상대 위치 기반의 데이터 영역 할당

산 후에 분할 정보와 데이터 영역의 크기, 영역 변경 주기 등의 정보를 각 센서에 전송한다. 그리고 각 센서들은 기지국에서 연산하여 결정한 데이터 영역의 분할 정보를 유지하고 그 정보를 바탕으로 감지한 데이터를 저장할 센서를 선택한다. 기지국에서 각 센서에 초기 할당된 영역번호(센서에 할당된 번호와 동일)와 영역 분할 정보를 전송하고 그 정보가 센서에서 유지되어 지는 데 소모되는 전송 비용과 저장 비용을 줄이기 위해 분할 정보를 압축하는 비트 패턴 기반의 영역 분할 정보 표현 기법을 제안한다. 제안하는 기법은 가장 작은 데이터 영역 셀의 크기를 기준으로 2x2 해상도를 가지는 데이터 영역에서 발생할 수 있는 모든 경우의 데이터 분할 영역을 정의하고, 이를 기반으로 실제 분할된 센서 네트워크의 데이터 영역의 분할 정보를 표현한다.

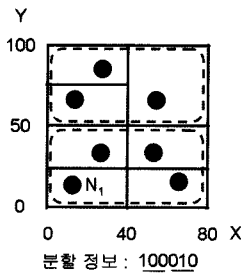
그림 6은 비트 패턴 기반의 데이터 영역 분할 정보 표현 기법의 한 예를 보여준다. 그림 6(a)는 2x2 해상도에서 데이터 영역 분할 정보를 비트로 표현하기 위한 표본을 보여준다. 2x2 해상도를 가지는 데이터 영역은 총 5개의 분할 영역을 가질 수 있으므로 3개의 비트로 모든 분할 영역 정보를 표현할 수 있다. K-D 트리 분할 정책에 따르면, 분할 후 작은 키 값을 담당하는 영역이 큰 키 값을 담당하는 영역에 대해 최대 1 많은 객체를 포함하므로 '010'과 '011'의 반대 경우는 발생하지 않는다.

각 센서들은 자신 및 다른 센서들이 관리하는 데이터 영역을 알기 위해서, 초기에 기지국으로부터 비트 패턴 기반의 데이터 영역 분할 정보와 그림 5(b)와 같이 Z-Ordering(Z-Ordering을 사용하는 이유는 3.4절에서 설명)을 통해 자신에게 할당된 영역 번호(zone number)를 획득한다. 예를 들어, 그림 6(b)에서 볼 수 있듯이 기지국으로부터 할당된 영역 번호 '0'과 데이터 영역 분할 정보 '100010'을 전달받은 센서  $N_1$ 은 데이터 영역이 그림 6(b)와 같이 분할된 것을 알 수 있고 영역 번호 '0'을 통해 자신이 X축에 대해 (0, 40], Y축에 대해 (0, 25]의 데이터 영역을 담당한다는 것을 알 수 있다. 이것은 데이터 영역이 균등 분할이 되어 있기 때문이다. 마찬가지로, 그림 6(c)의  $N_2$ 는 할당된 영역 번호 '1'과 데이터 영역 분할 정보 '001011010100'을 통해 자신이 X축에 대해 (0, 40], Y축에 대해 (25, 50]의 데이터 영역을 담당한다는 것을 알 수 있다. 게다가 다른 센서들이 담당하는 데이터 영역 정보도 알 수 있다. 데이터 영역 분할 정보는 분할 정보 뿐 만 아니라, 데이터 영역의 범위 정보와 전체 센서의 개수 등의 정보가 포함되어 각 센서에게 전송된다.

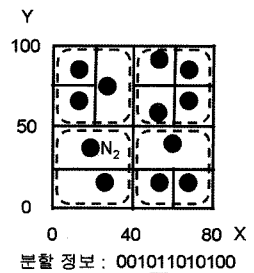
데이터 영역 분할 정보를 K-D 트리 구조 정보로 표



(a) 2x2 영역에서 영역 분할



(b) 2x4 해상도 영역에 대한 분할



(c) 4x4 해상도 영역에 대한 분할

그림 6 비트 패턴 기반의 데이터 영역 분할 정보 표현

현하면 식 (1)의 크기에 해당하는 저장 공간이 요구된다. K-D tree의 분할 축을 구성하는 정보를 두개의 자식 노드 포인터(4bytex2), 분할 축 구분자(1byte), 분할 축 위치(4byte)으로 가정하고 한 분할축의 크기를  $S_{split}$ 으로 표현하면, 식 (1)에 따라 그림 6(b)와 그림 6(c)의 분할 정보는 각각 78byte, 143byte로 표현된다.

$$S_{total\_split} = \sum_{i=0}^{\lfloor \log_2 SN \rfloor} (S_{split} \times 2^i) + S_{split} \times (SN - 2^{\lfloor \log_2 SN \rfloor}) \quad (1)$$

반면에 비트 패턴 기반의 분할 정보의 크기는 K-D tree형태로 분할 정보를 표현하는 것보다 훨씬 적은 저장 비용을 요구한다. 예를 들어, 그림 6(b)와 그림 6(c)는 그림 6(a)를 기반으로 2x4와 4x4 해상도를 가지는 데이터 분할 영역 정보를 표현한 것을 보여준다. 2x4 해상도를 가지는 데이터 분할 영역은 2x2 해상도를 가지는 데이터 분할 영역 2개로 표현되기 때문에 6비트로 표현된다. 마찬가지로, 4x4 해상도를 가지는 데이터 분할 영역은 12비트로 표현된다. 즉, 데이터 영역 분할 정보 표현을 위해 요구되는 비트의 데이터 크기  $S_{total\_split}$ 은 식 (2)와 같이 해상도를 결정하는 SN에 의해 결정된다.

$$S_{total\_split} = 2^{\lfloor \log_2 SN \rfloor - 2} \times 3(bit) \quad (2)$$

### 3.4 데이터 영역 재할당 알고리즘

TPDCS는 센서 네트워크에서의 저장소 핫스팟과 질의 핫스팟을 피하기 위해서, 임의의 시간 주기로 센서들이 담당하는 데이터 영역을 재할당하는 기법을 제안한다. 제안하는 데이터 영역 재할당 기법은 3.1절에서 기술된 TPDCS의 기본원리를 기반으로, 주어진 시간 주기  $T$ 가 되면 자신에게 할당된 영역 번호 다음 값에 해당하는

센서가 담당하는 데이터 영역을 할당받아 다음 시간 주기  $T$  동안 해당 데이터 영역에 대한 데이터를 저장하는 방법이다. 그림 7은 TPDCS의 데이터 영역 재할당 기법에 대한 예제를 보여준다. 그림 7(a)는 주기  $T_0$ 에 그림 5(b)에 대한 초기 데이터 영역이 매핑된 센서들을 보여준다. 예를 들어, 영역 번호 '0'을 할당받은 센서  $N_0$ 는  $T_0$ 에서  $T_1$ 까지 X축에 대해 (0, 20], Y축에 대해 (0, 20]의 데이터 영역을 담당한다. 그림 7(b)와 그림 7(c)는  $T_1$ 과  $T_n$ 에 대해 데이터 영역이 재할당된 센서들을 보여준다. 그림에서와 같이 센서들이 주기가 변경됨에 따라, 물리적 위치가 변경되는 것이 아니라 해당 주기에 담당하는 데이터 영역의 범위가 논리적으로 변경되는 것이다. 예를 들어, 그림 7(b)에서  $N_0$ 는  $T_1$ 에 영역 번호 '14'인 데이터 영역으로 위치가 변경되는 것이 아니라, 영역 번호 '0'인 곳에 위치하면서 다음 주기까지는 논리적으로 X축에 대해 (60, 80], Y축에 대해 (60, 80]의 데이터 영역을 담당한다.

TPDCS는 데이터 영역 재할당을 위해 사용되는 영역 번호를 부여하기 위해 Z-Ordering을 사용한다. TPDCS는 Z-Ordering을 통해 위치적으로 근접한 센서는 항상 유사한 데이터를 유지하여 질의 처리 시, 질의가 센서 네트워크에 광범위하게 내려지는 것을 피해 불필요한 통신비용을 줄인다. Z-Ordering은 Hilbert Curve를 위한 기법들 중에 한 기법으로 [9]에 따르면 지역성이 다른 기법들에 비해 가장 좋은 기법이기 때문에, TPDCS는 Z-Ordering을 사용했다.

센서 자신과 관련된 데이터 영역 분할 정보만 유지하는 DIM과 KDDCS와는 달리 TPDCS는 전체 데이터 영역 분할 정보를 유지하기 때문에 데이터 영역 재할당 시에 추가적인 비용 소모는 발생하지 않기 때문에 그에 따른 비용은 고려되지 않는다. TPDCS의 데이터 영역 재할당 기법에서 중요하게 고려될 사항은 주기  $T$  결정이다. 주기가 너무 길면 저장소 핫스팟과 질의 핫스팟의 문제가 발생하게 된다. 반면에 주기가 너무 짧으면 저장

공간의 활용성은 향상되지만, 질의 처리 시 참여하는 센서의 수가 증가하기 때문에 질의를 전송하는 비용에 따른 센서 네트워크의 전체 에너지 소모가 증가한다. 그러므로 TPDCS는 센서 네트워크의 에너지 효율을 가장 많이 향상시킬 수 있는 주기 결정이 요구된다.

일반적으로, 질의 처리를 위한 에너지 소모  $E_{QT}$ 는 식 (3)과 같이 질의에 참여하는 센서들에게 질의를 전송하는데 드는 비용  $E_{QD}$ , 질의 결과들을 전송하는데 드는 비용  $E_{QR}$  그리고 질의 처리를 위해 수행되는 센서들의 전체 연산 비용  $E_{QC}$ 의 총합이다.

$$E_{QT} = E_{QD} + E_{QR} + E_{QC} \quad (3)$$

DIM, KDDCS 그리고 TPDCS는 질의 처리 시 질의 결과들을 전달하는 평균 비용은 같다. 이것은 임의의 질의에 대해 DIM, KDDCS 그리고 TPDCS의 질의 결과의 개수는 동일하고 질의 결과를 전송하기 위한 평균 라우팅 비용은 같기 때문이다. 또한, 센서 네트워크 환경에서 데이터 전송 비용에 비해 센서의 연산 비용은 무시될 만큼 매우 적다. 그러므로 질의 처리를 위한 에너지 소모는 질의 전송 비용에 의해 결정된다. DIM과 KDDCS는 감지된 데이터들의 시간과는 관계없이 감지된 데이터 값을 포함하는 데이터 영역을 담당하는 센서에게 데이터를 저장하기 때문에 질의 전송 시, 비용 차이가 크게 발생하지 않는다. 하지만, TPDCS는 주기에 따라 각 센서의 데이터 영역이 변경되기 때문에 질의에 참여하는 센서의 개수는 주기 및 질의를 구성하는 시간 영역에 의해 결정된다. 식 (4)는 질의 전송 비용  $E_{QD}$ 를 보여준다.  $AE_{QD}$ 는 한 센서에게 질의를 전송하기 위한 평균 비용,  $SN_Q$ 는 질의에 참여하는 센서의 개수 그리고  $a$ 는  $SN_Q$ 의 증가에 따른 비용 계수를 나타낸다.  $a$ 는 센서들의 지역성에 따라 0에서 1의 값을 가지고 1의 값을 가지면 모든 센서에 질의를 각각 별도로 보내는 경우를 나타내고, 0에 가까울수록 질의에 참여하는 센서들의 지역성이 높아 공통된 질의 전송 경로를 많이 가지는 것을 나타낸다.

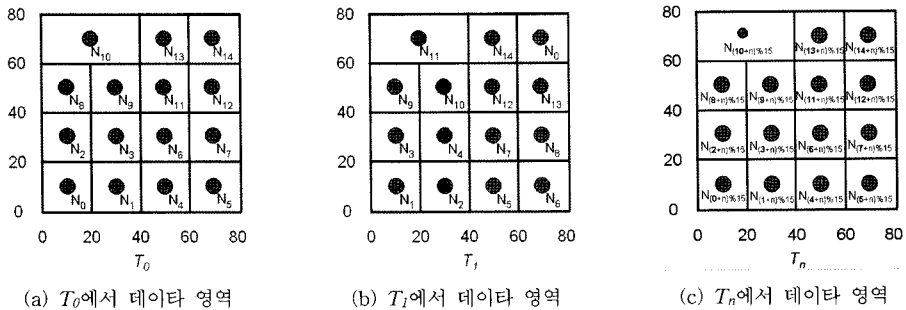


그림 7 데이터 영역 재할당 기법

$$E_{QD} = \alpha \cdot AE_{QD} \cdot SN_Q \quad (4)$$

DIM과 KDDCS에서는 SNQ는 식 (5)와 같이 질의 영역 QR과 데이터 영역 DR로 나타낸다. TPDCS에서는 식 (6)과 같이 질의를 구성하는 시간 영역 TRQ와 데이터 영역 재활당 주기 T로 나타낸다.

$$SN_Q = QR / DR \quad (5)$$

$$SN_Q = (QR / DR) \times (TR_Q / T) \quad (6)$$

DR은 TPDCS 데이터 영역 분할 정책에 의해 결정되며, QR과 TRQ는 사용자에게 의해 결정되므로 질의 전송 시 요구되는 에너지 비용은 T에 의해 결정된다. 그러므로 질의 전송 시 요구되는 에너지 비용을 줄이기 위해서는 T가 길수록 좋다. 즉, 에너지 소모 관점에서의 주기 결정은 식 (7)과 같은 조건식으로 나타낸다. ESE은 센서 네트워크에 포함된 모든 센서들의 에너지 총합 그리고 QN은 단위 시간당 요청되는 평균 질의 개수이다.

$$\frac{QR}{DR} \cdot \frac{TR_Q}{SN_Q} \leq T \leq \frac{E_{SE}}{E_{QT} \cdot QN \cdot SN} \quad (7)$$

식 (7)에서  $\frac{QR}{DR} \cdot \frac{TR_Q}{SN_Q}$ 을  $T_{MIN}$ ,  $\frac{E_{SE}}{E_{QT} \cdot QN \cdot SN}$ 을  $T_{MAX}$ 라고 표현했을 때, 그림 8은 SNQ에 따라 TPDCS가 선택해야할 주기 T를 보여준다.

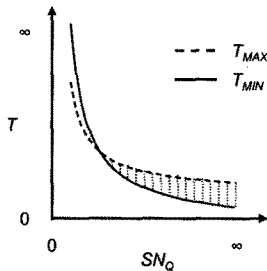


그림 8 에너지 소모 관점에서의 데이터 영역 재활당 주기

TPDCS는 주기 결정시 에너지 소모 관점에서 뿐만 아니라, 특정 데이터 영역에 대한 저장소 핫스팟을 해결하기 위해 저장 공간 소모 관점에서의 주기 결정이 요구된다. 식 (8)은 저장 공간 소모 관점에서의 주기 결정을 나타낸다. ESS는 센서 네트워크에 포함된 모든 센서들의 저장 공간의 총합 그리고 DN은 단위 시간당 감지되는 데이터 개수이다.

$$T \leq \frac{E_{SS}}{DN \cdot SN} \quad (8)$$

결론적으로, T가 길수록 저장소 핫스팟 문제가 발생할 수 있고 특정 데이터 영역에 대한 질의 핫스팟으로 인한 에너지 소모의 불균등 문제가 발생할 수 있어서 T

가 짧아야한다. 반면에 T가 너무 짧으면 저장소 핫스팟과 질의 핫스팟 문제는 향상되지만, 질의 처리 시 참여하는 센서의 수가 증가하기 때문에 질의를 전송하는 비용에 따른 센서 네트워크의 전체 에너지 소모가 증가한다. 그래서 TPDCS는 식 (7)과 식 (8)을 고려해 응용에 적합한 주기 선택이 요구된다. 이러한 주기 결정은 미국에서 센서 네트워크를 초기화할 때 함께 처리 된다. 주기 결정을 위해 이용되는 인자들은 대부분 응용에 따라 추정이 가능하며, 한 질의에 최대 포함되는 센서의 개수 SNQ는 그림 8에서 보여주는 것과 같이 주기 T에 의해 결정되며 단지 T를 결정하는데 참고되는 값으로 활용된다.

### 3.4 데이터 저장 및 전송 알고리즘

본 절에서는 센서를 통해 감지된 데이터와 시간을 이용하여 데이터를 저장할 센서를 선택하는 방법과 데이터를 전송하는 라우팅 기법에 대해 기술한다. 감지된 데이터를 저장하는 센서를 선택하기 위해서는 식 (9)와 같이 비트 패턴 기반의 데이터 영역 분할 정보와 영역 번호를 통해 감지된 데이터를 저장하는 데이터 영역의 영역 번호를 계산한다. DR(ND) 감지된 데이터를 포함하는 초기 데이터 영역 번호 DR(Ni), DR(NS)는 감지된 데이터가 저장될 데이터 영역 번호 그리고 TD는 데이터가 감지된 시간을 나타낸다.

$$DR(NS) = DR(ND) - (TD / T) \% SN \quad (9)$$

예를 들어, TPDCS의 T=2라고 가정했을 때, 그림 5(b)에서 DR(Ni)=0인 센서가 TD=10에 X=50 그리고 Y=50인 데이터를 감지하면, DR(Ni)=0인 센서는 비트 패턴 기반의 데이터 영역 분할 정보를 통해 감지된 데이터를 포함하는 데이터 영역 번호 DR(ND)=11을 획득한다. 하지만 TPDCS는 T를 기반으로 센서들이 담당하는 데이터 영역이 변경되므로, TD=10인 경우 데이터 영역 재활당이 (10/2)%15=5번 수행된 상황이므로 데이터 영역 번호가 DR(NS)=11-5=6을 가지는 센서에 감지된 데이터를 저장해야 한다.

데이터를 감지한 센서는 해당 데이터를 저장하는 센서로 데이터를 전송하기 위해 적절한 라우팅 전략을 선택해야한다. TPDCS는 DIM[7]과 마찬가지로 GPSR[19]를 통해 감지된 데이터를 전달한다. KDDCS[8]의 라우팅 전략은 DIM에 비해 최대 두 배의 비용이 요구되기 때문에 TPDCS는 DIM과 같은 라우팅 전략을 통해 데이터 전송 비용을 줄였다.

### 3.5 질의 처리 알고리즘

DIM과 KDDCS에서는 “어제와 오늘 온도가 30° 이상 50° 이하이면서, 습도가 50%인 곳은?”과 같은 질의를 처리하기 위해 온도가 30° 이상 50° 이하이면서, 습도가 50%인 데이터를 포함하는 데이터 영역을 담당하



는 센서를 선택해 요청받은 질의를 처리한다. 하지만, TPDCS는 저장소 핫스팟과 질의 핫스팟 문제를 해결하기 위해 센서들의 데이터 영역이 주기를 기준으로 재할당 되기 때문에, 질의를 처리하기 위한 센서 선택 시 질의를 구성하는 시간 영역이 고려되어야 한다. 즉, 질의의 데이터 영역을 포함하는 센서의 데이터 영역이 DIM과 KDDCS처럼 한 센서에 고정되는 것이 아니라, TPDCS에서는 다른 여러 센서에서 발생할 수 있다.

예를 들어, 그림 5(b)에서 X축을 온도 그리고 Y축을 습도라고 가정하고, 이와 같은 질의를 DIM, KDDCS 그리고 TPDCS에서 처리하는 과정을 살펴보자. DIM과 KDDCS는 습도가 50%이면서 온도가 (30°, 40°]과 (40°, 50°]인 데이터 영역을 담당하는 센서  $DR(N_D)=9$ 와  $DR(N_D)=11$ 을 질의 처리 센서로 선택하고, 각 기법에서 사용하는 라우팅 전략을 통해  $DR(N_D)=9$ 에는 “어제와 오늘 온도가 30° 이상 40° 이하이면서, 습도가 50%인 곳은?” 그리고  $DR(N_D)=11$ 에는 “어제와 오늘 온도가 40° 이상 50° 이하이면서, 습도가 50%인 곳은?” 질의를 전달한다. 하지만,  $T_0$ ='하루',  $T_0$ ='어제' 그리고  $T_1$ ='오늘'인 TPDCS에서  $T_0$ 인 시점에 대해서는 DIM과 KDDCS와 마찬가지로 습도가 50%이면서 온도가 (30°, 40°]과 (40°, 50°]인 데이터 영역을 담당하는 센서  $DR(N_D)=9$ 와  $DR(N_D)=11$ 을 질의 처리 센서로 선택한다. 그리고  $T_1$  시점에 데이터 영역이 재할당 되므로  $T_1$  시점에 대해서는 센서  $DR(N_D)=8$ 와  $DR(N_D)=9$ 가 질의 처리 센서로 선택된다. 결론적으로, TPDCS에서는  $DR(N_D)=9$ 에는 “어제 온도가 30° 이상 40° 이하이면서, 습도가 50%인 곳은?”,  $DR(N_D)=8$ 에는 “오늘 온도가 30° 이상 40° 이하이면서, 습도가 50%인 곳은?”,  $DR(N_D)=11$ 에는 “어제 온도가 40° 이상 50° 이하이면서, 습도가 50%인 곳은?” 그리고  $DR(N_D)=10$ 에는 “오늘 온도가 40° 이상 50° 이하이면서, 습도가 50%인 곳은?” 질의를 전달한다. 물론, TPDCS에서 질의 전달 및 결과 획득을 위한 라우팅 방법은 데이터 저장 방법과 같다.

TPDCS에서 요청된 질의를 데이터 영역과 주기를 고려해 분할하는 방법은 식 (10)과 같다.  $S$ 는 주어진 센서 집합 나타내고 내려진 질의의 형태가  $\langle q, T_s, T_e, D_{target} \rangle$  일 때, 식 (10)은 질의에 참여하는 센서들의 집합을 계산한다. 여기서,  $q$ 는 질의 식별자,  $T_s$ 와  $T_e$ 는 질의 시간 영역의 시작과 끝,  $D_{target}$ 은 질의 데이터 영역을 나타낸다.  $DCs(D_{target}, t)$ 는 시간  $t$ 에 데이터 영역  $D_{target}$ 과 겹치는 데이터 영역을 가지는 센서들의 집합을 나타낸다.

$$S_q(D_{target}, T_s, T_e) = \{s_i | DR(s_i) = dc - (t/T) \% N_{range} \quad (10)$$

$$, \forall s_i \in S, \forall dc \in DCs(D_{target}, t), T_s \leq t \leq T_e\}$$

## 4. 실험 및 성능 평가

### 4.1 실험 환경

본 논문에서 제안하는 TPDCS의 우수성을 보이기 위해 대표적인 DCS 기법인 DIM과 KDDCS를 비교 평가하였다. 성능 평가는 KDDCS의 실험 환경을 기반으로 하였다. 표 1은 실험 환경을 보여준다. 실험에서 각 센서는 0.5~1.0A의 제한된 에너지와 100개의 데이터를 저장할 수 있는 저장 공간을 갖는다고 가정하였다. 센서 네트워크의 크기는 100×100, 200×200 그리고 400×400으로, 센서 네트워크의 수는 16, 64 그리고 256으로 가정하고 다양한 평가를 수행하였다. 그리고 실험에 사용된 센서의 에너지 소비 모델은 1bit의 데이터 처리 시 소모되는 비용을 기준으로 단일 값을 지정하여 사용하였다. 또한, 센서의 통신 반경 내에는 평균 4개의 센서 노드가 존재하고 센서의 저장 공간이 제한된 크기를 가지므로 데이터 저장이 FIFO(First Input First Output) 방식으로 동작하도록 설정하였다. 그래서 센서의 저장 공간이 새로운 데이터를 여유가 없으면 가장 과거의 데이터를 삭제하고 새로운 데이터가 저장된다.

표 1 실험 환경 설정 값

항 목	환 경	
센서 수	16, 64, 256	
센서 네트워크의 크기	100×100, 200×200, 400×400	
센서의 초기 Energy	0.5~1.0 A	
센서의 저장 공간 크기	100 Unit	
데이터 크기	128bit (1 Unit 당 크기)	
데이터 범위	1~100	
에너지 소비 모델 (1bit 당 연산 비용)	데이터 연산 비용	0.5μA
	데이터 수신 비용	5μA
	데이터 송신 비용	5μA

실험 환경에 사용된 데이터 셋은 KDDCS의 성능 평가에서 사용된 데이터 셋으로 값의 범위가 [0, 100]인 2차원 데이터이다. 실험에서 저장소 핫스팟은 전체 센서 네트워크에서 매 라운드마다 특정 범위의 데이터를 일정 비율만큼 집중적으로 발생시켜 평가하였다. 그리고 질의 핫스팟은 저장소 핫스팟이 발생하는 센서 영역을 기준으로 매 라운드마다 일정 수의 질의를 수행하는 환경을 가정하여 평가하였다. 수집된 데이터와 질의 및 질의 결과는 GPSR[19]을 통해 전달된다. 또한 센서는 균일하게 분포시켜서 발생한 데이터와 질의에 따른 공정한 저장소 핫스팟과 질의 핫스팟에 의한 영향을 평가하였다. 센서가 균일하게 배포되면 KDDCS와 DIM의 초기 영역이 균등하게 구성 되고, 이는 초기 분할 영역의 불균등에 의한 성능차이가 아닌 순수하게 데이터 발생 성향에 따른 성능 차이를 확실하게 평가할 수 있기

때문이다. 그리고 기지국에서 데이터 영역을 분할하고 할당하기 위해 발생하는 데이터 전송을 위한 에너지 소모는 초기 구축 시에만 발생하기 때문에 성능 평가에는 포함하지 않았다.

4.2 성능 평가

먼저, 제안하는 TPDCS의 저장소 핫스팟 처리 성능을 비교하기 위해 각 라운드마다 발생하는 데이터의 일부가 센서 네트워크의 특정 센서들에 저장되도록 발생한 데이터의 영역을 조정하여 실험 평가하였다. 실험은 각 라운드마다 발생하는 전체 데이터의 80%를 전체 센서 네트워크의 5%와 10%에 해당하는 센서들에게 저장하는 방법으로 수행하였다. 실험 결과를 비교하기 위해 센서 수만큼의 라운드를 거친 후, 전체 센서 네트워크에 저장된 데이터와 손실된 데이터의 비율을 비교 평가하였다. 저장된 데이터와 손실된 데이터의 비율  $k$ 는 식 (11)으로 계산된다.

$$k = \frac{Data_{Dropped}}{Data_{Stored}} \quad (11)$$

그림 9는 저장소 핫스팟에 따른 실험 결과로, 손실된 데이터와 저장된 데이터의 비율을 보여준다. 실험 결과 제안하는 TPDCS는 저장소 핫스팟에 대해 센서 네트워크의 크기가 변하더라도 일정한 성능을 보이는 반면, DIM과 KDDCS는 센서 네트워크의 크기가 커질수록 데이터의 손실률이 증가하는 것을 확인하였다.

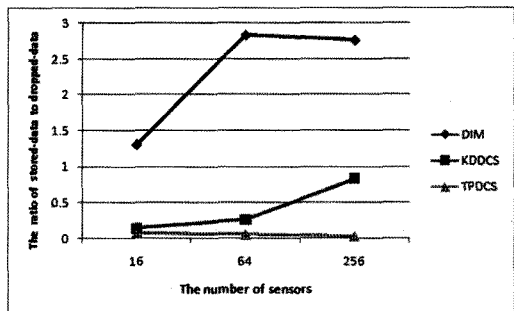
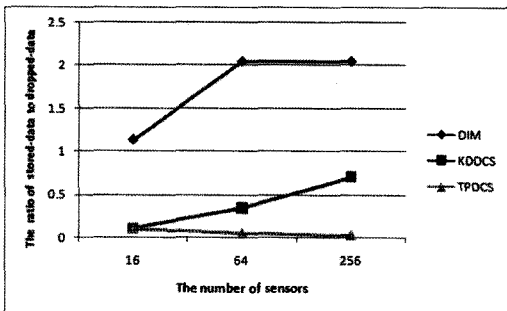
DIM은 TPDCS나 KDDCS와 같이 데이터 재분배를 하지 않기 때문에 일정 라운드 후 저장소 핫스팟이 발생하는 센서는 과거 데이터를 버리고 새로운 데이터를 저장한다. 따라서 저장소 핫스팟이 발생한 센서를 중심으로 높은 데이터 손실이 발생한다. KDDCS는 저장소 핫스팟이 발생한 센서가 분할 축을 재조정하여 데이터를 재분배시키므로 DIM과 비교할 때 좋은 성능을 보인다. 하지만, K-D 트리의 분할 축 조정이 일부 센서들을 기준으로 진행되기 때문에 전체 센서 네트워크의 모든

센서들이 효과적으로 데이터의 재분배를 수행할 수 없다. 반면, TPDCS는 저장소 핫스팟이 발생하지 않도록 시간 영역과 데이터 영역을 함께 고려해 분할된 시간-데이터 영역을 센서에게 할당하기 때문에 전체 센서 네트워크의 저장 공간을 고르게 사용한다.

다음으로, 저장소 핫스팟에 따른 질의 결과의 QoD (Quality of Data)를 통해 TPDCS, DIM 그리고 KDDCS의 성능을 평가하였다. 질의 결과의 QoD는 데이터 손실률에 따라 크게 좌우된다. 실험은 256개의 센서로 구성된 400×400 크기의 센서 네트워크 환경에서 수행되었다. 실험에 사용된 데이터는 각 데이터 수집 라운드마다 발생하는 데이터의 80%를 전체 센서 네트워크의 5%와 10%에 해당하는 센서들에게 저장하는 저장소 핫스팟 환경에서 센서 수만큼 라운드를 수행하여 수집하였다. 그리고 데이터 수집 후, 저장소 핫스팟 영역을 중심으로 전체 센서 네트워크의 25%에 해당하는 영역에 전체 라운드를 질의 시간 영역으로 가지는 질의를 수행한 후 질의 결과로 반환되는 데이터의 양을 검사하였다.

그림 10은 저장소 핫스팟이 발생한 상황에서 질의 시간 영역을 일정한 구간 단위로 가지는 질의 처리에 대한 QoD를 보여준다. 실험 결과 DIM은 가장 최근 데이터만이 반환된다. KDDCS의 경우 DIM보다 좋은 QoD를 가지지만, 최근의 데이터가 압도적으로 많이 반환된다. 반면, TPDCS는 모든 시간 구간에서 균등한 수의 데이터를 반환하는 것을 확인하였다. TPDCS는 다른 DCS 기법들과 달리 데이터 영역 재조정 알고리즘에 따라 센서 네트워크에 존재하는 모든 센서들이 저장 공간을 고르게 사용하기 때문에 과거 데이터가 최대한 보존되는 특성을 가진다. 따라서 최신 데이터는 물론, 과거 데이터에 대한 질의 역시 높은 QoD를 보장한다.

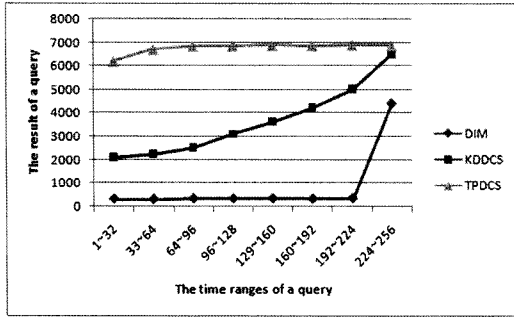
마지막으로, DIM, KDDCS 그리고 TPDCS에서 저장소 핫스팟과 질의 핫스팟을 함께 발생시켜 센서 네트워크의 수명과 각 센서의 에너지 소모율에 대한 성능을



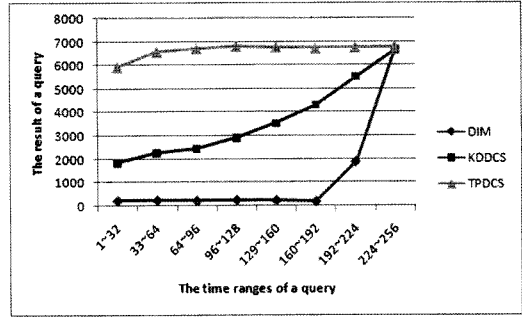
(a) 80%-5%의 저장소 핫스팟

(b) 80%-10%의 저장소 핫스팟

그림 9 저장된 데이터와 손실된 데이터의 비율



(a) 80%~5%에서 질의 처리



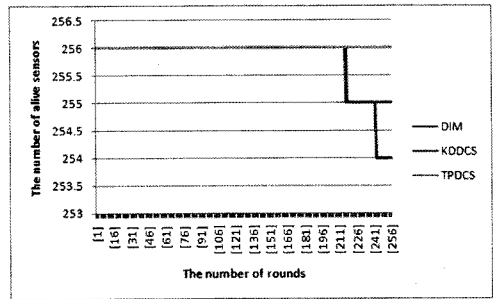
(b) 80%~10%에서 질의 처리

그림 10 저장소 핫스팟 환경에서 질의 처리 시 반환되는 데이터 수

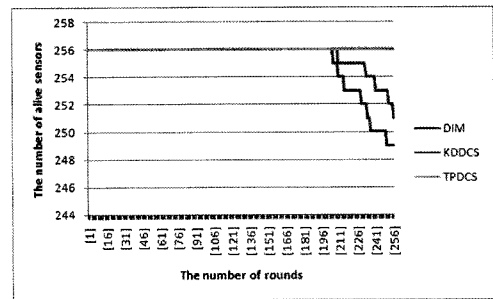
평가하였다. 실험에 사용된 데이터 셋은 각 라운드마다 발생하는 전체 데이터의 80%를 전체 센서 네트워크의 10%에 해당하는 센서들에게 저장하는 저장소 핫스팟 환경에서 센서 수만큼 데이터 수집 라운드를 수행하여 데이터를 수집하였다. 그리고 각 라운드마다 저장소 핫스팟 영역을 중심으로 질의 영역의 크기를 변경하며 일정 수의 질의를 반복 수행하였다. 질의 시구간은 현재 데이터 수집 라운드가  $k$ 인 경우,  $k-1$ 번째 데이터 수집 라운드를 대상으로 설정하였다. 또한, 센서 네트워크의 수명은 전체 센서 네트워크의 센서 중 하나의 센서 에너지가 0이 되는 시점까지로 가정하였다.

그림 11은 저장소 핫스팟 환경에서 질의 영역의 크기를 달리하며 질의 핫스팟을 발생시켰을 때, 센서 네트워크의 수명을 보여준다. 실험 결과 DIM과 KDDCS의 경우 30%와 40%의 질의 영역을 가지는 질의 핫스팟 환경에서 200라운드 이후 일부 센서의 에너지가 모두 소모된 것을 확인할 수 있다. 이와 같은 결과는 DIM이 저장소 핫스팟과 질의 핫스팟이 함께 발생하면 특정 센서의 에너지가 빠르게 소모하기 때문이다. KDDCS는 DIM보다 나쁜 성능을 보인다. 이는 DIM과 달리 KDDCS는 저장소 핫스팟이 발생한 센서에서 데이터의 재분배를 위해, 추가적인 에너지 소모가 요구되는 K-D 트리의 분할 축소 및 데이터 이동이 발생하기 때문이다. 반면에, TPDCS는 질의 처리 시 질의 데이터 영역이 저장소 핫스팟이 발생한 센서일지라도 질의 시간 영역에 따라 해당 질의가 다른 센서들에서 수행될 수 있기 때문에 DIM과 KDDCS에 비해 질의 핫스팟 문제가 줄어든다.

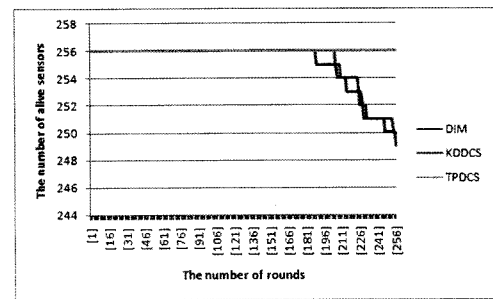
그림 12는 저장소 핫스팟 환경에서 질의 영역의 크기를 달리하며 질의 핫스팟을 발생시켰을 때, 센서의 에너지 비율을 보여준다. 실험 환경은 그림 11을 위한 실험 환경과 같다. DIM과 KDDCS는 전체 센서 네트워크의 평균 에너지 측면에서는 TPDCS보다 좋은 성능을 보여준다. 그러나 최소 에너지 측면에서는 TPDCS가



(a) 20%의 질의 영역

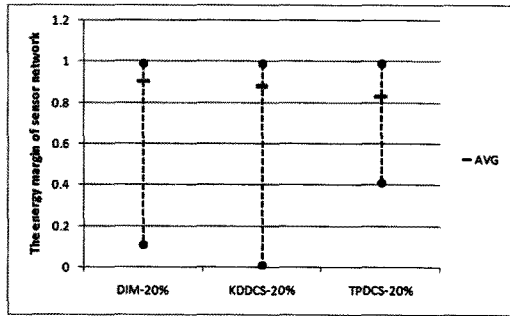


(b) 30%의 질의 영역

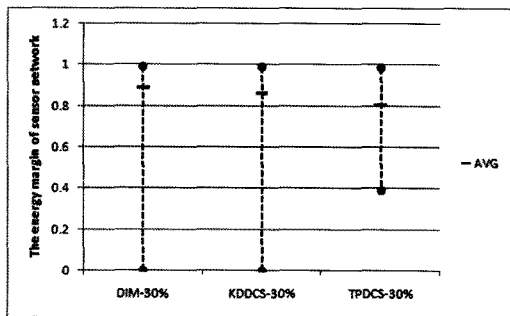


(c) 40%의 질의 영역

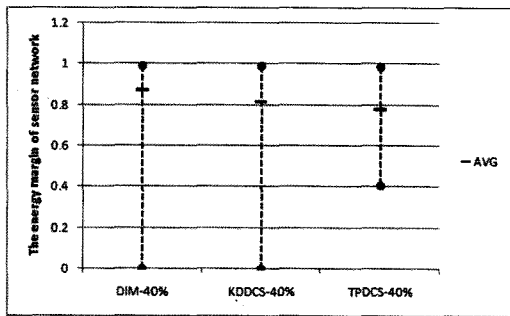
그림 11 80%~10% 저장소 핫스팟과 질의 핫스팟 발생 시 센서 네트워크의 수명



(a) 20%의 질의 영역



(b) 30%의 질의 영역



(c) 40%의 질의 영역

그림 12 80%~10% 저장소 핫스팟과 질의 핫스팟 발생 시 센서 에너지의 비율

보다 좋은 성능을 보여준다. TPDCS는 질의 처리 시, 질의 데이터 영역의 크기와 시간 범위에 따라 전체 센서 네트워크의 센서를 방문할 수도 있다. 따라서 센서 네트워크의 평균 에너지 소모에 있어서는 다른 기법들에 비해 많은 에너지를 소모한다. 하지만, 저장소 핫스팟과 질의 핫스팟을 효과적으로 분리하여 처리하므로 특정 센서의 불균등한 에너지 소모를 최소화 할 수 있다. 따라서 TPDCS는 DIM이나 KDDCS에 비해, 전체 센서 네트워크의 에너지 소모를 최대한 균등하게 유지시켜 네트워크의 수명을 오래 유지할 수 있다.

### 5. 결론 및 향후 연구

본 논문에서는 DIM과 KDDCS와 같이 DCS 기반의 기법들이 데이터 영역만을 고려한 데이터 분할 정책을 사용하기 때문에 발생하는 저장소 핫스팟과 질의 핫스팟 문제를 해결하기 위한 새로운 DCS 기법을 제안했다. 제안하는 기법인 TPDCS는 시간 영역과 데이터 영역을 함께 고려해 분할된 시간-데이터 영역을 센서에게 할당하여 기존 DCS 기반의 기법들에 비해 센서 네트워크의 저장 공간 활용률과 에너지 효율을 크게 향상시켰다. 마지막으로, 본 논문은 DIM, KDDCS와 다양한 성능 평가를 통해 제안하는 TPDCS의 우수성을 입증하였다.

향후 연구는 TPDCS의 기반의 센서 네트워크 환경에 적합한 클러스터링 기법, 집합(Aggregation) 기법뿐만 아니라 Skyline 질의, k-NN 질의 그리고 Top-K 질의와 같은 다양한 질의 처리 기법을 연구하여 센서 네트워크의 에너지 효율을 극대화할 것이다.

### 참고 문헌

- [1] National Research Council, "Embedded Everywhere: A Research Agenda for Networked Systems of Embedded Computers," National Academy Press, 2001.
- [2] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao, "Habitat Monitoring: Application Driver for Wireless Communications Technology," In ACM SIGCOM Workshop on Data Communications in Latin America and the Caribbean, 2001.
- [3] G. Pottie and W. Kaiser, "Wireless Integrated Sensor Networks(WINS)," In Communications of the ACM, Vol.43, No.5, 2000.
- [4] T. Yan, T. He, and J. A. Stankovic. Differentiated surveillance for sensor networks," In Proc. of SenSys, 2003.
- [5] 임용훈, 정연동, 김명호, "데이터 기반 센서 네트워크에서 다차원 영역 질의를 위한 동적 데이터 분산", 정보과학회논문지, 제33권, 제1호, pp. 32-41, 2006.
- [6] S. Shenker, S. Ratnasamy, B. Karp, R. Govidan, and D. Estrin. "Data-centric storage in sensornets," In Proc. of HotNets-I, 2002.
- [7] Xin Li, Young-Jin Kim, Ramesh Govindan, and Wei Hong, "Multi-dimensional range queries in sensor networks," In Proc. of SenSys, pp. 63-75, 2005.
- [8] Mohamed Aly, Kirk Pruhs, Panos K. and Chrysanthos, "KDDCS: a load-balanced in-network data-centric storage scheme for sensor networks," In Proc. of CIKM, pp. 317-326, 2006.
- [9] B. Moon, H. V. Jagadish, C. Faloutsos, and J. H. Saltz, "Analysis of the Clustering Properties of the Hilbert Space-Filling Curve," Journal of TKDE, pp. 124-141, 2006.

[10] S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Extrin, R. Govidan, and S. Shenker. "GHT: A Geographic Hash Table for Data-centric Storage," In Proc. of WSN, 2002.

[11] Y. Lai, H. Chen, and Y. Wang. "Dynamic Balanced Storage in Wireless Sensor Networks," In Proc. of DMSN, 2007.

[12] M. Aly, P. Chrysanthis, and K. Pruhs. "Decomposing Data-centric Storage Query Hot-Spots in Sensor Networks," In Proc. of MOBIQUITOUS, 2006.

[13] CT. Ee, S. Ratnasamy, and S. Shenker, "Practical Data-centric Storage," In Proc. of NSDI, 2006.

[14] M. Caesar, M. Castro, EB. Nightingale, G. O'Shea, and A. Rowstron, "Virtual Ring Routing: Network Routing Inspired by DHTs," In Proc. of SIGCOMM, 2006.

[15] J. Newsome, and D. Song, "GEM: Graph Embedding for Routing and Data-centric Storage in Sensor Networks without Geographic Information," In Proc. of SenSys, 2003.

[16] Y. Mao, F. Wang, L. Qiu, SS. Lam, JM. Smith, "S4: Small State and Small Stretch Routing Protocol for Large Wireless Sensor networks," In Proc. of NSDI, 2007.

[17] Y. Zhao, Y. Chen, and S. Ratnasamy, "Load Balanced and Efficient Hierarchical Data-centric Storage in Sensor Networks," In Proc. of SECON, 2008.

[18] A. Henrich, H. S. Six, and P. Widmayer, "The LSD-tree: Spatial Access to Multi-dimensional Point and Non-point Objects," In Proc. of International Conference on VLDB, pp. 45-53, 1989.

[19] B. Karp and H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Sensor Networks," In Proc. of ACM Mobicom, 2000.



서 동 민

2002년 충북대학교 정보통신공학과(공학사). 2004년 충북대학교 정보통신공학과(공학석사). 2008년 충북대학교 정보통신공학과(공학박사). 2008년 3월~현재 한국과학기술원 전산학과 연수연구원. 관심분야는 데이터베이스 시스템, 에이전트 시스템, XML, 이동 객체 데이터베이스, 시공간 색인 구조, 센서 네트워크 등



김 준

1982년 2월 부산대학교 계산통계학과(공학사). 1986년 한국과학기술원 전산학과(공학석사). 1986년~2008년 한국전자통신연구원 저장시스템연구팀 책임연구원 2008년 3월~현재 (주)테스트마이더스 대표이사. 관심분야는 데이터베이스 관리 처리, 파일 시스템, 센서 네트워크, SW 테스트

유 재 수

정보과학회 논문지 : 데이터베이스 제 36 권 제 1 호 참조



박 용 훈

2005년 호원대학교 정보통신공학과 및 건축공학과(공학사). 2007년 충북대학교 정보통신공학과(공학석사). 2007년~현재 충북대학교 정보통신공학과 박사과정. 관심분야는 데이터베이스 시스템, 정보검색, 시공간 데이터베이스, 센서 네트워크 등



윤 중 현

2003년 충북대학교 전기전자및컴퓨터공학부 정보통신공(공학사). 2005년 충북대학교 대학원 정보통신공학과(공학석사) 2005년 3월~현재 충북대학교 정보통신공학과 박사과정. 관심분야는 DBMS, 저장시스템, 시공간색인구조, 이동객체, 센서 네트워크 등

서 네트워크 등